ComplexLogicalQA: A Comprehensive Benchmark for Complex Logical Question Answering over Knowledge Graphs

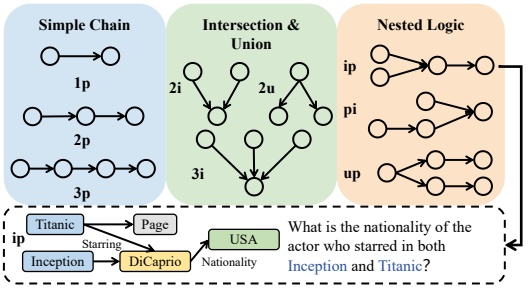
Anonymous ACL submission

Abstract

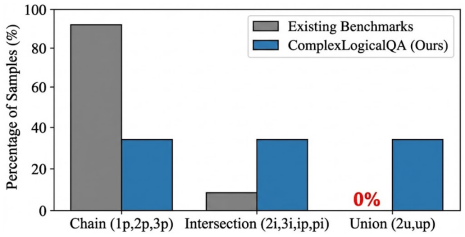
Knowledge Graph Question Answering (KGQA) has evolved significantly with the advent of Large Language Models (LLMs). However, current benchmarks suffer from a severe structural bias. They are dominated by simple linear paths, while complex logical operations such as Disjunction (Union) and Nested Logic are noticeably scarce. Our quantitative analysis of mainstream KGQA datasets reveals that 94% of questions are limited to chain-like reasoning, yet Union operations are completely absent. To bridge this gap, we propose **ComplexLogicalQA**, a comprehensive benchmark encompassing nine distinct Existential Positive First-Order Logic (EPFO) structures. We develop a novel logic-driven reverse-construction pipeline that leverages LLMs to verbalize sampled subgraphs, ensuring both structural complexity and linguistic diversity. Extensive evaluations across four paradigms reveal a significant reasoning illusion: while models excel at linear pattern matching, their performance collapses on Union and Nested logic. Further analysis identifies fundamental limitations in current paradigms, such as disjunction blindness in retrieval and premature pruning in agent-based search.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide spectrum of Natural Language Processing (NLP) tasks (OpenAI, 2025; Achiam et al., 2023; DeepSeek-AI et al., 2024). However, their deployment in knowledge-intensive scenarios is often hindered by critical limitations, most notably factual hallucinations and reasoning deficiencies(Huang et al., 2025a). To mitigate these issues, integrating external Knowledge Bases (KBs) to augment LLM generation has become a mainstream research direction (Lewis et al., 2020). Among various knowledge sources, Knowledge Graphs (KGs), characterized by their rich structured semantics and multi-hop connectivity, provide an essential structural foundation for enabling and grounding complex logical reasoning. Leveraging KGs to answer natural language questions defines the task of Knowledge Graph Question Answering (KGQA), which challenges models to derive answers by executing multi-hop reasoning over structured graph data (Berant et al., 2013).



(a) Question Logical Structures



(b) Structural Distribution across Benchmarks

Figure 1: (a) introduces diverse logical forms of KGQA questions with one example, and (b) presents a statistical Existing benchmarks and our ComplexLogicalQA.

While existing KGQA benchmarks have successfully evaluated the multi-hop reasoning abilities of LLMs to some extent, they suffer from a critical lack of diversity in *logical structures*. Drawing upon established formalisms in Complex Query Task(Ren* et al., 2020), questions on KGs can be categorized based on Existential Positive First-Order Logic (EPFO) operations, including multi-hop projection ($1p, 2p, 3p$), intersection ($2i, 3i$), union ($2u$), and complex nested structures (ip, pi, up). Unlike linear projections which primarily assess sequential

Table 1: Summary of the 9 logical query structures in our dataset. The notation P denotes the projection operator, while $\cap$ and $\cup$ denote intersection and union operations, respectively.

