GROUNDING IS NOT ENOUGH: TESTING EVIDENCE SUFFICIENCY IN LLM REASONING OVER KNOWLEDGE GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Grounded large language model (LLM) systems are expected to justify predictions with evidence, making answer correctness alone insufficient. Knowledge graph question answering (KGQA) provides a concrete setting for testing this requirement since returned graph evidence can be checked against the conditions needed to support a prediction. Yet existing KGQA evaluations largely emphasize answer correctness, leaving two questions insufficiently examined: whether the returned evidence sufficiently supports the prediction, and whether systems return UNKNOWN when the available knowledge cannot support an answer. We introduce an executable evaluation protocol that derives constraints from structured queries, verifies whether the returned evidence satisfies those constraints, and uses controlled graph edits to test system behavior after answer-related evidence is removed. We apply this protocol on *WebQSP*, *CWQ*, and *2WikiMultiHopQA*. Across existing grounded LLM-based KGQA systems, the required supporting evidence is unavailable in up to 76.77% of cases, while systems correctly return UNKNOWN in only 0.00–30.53% of cases where support is unavailable. After the removal of answer-critical evidence, 53.98% of initially correct predictions still persist, whereas only 18.31% transition to UNKNOWN. These gaps persist even with Codex (GPT-6), suggesting that stronger language models alone do not eliminate the problem. Motivated by these findings, we introduce Grounding with Requirement-Aware Shared-binding Preservation (GRASP) as a reference mitigation that preserves question constraints, reducing the EVIDENCEGAP to 27.47%, improving the correct UNKNOWN up to 88.01%. Overall, our results reveal a key limitation of answer-based evaluation for grounded LLMs: it does not capture whether returned evidence sufficiently supports an answer or whether models abstain when such support is unavailable.

1 INTRODUCTION

Grounded large language model (LLM) systems (Zhang et al., 2022; Gao et al., 2023; Hsu et al., 2024) are expected to justify predictions with designated external evidence, e.g., retrieved documents, or knowledge graphs. In clinical or access-controlled workflows that require such justification, memorized knowledge cannot substitute for a required laboratory result or an authorized record. Reliable evidence-bounded answering therefore requires both supporting answers from the permitted source and recognizing when that source is insufficient.

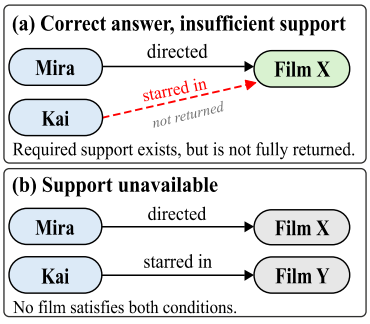


Figure 1. Question for *Which film did Mira direct and Kai star in?* **(a)** Film X satisfies both question constraints, but returned evidence supports one of them, yielding insufficient support for the answer. **(b)** No single film satisfies both constraints in the available knowledge, so sufficient support is unavailable and the system should return UNKNOWN.

However, answer correctness and the validity of individual evidence items do not establish sufficient support for the answer (Gao et al., 2023; Ru et al., 2024; Joren et al., 2025). Prior studies have documented related gaps between correctness and faithfulness, as well as failures under insufficient or altered context (Nguyen et al., 2024; Ming et al., 2025; Matton et al., 2025). These issues become particularly concrete in knowledge graph question answering (KGQA), where evidence can be

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.

represented as explicit graph paths or subgraphs and checked against relational requirements (see Figure 1). KGQA therefore provides an executable setting for studying evidence-bounded answering: queries make answer requirements testable, while controlled graph edits allow answer-critical evidence to be removed. Despite these advantages, two properties remain insufficiently evaluated: *among predictions counted as correct, does the returned evidence provide sufficient support for the answer, and does the system return* UNKNOWN *when sufficient support is unavailable?*

To operationalize these two properties, *we exploit the executable structure of KGQA and introduce an evaluation protocol that derives question constraints from structured queries and compiles them into executable checks.* These checks require all conditions for an answer to be jointly satisfied, allowing us to determine whether the returned evidence sufficiently supports a prediction. We then remove evidence critical to the answer and apply the same checks to the remaining input knowledge to verify that sufficient support has been fully removed.

We instantiate the protocol on WebQSP (Yih et al., 2016), CWQ (Talmor & Berant, 2018), and 2WikiMultiHopQA (Ho et al., 2020). For questions where the input knowledge contains sufficient evidence for a gold answer, we introduce EVIDENCEHIT@1, which measures correct answers accompanied by sufficient returned evidence, and EVIDENCEGAP, which quantifies how often correct answers lack sufficient returned evidence. For questions where sufficient support is unavailable, we evaluate whether systems appropriately return UNKNOWN. We further conduct paired graph interventions that remove answer-critical evidence and verify that no alternative support remains, allowing us to measure whether initially correct predictions persist, switch to another unsupported answer, or transition to UNKNOWN. Together, these measures test whether systems return sufficient evidence when answering and return UNKNOWN when sufficient evidence is unavailable.

Using this protocol, we evaluate existing grounded LLM systems and uncover EVIDENCEGAP values of up to 76.77%, showing that correct predictions are not accompanied by sufficient returned evidence. When sufficient support is unavailable, these systems return UNKNOWN correctly on only 0.00—30.53% of questions, revealing substantial failures to recognize unsupported cases. To mitigate these failures, we introduce Grounding with Requirement-Aware Shared-binding Preservation (GRASP), a training-free approach that operationalizes an explicit support specification through relation guidance, binding-consistent grounding, and specification-aware answering. GRASP reduces EVIDENCEGAP to 27.47% and improves correct UNKNOWN responses up to 88.01%, demonstrating that explicit support specifications and binding-consistent grounding can improve both evidence sufficiency and behavior under insufficient evidence.

In summary, our contributions are listed as follows:

- We introduce an executable evaluation protocol for grounded LLM systems that goes beyond answer correctness to assess whether returned evidence is sufficient and whether systems return UNKNOWN when sufficient support is unavailable (Sections 3 and 4).
- We conduct a large-scale evaluation of existing grounded LLM KGQA systems across multiple testbeds, uncovering EVIDENCEGAP up to 76.77% and substantial failures to return UNKNOWN when sufficient support is unavailable (Sections 5.1 and 5.2).
- We introduce GRASP, a training-free reference mitigation based on relation guidance, binding consistency, and specification-aware answering, and show that it substantially improves evidence-bounded behavior in grounded LLM systems (Section 5.3).
- Our study shows that answer correctness alone can substantially overstate grounded performance, providing a new lens for evaluating whether grounded LLM systems actually rely on sufficient support rather than merely produce correct answers (Section 5).

2 RELATED WORK

Evaluation of grounded LLM systems increasingly goes beyond answer correctness, i.e., whether answers exactly match real-world facts, to assess faithfulness, i.e., whether answers are supported by provided knowledge (Gao et al., 2023; Ru et al., 2024; Adlakha et al., 2024). Recent work studies information sufficiency (Joren et al., 2025; Jain & Garimella, 2026; Li et al., 2026), faithfulness under altered or unanswerable contexts (Ming et al., 2025; Matton et al., 2025), and abstention (Zhou et al., 2026b). These directions motivate evaluating not only whether a model is correct, but also whether the available evidence can support the answer.

Table 1. Comparison with existing work. **Evidence** indicates the use of explicit graph-based reasoning evidence, such as paths, subgraphs, etc. **Support** indicates assessment of whether the available evidence is sufficient to answer the question. **UNKNOWN** indicates support for or evaluation of abstention when evidence is insufficient.

Work	Evaluation target	Intervention target	Evidence	Support	UNKNOWN?
Sufficient Context (Joren et al., 2025)	Context sufficiency	–	✗	✓	✓
S2G-RAG (Li et al., 2026)	Retrieval sufficiency	–	✗	✓	✗
FaithEval (Ming et al., 2025)	Context faithfulness	Context perturbation	✗	✗	✓
Walk the Talk (Matton et al., 2025)	Explanation faithfulness	Concept perturbation	✗	✗	✗
When Silence Is Golden (Zhou et al., 2026b)	Abstention	–	✗	✗	✓
SubgraphRAG (Li et al., 2025b)	Subgraph grounding	–	✓	✗	✓
RoG (Luo et al., 2024)	Graph-valid reasoning	–	✓	✗	✗
GCR (Luo et al., 2025)	Graph-valid reasoning	–	✓	✗	✗
DoG (Li et al., 2025a)	Graph-valid reasoning	–	✓	✗	✗
RJE (Lin et al., 2025)	Path sufficiency	–	✓	✓	✗
CoR (Liu et al., 2026)	Graph-valid reasoning	–	✓	✗	✗
CogGRAG (Cheng et al., 2026)	Graph-valid reasoning	–	✓	✗	✓
STEM (Yu et al., 2026)	Evidence coverage	–	✓	✗	✗
VoG (Zhao et al., 2026)	Stepwise verification	–	✓	✗	✗
BRINK (Zhou et al., 2026a)	Incomplete-KG reasoning	Support preservation	✗	✗	✗
Our study	**Evidence-bounded QA**	**Support removal**	✓	✓	✓

Knowledge graphs provide an executable setting for this evaluation. Existing KGQA methods retrieve paths or subgraphs, enforce graph-valid traversal, improve evidence coverage, or verify intermediate reasoning, including RoG (Luo et al., 2024), ToG (Sun et al., 2024), SubgraphRAG (Li et al., 2025b), DoG (Li et al., 2025a), GCR (Luo et al., 2025), RJE (Lin et al., 2025), STEM (Yu et al., 2026), and VoG (Zhao et al., 2026). Our evaluation tests whether the returned evidence contains a support for a correct prediction and whether a system returns UNKNOWN when support is absent. BRINK (Zhou et al., 2026a) studies a complementary intervention that removes direct answer facts while preserving alternative support; our edits remove support and verify that none remains in the input knowledge. Table 1 summarizes these distinctions.

