LLM–KG COLLABORATIVE REASONING WITH KNOWLEDGE PROBES

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) excel at reasoning but suffer from static knowledge and hallucinations in fact-intensive tasks. Knowledge Graphs (KGs) offer trustworthy evidence, yet existing KG-centric agents remain brittle: relevant evidence may be structurally unavailable due to missing entities or relations, or computationally unavailable when buried in dense neighborhoods that are too costly to traverse exhaustively. We propose a KG-first collaborative reasoning framework that improves access to existing KG evidence in dense neighborhoods while recovering from missing KG links without relinquishing factual control to the LLM. We introduce the **Guidance Graph of Thought (GGoT)** as the framework's state interface, exposing unresolved reasoning slots as probeable states, and build on this interface with a conditional KG-first protocol. This protocol preserves reasoning continuity when required KG links are absent, reduces unnecessary exploration when relevant facts exist in large neighborhoods, and admits LLM-generated knowledge only under strict factual control. Experiments on Freebase- and Wikidata-based QA benchmarks show consistent improvements over strong KG-LLM reasoning baselines across both large and smaller LLMs. Efficiency and ablation analyses further confirm that the framework reduces dense-neighborhood exploration while making controlled recovery from KG dead ends beneficial rather than permissive.

1 INTRODUCTION

Large Language Models (LLMs) have achieved impressive performance across diverse natural language processing tasks, such as question answering (Wang et al., 2024; Li et al., 2024) and commonsense reasoning. By pre-training on massive text corpora, they can generate coherent and contextually appropriate responses (Ji et al., 2024; Chen et al., 2023b;a; Gong & Sun, 2024). However, their knowledge remains frozen after training, and they are prone to producing plausible yet factually incorrect outputs—a phenomenon known as hallucination (Bang et al., 2023; Ji et al., 2023; Luo et al., 2023). Knowledge Graphs (KGs), on the other hand, offer structured, explicit, and verifiable representations of entities and their relationships, making them well-suited to compensate for these limitations of LLMs (Zhang et al., 2019; Yao et al., 2019; Wang et al., 2021; Luo et al., 2024a).

Recent studies have explored integrating LLMs and KGs through an agent-based exploration paradigm. Approaches such as ToG (Sun et al., 2024), PoG (Chen et al., 2024), FiSKE (Tao et al., 2025b), and GG-explore (Tao et al., 2025a) iteratively expand from selected nodes to their neighbors, retrieving facts at each step and progressively building an interlinked chain of knowledge to ground LLM reasoning. These paradigms often suppress the LLM's internal knowledge to minimize hallucinations, relying on the KG as the sole source of truth. While this ensures factual grounding, it exposes a broader limitation: relevant KG evidence may become unavailable to the reasoning process in different ways.

First, evidence may be structurally unavailable because no KG can contain all entities and relations. If a crucial node or edge is missing, reasoning breaks–exploration halts and relevant downstream facts remain inaccessible. Second, evidence may be computationally unavailable even when it exists in the KG. Large-scale KGs like Freebase can link a single node (e.g., France) to hundreds of relations and over 100,000 entities. Exhaustive traversal is computationally prohibitive, so existing

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

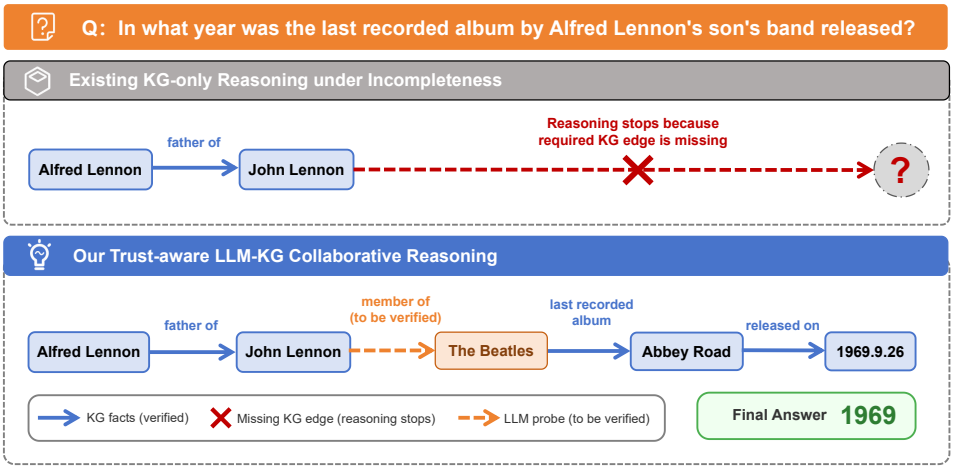

Figure 1: Illustration of KG Reasoning under Incompleteness: Existing Approaches vs Trust-aware Bridging in Our Framework.

methods often adopt random or heuristic edge selection. Consequently, useful facts may be skipped not because they are absent, but because they are too costly to access reliably.

To improve both the coverage and accessibility of KG evidence, we propose a KG-first collaborative reasoning framework that keeps the KG as the primary evidence source while allowing LLM-generated knowledge to assist only at explicitly exposed unresolved reasoning slots. The key interface is the Guidance Graph of Thought (GGoT), which represents these slots as typed triples and turns LLM knowledge into probes rather than accepted facts.

The resulting protocol uses probes in two targeted ways. In dense KG neighborhoods, probes serve as semantic handles for locating a small KG-grounded candidate region, reducing unnecessary exploration without leaving the KG neighborhood. When KG-local exploration reaches a dead end, probes may propose bridges over missing entities or relations, but only as tentative recoveries subject to trust validation and later retraction.

This design separates probe production from probe admissibility. GGoT exposes where hypotheses may enter the reasoning state, while the KG-first protocol decides whether they can be grounded, committed, or rolled back. Thus, the framework exploits the LLM's internal knowledge without making it the main factual authority.

Our **contributions** are threefold: (1) we introduce the Guidance Graph of Thought (GGoT), an extensible interface that exposes unresolved reasoning slots for controlled interaction between LLM hypotheses and external factual sources; (2) we develop a KG-first collaborative reasoning protocol with conditional probe operators for two evidence-unavailability regimes: KG-local acceleration for computationally inaccessible evidence in dense neighborhoods and trust-aware bridging for structurally missing evidence under KG dead ends; (3) we demonstrate consistent gains on Freebase- and Wikidata-based benchmarks, with efficiency and ablation analyses showing reduced dense-neighborhood exploration and controlled recovery from KG dead ends.

2 Related Work

Reasoning with Large Language Models. Structured prompting improves LLM reasoning by decomposing complex tasks. Chain-of-Thought (CoT) (Wei et al., 2022) elicits sequential reasoning, while extensions such as Tree-of-Thought (ToT) (Yao et al., 2023) and Graph-of-Thought (GoT) (Besta et al., 2024) introduce non-linear reasoning structures, while Self-Consistency (Wang et al., 2022), Least-to-Most prompting (Zhou et al., 2022), Memory-of-Thought (Li & Qiu, 2023), Program-Aided Reasoning (Gao et al., 2023), Plan-and-Solve prompting (Wang et al., 2023a), and Reflexion (Shinn et al., 2023) improve robustness, planning, or self-correction.