Category	Type	Formal Definition	Natural Language Question
Chain	1p	$P(\{e\}, r_1)$	Who directed the movie *Inception*?
	2p	$P(P(\{e\}, r_1), r_2)$	What is the birth country of the director of *Inception*?
	3p	$P(P(P(\{e\}, r_1), r_2), r_3)$	What awards were won by actors who starred in movies directed by *Nolan*?
Intersection	2i	$P(\{e_1\}, r_1) \cap P(\{e_2\}, r_2)$	Who directed both *Inception* and *Interstellar*?
	3i	$P(\{e_1\}, r_1) \cap P(\{e_2\}, r_2) \cap P(\{e_3\}, r_3)$	Who is the director of *Inception*, *Dunkirk*, and *Tenet*?
	ip	$P(P(\{e_1\}, r_1) \cap P(\{e_2\}, r_2), r_3)$	What is the nationality of the actor who starred in both *Inception* and *Titanic*
	pi	$P(P(\{e_1\}, r_1), r_2) \cap P(\{e_2\}, r_3)$	Who was born in a city located in the *UK* and also starred in *Inception*?
Union	2u	$P(\{e_1\}, r_1) \cup P(\{e_2\}, r_2)$	Who starred in *Inception* or *Titanic*?
	up	$P(P(\{e_1\}, r_1) \cup P(\{e_2\}, r_2), r_3)$	What awards were won by movies directed by *Nolan* or *Cameron*?

reasoning depth, these complex structures are indispensable for evaluating a model's ability to handle parallel constraints (Intersection) and disjunctive search spaces (Union). Detailed definitions and examples are presented in Table 1.

Unfortunately, current benchmarks exhibit a severe bias towards simple multi-hop projection questions. To scientifically quantify this bias, we conducted a large-scale structural analysis on 3,000 randomly sampled questions (1,000 each from WebQSP(Yih et al., 2016), CWQ(Talmor and Berant, 2018), and GrailQA(Gu et al., 2021)). As shown in Table 2 and Figure 1b, WebQSP and GrailQA are dominated by linear paths ($1p, 2p$), accounting for 94.0% and 91.1% of their respective samples. While CWQ introduces some intersection structures ($2i$), it severely lacks nested logic. Most critically, Disjunction (Union) operations ($2u, up$) are completely absent (0%) across all three mainstream benchmarks. This structural void fundamentally compromises the evaluation of reasoning algorithms. State-of-the-Art (SOTA) methods that excel on these leaderboards may essentially be overfitting to linear pattern matching.

To bridge this gap and provide a rigorous testbed for logical generalization, we propose **ComplexLogicalQA**, a comprehensive benchmark encompassing nine distinct logical structures. Departing from previous approaches that rely on rigid templates, we adopt a logic-driven reverse-construction pipeline. Specifically, we sample subgraphs corresponding to the target logical structures from Freebase (Bollacker et al., 2008) and employ LLMs to translate these structured queries into diverse natural language questions. To guarantee data validity, we enforce a strict multi-stage verification protocol featuring anti-leakage constraints, such as intermediate secrecy. In addition, we employ a cross-model auditing loop where a secondary LLM iteratively regenerates questions to resolve ambiguities. Human verification confirms the high quality of our dataset, with 96% of sampled questions deemed fluent and logically consistent.

Leveraging this benchmark, we conduct an extensive evaluation across four distinct KGQA paradigms: LLM-based methods (Wei et al., 2022; Wang et al., 2023), Retrieval-based methods (Mavromatis et al., 2025), Agent-based methods (Sun et al., 2024; Xu et al., 2024), and Semantic Parsing methods (Huang et al., 2025b; Yang et al., 2025). Our experiments reveal a significant reasoning illusion: while current models excel on simple linear paths, they suffer a severe performance collapse when confronted with complex logical forms. This finding highlights that reasoning in current LLM-based systems remains fragile and lacks the robustness required for Union and Nested logic. Furthermore, we demonstrate that these failures are not random but stem from fundamental limitations in how current paradigms represent and retrieve knowledge.

In summary, our contributions are as follows:

- We provide the first quantitative analysis of logical structure distribution in mainstream KGQA benchmarks, revealing a critical lack of complex operations such as Union and nested logic.
- We introduce ComplexLogicalQA, a high-quality dataset covering nine First-Order Logic structures, constructed via a novel logic-driven reverse-construction pipeline.
- We benchmark four categories of KGQA methods, systematically exposing their limitations in handling complex logic and offering new insights for future research in logical reasoning over Knowledge Graphs.