3 PROBLEM FORMULATION: EVIDENCE-BOUNDED KGQA

Grounded KGQA systems should not only predict the correct answer, but also establish that the answer is supported by the information they are permitted to use. We therefore formulate KGQA relative to a designated evidence source. For a question q, let G_q denote the fixed input knowledge, represented as a set of triples. A system returns either a prediction $\hat{y}$ with evidence $E \subseteq G_q$, or UNKNOWN. The input knowledge G_q defines the *evidence boundary*: only information within this designated source may support the prediction. We use *evidence* to refer to the available or returned triples and *support* to denote whether those triples jointly establish an answer.

To determine when evidence supports an answer requires, we derive a set of constraints $\Phi(q)$ from the question. These constraints capture the relations, filters, comparisons, aggregations, and binding dependencies expressed by the question. Constraints involving the same entity or value must be satisfied under a binding-consistent assignment. Returned evidence $E \subseteq G_q$ provides *sufficient support* for an answer a if it contains evidence that jointly satisfies every constraint in $\Phi(q)$:

$$\textsc{SufficientSupport}(a, q, E) \iff \exists E' \subseteq E : (E', a) \models \Phi(q). \tag{1}$$

Any such subset E' is a *support* for a. This definition allows the same executable criterion to be applied at two levels. Applying Equation 1 to G_q determines whether the input knowledge contains sufficient support for a gold answer, whereas applying it to E determines whether the system's prediction is sufficiently supported by its returned evidence. Let $A_q^\star$ denote the gold-answer set. We partition questions according to support availability in the input knowledge:

$$\begin{aligned} q \in \mathcal{S} &\iff \exists a \in A_q^\star : \textsc{SufficientSupport}(a, q, G_q), \\ q \in \mathcal{I} &\iff \nexists a \in A_q^\star : \textsc{SufficientSupport}(a, q, G_q). \end{aligned} \tag{2}$$

Accordingly, evidence-bounded answering requires a correct prediction with sufficient returned evidence on $\mathcal{S}$, and an UNKNOWN response on $\mathcal{I}$. Section 4 instantiates these definitions for each benchmark and introduces the corresponding evaluation measures.

4 STUDY DESIGN

In this section, we instantiate the preceding framework on three KGQA datasets.

Table 2. Benchmark composition by support availability.

Dataset	$\mathcal{S}$	$\mathcal{I}$	Total
WebQSP	1,335	286	1,621
CWQ	1,679	1,459	3,138
2WikiMultiHopQA	1,383	2,614	3,997

4.1 Dataset Construction

WebQSP contains questions over Freebase paired with SPARQL queries (Yih et al., 2016). CWQ extends WebQSP with questions involving function composition, conjunction, comparison, and superlatives (Talmor & Berant, 2018). 2WikiMultiHopQA contains questions constructed from Wikipedia and Wikidata, with evidence triples (Ho et al., 2020). For WebQSP and CWQ, we use the graph snapshots released by RoG (Luo et al., 2024) as G_q and derive $\Phi(q)$ from the official gold SPARQL queries. We replay each query over G_q, enforcing binding consistency and applying its filters, comparisons, and ordering operations; inverse relations declared in the GrailQA ontology are also accepted (Gu et al., 2021). For 2WikiMultiHopQA, we used the DoG graph snapshot (Li et al., 2025a; Ho et al., 2020) as G_q and derive $\Phi(q)$ from official evidence triples rather than SPARQL queries. Each required triple must align with an edge in G_q under official entity identities or aliases, with exact relation and direction. Appendix A.2 details the benchmark-specific mappings. Table 2 reports the split statistics for the evaluated datasets.

4.2 Systems and Metrics

Systems. We evaluate CoT (Wei et al., 2022), DoG (Li et al., 2025a), GCR* (Luo et al., 2025), and Codex (GPT-6) with *xhigh* reasoning efforts. We select these systems to span complementary reasoning paradigms: CoT provides a standard baseline, DoG and GCR represent graph-constrained KGQA, and Codex probes whether the observed failures persist with stronger reasoning and tool-assisted graph access. CoT, DoG, and GCR* are evaluated with Qwen2.5-7B, Gemma-2-9B, and Llama-3.1-8B, whereas Codex uses GPT-6 with *xhigh* reasoning effort. GCR* augments the original answer-only GCR with GPT-OSS-120B to select supporting paths for evidence evaluation For each question, all systems receive the same input knowledge G_q. We construct the returned evidence set E from the graph triples explicitly provided by each system as evidence for its prediction.

Table 3. Evaluation metrics. Arrows indicate the preferred direction.

Metric	Meaning
Support available ($\mathcal{S}$)	
Hit@1 ↑	Correct answer.
E-Hit ↑	Correct with sufficient evidence.
EvidenceGap ↓	Correct without sufficient evidence.
$\mathrm{OAR}_{\mathcal{S}}$ ↓	Unnecessary Unknown.
Support unavailable ($\mathcal{I}$)	
$\mathrm{CAR}_{\mathcal{I}}$ ↑	Correct Unknown.
All questions ($\mathcal{S} \cup \mathcal{I}$)	
EBA ↑	Supported answer or correct abstention.
After support removal	
APR ↓	Original answer retained.
ATR_o	Transition to outcome o.

Supported answering. For question q_i, let $\hat{a}_i$, E_i, and $A_i^\star$ denote the prediction, returned evidence, and gold-answer set. On $\mathcal{S}$, Hit@1 measures answer accuracy, while Evidence-Hit@1 (E-Hit) measures the fraction of questions answered correctly with sufficient returned evidence:

$$\text{E-Hit} = \frac{1}{|\mathcal{S}|} \sum_{i:q_i \in \mathcal{S}} \left[\begin{array}{c} \hat{a}_i \in A_i^\star \wedge \\ \textsc{SufficientSupport}(\hat{a}_i, q_i, E_i) \end{array} \right]. \tag{3}$$

Among correct predictions on $\mathcal{S}$, EvidenceGap is the fraction whose returned evidence lacks sufficient support:

$$\Delta = \textsc{EvidenceGap} = 1 - \frac{\text{E-Hit}}{\text{Hit@1}}. \tag{4}$$

A lower EvidenceGap indicates closer alignment between answer correctness and evidence sufficiency; a value of zero means that every correct prediction is accompanied by sufficient returned evidence. It is undefined when Hit@1 $= 0$.

Abstention and overall accuracy. The *correct abstention rate* ($\mathrm{CAR}_{\mathcal{I}}$) and *over-abstention rate* ($\mathrm{OAR}_{\mathcal{S}}$) measure the fractions of questions answered with Unknown on $\mathcal{I}$ and $\mathcal{S}$, respectively:

$$\mathrm{CAR}_{\mathcal{I}} = \frac{1}{|\mathcal{I}|} \sum_{i:q_i \in \mathcal{I}} [\hat{a}_i = \textsc{Unknown}], \tag{5}$$

$$\mathrm{OAR}_{\mathcal{S}} = \frac{1}{|\mathcal{S}|} \sum_{i:q_i \in \mathcal{S}} [\hat{a}_i = \textsc{Unknown}]. \tag{6}$$

A higher $\mathrm{CAR}_{\mathcal{I}}$ indicates better recognition of unavailable support, whereas a lower $\mathrm{OAR}_{\mathcal{S}}$ indicates fewer unnecessary abstentions.

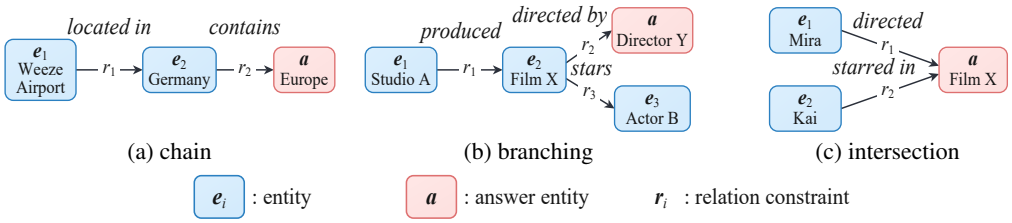


Figure 2. Question structures.

Evidence-Bounded Accuracy (EBA) is the fraction of questions in $\mathcal{S} \cup \mathcal{I}$ with either a correct prediction supported by returned evidence on $\mathcal{S}$ or an UNKNOWN response on $\mathcal{I}$:

$$\text{EBA} = \frac{|\mathcal{S}| \cdot \text{E-HIT} + |\mathcal{I}| \cdot \text{CAR}_{\mathcal{I}}}{|\mathcal{S}| + |\mathcal{I}|}. \tag{7}$$

Thus, EBA measures overall performance under the evidence boundary, combining supported answering with appropriate abstention.

4.3 EVIDENCE-SUFFICIENCY ANALYSIS

To characterize evidence-sufficiency failures, we conduct two analyses on CWQ-$\mathcal{S}$. First, we stratify questions by constraint count and structural pattern to examine how evidence sufficiency varies with question complexity. Second, we perform a semantic-relaxation analysis to assess the sensitivity of EVIDENCEGAP to relation matching.

Question complexity and structure. We characterize each question by its number of constraints and the structure induced by its gold query. Constraint count is the number of positive relational conditions that must be jointly satisfied. We additionally consider three common structures illustrated in Figure 2: *chain*, *branching*, and *intersection*.

Relaxed graph-relation matching. For CWQ, our evaluation constraints are derived from the gold SPARQL query, which specifies the exact graph relations expected by the benchmark. The evaluated systems, however, retrieve evidence from the graph based only on the natural-language question. A system may therefore select a graph relation that is semantically similar to, but not exactly the same as, the graph relation specified in the gold SPARQL query. Under exact matching, evidence containing such a relation would be treated as unsupported. To test whether this mismatch contributes to EVIDENCEGAP, we relax only the graph-relation matching criterion while keeping the CWQ-$\mathcal{S}$ questions, predictions, and returned evidence fixed. In addition to exact matches, we accept pairs of graph relations whose `gte-large-en-v1.5` (Zhang et al., 2024) cosine similarity is at least τ.