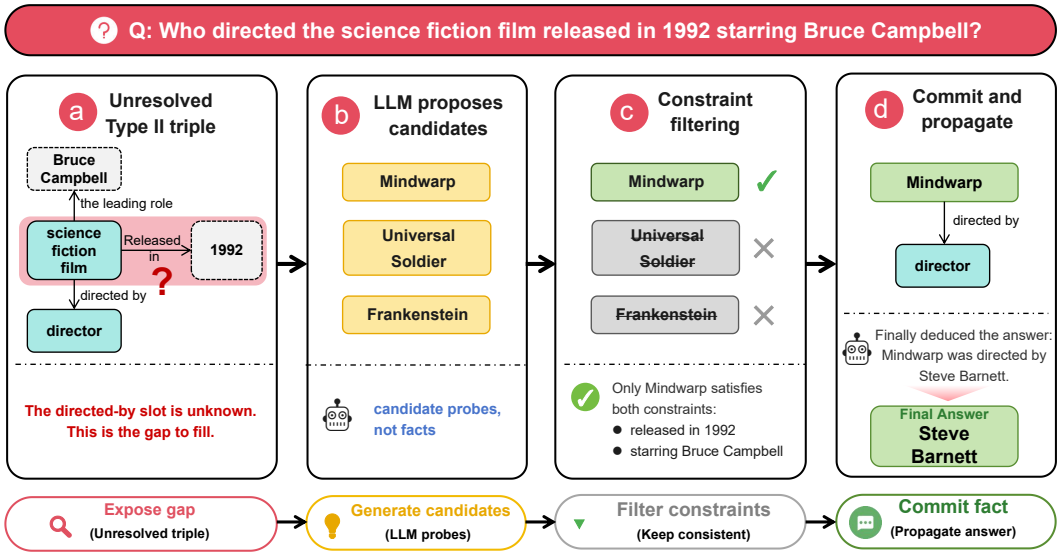

Figure 2: Example of probeable state transitions in GGoT: Type II triples expose candidate hypotheses, neighboring constraints filter admissible commitments, and committed nodes propagate to subsequent triples.

Integrating Knowledge Graphs with LLMs. KG integration methods use structured knowledge through input augmentation, model adaptation, or external access during inference. K-BERT (Liu et al., 2020) injects KG triples into BERT inputs, while StructLM (Zhuang et al., 2024) instruction-tunes models across structured-knowledge tasks. In KGQA, KAPING (Baek et al., 2023) retrieves facts into prompts without model training; KB-BINDER (Li et al., 2023) generates and binds logical forms. Other interfaces include programmatic KB access (Wang et al., 2023b), graph retrieval (Mavromatis & Karypis, 2025), and iterative reasoning in RoG (Luo et al., 2024a), ToG (Sun et al., 2024), PoG (Chen et al., 2024), and FiSKE (Tao et al., 2025b). ChatDB (Hu et al., 2023) instead uses SQL databases as symbolic memory. Our work focuses on inference-time KG collaboration under both sparse connections and dense neighborhoods. Reasoning-intensive agent frameworks treat KGs as environments for iterative multi-hop exploration, which makes them especially relevant to our setting.

Recent graph-exploration methods also include Search-on-Graph (Sun et al., 2026), which couples navigation to graph structure and reasoning history, Generate-on-Graph (Xu et al., 2024), which generates and verifies triples for incomplete-KG QA, and Explore-on-Graph (Yan et al., 2026), which learns exploration with path-refined rewards. These works provide context for distinguishing the conditions under which our probes localize existing evidence or propose tentative continuations.

3 Guidance Graph of Thought as an Extensible Interface

The Guidance Graph of Thought (GGoT) makes unresolved reasoning slots explicit. It provides an interface for generating candidate hypotheses; the KG-first protocol determines whether they can be committed.

3.1 Guidance Graph Representation

Natural language queries often contain a mix of concrete entities (*e.g.*, "Beijing") and generic concepts (*e.g.*, "city"), making purely text-based reasoning hard to control or verify. To separate factual anchors from reasoning targets, we represent the query as a structured graph of entities and relations, following Tao et al. (Tao et al., 2025a). The resulting Guidance Graph has nodes that are specific or generic entities, and edges encoding query. Construction rules are in AppendixA.3, with prompt templates in Tables 8 and 9.

3.2 TYPED INTERVENTION POINTS

Triples in the Guidance Graph (h, r, t) differ in their specificity. Some express fully grounded knowledge, while others contain generic placeholders requiring inference. GGoT organizes these states as typed intervention points rather than as an unconstrained LLM reasoning chain.

We distinguish three mutually exclusive triple types: **Type I** — both head and tail are specific entities (e.g., `China, Capital, Beijing`), representing fully specified entity slots; **Type II** — one side specific, the other generic (e.g., `China, Capital, City`); the specific side anchors a probeable slot for the generic entity; **Type III** — both head and tail generic (e.g., `Country, Capital, City`), which rely on other triples before resolution.

The taxonomy exhausts the specificity states of the two entity slots: the four ordered specific/generic combinations become three classes when the two one-specific cases are grouped. The relation remains a predicate constraint. Type II supplies both a concrete conditioning entity and an unresolved target; Type I needs no entity completion, and Type III becomes probeable after a neighboring instantiation supplies an anchor. Specificity and factual support are assessed separately.

Type II triples are the only places where probes can be produced. A probe may suggest a concrete instantiation for the generic node, but the triple becomes Type I only after the candidate is committed by an admissibility rule. As committed nodes propagate, previously generic connections (Type III) become partially grounded (Type II) and expose new intervention points. Figure 2 illustrates this state transition.

3.3 PROBE OPERATORS AND CONSTRAINT-BASED COMMITMENT

The transformation from Type II to Type I triples is therefore separated into four interface operations: generating candidates, filtering them with available constraints, committing admissible candidates, and retracting tentative commitments when later evidence contradicts them.

Probe Generation: For a given Type II triple (s, r, g), where s is a specific entity, r a relation, and g a generic entity, the LLM uses s and r as cues to propose candidate entities instantiating g. For example, given `(China, Capital, City)`, candidates such as `Beijing` may be drawn from the model's internal knowledge (see prompts in Appendix 10). These candidates serve only as initial probes–hypotheses that still require downstream admissibility checks before being accepted as factual.

Constraint-Based Commitment: Since generic entities often occur in multiple triples–e.g., (s_1, r_1, g) and (s_2, r_2, g)–a candidate committed to g must satisfy all incident constraints. In an LLM-only instantiation, this check can be performed by semantic prompting with contextual information from neighboring triples (Appendix 11). In our KG-collaborative framework, Section 4 instantiates the same interface with KG-local candidate consistency, necessity-gated bridging, and GGoT-style semantic fallback for candidates whose KG structure is incomplete. Thus, GGoT defines where candidates can enter the reasoning state, while the KG-first protocol defines whether they may be used, committed, or retracted.