2 Related Work

2.1 Methodologies for KGQA

Existing KGQA approaches can be broadly categorized into four paradigms based on their utilization of LLMs and graph structures. LLM-based methods leverage the internal parametric knowledge of models like GPT-5 and DeepSeek-V3 (OpenAI, 2025; DeepSeek-AI et al., 2024) through strategies like CoT (Wei et al., 2022) and Self-Consistency (Wang et al., 2023). However, without external grounding, they remain prone to hallucination. Retrieval-based methods address this by fetching subgraph contexts, progressing from sparse (BM25) or dense (BERT, MiniLM) retrieval to advanced frameworks like G-Retrieval (He et al., 2024) and BYOKG-RAG (Mavromatis et al., 2025). Nevertheless, these approaches suffer from retrieval incompleteness in Union and nested queries. Their reliance on semantic similarity creates a bottleneck where semantically distant triples located in disparate logical branches are often overlooked, leading to reasoning failures.

Alternatively, Agent-based methods such as Think-on-Graph (Sun et al., 2024) and Generate-on-Graph (Xu et al., 2024) treat KGs as environments for autonomous navigation. While they ground reasoning in the graph structure, they are susceptible to pruning errors in branching logic, where valid but low-probability intermediate paths in Union or nested queries are discarded during beam search. Semantic Parsing (SP) approaches like ChatKBQA (Luo et al., 2024) and TARGA (Huang et al., 2025b), attempt to resolve these completeness issues by translating questions into executable logical forms. Although SP guarantees precise results for certain complex operations, it remains hindered by sophisticated nested structures, where the structural gap between natural language and formal syntax frequently leads to parsing errors.

2.2 KGQA Datasets

The development of KGQA benchmarks has evolved from single-relation retrieval in WebQuestions (Berant et al., 2013) and SimpleQuestions (Bordes et al., 2015) to multi-hop reasoning in WebQSP (Yih et al., 2016) and CWQ (Talmor and Berant, 2018). Recent benchmarks have further targeted specialized skills, such as compositional generalization in GrailQA (Gu et al., 2021), numerical reasoning in MarkQA (Huang et al., 2023), and retrieval robustness in M^{3}GQA (Peng et al., 2025). Despite this progress, a critical structural bias persists: existing KGQA datasets are heavily dominated by linear chain structures ($1p, 2p$), while Union and complex nested structures remains severely underrepresented.

Dataset	1p	2p	3p	2i	3i	ip	pi	2u	up
WebQSP	649	291	0	45	0	7	8	0	0
CWQ	55	504	201	182	8	7	43	0	0
GrailQA	784	127	22	36	20	9	2	0	0

Table 2: Distribution of logical structures in 1,000 randomly sampled questions from major benchmarks.

A parallel field, Complex Query Answering (CQA) (Ren* et al., 2020; Luus et al., 2021), also targets FOL reasoning but is unsuitable for KGQA due to two limitations: (1) it lacks natural language questions, precluding evaluation of linguistic-to-logical translation; and (2) it lacks realistic semantics, as sampling based purely on connectivity often groups unrelated entities. **ComplexLogicalQA** bridges this gap by providing a benchmark that combines CQA's structural diversity with semantically coherent natural language questions.

3 Preliminaries

To establish a formal foundation for our benchmark, we define KGQA through the lens of Existential Positive First-Order (EPFO) logic queries.

3.1 Knowledge Graph and Logical Queries

We define a Knowledge Graph as $\mathcal{G} = (\mathcal{V}, \mathcal{R}, \mathcal{E})$, where $\mathcal{V}$ is the set of entities, $\mathcal{R}$ is the set of binary relations, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{R} \times \mathcal{V}$ is the set of factual triples. A triple $(s, r, o) \in \mathcal{E}$ indicates that the relationship r holds between the subject entity s and the object entity o. A logical query q aims to find a set of answer entities $[\![q]\!] \subseteq \mathcal{V}$ that satisfy a specific logical constraint.