4.4 SUPPORT-REMOVAL INTERVENTIONS

We test how systems respond when sufficient support is removed while keeping the question fixed. We focus on CWQ since its multi-hop questions enable targeted and progressive support-removal interventions. We select questions for which DoG, GCR*, and Codex all initially return a correct prediction with sufficient returned evidence. For each question, we remove evidence from G_q to construct G_q^- and retain only interventions for which gold-query replay confirms that support exists in G_q but not in G_q^-. This yields 295 condition-loss pairs and 120 binding-mismatch pairs, shared across all evaluated systems.

We consider two complementary interventions. *Constraint loss* removes the minimum number of supporting triples needed to make one constraint in $\Phi(q)$ unsatisfied while keeping the answer entity in the graph; for queries with alternative branches, we ensure that no branch still provides sufficient support. *Binding mismatch* targets conjunctive questions by removing triples so that each constraint remains individually satisfiable, but no binding-consistent assignment satisfies all constraints jointly. Figure 3 illustrates the two intervention types.

Answer transitions. For each edited instance, let $\hat{a}_i$ and $\hat{a}_i^-$ denote the system predictions on G_i and G_i^-, respectively. We categorize the outcome as SAME if the original answer persists, UNKNOWN if the system abstains, and OTHER otherwise. For outcome $o \in \{\text{SAME}, \text{OTHER}, \text{UNKNOWN}\}$, the

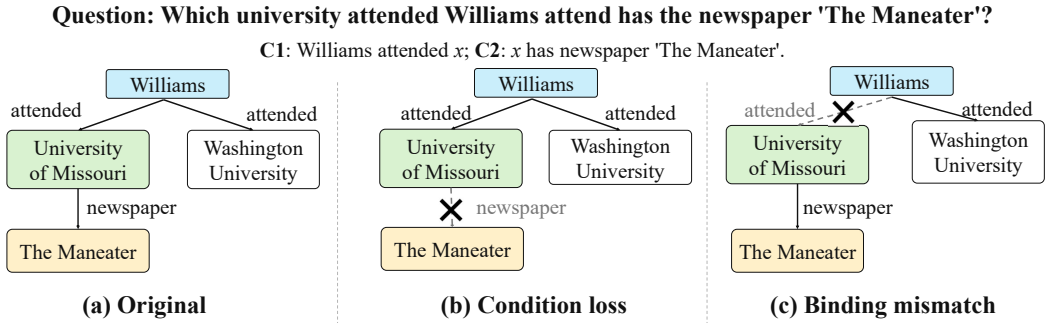


Figure 3. Two interventions to remove support for the same question. **(a)** The original graph contains sufficient support: University of Missouri satisfies both constraints. **(b) Condition loss:** removing its newspaper edge leaves C_2 unsatisfied. **(c) Binding mismatch:** removing Williams's attendance edge to University of Missouri leaves individual matches for both C_1 and C_2, but no single university satisfies them jointly.

answer transition rate is $\mathrm{ATR}_o = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\left[\tau(\hat{a}_i, \hat{a}_i^-) = o\right]$, where N is the number of evaluated edits and

$$\tau(\hat{a}_i, \hat{a}_i^-) = \begin{cases} \textsc{Same}, & \hat{a}_i^- = \hat{a}_i, \\ \textsc{Unknown}, & \hat{a}_i^- = \textsc{Unknown}, \\ \textsc{Other}, & \text{otherwise}. \end{cases}$$

We define the *answer persistence rate* as $\mathrm{APR} = \mathrm{ATR}_{\textsc{Same}}$. The transition rates distinguish whether a system preserves its original answer, switches to another answer, or abstains after sufficient support is removed.

5 Main Results

We organize the results around three questions. RQ1 asks whether the evidence returned by the system actually supports correct predictions. RQ2 then tests how predictions change when sufficient support is unavailable or deliberately removed. Finally, RQ3 tests whether explicit support specifications can improve evidence-bounded behavior in RQ1 and RQ2.

5.1 RQ1: Among correct predictions, does the returned evidence support correct predictions?

Results and Analysis. EvidenceGap is substantially large in the existing grounded LLM system. As shown in Table 4, EvidenceGap frequently exceeds 50%, reaching 76.77% for CoT, 57.83% for GCR*, and 54.46% for DoG. The gap remains substantial at high answer accuracy, e.g, GCR* with Llama-3.1-8B achieves 73.32% Hit@1 but only 34.19% E-Hit on CWQ-$\mathcal{S}$. Even Codex (GPT-6) retains a nonzero gap. Thus, answering correctly does not ensure that the returned evidence sufficiently supports the prediction.

Table 4. Answer accuracy and evidence-supported accuracy on data split $\mathcal{S}$ (%).

Model	Method	WebQSP			CWQ			2WikiMultiHopQA		
		Hit@1 ↑	E-Hit ↑	Δ ↓	Hit@1 ↑	E-Hit ↑	Δ ↓	Hit@1 ↑	E-Hit ↑	Δ ↓
Qwen2.5-7B	CoT	59.10	15.43	73.89	63.73	27.99	56.07	46.49	14.03	69.83
	GCR*	86.37	66.44	23.07	72.13	31.33	56.56	75.20	36.59	51.35
	DoG	73.11	56.10	23.26	61.94	30.73	50.38	87.13	79.68	8.55
Gemma-2-9B	CoT	43.52	11.01	74.70	49.97	13.70	72.59	32.68	7.59	76.77
	GCR*	86.59	66.22	23.53	72.25	33.00	54.33	73.90	31.16	57.83
	DoG	62.77	49.06	21.84	58.61	30.73	47.56	83.73	70.93	15.28
Llama-3.1-8B	CoT	54.16	23.90	55.88	46.81	25.13	46.31	39.91	15.11	62.14
	GCR*	88.54	67.79	23.43	73.32	34.19	53.37	76.07	34.92	54.09
	DoG	66.22	53.71	18.89	57.42	26.15	54.46	84.16	74.91	11.00
Codex (GPT-6)	Agent	84.12	74.76	11.13	89.52	67.30	24.82	97.47	96.75	0.74

EvidenceGap increases with question complexity. On CWQ-$\mathcal{S}$, the gap generally grows with constraint count (Figure 4), reaching 94.87% for GCR* and 97.22% for DoG on questions with five or more constraints. Branching questions show the largest gaps across systems, indicating that

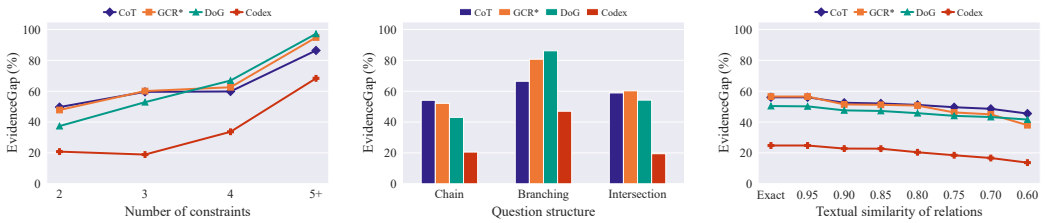


Figure 4. EVIDENCEGAP on CWQ-$\mathcal{S}$ by constraint count (left), question structure (center), and textual similarity of relations (right). CoT, GCR*, and DoG use Qwen2.5-7B; Codex uses GPT-6 xhigh.

evidence sufficiency becomes particularly challenging when multiple constraints must be satisfied under consistent bindings.

EVIDENCEGAP persists under semantic relaxation. Relaxing textual simiarity of relations reduces the gap only modestly (Figure 9 in Appendix A.5). Even at $\tau = 0.60$, substantial gaps remain, indicating that variation in relation wording cannot explain the observed mismatch between answer correctness and evidence sufficiency.

> **RQ1 Takeaway.** EVIDENCEGAP reveals that correct predictions can lack evidence that jointly satisfies the question constraints, especially on high-constraint and branching questions. This mismatch persists under relaxed semantic relation matching.

5.2 RQ2: DO GROUNDED LLMS RESPECT THE EVIDENCE BOUNDARY WHEN SUPPORT IS UNAVAILABLE?

Table 5. Evidence-bounded decisions on the $\mathcal{S}/\mathcal{I}$ splits (%).

Model	Method	WebQSP			CWQ			2WikiMultiHopQA		
		$\text{CAR}_{\mathcal{I}}\uparrow$	$\text{OAR}_{\mathcal{S}}\downarrow$	EBA ↑	$\text{CAR}_{\mathcal{I}}\uparrow$	$\text{OAR}_{\mathcal{S}}\downarrow$	EBA ↑	$\text{CAR}_{\mathcal{I}}\uparrow$	$\text{OAR}_{\mathcal{S}}\downarrow$	EBA ↑
Qwen2.5-7B	CoT	0.35	0.15	12.77	0.00	0.00	14.98	0.42	0.00	5.13
	GCR*	5.59	1.35	55.71	11.58	12.63	22.15	27.31	7.52	30.52
	DoG	4.90	0.37	47.07	0.75	0.12	16.79	9.41	0.00	33.73
Gemma-2-9B	CoT	0.70	0.15	9.19	0.00	0.00	7.33	0.65	0.00	3.05
	GCR*	5.59	0.82	55.52	11.93	13.52	23.20	30.53	9.91	30.75
	DoG	11.19	0.97	42.38	0.07	0.00	16.48	18.90	0.43	36.90
Llama-3.1-8B	CoT	0.00	0.00	19.68	0.00	0.00	13.45	0.04	0.00	5.25
	GCR*	5.94	1.20	56.88	11.45	12.69	23.61	26.43	8.24	29.37
	DoG	0.35	0.00	44.29	0.00	0.00	13.99	4.17	0.14	28.65
Codex (GPT-6)	Agent	45.10	9.21	69.52	88.14	7.50	76.99	62.47	0.43	74.33

Table 6. Prediction responses to support removal on CWQ (%).