4 KG-FIRST COLLABORATIVE REASONING WITH CONDITIONAL PROBES

For each Type II triple, the protocol first explores the current KG anchor and its neighborhood. Probes support two conditional operations: acceleration localizes candidates within dense neighborhoods, while bridging proposes tentative continuations only after KG-local exploration fails. Appendix A.1 summarizes the protocol.

4.1 PROBE-ASSISTED ANCHOR DISCOVERY

The protocol starts by linking the specific side of a Type II triple to the KG. Existing KG exploration methods commonly rely on dataset-provided topic entities or assume that the query exposes an exact KG surface form for entity linking. Standard label matching remains the default mechanism in our framework, but probeable slots provide an additional pool of semantically close entity names

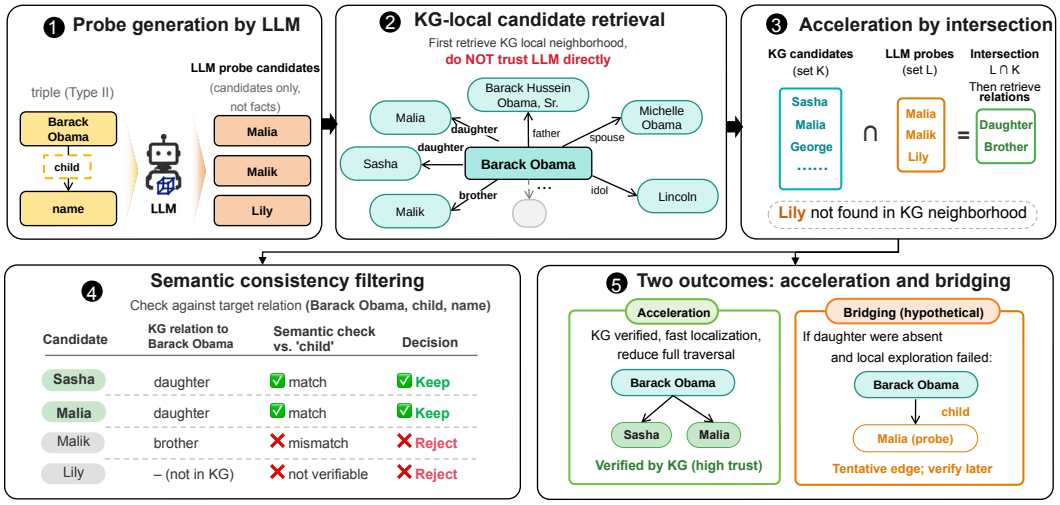

Figure 3: Knowledge probes in KG–LLM collaboration. With *daughter* present, Malia supports acceleration and expansion retrieves Sasha; Malik fails the *child* constraint. Bridging is hypothetical: if the *daughter* connection were absent and local exploration failed, Malia would serve as the endpoint of a tentative *child* bridge from Barack Obama, subject to subsequent validation.

for anchor matching. These probes do not become facts by themselves: only entities that can be matched to KG nodes are used as anchors for KG-local exploration.

This probe-assisted anchor discovery improves coverage when the query does not expose an exact KG surface form or when aliases make literal matching brittle. If no valid KG anchor emerges after several rounds, the system does not force a probe into the KG; it reverts to the LLM-only GGoT fallback, preserving continuity without fabricating KG evidence.

4.2 KG-Local Acceleration in Dense Neighborhoods

Given a Type II triple $(\text{Specific_s}, r, \text{Generic_g})$, we identify the corresponding KG node N for Specific_s and collect all directly connected neighbors: $\text{Candidate_KG} = \{e \mid (N, r', e) \in \text{KG}\}$. In the acceleration regime, a probe is not adopted directly. We first intersect the probe set $P = \{\text{p_e}_1, \text{p_e}_2, \dots\}$ with Candidate_KG.

This intersection acts as a targeted accelerator: it localizes a small, KG-verified candidate region before expensive relation-level filtering, rather than scoring or pruning the full neighborhood directly. Since the adopted entities are a subset of Candidate_KG, this branch cannot introduce entities outside the KG neighborhood; its remaining risk is relevance error, which is handled by the semantic filtering step below.

Because probes are finite and may not cover all relevant nodes, we then perform a controlled *expansion*: each KG-matched probe is extended to include all entities connected to N under the same relation. This step compensates for limited probe coverage while remaining restricted to the same semantic relation context. Crucially, if the intersection yields no matches (i.e., acceleration fails), the process falls back to the full neighborhood of N as the candidate set.

For nonempty seeds $S = P \cap \text{Candidate_KG}$, let $R_S = \{r' : \exists e \in S, (N, r', e) \in \text{KG}\}$ and let $C_S = \{e : \exists r' \in R_S, (N, r', e) \in \text{KG}\}$. Then $S \subseteq C_S \subseteq \text{Candidate_KG}$: a seed identifies a relation context, and expansion can recover other neighbors under that context. This set property concerns KG membership; semantic relevance is checked separately.

To guard against *semantic hallucinations*—where a triple is factual but contextually mismatched—we perform **semantic consistency filtering** with the LLM. To further reduce computational overhead from one-to-many relation contexts in the KG, we group candidates by their $(N, \text{relation})$ context and represent them as $(N, \text{relation}, \text{generic})$. If the semantics of this triple aligns with the original $(\text{Specific_s}, r, \text{Generic_g})$, the relation context is retained, and its candidate entities are eval-

uated against the other incident query constraints. The filtering prompts are given in Appendix 12. The selected entity then replaces Generic_g in the Guidance Graph, fully instantiating the triple.

4.3 Trust-aware Bridging under KG Dead Ends

Despite careful curation, KGs inevitably have bounded coverage: some entities or relations needed for answering may be absent, causing premature termination when a crucial local link is missing. Bridging addresses this failure mode only after KG-local exploration on the current Type II triple cannot produce a viable continuation. This *necessity condition* is the gate for all bridges: a probe cannot override an available KG path and is evaluated only against a locally failing branch.

Within this necessity-gated regime, trust depends on the grounding level of the proposed bridge. If a probe entity p_e is absent from Candidate_KG, it may signal either local KG incompleteness or hallucination. We therefore distinguish existence-grounded bridges from provisional virtual bridges.

Existence-grounded bridging. If p_e exists elsewhere in the KG, the entity is grounded, but the required relation to the current anchor remains a hypothesis. Its absence from the explored neighborhood can reflect a missing connection or a failure of local retrieval. Under the necessity condition, KG presence gives ontological grounding, and the probe becomes a semantic bridge: we temporarily link N to p_e via r and continue exploration through p_e. The link is retained only if subsequent steps return coherent, verifiable facts; otherwise it is removed.

Provisional virtual bridging. If p_e is absent both locally and globally, it has no KG entity grounding. It can still be used only after the same necessity condition holds, and only as a provisional virtual node that maintains connectivity for subsequent reasoning. Such a node is considered only when the alternative is terminating the current reasoning branch; it is never used to replace a KG-local continuation. Later exploration validates or retracts this link based on whether coherent facts emerge.