Model	Method	Answer persistence (APR)		Transitions after support removal (ATR)		
		Condition loss ↓	Binding mismatch ↓	SAME ↓	UNKNOWN ↑	OTHER ↓
Qwen2.5-7B	CoT	46.10	63.33	51.08	0.24	48.67
	GCR*	55.25	50.83	53.98	18.31	27.71
	DoG	53.56	50.00	52.53	3.37	44.1
Gemma-2-9B	CoT	63.39	62.50	63.13	0.00	36.87
	GCR*	52.54	47.50	51.08	18.31	30.60
	DoG	59.66	61.67	60.24	5.78	33.98
Llama-3.1-8B	CoT	61.36	72.50	64.58	0.00	35.42
	GCR*	55.59	49.17	53.73	16.39	29.88
	DoG	51.86	55.00	52.77	0.72	46.51
Codex (GPT-6)	Agent	8.81	20.83	12.29	85.30	2.41

Results and Analysis. Grounded LLMs often fail to return UNKNOWN when support is unavailable. On CWQ-$\mathcal{I}$, CoT, GCR*, and DoG rarely abstain despite the absence of sufficient support (Table 5). Across the three models, GCR* reaches only 11.45–11.93% $\text{CAR}_{\mathcal{I}}$, while CoT and DoG are at or near zero. Moreover, GCR* abstains at similar rates on $\mathcal{S}$ and $\mathcal{I}$, indicating that its abstention behavior is weakly aligned with support availability. Codex (GPT-6) performs substantially better, but still does not eliminate the failure across datasets.

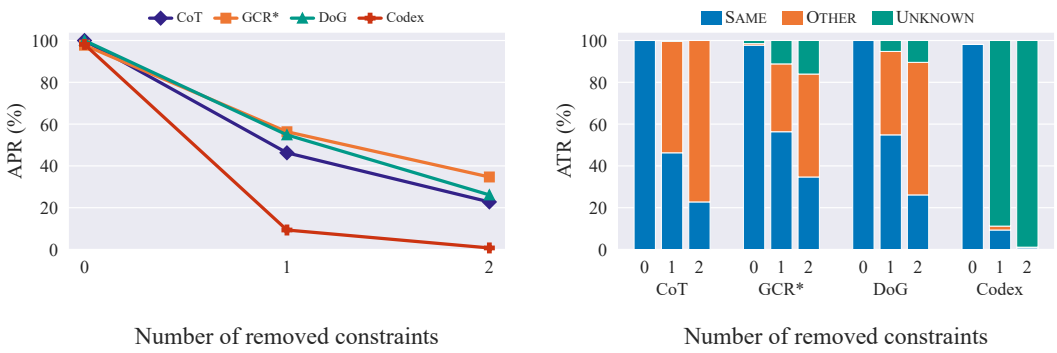


Figure 5. Prediction responses under progressive support removal on CWQ. Left: Answer persistence (APR). Right:SAME/OTHER/UNKNOWN transition rates. k denotes the number of removed constraints.

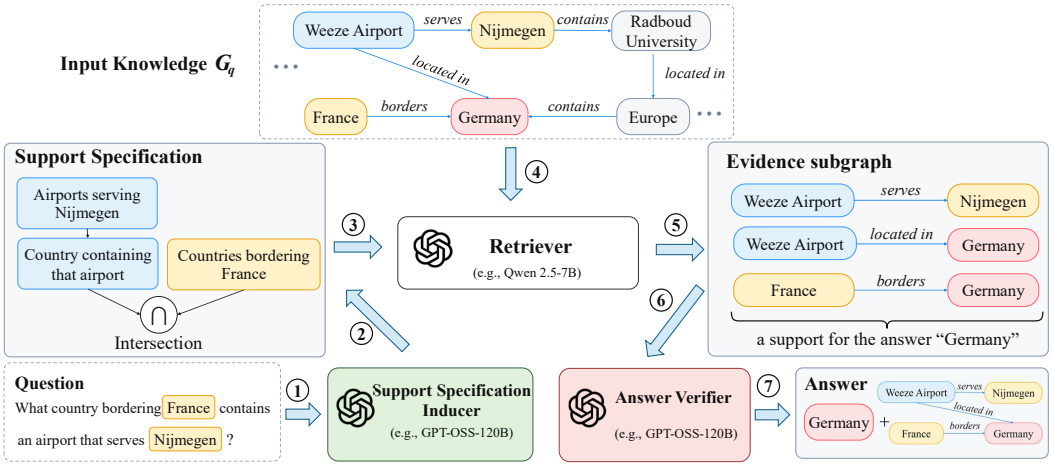


Figure 6. GRASP overview. ① The question is passed to the support specification inducer. ② The inducer predicts a support specification encoding evidence requirements and shared bindings. ③ The specification guides the retriever. ④ The input graph G_q provides the evidence available for retrieval. ⑤ The retriever constructs an evidence subgraph using relation guidance and binding consistency. ⑥ The retrieved evidence is passed to the answer verifier for assessment against the specification. ⑦ The verifier returns an answer with supporting evidence when it judges the evidence sufficient; otherwise, it returns UNKNOWN.

Grounded LLMs exhibit high answer persistence after support loss. After controlled support removal, CoT, GCR*, and DoG retain their original answers in 51.08–64.58% of cases (Table 6). As shown in Figure 5, APR decreases as more constraints loose support, yet substantial answer persistence remains. Thus, many predictions remain unchanged even after the evidence required to support them has been removed.

Prediction changes rarely translate into UNKNOWN after support removal. When CoT, GCR*, and DoG change their predictions after support removal, they more often switch to OTHER than to UNKNOWN (Table 6). Progressive removal reinforces this pattern: the decline in SAME is primarily absorbed by OTHER, rather than UNKNOWN (Figure 5). Together, these results show that predictions are weakly aligned with support availability: systems often continue answering even when sufficient evidence is no longer available.

> **RQ2 Takeaway.** Grounded LLMs often continue answering when sufficient support is unavailable, either preserving the original answer or switching to another one rather than returning UNKNOWN.

5.3 RQ3: DO EXPLICIT SUPPORT SPECIFICATIONS IMPROVE EVIDENCE-BOUNDED BEHAVIOR?

Motivation. RQ1 reveals a substantial gap between answer correctness and returned support, while RQ2 shows that grounded LLMs often continue answering when sufficient support is unavailable. We therefore examine whether making the conditions for sufficient support explicit and preserving them throughout grounding and answering can mitigate these two failures.

Approach. To mitigate the failures identified in RQ1 and RQ2, we introduce GRASP, i.e., a training-free framework that makes the evidence requirements of a question explicit and preserves them throughout grounding and answering. Given a natural-language question, linked entities, and a question-local knowledge graph, GRASP first predicts a *support specification* describing the evidence required to answer the question. It then jointly grounds this specification over the graph using relation guidance and binding consistency, producing step-organized evidence aligned with the predicted requirements. Finally, a specification-aware answerer returns an answer only when the grounded evidence sufficiently supports the specification; otherwise, it returns UNKNOWN. This design directly targets the two failures identified above. Explicit support specifications and binding-consistent grounding reduce the risk that a correct prediction is accompanied by incomplete or binding-inconsistent evidence, addressing the evidence gap in RQ1. Specification-aware answering, in turn, conditions the decision to answer on whether the required support has been established, addressing the persistent answering observed in RQ2. All model components remain frozen, requiring no task-specific training. The complete GRASP procedure is provided in Algorithm 1 in Appendix A.7.

Results and Analysis. GRASP substantially improves evidence-supported answering. On CWQ, GRASP with Qwen2.5-7B as the retriever reaches 81.30% HIT@1 and 58.96% EVIDENCEHIT@1, while reducing EVIDENCEGAP from 56.56% for GCR* to 27.47% (Figure 7). Consistent improvements are observed on WebQSP and 2WikiMultiHopQA, with detailed results reported in Appendix A.8. These results show that explicit support requirements and binding-consistent grounding reduce the mismatch between answer correctness and sufficient returned evidence identified in RQ1.

GRASP substantially improves recognition of unavailable support. On CWQ-$\mathcal{I}$, GRASP reaches 85.74% $\mathrm{CAR}_{\mathcal{I}}$; after verified support removal, 86.02% of edited inputs transition to UNKNOWN, while only 9.64% retain the original answer and 4.34% switch to another answer. This directly mitigates the persistent and unsupported answering observed in RQ2, although grounding failures still cause some over-abstention on support-available questions ($\mathrm{OAR}_{\mathcal{S}} = 12.92\%$).

RQ3 Takeaway. By preserving the support specification through grounding and answering, GRASP narrows the evidence gap and improves recognition of unavailable support.

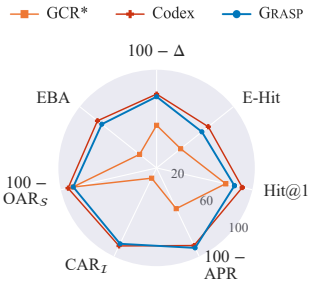


Figure 7. Evidence-bounded performance on CWQ. GCR*, GRASP use GPT-OSS-120B for answering, while Codex uses GPT-6.

6 LIMITATIONS

Our support judgments treat structured benchmark annotations as operational specifications of question semantics. The evaluation verifies whether returned evidence satisfies constraints derived from gold SPARQL queries or annotated evidence triples, but it cannot determine whether these annotations fully capture the intended meaning of the question. Annotation errors, question–query mismatches, and machine-translation artifacts may therefore introduce residual evaluation noise. The support-removal analysis is restricted to CWQ and to instances that are initially answered correctly with sufficient evidence, so its findings do not necessarily generalize to the full benchmark distribution. In addition, GRASP is a reference mitigation rather than a complete solution: errors in support-specification induction or relation grounding, as well as finite search , may still cause valid support to be missed.

7 CONCLUSION

Grounded LLMs must provide sufficient evidence and recognize insufficient support, not merely answer correctly. On KGQA, we find gaps between answer accuracy and evidence sufficiency, as well as persistent answering after support removal. GRASP mitigates both failures by preserving an explicit support specification through binding-consistent grounding and specification-aware answering. Beyond testing whether returned evidence supports a prediction, our results motivate evaluating whether systems appropriately return UNKNOWN when support is unavailable.

REPRODUCIBILITV STATEMENT

The replication package can be found here: https://anonymous.4open.science/r/grasp-code/README.md.

AI USE STATEMENT

Generative AI tools assisted with language checking, literature search and CWQ answer-granularity screening and sensitivity analysis. The authors independently read and verified the cited literature and reviewed the visualization code and core implementation. The curation procedure is described in Appendix A.6. The authors take responsibility for the manuscript, code, results, and conclusions.

REFERENCES

Vaibhav Adlakha, Parishad BehnamGhader, Xing Han Lu, Nicholas Meade, and Siva Reddy. Evaluating correctness and faithfulness of instruction-following models for question answering. *Transactions of the Association for Computational Linguistics*, 12:681–699, 2024. doi: 10.1162/tacl_a_00667. URL https://aclanthology.org/2024.tacl-1.38/.

Yao Cheng, Yibo Zhao, Jiapeng Zhu, Yao Liu, Xing Sun, and Xiang Li. Human cognition inspired RAG with knowledge graph for complex problem solving. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(36):30386–30394, 2026. doi: 10.1609/aaai.v40i36.40291. URL https://ojs.aaai.org/index.php/AAAI/article/view/40291.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6465–6488. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.398. URL https://aclanthology.org/2023.emnlp-main.398/.

Yu Gu, Sue Kase, Michelle Vanni, Brian Sadler, Percy Liang, Xifeng Yan, and Yu Su. Beyond I.I.D.: Three levels of generalization for question answering on knowledge bases. In *Proceedings of the Web Conference 2021*, pp. 3477–3488. Association for Computing Machinery, 2021. doi: 10.1145/3442381.3449992. URL https://doi.org/10.1145/3442381.3449992.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625. International Committee on Computational Linguistics, 2020. doi: 10.18653/v1/2020.coling-main.580. URL https://aclanthology.org/2020.coling-main.580/.

I-Hung Hsu, Zifeng Wang, Long Le, Lesly Miculicich, Nanyun Peng, Chen-Yu Lee, and Tomas Pfister. CaLM: Contrasting large and small language models to verify grounded generation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 12782–12803. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.759. URL https://aclanthology.org/2024.findings-acl.759/.

Akriti Jain and Aparna Garimella. Knowing what's missing: Assessing information sufficiency in question answering. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 4163–4174. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-eacl.217. URL https://aclanthology.org/2026.findings-eacl.217/.

Hailey Joren, Jianyi Zhang, Chun-Sung Ferng, Da-Cheng Juan, Ankur Taly, and Cyrus Rashtchian. Sufficient context: A new lens on retrieval augmented generation systems. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 20310–20334, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/33dffa2e3d2ab74a783d1a8c292f66d9-Paper-Conference.pdf.

Kun Li, Tianhua Zhang, Xixin Wu, Hongyin Luo, James R. Glass, and Helen M. Meng. Decoding on graphs: Faithful and sound reasoning on knowledge graphs through generation of well-formed

chains. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24349–24364. Association for Computational Linguistics, 2025a. doi: 10.18653/v1/2025.acl-long.1186. URL https://aclanthology.org/2025.acl-long.1186/.

Minghan Li, Junjie Zou, Xinxuan Lv, Chao Zhang, and Guodong Zhou. S2G-RAG: Structured sufficiency and gap judging for iterative retrieval-augmented QA. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25846–25862. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1185. URL https://aclanthology.org/2026.acl-long.1185/.

Mufei Li, Siqi Miao, and Pan Li. Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation. In *International Conference on Learning Representations*, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/11e1900e680f5fe1893a8e27362dbe2c-Abstract-Conference.html.

Can Lin, Zhengwang Jiang, Ling Zheng, Qi Zhao, Yuhang Zhang, Qi Song, and Wangqiu Zhou. RJE: A retrieval-judgment-exploration framework for efficient knowledge graph question answering with LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 17277–17294. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.873. URL https://aclanthology.org/2025.emnlp-main.873/.

Chenhui Liu, Jianpeng Zhou, and Jiahai Wang. Chain-of-relations: Faithful and efficient LLM reasoning over knowledge graphs via relation-centric exploration. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 43108–43119. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.2138. URL https://aclanthology.org/2026.findings-acl.2138/.

Linhao Luo, Yuan-Fang Li, Reza Haffari, and Shirui Pan. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/3e2aeb66481dd63a32421bf032b70384-Abstract-Conference.html.

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. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 41540–41565. PMLR, 2025. URL https://proceedings.mlr.press/v267/luo25t.html.

Katie Matton, Robert Ness, John Guttag, and Emre Kiciman. Walk the talk? measuring the faithfulness of large language model explanations. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/b5ec50eb177908f21f78ed0d76ed525c-Abstract-Conference.html.

Yifei Ming, Senthil Purushwalkam, Shrey Pandit, Zixuan Ke, Xuan-Phi Nguyen, Caiming Xiong, and Shafiq Joty. FaithEval: Can your language model stay faithful to context, even if "the moon is made of marshmallows". In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/48404cd9ce03946c6b7177691f3267a1-Abstract-Conference.html.

Minh-Vuong Nguyen, Linhao Luo, Fatemeh Shiri, Dinh Phung, Yuan-Fang Li, Thuy-Trang Vu, and Gholamreza Haffari. Direct evaluation of chain-of-thought in multi-hop reasoning with knowledge graphs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 2862–2883. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.168. URL https://aclanthology.org/2024.findings-acl.168/.

Dongyu Ru, Lin Qiu, Xiangkun Hu, Tianhang Zhang, Peng Shi, Shuaichen Chang, Cheng Jiayang, Cunxiang Wang, Shichao Sun, Huanyu Li, Zizhao Zhang, Binjie Wang, Jiarong Jiang, Tong He, Zhiguo Wang, Pengfei Liu, Yue Zhang, and Zheng Zhang. RAGChecker: A fine-grained framework for diagnosing retrieval-augmented generation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 21999–22027. Curran Associates, Inc., 2024. doi: 10.52202/079017-0692.

URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/27245589131d17368cccdfa990cbf16e-Abstract-Datasets_and_Benchmarks_Track.html.

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. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/10a6bdcabbd5a3d36b760daa295f63c1-Abstract-Conference.html.

Alon Talmor and Jonathan Berant. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 641–651. Association for Computational Linguistics, 2018. doi: 10.18653/v1/N18-1059. URL https://aclanthology.org/N18-1059/.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. doi: 10.52202/068431-1800. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Wen-tau Yih, Matthew Richardson, Chris 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. Association for Computational Linguistics, 2016. doi: 10.18653/v1/P16-2033. URL https://aclanthology.org/P16-2033/.

Peng Yu, En Xu, Bin Chen, Haibiao Chen, and Yinfei Xu. STEM: Structure-tracing evidence mining for knowledge graphs-driven retrieval-augmented generation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7252–7285. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.329. URL https://aclanthology.org/2026.acl-long.329/.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang. mGTE: Generalized long-context text representation and reranking models for multilingual text retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 1393–1412. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-industry.103. URL https://aclanthology.org/2024.emnlp-industry.103/.

Yizhe Zhang, Siqi Sun, Xiang Gao, Yuwei Fang, Chris Brockett, Michel Galley, Jianfeng Gao, and Bill Dolan. RetGen: A joint framework for retrieval and grounded text generation modeling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11739–11747, 2022. doi: 10.1609/aaai.v36i10.21429. URL https://ojs.aaai.org/index.php/AAAI/article/view/21429.

Wenxin Zhao, Jiachuan Wang, Yongqi Zhang, Shuangyin Li, Cheng Deng, Jun Wang, and Lei Chen. VoG: Enhancing LLM reasoning through stepwise verification on knowledge graphs. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/ab1ee157f7804a13f980414b644a9460-Abstract-Conference.html.

Dongzhuoran Zhou, Yuqicheng Zhu, Xiaxia Wang, Hongkuan Zhou, Yuan He, Jiaoyan Chen, Steffen Staab, and Evgeny Kharlamov. What breaks knowledge graph based RAG? benchmarking and empirical insights into reasoning under incomplete knowledge. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2522–2538. Association for Computational Linguistics, 2026a. doi: 10.18653/v1/2026.eacl-long.114. URL https://aclanthology.org/2026.eacl-long.114/.

Xinyu Zhou, Chang Jin, Carsten Eickhoff, Zhijiang Guo, and Seyed Ali Bahrainian. When silence is golden: Can LLMs learn to abstain in temporal QA and beyond? In *International Conference on Learning Representations*, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/7718914dfe7d5a657bf6261b5f431021-Abstract-Conference.html.

A Appendix

A.1 Replication Package

An anonymized replication package for this work is available at `https://anonymous.4open.science/r/grasp-code/README.md`. The package provides the source code and supporting materials for reproducing our experiments.

A.2 Dataset Construction

This section details how we construct the $\mathcal{S}/\mathcal{I}$ partitions from the graph snapshots and structured benchmark supervision used in our evaluation.

A.2.1 Data Sources and Partition Sizes

For WebQSP and CWQ, we use the graph snapshots released with RoG (Luo et al., 2024), containing 3,531 and 1,628 question–graph pairs, respectively. Their directed Freebase graphs contain relations formed by folding compound value type (CVT) nodes. We align each instance by question ID with the corresponding official gold SPARQL query: the CWQ query or the eligible WebQSP parses.