4.4 Cross-Triple Candidate Consistency in KG–LLM Collaboration

As discussed in Section 3, the same generic node may appear in multiple triples, so a candidate proposed from one Type II triple must also satisfy the other incident triples before commitment. For KG-local candidates, we verify this consistency structurally: the candidate node should also connect to the specific nodes in the other Type II triples involving the same generic node. For non-local candidates whose required KG structure is unavailable, we reuse the GGoT-style semantic consistency check from Section 3.

5 Experiments

5.1 Experimental Settings

Datasets and Evaluation Metrics. We use Freebase (Bollacker et al., 2008) as the main external KG and evaluate on WebQSP (Yih et al., 2016) and CWQ (Talmor & Berant, 2018); Wikidata (Vrandečić & Krötzsch, 2014) is used to test generalization across KGs. For multi-answer questions, we report partial match (at least one gold answer retrieved) and complete match (all gold answers retrieved).

Baselines. We compare against standard prompting (Brown et al., 2020), CoT (Wei et al., 2022), fine-tuned KG reasoning methods (Luo et al., 2024a;b; Mavromatis & Karypis, 2025; Long et al., 2025), and inference-time KG-LLM systems including StructGPT, KB-BINDER, ReKnos, FiDeLis, FastToG, ReKG, ToG, PoG, FiSKE, and GG-explore (Jiang et al., 2023; Li et al., 2023; Wang et al., 2025; Sui et al., 2025; Liang & Gu, 2025; Song et al., 2025; Sun et al., 2024; Chen et al., 2024; Tao et al., 2025b;a).

5.2 Performance Comparison

Table 1 compares backbone models and KG collaboration. With DeepSeek-V3, our framework scores 85.7 on WebQSP partial match and 75.2 on CWQ, versus 81.8 and 71.8 for GG-explore. With LLaMA3-8B, the corresponding scores are 81.3 and 59.3, versus 79.3 and 56.7. The LLM-

only GGoT variant is more sensitive to the backbone: its WebQSP partial match rises from 52.1 to 72.5 when switching to DeepSeek-V3, while KG collaboration improves both backbones.

Table 1: Results for WebQSP and CWQ. Partial and complete refer to the matching metrics described in Section 5.1. Bold results indicate the highest reported values among the listed methods without finetuning.

Method	WebQSP		CWQ	
	partial	complete	partial	complete
without external knowledge				
IO prompt (Brown et al., 2020) w/DeepSeek-V3	63.3	35.3	44.8	44.8
COT (Wei et al., 2022) w/DeepSeek-V3	70.5	41.4	46.7	46.7
with finetuning				
RoG (Luo et al., 2024a) w/Llama2-7B	85.7	–	62.6	62.6
GCR (Luo et al., 2024b) w/Llama3.1-8B+ChatGPT	92.6	–	72.7	72.7
GNN-RAG (Mavromatis & Karypis, 2025) w/Llama2-7B	85.7	–	66.8	66.8
EPERM (Long et al., 2025) w/Llama2-7B	88.8	–	66.2	66.2
without finetuning				
StructGPT (Jiang et al., 2023) w/GPT-3.5	72.6	–	54.3	54.3
KB-BINDER (Li et al., 2023) w/GPT-3.5	74.4	–	–	–
ReKnos (Wang et al., 2025) w/GPT-3.5	81.1	–	58.5	58.5
FiDeLis (Sui et al., 2025) w/GPT-4-turbo	84.4	–	71.5	71.5
FastToG (Liang & Gu, 2025) w/GPT-4o-mini	65.8	–	45.0	45.0
FastToG (Liang & Gu, 2025) w/Llama3-70B	66.4	–	46.2	46.2
ReKG (Song et al., 2025) w/Llama3-8B	72.2	–	59.8	59.8
ReKG (Song et al., 2025) w/Llama3-70B	78.9	–	64.4	64.4
ToG (Sun et al., 2024) w/Llama3-8B	55.6	32.3	–	–
ToG (Sun et al., 2024) w/GPT-3.5	76.2	–	57.1	57.1
PoG (Chen et al., 2024) w/Llama3-8B	63.4	34.4	–	–
PoG (Chen et al., 2024) w/DeepSeek-V3	81.9	60.7	55.7	55.7
FiSKE (Tao et al., 2025b) w/Llama3-8B	70.8	40.4	–	–
FiSKE (Tao et al., 2025b) w/DeepSeek-V3	82.5	61.1	50.2	50.2
GG-explore (Tao et al., 2025a) w/Llama3-8B	79.3	54.1	56.7	56.7
GG-explore (Tao et al., 2025a) w/DeepSeek-V3	81.8	64.5	71.8	71.8
GGoT-LLM w/Llama3-8B	52.1	21.2	35.9	35.9
GGoT-LLM w/DeepSeek-V3	72.5	46.4	53.4	53.4
GGoT-LLM+KG w/Llama3-8B	81.3	55.2	59.3	59.3
GGoT-LLM+KG w/DeepSeek-V3	**85.7**	**65.5**	**75.2**	**75.2**

5.3 Computational Cost

Table 2 reports average calls and tokens per question, not end-to-end latency. Relative to GG-explore, total tokens decrease by 9.4% on WebQSP (3848.5 to 3486.4), but increase by 23.1% on CWQ (4760.6 to 5859.4), alongside a CWQ score increase from 71.8 to 75.2. Thus, the framework offers a dataset-dependent accuracy–cost trade-off rather than a uniform cost reduction.

Table 2: Efficiency comparison across datasets.

Dataset	Method	LLM Call	Input Token	Output Token	Total Token
WebQSP	PoG	11.3	6590.2	427.0	7017.2
	FiSKE	7.9	3079.0	1379.7	4458.7
	GG-explore	8.6	3264.5	584.0	3848.5
	Ours	8.13	3077.5	408.9	3486.4
CWQ	PoG	23.4	15483.4	694.8	16178.2
	FiSKE	9.4	3578.8	1828.7	5407.5
	GG-explore	10.2	4052.0	708.6	4760.6
	Ours	11.0	5239.5	619.9	5859.4

5.4 Ablation Studies

Table 3 reports three component ablations. Without probe-driven anchor discovery, starting points rely on provided or literally matched anchors. Removing the bridging module disables dead-end recovery and bridge rejection; removing knowledge probes disables all probe-dependent operators. The full configuration scores 75.2, compared with 69.3, 69.3, and 70.7 for these variants. This supports their contribution within the tested configuration, but the non-additive ordering does not isolate acceleration or establish the cause of component interactions.

Table 3: CWQ module ablations. Calls and input/output/total tokens are averaged per query.

Variant	Partial	Calls	Input	Output	Total
Ours	75.2	11.0	5239.5	619.9	5859.4
w/o probe-driven anchor discovery	69.3	12.2	3850.2	514.7	4364.9
w/o trust-aware bridging module	69.3	9.2	3631.5	483.0	4114.5
w/o knowledge probe	70.7	10.7	4136.1	691.8	4827.9

5.5 Studies on Trust-aware Bridging

Table 4 distinguishes bridge adoption from trust filtering at answer time. Ours* retains untrusted hypotheses; w/o TB removes accepted bridging triples before answering while retaining non-bridged facts and trust decisions. This differs from disabling the entire module in Table 3.