For 2WikiData, we use the graph snapshot of 2WikiMultiHopQA released with DoG (Li et al., 2025a), containing 4,016 question–graph pairs, and align it with the official development annotations of 2WikiMultiHopQA (Ho et al., 2020). The official evidence triples define the question constraints used to determine support availability.

The resolved evaluation populations are reported in Table 2. We exclude 7 WebQSP questions and 19 2WikiData questions due to inconsistencies between the graph snapshots and the corresponding gold SPARQL queries or evidence annotations.

A.2.2 Constructing the $\mathcal{S}/\mathcal{I}$ Partitions

For WebQSP and CWQ, we derive $\Phi(q)$ from the official gold SPARQL queries and replay them over G_q. A gold answer is sufficiently supported when the query constraints can be jointly satisfied in G_q under binding-consistent assignments. Following Section 3, a question belongs to $\mathcal{S}$ when at least one gold answer is sufficiently supported by G_q; otherwise, a completed replay places it in $\mathcal{I}$. Questions with unresolved eligible parses are excluded.

For 2WikiData, $\Phi(q)$ is defined by the official evidence triples. A question belongs to $\mathcal{S}$ when every required triple can be aligned with an edge in G_q under the entity, relation, and direction criteria described below. If at least one required triple is unavailable, the question belongs to $\mathcal{I}$. Questions with conflicting official entity identities are excluded.

A.2.3 WebQSP and CWQ: Gold SPARQL Replay

Entity alignment. For WebQSP, we use the official topic and constraint entities, supplemented by a fixed MID-to-name mapping. Entity names are compared after Unicode, whitespace, and case normalization. Ambiguous entity alignment leaves the corresponding parse unresolved. For CWQ, we use the provided entity mappings to align Freebase machine identifiers with entity names in the RoG graph snapshot.

Constraint execution and binding consistency. The replayer retains positive triple patterns, shared variables, `UNION` alternatives, and supported filters, comparisons, temporal conditions, and ordering operations. Constraints involving the same variable must be satisfied under binding-consistent

assignments. For `UNION`, one successfully supported branch is sufficient. The resulting answers are compared with the gold-answer set $A_q^\star$ using normalized exact matching.

Folded relations and inverse relations. The RoG snapshots may represent a CVT-mediated relation chain as a folded edge. For example, $u \xrightarrow{r_1 - r_2} v$ represents the ordered chain $u \xrightarrow{r_1} c \xrightarrow{r_2} v$. Replay matches the folded relation sequence while preserving the observable endpoints and their bindings; hidden CVT identities are not reconstructed.

To account for support reachable through inverse relations, we additionally accept relation pairs declared as inverses in the GrailQA Freebase ontology (Gu et al., 2021). For folded relations, inversion reverses both direction and relation order:

$$\mathrm{inv}(r_1 - r_2) = \mathrm{inv}(r_2) - \mathrm{inv}(r_1). \tag{8}$$

Relations without a declared inverse retain their original direction.

Comparison and ordering constraints. Temporal and numeric comparisons require values supported by explicit attribute triples in G_q. For ordering queries, replay first enumerates feasible candidates in G_q, applies the requested ordering operation, and then checks gold-answer membership. Gold answers are not used to prefilter candidate entities.

Some official temporal queries contain `NOT EXISTS` alternatives. Because our evaluation requires explicit supporting evidence, the absence of an attribute alone does not establish a temporal condition. We therefore require an observed attribute value that satisfies the corresponding comparison.

Support-availability decision. WebQSP uses official parses with `ParseQuality=Complete`. A successful eligible parse places the question in $\mathcal{S}$. If every eligible parse is successfully evaluated but none sufficiently supports a gold answer, the question is placed in $\mathcal{I}$. Questions with an eligible parse that cannot be parsed, aligned, or executed are treated as unresolved. All CWQ questions in the reported population receive resolved support-availability labels.

A.2.4 2WIKIDATA: EVIDENCE-TRIPLE ALIGNMENT

Triple normalization. We normalize the official evidence triples by lowercasing text, removing ASCII punctuation, and collapsing whitespace while preserving articles. Relations must agree after normalization; semantic paraphrases and year-to-date projections are not introduced during this alignment.

Entity alignment. When an official evidence triple provides endpoint QIDs, those identities determine the permitted entities. The original surface form and official aliases or demonyms of the same entity may match graph endpoints. A conflict between an annotated surface form and its explicit QID is treated as an annotation conflict.

When no endpoint QID is available, a surface form is mapped to a QID only when it uniquely identifies an entity among the official entities and answers for that question. Alternative aliases are accepted only when they uniquely resolve to the same entity.

Evidence-triple alignment. Each required triple (s, r, o) must align with a directed edge in G_q whose endpoints correspond to the permitted entity identities and whose normalized relation agrees with r. Endpoint presence alone is insufficient: relation and direction must also agree. The 2WikiData alignment does not apply typed-literal projection, relation-schema projection, or Freebase inverse rules.

A.3 RQ1 STRATIFICATION BY CONSTRAINT COUNT AND STRUCTURE

We stratify CWQ-$\mathcal{S}$ questions by constraint count and query structure. Constraint count is the number of positive triple patterns in the gold SPARQL query, i.e., relational facts that must be jointly satisfied. We further identify three structural patterns: *chain*, which composes relations through an intermediate entity; *branching*, which requires multiple constraints to share an intermediate binding; and *intersection*, which requires multiple constraints to share the same answer binding. These patterns may overlap within a question (See Figure 2). Figure 4 reports results for CoT, GCR*, and DoG with Qwen2.5-7B, together with Codex GPT-6 xhigh.

A.4 CWQ PARTITION STABILITY UNDER SEMANTIC RELAXATION

We recompute the $\mathcal{S}/\mathcal{I}$ partition on all CWQ questions in Table 2 under exact relation matching and seven semantic relaxation thresholds from $\tau = 0.95$ to 0.60 (Table 7 and Figure 8). Replay on the full input knowledge G_q for each question retains exact relation matches and additionally accepts relation pairs whose `gte-large-en-v1.5` cosine similarity is at least τ. Lowering τ enlarges the set of admissible relation matches. Shared entity bindings, folded-path length and order, declared inverse relations, and comparison and ordering criteria remain unchanged.

Table 7. CWQ $\mathcal{S}/\mathcal{I}$ partition under semantic relation relaxation. Transitions are measured relative to the exact split; percentages use all 3,138 questions as the denominator.

Matching	$\lvert\mathcal{S}\rvert$	$\lvert\mathcal{I}\rvert$	$\mathcal{I}\to\mathcal{S}$	Changed (%)
Exact	1,679	1,459	0	0.00
$\tau = 0.95$	1,688	1,450	9	0.29
$\tau = 0.90$	1,696	1,442	17	0.54
$\tau = 0.85$	1,706	1,432	27	0.86
$\tau = 0.80$	1,718	1,420	39	1.24
$\tau = 0.75$	1,725	1,413	46	1.47
$\tau = 0.70$	1,728	1,410	49	1.56
$\tau = 0.60$	1,750	1,388	71	2.26

All 1,679 original $\mathcal{S}$ questions remain in $\mathcal{S}$ at every tested threshold. At $\tau = 0.60$, 71 original $\mathcal{I}$ questions move to $\mathcal{S}$, giving 1,750 $\mathcal{S}$ and 1,388 $\mathcal{I}$ questions. Thus, only 2.26% of the full CWQ population changes its partition label under the most permissive setting tested. This input-knowledge analysis recomputes partition membership; Figure 9 keeps the original 1,679-question $\mathcal{S}$ split fixed to measure EVIDENCEGAP robustness in returned evidence.

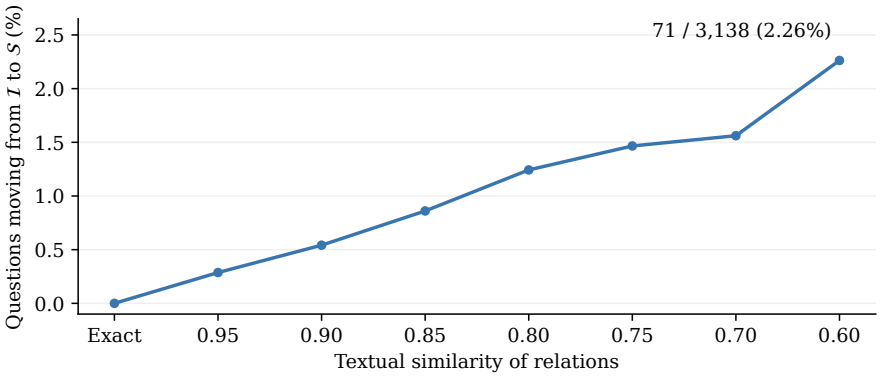
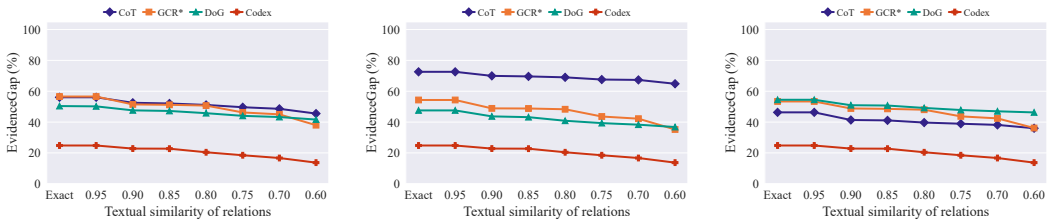


Figure 8. CWQ partition changes under semantic relation relaxation. Percentages use all 3,138 questions; only transitions from $\mathcal{I}$ to $\mathcal{S}$ occur. All 1,679 original $\mathcal{S}$ questions remain supported at every threshold.

A.5 SEMANTIC RELAXATION ACROSS MODELS

Figure 9 reports semantic relaxation for Qwen2.5-7B, Gemma-2-9B, and Llama-3.1-8B. The analysis keeps the original 1,679 CWQ-$\mathcal{S}$ questions, predictions, and returned evidence fixed. In addition to exact relation identifiers, replay accepts relation pairs whose `gte-large-en-v1.5` cosine similarity reaches the displayed threshold. Other replay criteria, including shared entity bindings, remain unchanged. All configurations retain a nonzero EVIDENCEGAP at $\tau = 0.60$.



(a) Qwen2.5-7B **(b)** Gemma-2-9B **(c)** Llama-3.1-8B

Figure 9. EVIDENCEGAP under semantic relation relaxation on CWQ-$\mathcal{S}$. The original 1,679-question $\mathcal{S}$ split, predictions, and returned evidence remain fixed. Lower thresholds admit more relation matches. The Codex GPT-6 xhigh reference is repeated across panels.

A.6 CURATION OF ANSWER GRANULARITY

Answer granularity. We examine whether CWQ answers and query outputs have the type, explicit unit, and granularity requested by the natural-language question. For example, a question asking for a year requires a year-valued answer, and a question asking for a count requires a number. The analysis uses the question text, benchmark answers, and the output variables of the corresponding SPARQL queries. GPT-6 Astra (xhigh) and Claude 5.1 independently screened all 3,531 CWQ questions for potential answer-granularity mismatches. We then manually audited all cases on which the two models disagreed and resolved these discrepancies based on the question, benchmark answer, and SPARQL output. This process identified 393 answer-granularity mismatches. The reported CWQ results exclude these mismatches and uncertain cases, leaving 3,138 questions. We assess sensitivity by comparing this curated population with all 3,531 questions(Table 8).

Table 8. Curated CWQ populations answer-granularity screening. Condition loss and binding mismatch count intervention pairs; progressive removal counts questions shared across removal levels.

N	$\lvert\mathcal{S}\rvert$	$\lvert\mathcal{I}\rvert$	Condition loss	Binding mismatch
3,138	1,679	1,459	295	120

A.7 PROPOSED METHOD: GRASP

Algorithm 1 gives the complete procedure for the three stages introduced in Section 5.3 and illustrated in Figure 6.

Step 1: Support Specification Induction. Given question q and linked entities A_q, GPT-OSS-120B generates a support specification P. Its steps P_{steps} form a DAG of retrieval and intersection operations. Each retrieval step specifies a relation constraint, an input entity or earlier result, and an output role. Dependencies express composition, while intersections require shared bindings across branches. A final answering instruction specifies how to combine the results, including filtering, comparison, or aggregation.

Step 2: Specification-Guided Grounding. We ground P in G_q by combining symbolic execution with neural relation scoring. Each search state (ρ, D, s) contains the directed relation assignments ρ, binding records D with evidence triples for each step, and cumulative score s. The initial state binds the linked entities to graph nodes. For each retrieval step, ADJACENTRELATIONS enumerates relations adjacent to the current input bindings and retains those with a direction-specific domain–range signature in $\mathcal{O}$.

EXECUTE extends the bindings and records the traversed triples. APPLYINTERSECTIONS then executes the intersection steps following the retrieval, retaining records whose shared entity or value bindings agree. Empty extensions are discarded before scoring. In Figure 6, the same country must satisfy the border and airport-location conditions, and the same airport must satisfy the location and service conditions. The compatible Germany/Weeze Airport bindings supply supporting triples for Germany.

For each surviving extension, RELATIONSCORE assesses how well the directed relation r and its executed outputs match retrieval step j. Its context $c_j(r)$ includes q, P_{steps}, prior relation assignments, current input bindings, the candidate's direction and domain–range signature, and its outputs. The

final answering instruction is omitted from this context. A frozen language model scores compatibility using the next-token logits of YES and NO:

$$p_j(r) = \frac{\exp \ell_\theta(\texttt{YES} \mid c_j(r))}{\exp \ell_\theta(\texttt{YES} \mid c_j(r)) + \exp \ell_\theta(\texttt{NO} \mid c_j(r))}, \qquad S_m = \sum_{j=1}^{m} \log p_j(r_j). \tag{9}$$

Here ℓ_θ denotes the model's next-token logit and S_m the cumulative score after m retrieval choices. TOPB retains the B highest-scoring extensions across all current states. The operator $\|$ appends a relation assignment to ρ. Search continues until all retrieval steps have been processed or the beam becomes empty; an empty beam yields UNKNOWN.

Step 3: Evidence-Bounded Answering. PACKBYSTEP organizes triples from the retained complete groundings into an evidence subgraph H. It selects unique triples in round-robin order across steps so that the full answer-generator input fits token budget L. Given q, P, and the step-organized evidence, ANSWERLLM predicts a sufficiency flag u, an answer $\hat{a}$, and citation identifiers C. If it judges the evidence insufficient, the system returns UNKNOWN; otherwise, RESOLVECITATIONS maps the identifiers to returned evidence E. Model configurations and prompts appear in Appendices A.9 and A.10.

Algorithm 1 GRASP: support specification, grounding, and answering

Require: Question q, input knowledge G_q, linked entities A_q
Require: Relation-signature index $\mathcal{O}$, beam width B, input token budget L
Ensure: Predicted answer $\hat{a}$ with returned evidence E, or UNKNOWN

1. Support Specification Induction
$P \leftarrow \textsc{InduceSpecification}(q, A_q)$
2. Specification-Guided Grounding
$\mathcal{B} \leftarrow \{(\varnothing, \textsc{BindAnchors}(P, A_q, G_q), 0)\}$
for each retrieval step j in dependency order of P **do**
  $\mathcal{C} \leftarrow \varnothing$
  for each state $(\rho, D, s) \in \mathcal{B}$ **do**
    $R \leftarrow \textsc{AdjacentRelations}(G_q, D, j, \mathcal{O})$
    for each directed relation $r \in R$ **do**
      $D' \leftarrow \textsc{Execute}(G_q, D, j, r)$
      $D' \leftarrow \textsc{ApplyIntersections}(P, j, D')$
      if $D' \neq \varnothing$ **then**
        $p_j(r) \leftarrow \textsc{RelationScore}(q, P_{\text{steps}}, j, \rho, D, r, D', \mathcal{O})$
        $\mathcal{C} \leftarrow \mathcal{C} \cup \{(\rho \,\|\, (j, r), D', s + \log p_j(r))\}$
      end if
    end for
  end for
  $\mathcal{B} \leftarrow \textsc{TopB}(\mathcal{C}, B)$
  if $\mathcal{B} = \varnothing$ **then return** UNKNOWN
end for
3. Evidence-Bounded Answering
$H \leftarrow \textsc{PackByStep}(q, P, \mathcal{B}, L)$
$(u, \hat{a}, C) \leftarrow \textsc{AnswerLLM}(q, P, H)$
if $u =$ false **then return** UNKNOWN
$E \leftarrow \textsc{ResolveCitations}(C, H)$
return $(\hat{a}, E)$

A.8 ADDITIONAL EVALUATION OF GRASP

A.8.1 EVIDENCE-BOUNDED PERFORMANCE ACROSS DATASETS

Figure 10 summarizes the evidence-bounded performance of GRASP across datasets and models. The WebQSP and 2WikiData results complement the CWQ results presented in the main text. Across these datasets, the improvements in evidence-supported answering and recognition of unavailable support remain consistent, indicating that the observed gains are not specific to CWQ.

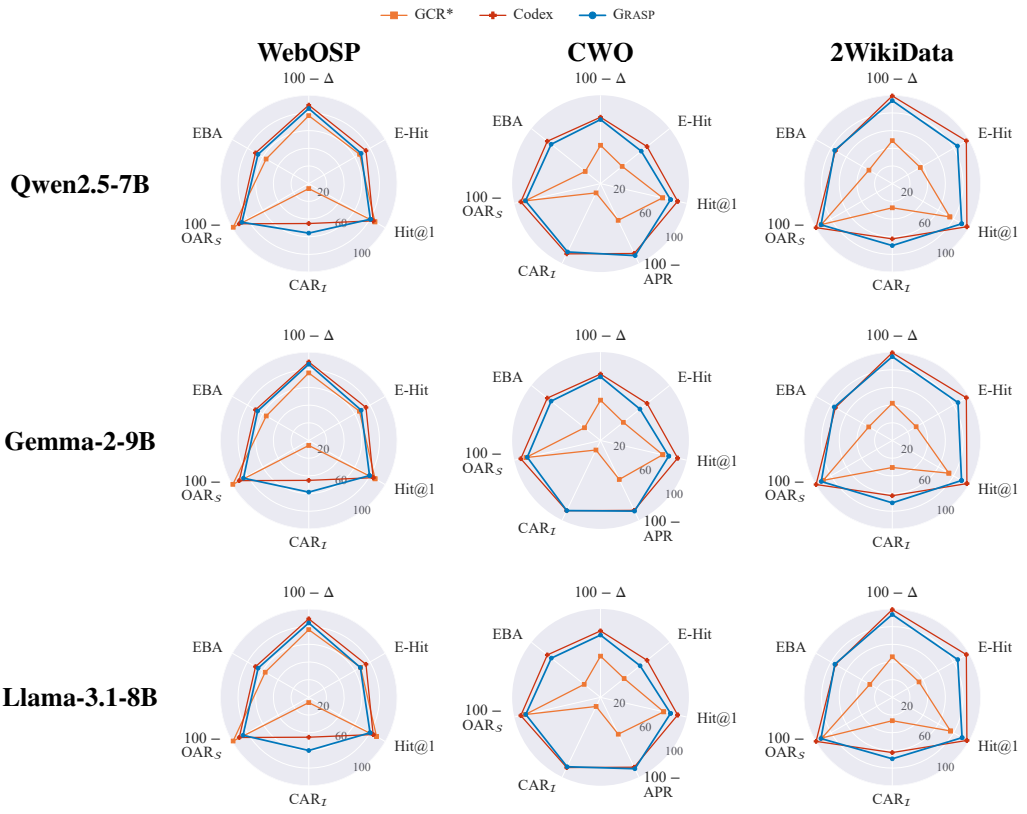


Figure 10. Evidence-bounded performance across datasets and models. Columns show WebQSP, CWQ, and 2WikiData; rows show Qwen2.5-7B, Gemma-2-9B, and Llama-3.1-8B.