Retaining untrusted hypotheses reduces the score from 75.2 to 72.6; removing accepted bridges under the same trust filter yields 72.2. Trusted chains occur for 90.5% of CWQ questions, with trusted bridging in 21.8%. These are answer-level results, not edge-precision estimates. The selected fully trusted subset scores 77.4 and should not be interpreted as an independent causal test.

Table 4: CWQ trust variants. Ours* retains untrusted bridges; w/o TB removes accepted bridge triples at answer time. Filtered-CWQ is the trusted subset. Partial and complete match coincide for these reported rows.

Dataset	Variant	Trust (%)	Bridge (%)	Calls	Tokens	Score
CWQ	Ours	90.5	21.8	11.0	5859.4	75.2
CWQ	Ours*	90.5	21.8	11.0	5840.3	72.6
CWQ	Ours w/o TB	90.5	0	9.2	4114.5	72.2
CWQ	Ours* w/o TB	90.5	0	9.2	4099.1	69.3
Filtered-CWQ	Ours	100	24.1	11.2	6100.8	77.4

5.6 Evaluation on Wikidata

Table 5 shows the highest reported scores for our method on transferred CWQ and Wikidata-native QALD10-en. Baseline results are taken from their original paper; ToG uses ChatGPT for answer generation and ours uses DeepSeek-V3, so backbone differences remain. Our CWQ score drops by 13.0% relative to Freebase, while trusted-bridging usage rises from 21.8% to 30.9%. Transfer changes schema, entity alignment, and coverage together, rather than isolating missing edges.

Table 5: Cross-KG results. CWQ drop is relative to Freebase. ToG and ToG-R scores are from their original paper.

Method	CWQ/Freebase	CWQ/Wikidata	Drop	QALD10-en
ToG	58.8	54.9	6.6%	50.2
ToG-R	59.2	51.9	12.3%	48.6
Ours	**75.2**	**65.4**	13.0%	**70.7**

5.7 Qualitative Analysis

Table 6 traces two questions: which country bordering France contains an airport serving Nijmegen, and where the Country Nation World Tour artist attended college. The first illustrates relation expansion followed by a shared constraint; the second illustrates recovery of correct KG evidence despite noisy probes. Complete traces are provided in Appendix A.5.

Table 6: Qualitative traces of relation expansion and recovery from probe noise. Complete candidate lists and relation identifiers appear in Appendix A.5.

Case / step	**Recorded operation**	**Evidence or outcome**
France–Nijmegen: country step	From 498 relations at France, country probes such as Germany identify the adjoining-country relation path.	Relation expansion retrieves 19 candidates, extending retrieval beyond the generated names.
Airport step	Airport probes locate the containment relation from the country candidates.	Expansion yields 15,184 candidates; probe generation alone has not resolved the answer.
Shared constraint	Apply the requirement that the airport serves Nijmegen.	Weeze Airport remains, yielding the France–Germany–Weeze Airport–Nijmegen path and answer Germany.
Country Nation World Tour: artist step	Initial probes include Taylor Swift, Luke Bryan, Carrie Underwood, Blake Shelton, and Keith Urban.	None is connected to the tour. Matching the artist relation among 17 relations retrieves Brad Paisley.
Education step	Belmont University is the college probe connected to Brad Paisley; expand the education relation path.	Retrieved institutions include Belmont University, West Liberty University, and John Marshall High School.
Answer check	Reference answer: Belmont University. Reported output: Belmont University and West Liberty University.	The trace recovers correct KG evidence amid noisy probes, but does not establish exact answer-set recovery or missing-edge repair.

The dense-neighborhood trace also shows why the probe set is not the final answer space: Weeze Airport is absent from the generated airport names but is recovered by relation expansion and the Nijmegen constraint. Conversely, the tour trace recovers a correct KG path but outputs an additional institution beyond the reference answer. These observations motivate separating candidate generation, graph-based retrieval, and final answer validation.

6 Conclusion

Existing LLM–KG reasoning methods remain brittle because relevant KG evidence can be unavailable in two ways: missing entities or relations can break reasoning chains, while dense neighborhoods can make existing facts too costly to access reliably.

We presented a KG-first collaborative reasoning framework that addresses both cases without transferring factual authority to the LLM. GGoT serves as the framework's state interface, exposing unresolved reasoning slots as probeable states, while the KG-first protocol admits probes only as conditional operators for KG-local acceleration and trust-aware dead-end recovery. In this design, probes help locate useful KG evidence in dense neighborhoods and preserve reasoning continuity under missing links, but grounding, commitment, and rollback remain governed by KG-centered validation. Experiments across Freebase- and Wikidata-based benchmarks, together with efficiency and ablation analyses, show that the framework improves KG-grounded reasoning while remaining efficient and tightly controlled.

AI USE STATEMENT

LLM assistance was used for language editing and manuscript revision, including the organization of reviewer feedback and suggestions for clarifying the method and experiments.

REPRODUCIBILITY STATEMENT

Sections 3 and 4 describe the query representation and collaborative operations. Algorithm 1 summarizes the control flow; Appendix A.2 lists the prompts. Section 5 pairs each experimental analysis with its results table. Model-scale results and complete qualitative traces are provided with their analyses in Table 14 and Appendix A.5.

REFERENCES

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. In *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)*, pp. 78–106, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.nlrse-1.7. URL https://aclanthology.org/2023.nlrse-1.7/.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*, 2023.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 17682–17690, 2024.

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 2008 ACM SIGMOD international conference on Management of data*, pp. 1247–1250, 2008.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901, 2020.

Liyi Chen, Zhi Li, Weidong He, Gong Cheng, Tong Xu, Nicholas Jing Yuan, and Enhong Chen. Entity summarization via exploiting description complementarity and salience. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):8297–8309, 2023a.

Liyi Chen, Panrong Tong, Zhongming Jin, Ying Sun, Jieping Ye, and Hui Xiong. Plan-on-graph: Self-correcting adaptive planning of large language model on knowledge graphs. *arXiv preprint arXiv:2410.23875*, 2024.

Xi Chen, Xinjiang Lu, Haoran Xin, Wenjun Peng, Haoyang Duan, Feihu Jiang, Jingbo Zhou, and Hui Xiong. A table-to-text framework with heterogeneous multidominance attention and self-evaluated multi-pass deliberation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 607–620, 2023b.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. Pal: Program-aided language models. In *International Conference on Machine Learning*, pp. 10764–10799. PMLR, 2023.

Zheng Gong and Ying Sun. Graph reasoning enhanced language models for text-to-sql. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2447–2451, 2024.

Chenxu Hu, Jie Fu, Chenzhuang Du, Simian Luo, Junbo Zhao, and Hang Zhao. Chatdb: Augmenting llms with databases as their symbolic memory. *arXiv preprint arXiv:2306.03901*, 2023.

Bin Ji, Huijun Liu, Mingzhe Du, and See-Kiong Ng. Chain-of-thought improves text generation with citations in large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 18345–18353, 2024.

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

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. Structgpt: A general framework for large language model to reason over structured data. *arXiv preprint arXiv:2305.09645*, 2023.

Tianle Li, Xueguang Ma, Alex Zhuang, Yu Gu, Yu Su, and Wenhu Chen. Few-shot in-context learning for knowledge base question answering. *arXiv preprint arXiv:2305.01750*, 2023.

Xiaonan Li and Xipeng Qiu. Mot: Memory-of-thought enables chatgpt to self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6354–6374, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.392. URL https://aclanthology.org/2023.emnlp-main.392/.

Xiaoxi Li, Yujia Zhou, and Zhicheng Dou. Unigen: A unified generative framework for retrieval and question answering with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 8688–8696, 2024.

Xujian Liang and Zhaoquan Gu. Fast think-on-graph: Wider, deeper and faster reasoning of large language model on knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 24558–24566, 2025.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. K-bert: Enabling language representation with knowledge graph. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 2901–2908, 2020.

Xiao Long, Liansheng Zhuang, Aodi Li, Minghong Yao, and Shafei Wang. Eperm: An evidence path enhanced reasoning model for knowledge graph question and answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 12282–12290, 2025.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *The Twelfth International Conference on Learning Representations*. ICLR, 2024a. URL https://openreview.net/forum?id=ZGNWW7xZ6Q.

Linhao Luo, Zicheng Zhao, Gholamreza Haffari, Yuan-Fang Li, Chen Gong, and Shirui Pan. Graph-constrained reasoning: Faithful reasoning on knowledge graphs with large language models. *arXiv preprint arXiv:2410.13080*, 2024b.

Ziyang Luo, Can Xu, Pu Zhao, Xiubo Geng, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. Augmented large language models with parametric knowledge guiding. *arXiv preprint arXiv:2305.04757*, 2023.

Costas Mavromatis and George Karypis. Gnn-rag: Graph neural retrieval for efficient large language model reasoning on knowledge graphs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 16682–16699, 2025.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

Xiaozhuang Song, Shufei Zhang, and Tianshu Yu. Rekg-mcts: Reinforcing llm reasoning on knowledge graphs via training-free monte carlo tree search. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 9288–9306, 2025.

Yuan Sui, Yufei He, Nian Liu, Xiaoxin He, Kun Wang, and Bryan Hooi. Fidelis: Faithful reasoning in large language models for knowledge graph question answering. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 8315–8330, 2025.

Jia Ao Sun, Hao Yu, Fabrizio Gotti, Fengran Mo, Yihong Wu, Yuchen Hui, Zhan Su, Lingfeng Xiao, and Jian-Yun Nie. Search-on-graph: Iterative informed navigation for large language model reasoning on knowledge graphs, 2026. URL https://arxiv.org/abs/2510.08825v2. Version 2; accepted to KDD 2026.

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 *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=nnVO1PvbTv.

Alon Talmor and Jonathan Berant. The web as a knowledge-base for answering complex questions. *arXiv preprint arXiv:1803.06643*, 2018.

Dehao Tao, Guangjie Liu, Weizheng, Yongfeng Huang, and Minghu jiang. Guided navigation in knowledge-dense environments: Structured semantic exploration with guidance graphs, 2025a. URL https://arxiv.org/abs/2508.10012v1.

Dehao Tao, Congqi Wang, Feng Huang, Junhao Chen, Yongfeng Huang, and Minghu Jiang. Fine-grained stateful knowledge exploration: Effective and efficient graph retrieval with large language models. *Knowledge-Based Systems*, 326:114011, 2025b. ISSN 0950-7051. doi: 10.1016/j.knosys.2025.114011. URL https://www.sciencedirect.com/science/article/pii/S0950705125010561.

Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *arXiv preprint arXiv:2305.04091*, 2023a.

Lei Wang, Yi Hu, Jiabang He, Xing Xu, Ning Liu, Hui Liu, and Heng Tao Shen. T-sciq: Teaching multimodal chain-of-thought reasoning via large language model signals for science question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19162–19170, 2024.

Song Wang, Junhong Lin, Xiaojie Guo, Julian Shun, Jundong Li, and Yada Zhu. Reasoning of large language models over knowledge graphs with super-relations. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=rTCJ29pkuA.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194, 2021. doi: 10.1162/tacl_a_00360. URL https://aclanthology.org/2021.tacl-1.11/.

Xintao Wang, Qianwen Yang, Yongting Qiu, Jiaqing Liang, Qianyu He, Zhouhong Gu, Yanghua Xiao, and Wei Wang. Knowledgpt: Enhancing large language models with retrieval and storage access on knowledge bases. *arXiv preprint arXiv:2308.11761*, 2023b.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. URL https://papers.nips.cc/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Yao Xu, Shizhu He, Jiabei Chen, Zihao Wang, Yangqiu Song, Hanghang Tong, Guang Liu, Jun Zhao, and Kang Liu. Generate-on-graph: Treat LLM as both agent and KG for incomplete

knowledge graph question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 18410–18430. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.1023. URL https://aclanthology.org/2024.emnlp-main.1023/.

Shiqi Yan, Yubo Chen, Ruiqi Zhou, Zhengxi Yao, Shuai Chen, Tianyi Zhang, Shijie Zhang, Wei-Qiang Zhang, Yongfeng Huang, Haixin Duan, and Yunqi Zhang. Explore-on-graph: Incentivizing autonomous exploration of large language models on knowledge graphs with path-refined reward modeling. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/5d2e24df9cfaad3189833b819c40b392-Abstract-Conference.html.

Liang Yao, Chengsheng Mao, and Yuan Luo. Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*, 2019.

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. *Advances in neural information processing systems*, 36:11809–11822, 2023.

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 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 201–206, 2016.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*, 2019.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

Alex Zhuang, Ge Zhang, Tianyu Zheng, Xinrun Du, Junjie Wang, Weiming Ren, Stephen W. Huang, Jie Fu, Xiang Yue, and Wenhu Chen. Structlm: Towards building generalist models for structured knowledge grounding, 2024. URL https://arxiv.org/abs/2402.16671.

A APPENDIX

A.1 KG-FIRST PROTOCOL

Algorithm 1 Probe-guided triple conversion in KG–LLM collaboration

Require: Type II triple (s, r, g), KG, dense-neighborhood criterion DENSE
Ensure: Candidate instantiations for g

$N \leftarrow \text{LINKTOKG}(s)$
Candidate_KG $\leftarrow \{e \mid (N, r', e) \in \text{KG}\}$
if $\text{DENSE}(N)$ **then**
  $P \leftarrow \text{LLMPROBE}(s, r, g)$ {KG-local acceleration}
  $C \leftarrow P \cap$ Candidate_KG
  if $C = \emptyset$ **then**
    $C \leftarrow$ Candidate_KG {safe fallback}
  end if
else
  $C \leftarrow$ Candidate_KG
end if
$C \leftarrow \text{CROSSTRIPLECONSISTENCY}(C, g)$
if C contains a viable continuation **then**
  return C {KG-local commitment}
end if
$P \leftarrow \text{LLMPROBE}(s, r, g)$ {dead-end bridging}
for all $p_e \in P$ **do**
  if $p_e \in$ KG **then**
    conditionally link N to p_e as an existence-grounded bridge
  else
    retain p_e only as a provisional virtual bridge
  end if
end for
retract provisional links that fail later consistency checks
return trusted continuations

Table 7: Grounding status of probe-assisted continuations. Entity lookup and support for a query relation are separate checks.