A.8.2 COMPONENT ANALYSIS

We examine three design elements of GRASP: *specification-aware answering*, which keeps the predicted support specification explicit during answer generation; *relation guidance*, which uses relation direction and type information to guide grounding; and *binding consistency*, which preserves compatible entity and value bindings across dependent constraints. Each ablation removes one element while keeping the remaining configuration fixed.

Table 9. Component analysis of GRASP on CWQ (%).

Model	Configuration	HIT@1 ↑	E-HIT ↑	$\text{OAR}_{\mathcal{S}}$ ↓	$\text{CAR}_{\mathcal{I}}$ ↑	EBA ↑
Qwen2.5-7B	w/o specification-aware answering	78.20	52.23	17.09	86.29	68.07
	w/o relation guidance	79.39	57.24	14.83	**87.05**	71.10
	w/o binding consistency	74.39	53.72	15.37	86.57	68.99
	GRASP (full)	**81.30**	**58.96**	**12.92**	85.74	**71.41**
Llama-3.1-8B	w/o specification-aware answering	78.98	49.97	15.90	86.36	66.89
	w/o relation guidance	78.62	56.76	15.31	86.36	70.52
	w/o binding consistency	75.40	51.94	14.77	85.54	67.56
	GRASP (full)	**81.30**	**57.30**	**13.22**	**87.11**	**71.16**
Gemma-2-9B	w/o specification-aware answering	77.07	49.49	18.11	87.39	67.11
	w/o relation guidance	77.78	55.99	15.90	**88.14**	70.94
	w/o binding consistency	68.37	47.89	19.89	86.63	65.90
	GRASP (full)	**79.21**	**57.06**	**15.01**	88.01	**71.45**

Removing specification-aware answering reduces EVIDENCEHIT@1 by 6.73–7.56% and EBA by 3.35–4.33% across models. Removing binding consistency produces similarly large degradation, reducing EVIDENCEHIT@1 by 5.24–9.17% and HIT@1 by 5.90–10.84%. By comparison, removing relation guidance has a smaller effect on EVIDENCEHIT@1 and EBA.

Across all three models, the full GRASP configuration achieves the highest HIT@1, EVIDENCEHIT@1, and EBA, while also yielding the lowest $\text{OAR}_{\mathcal{S}}$. Some ablations slightly increase $\text{CAR}_{\mathcal{I}}$, but at the cost of weaker evidence-supported answering or greater over-abstention. These results iden-

tify specification-aware answering and binding consistency as the strongest contributors to GRASP's evidence-bounded behavior.

A.8.3 SENSITIVITY TO GROUNDING MODEL SIZE

We vary the Qwen2.5 grounding model from 1.5B to 7B and 32B on CWQ while keeping the Support Specification Induction outputs, Specification-Guided Grounding procedure, Evidence-Bounded Answering stage, and evaluation setup fixed.

Table 10. Sensitivity to grounding model size on CWQ (%).

Grounding model	Evidence support on $\mathcal{S}$			Evidence-bounded decisions		
	HIT@1 ↑	E-HIT ↑	Δ ↓	$\text{CAR}_{\mathcal{I}}$ ↑	$\text{OAR}_{\mathcal{S}}$ ↓	EBA ↑
Qwen2.5-1.5B	68.08	47.17	30.71	88.42	26.03	66.35
Qwen2.5-7B	81.30	58.96	27.47	85.74	12.92	71.41
Qwen2.5-32B	84.46	60.51	28.35	85.47	10.84	72.12

Scaling the grounding model from 1.5B to 7B substantially improves both HIT@1 and EVIDENCE-HIT@1 while reducing over-abstention. Scaling further to 32B yields only modest additional gains: EBA increases from 71.41% to 72.12%, while EVIDENCEGAP remains similar at 27.47% and 28.35%. Thus, larger grounding models improve answer accuracy, but model scale alone does not account for the evidence-sufficiency gains of the full GRASP design.

A.9 EXPERIMENTAL SETTINGS

Table 11 summarizes the GRASP configurations. Only the grounding model varies across the Qwen, Gemma, and Llama configurations.

Table 11. Models and inference settings for GRASP.

Component / setting	Configuration
Support Specification Induction and Evidence-Bounded Answering	GPT-OSS-120B

Grounding model (Qwen)	`Qwen/Qwen2.5-1.5B-Instruct`
	`Qwen/Qwen2.5-7B-Instruct`
	`Qwen/Qwen2.5-32B-Instruct`

Grounding model (Gemma)	`google/gemma-2-9b-it`
Grounding model (Llama)	`meta-llama/Meta-Llama-3.1-8B-Instruct`
Support Specification Induction and Evidence-Bounded Answering	Temperature 0; seed 0; medium reasoning effort
Output budget	At most 4,096 tokens per call
Answer-generator input	At most 4,096 tokens, including instructions and evidence
Support Specification Induction	Five fixed demonstrations; one call per question
Specification-Guided Grounding	Global beam size 8; at most 32 output values shown per candidate

Evidence packing selects unique triples in round-robin order across support-specification steps within the input budget.

Comparison configurations. CoT and DoG generate answers with their configured language model. GCR* uses its GCR path generator (beam size 10, index path length 2) followed by GPT-OSS-120B, which selects paths as evidence. Codex uses GPT-6 with xhigh reasoning effort. It answers each graph in an independent session with tools for listing and filtering the supplied triples. The agent receives the question and linked entities, and returns an answer with supporting paths or UNKNOWN. Its tools access only the question's input knowledge. The prompt permits graph-based deductions and requires UNKNOWN when factual premises are missing.

A.10 PROMPT TEMPLATES

We show prompts for the three stages of GRASP. The source code provides complete templates, all five fixed demonstrations, and output schemas.

The prompt excerpts retain the implementation name `Plan` for the complete support specification P. In the grounding prompt, `Evidence DAG` denotes only its retrieval and intersection steps P_{steps}, excluding the final answering instruction.

A.10.1 Support Specification Induction

```
You plan evidence retrieval for a knowledge-graph question answering
    system. Given a question and linked entities, create the smallest
    complete Plan that retrieves the evidence a separate final reasoner
    needs. The Plan will be grounded and executed on the graph.
# Plan semantics
- Identify the requested answer role and the evidence needed to
    determine it.
- Each Retrieve asks one answer-relevant question-level semantic
    relation from its `input` to the values named by `result_role`.
    Runtime grounds it as one directed dataset relation; do not expose
    CVTs, mediator records, or storage paths as Steps.
- Start each branch from a linked entity. Chain Retrieves only when a
    distinct answer-changing relation needs an earlier Step's returned
    values as input; otherwise stop when the requested answer role is
    returned.
- Do not add a Retrieve merely to verify or describe values already
    returned by the answer relation. Retrieve a property separately only
     when the question asks for that value or the decision must compare,
     order, or aggregate it.
- Use Intersect when independent branches return the same kind of entity
     or value and only their shared values satisfy the question.
- Write `decision` as a self-contained instruction for the final
    reasoner. Explain how to produce the answer from the Step results,
    including what to return and any condition, aggregation, or
    comparison to apply.
```

One of the five demonstrations asks which films Mira Stone directed and Kai Lee starred in. The support specification retrieves each person's films in a separate branch and intersects the results. This example illustrates the shared answer binding; the other demonstrations cover a direct relation, relation composition, a property condition, and comparison.

A.10.2 Specification-Guided Grounding

The relation scorer uses the evaluated model's chat template.

```
Judge the proposed relation for the current step.
Following it from the current input should produce the values requested
    by this step and fit the question and full plan.
Reply with exactly YES or NO.
Question: {question}
Evidence DAG: {"steps":{plan_steps}}
Relations chosen for earlier steps: {prior_relation_assignments}
Current step asks for: {instruction_with_current_input}
Current result role: {result_role}
Proposed directed relation: {typed_directed_relation}
Deterministic output values shown: {first_32_output_values}
Actual output count: {full_output_count}
Answer:
```

A.10.3 Evidence-Bounded Answering

```
# Role and task
You are the Final Reasoner in a knowledge-graph question-answering
    pipeline. Answer QUESTION from the supplied Step Requirements and
    evidence. Verify that the evidence jointly satisfies every required
    Step and operation before answering. QUESTION defines the requested
    result. DECISION HINT describes how the Step results may be combined.
```

```
     Step Requirements and DECISION HINT are context; only the directed
    triples establish facts.
# Evidence semantics
- `[sN.tM] ["head","relation","tail"]` is one directed KG fact retrieved
     by Step `sN`. Only the displayed relation and direction determine
    what it proves.
- Cite a fact in `reason` by stating it in natural language and placing
    its evidence ID immediately after it, for example: `France borders
    Germany [s1.t1].`
- An Intersect Step keeps values shared by all listed input Steps.
# Output
Return one object matching the supplied schema. Set `sufficient` to true
     only when every fact and operation needed by QUESTION is supported.
# Input
QUESTION
{question}
DECISION HINT
{decision}
STEP REQUIREMENTS AND EVIDENCE
{rendered_evidence}
```

Removing specification-aware answering. For *w/o specification-aware answering*, the prompt retains the same triple sequence but omits the support specification's constraints, dependencies, intersections, step grouping, and answering instruction. It asks for an answer supported by the displayed triples.