Use of probe	Available KG support	Remaining check
Local acceleration	Candidate belongs to the anchor's KG neighborhood.	Relation relevance and incident query constraints.
Existence-grounded bridge	Candidate exists elsewhere in the KG; the proposed link is tentative.	Support for the proposed relation and consistency of the continuation.
Virtual bridge	No matching KG entity is found; the endpoint and link are provisional.	Subsequent evidence for retaining or retracting the continuation.

Table 7 summarizes the evidence available at each entry point. Bridging is considered only after the configured local exploration fails. Entity existence supports grounding of an endpoint; it does not by itself validate the relation proposed by the probe. This distinction separates the gate for exploration from the later decision to retain a continuation.

A.2 PROMPT TEMPLATES

In this subsection, we introduce all prompt templates used in this paper. The complete prompt 8 and structure prompt 9 are employed for constructing the guidance graph. The internal prompt 10 is used in GGoT to mine the internal knowledge of the LLM, while the specific_internal prompt 11 serves cross-triple semantic consistency checks. The prune prompt is 12 applied for semantic consistency filtering, and answer_by_triples 13 is designed for generating answers based on knowledge chains.

Table 8: Prompt for completing and decomposing a question (`complete_prompt`).

complete	Rephrase the question as a statement, ensuring: 1. All explicit and logically implied information is included (e.g., location, time, scope if hinted in the question). 2. Split compound noun phrases into individual entities. 3. Classify each entity as generic/specific. Do not answer the question. question: What team did Payton Manning's father play for? answer: statement: Identify the team Payton Manning's father played for. entities: - team (generic) - Payton Manning (specific) - father (generic) question: what did James K. Polk do before he was president? answer: statement: Identify the job of James K. Polk before he became president. entities: - job (generic) - James K. Polk (specific) - president (generic) question: Where did the "Country Nation World Tour" concert artist go to college? answer: statement: Identify the college attended by the concert artist of the "Country Nation World Tour." entities: - college (generic) - Country Nation World Tour (specific) - concert artist (generic) Now answer with the format of the example above. Be brief and precise. question: {} answer:

Table 9: Prompt for constructing the guidance graph (`structure_prompt`).

structure	Please analyze the relationships between the following entities in the given sentence. Represent each relationship as a triple in the format (subject, relation, object). Important Rules: - Do not alter the provided keywords. Use them exactly as given (no paraphrase, no pluralization). You may add other words only if they appear in the sentence. - If a generic keyword and another keyword refer to the same entity, the generic keyword must be treated as a relation rather than an entity. - Ensure the triples are logically connected. sentence: Identify the father of Keyshia Cole. generic keywords: father specific keywords: Keyshia Cole answer: triples: [(father, of, Keyshia Cole)] sentence: Identify the main trading partner of China that appointed Abdelaziz Bouteflika to a governmental position. generic keywords: main trading partner; governmental position specific keywords: China; Abdelaziz Bouteflika answer: triples: [(main trading partner, of, China), (main trading partner, appointed, Abdelaziz Bouteflika), (Abdelaziz Bouteflika, appointed to, governmental position)] Now answer with the format of the example above. Be brief and precise. sentence: {} generic keywords: {} specific keywords: {} answer:

Table 10: Prompt for generating candidate entities (`internal_prompt`).

internal	Task: Infer the entity that fills the target slot in the triples using only the semantics of the relation and general world knowledge. Critical rules: - non_target_context contains information about the other side of the triple (the complementary entity), NOT the target. - Never output any string that appears as a key or value in non_target_context. If you include any item from non_target_context in your answer, it is wrong. - Output concrete, specific entities that can occupy the target slot (e.g., for target=award: award names; for target=year: a 4-digit year). - Multiple correct answers may exist; list the most likely 1–5. - If you cannot reasonably infer any specific target, output [] - Output format must be exactly: [answer1, answer2, ...] with no other text. Examples: triples: [(Marie Curie, winner of, award)] target slot: award non_target_context: {} answer: [Nobel Prize] triples: [('team', 'lastWonWorldSeries', 'year')] target slot: year non_target_context: {{'team': 'San Francisco Giants'}} answer: [2014] triples: [('director', 'directed', 'film')] target slot: film non_target_context: {{'director': 'Greta Gerwig'}} answer: [Lady Bird, Little Women] triples: [('author', 'wrote', 'book')] target slot: book non_target_context: {{'author': 'Haruki Murakami'}} answer: [Norwegian Wood, Kafka on the Shore] Now answer this: triples: {triples} target slot: {target_entity} non_target_context: {previous_knowledge} answer:

Table 11: Prompt for cross-triple consistency checking (`specific_internal_prompt`).

specific_internal	Please rely on your own knowledge to find the answer to the target entity based on the given triples and previous knowledge. There may be more than one answer. Here are some examples. triples: [(Marie Curie, winner of, award)] previous knowledge: {{award: [Nobel Prize, Fields Medal, Turing Award]}} answer: [Nobel Prize] triples: [(programming language, named after, comedy group)] previous knowledge: {{programming language: [Python, C++]}} answer: [Python] Now answer with the format of the example above. triples: {triples} previous knowledge: {previous_knowledge} answer:

Table 12: Prompt for pruning relation candidates (`prune_prompt`).

prune	Select the most semantically matching relation from the candidate relations to replace the given relation in the theme information while maintaining its essential meaning. Rate it on a scale of 0 to 10. Be brief and precise.
	
	Below is an example:
	theme information: main spoken language of country
	relation: of
	candidate relations: {
	'language.human_language.main_country',
	'language.human_language.human_language',
	'language.human_language.language_family',
	'language.human_language.iso_639_3_code',
	'base.rosetta.languoid.parent',
	'language.human_language.writing_system',
	'base.rosetta.languoid.languoid_class'
	}
	assess:
	- 'language.human_language.human_language' can perfectly replace the relation, (9) score.
	- 'language.human_language.main_country' can replace the relation, (8) score.
	- other relations are not related to any information in the sentence, (0) score.
	
	Now answer with the format of the example above. Be brief and precise.
	theme information: {}
	relation: {}
	candidate relations: {}
	assess:

Table 13: Prompt for answer generation from retrieved triples (`answer_by_triples_prompt`).

answer_by_triples	Please answer the question. Triples are available for reference. If there is not enough information in the triples, please answer with your own knowledge. question: Which place is the madam satan located? triples: [('madam satan', 'film.film.country', 'the USA'), ('madam satan', 'film.film.language', 'English')] answer: 'madam satan' is located in 'the USA'. question: {} triples: {} answer:

A.3 Guidance Graph Construction Rules

1. Specific keywords exclusively serve as graph nodes (i.e., entities in triples) rather than edges.
2. When a generic keyword co-refers to the same entity as a specific keyword, it functions as a triple's relation rather than an entity.
3. For two distinct generic keywords referencing the same entity, one must be assigned as the relation in the triple.
4. For associated keywords referring to different entities, we construct triples where the keywords become head/tail entities and their association forms the relational edge.

A.4 Impact of Model Scale and Family

Table 14 reports existing CWQ results with the Qwen2.5-instruct series and Qwen3-32B. Within Qwen2.5, complete match increases from 59.6% at 7B to 61.0% at 32B and 63.9% at 72B. Qwen3-32B scores 61.9%, compared with 61.0% for Qwen2.5-instruct-32B. These results extend the evaluation across model sizes and families. Since the cross-family comparison changes more than pretraining data volume, the 0.9-point difference does not isolate the effect of internal knowledge density.

Table 14: Performance comparison across different model sizes and families on the CWQ dataset. "LLM Call" denotes the average number of calls per query. Token metrics (Input/Output/Total) represent the average token usage per query.

Model	**Params (Size)**	**Complete Match (%)**	**LLM Call**	**Input Tokens**	**Output Tokens**	**Total Tokens**
Qwen2.5-instruct-7B	7B	59.6	9.50	3379.90	389.30	3769.20
Qwen2.5-instruct-32B	32B	61.0	9.19	3348.53	581.90	3930.43
Qwen2.5-instruct-72B	72B	**63.9**	9.40	4644.70	1201.00	5845.70
Qwen3-32B	32B	61.9	9.80	3610.15	668.58	4278.73

A.5 QUALITATIVE CASE STUDIES

Table 15: **Case study 1:** Reasoning process for a complex query involving dense subgraphs. The comparison highlights our method's efficiency in handling "noisy" nodes compared to baseline pruning.

Steps	Content
Question	What country bordering France contains an airport that serves Nijmegen?
Guidance Graph Construction	Guidance Graph: (country, borders, France), (country, contains, airport), (airport, serves, Nijmegen) **Specific entities:** France; Nijmegen **Note:** France is retrievable in the KG and used as the starting point.
First Round	**Baseline Method:** Retrieves all relations associated with France (498 in total), randomly selects a subset, and uses an LLM for pruning. **Our Method:** Triplet to parse: (country, borders, France) **Probe Generation:** [Germany, Spain, Italy, Belgium, Switzerland] **KG verification & connectivity check:** France is connected to the probes via the path `location.location.adjoin_s` → `location.adjoining_relationship.adjoins`. **Candidate acquisition:** All nodes connected by the same relation path are collected as country candidates: ['Germany', 'United Kingdom', 'Italy', ..., 'Bay of Biscay', 'Dover'] (19 nodes in total).
Second Round	**Baseline Method:** Retrieves relations associated with the previous round's nodes, randomly selects a subset, and uses an LLM for pruning. **Our Method:** Triplet to parse: (country, contains, airport) **Probe Generation:** [Heathrow Airport, Charles de Gaulle Airport, Frankfurt Airport, Schiphol Airport, Munich Airport] **KG verification & connectivity check:** The nodes from the previous round are connected to the probes via `location.location.contains`. **Candidate acquisition:** 15,184 nodes connected by the same relation path are collected as airport candidates.
Third Round	**Baseline Method:** Retrieves relations associated with the previous round's nodes and uses an LLM for pruning. **Our Method:** Triplet to parse: (airport, serves, Nijmegen) **Credible Knowledge Disambiguation:** Triggered because Nijmegen is a specific entity. **Disambiguation process:** Among the 15,184 candidate nodes from the previous round, only Weeze Airport is connected to Nijmegen via `aviation.airport.serves`. All other unconnected nodes are discarded.
Knowledge Path	('France', `'location.location.adjoin_s→location.adjoining_relationship.adjoins'`, 'Germany') ∧ ('Germany', `'location.location.contains'`, 'Weeze Airport') ∧ ('Weeze Airport', `'aviation.airport.serves'`, 'Nijmegen')
Answer	Germany

Analysis: In this case, the underlying subgraph is extremely dense. The first reasoning step involves 498 distinct relations from France, making full pruning computationally prohibitive. Our method circumvents this by leveraging the LLM's internal knowledge to generate targeted probes (e.g., neighboring countries like Germany), which rapidly identify the unique relevant relation path without exhaustive search. Finally, our Credible Knowledge Disambiguation mechanism filters 15,184 airport candidates down to a single node—Weeze Airport—that actually serves Nijmegen.

Table 16: **Case study 2:** Correct KG evidence recovered despite noisy probes. This trace illustrates retrieval recovery, not a controlled missing-edge experiment.

Steps	Content
Question	Where did the "Country Nation World Tour" concert artist go to college?
Guidance Graph	Guidance Graph: (concert artist, of, country nation world tour), (concert artist, attended, college)
Construction	**Specific entities:** country nation world tour **Note:** country nation world tour is retrievable in the KG and serves as the starting point.
First Round	**Triple to parse:** (concert artist, of, country nation world tour) **Probe Generation:** [Taylor Swift, Luke Bryan, Carrie Underwood, Blake Shelton, Keith Urban] **Verify probes in KG & check connectivity:** country nation world tour has no connecting edge to the probes. **Semantic consistency filtering:** Among all 17 relations, `music.concert_tour.artist` is semantically consistent with the triple to parse. **Get all nodes connected via the same relation:** `music.concert_tour.artist` connects to ['Brad Paisley']. **KG-existing probes also serve as candidate set:** ['Taylor Swift', 'Luke Bryan', 'Carrie Underwood', 'Blake Shelton', 'Keith Urban'] (KG existence does not establish a tour–artist connection).
Second Round	**Triple to parse:** (concert artist, attended, college) **Probe Generation:** [Belmont University, UNC Chapel Hill, Oklahoma State, Lipscomb University, UCO] **Verify probes in KG & check connectivity:** Only Brad Paisley connects to the probe Belmont University via `people.person.education`→`education.education.institution`. **Get 3 nodes from the same relation path:** ['Belmont University', 'West Liberty University', 'John Marshall High School']
Knowledge Path	('Country Nation World Tour', '`music.concert_tour.artist`', 'Brad Paisley'), ('Brad Paisley', '`people.person.education`→`education.education.institution`', 'Belmont University'), ('Brad Paisley', ..., 'West Liberty University')
Reference answer	Belmont University
Reported output	Belmont University, West Liberty University

Analysis: The initial artist probes do not connect to the tour. Relation matching nevertheless retrieves Brad Paisley, and subsequent KG exploration finds Belmont University. Noisy candidates can coexist with correct evidence; this trace does not establish that every wrong probe was removed. The output contains the reference answer and an additional institution, so reference-answer coverage should be distinguished from exact answer-set correctness. This case supports recovery through an existing KG path, not the repair of a missing edge.