3.2 Atomic Logical Operators

Following established formalisms (Ren* et al., 2020), we decompose complex queries into three atomic operators: **Projection** (P), which retrieves adjacent entities; **Intersection** (I), representing logical conjunction ($\cap$); and **Union** (U), representing disjunction ($\cup$). We intentionally focus on EPFO logic, encompassing projection, intersection, and union. Negation is excluded to avoid the ambiguity inherent in the Open World Assumption of KGs, where incomplete data can lead to

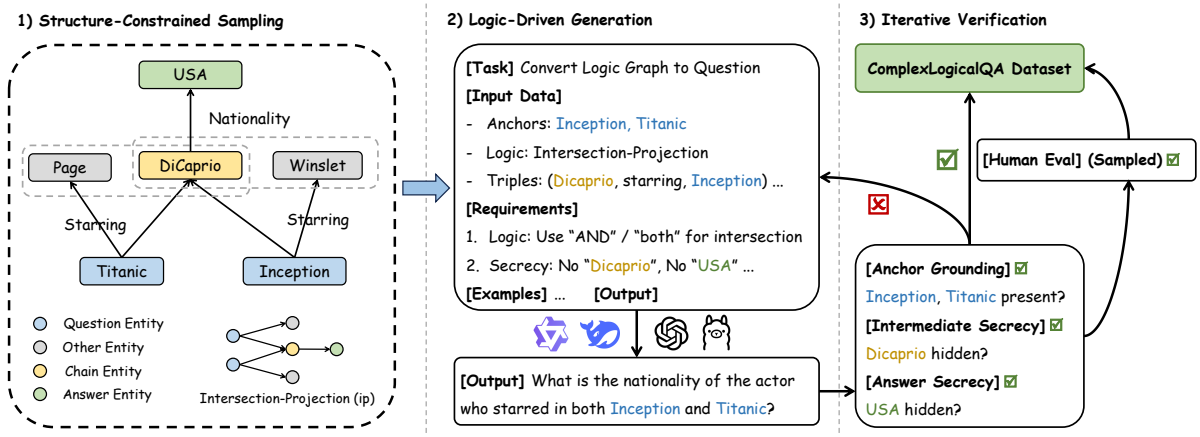


Figure 2: **Data Construction Pipeline.** (1) **Structure-Constrained Sampling:** Subgraphs are extracted based on specific logical topologies. (2) **Logic-Driven Generation:** LLMs translate structured triples into natural language questions under strict constraints. (3) **Iterative Verification:** A multi-stage quality control loop combining automated auditing.

unreliable negative gold standards. This ensures that our benchmark remains a rigorous testbed for multi-branch structural reasoning.

3.3 Logical Structure

By composing these atomic operators, we define nine distinct logical structures, as summarized in Table 1. While simple Chain queries ($1p, 2p, 3p$) involve linear projections to evaluate sequential reasoning depth, Intersection ($2i, 3i, ip, pi$) and Union ($2u, up$) structures introduce complex branching and nested logic. This rigorous categorization allows us to systematically benchmark the model's capability to handle parallel constraints and disjunctive paths, ensuring a comprehensive evaluation of Existential Positive First-Order (EPFO) logic beyond simple pattern matching.

4 Data Construction

To systematically evaluate the logical reasoning capabilities of KGQA models, we introduce **ComplexLogicalQA**. Unlike previous datasets constructed via rigid sentence templates (which limit linguistic diversity) or user logs (which lack complex logical structures), our construction follows a **logic-driven reverse-generation pipeline**. This paradigm ensures rigorous control over logical complexity while leveraging the linguistic generalization capabilities of LLMs. As illustrated in Figure 2, the pipeline integrates anchor-centric sampling, question verbalization, and an iterative multi-stage verification loop.

4.1 Anchor-Centric Structure Sampling

To ensure that our benchmark covers information-rich and realistic regions of the Knowledge Graph, we first collect all anchor entities from mainstream datasets, including WebQSP, CWQ, and GrailQA. These entities and their local neighborhoods serve as the primary seeds for our structure-constrained sampling process.

Target-Driven Backward Traversal for Intersections ($2i, 3i$). Instead of random forward sampling, we iterate through the collected anchor list to identify candidate answer entities $e_{ans} \in \mathcal{V}$. From a chosen e_{ans}, we perform a backward traversal along incoming relations to identify other potential anchor entities. To ensure the reasoning task is non-trivial, we enforce a non-overlap constraint: the sets of entities reachable from different branches must not be identical ($P(e_1, r_1) \neq P(e_2, r_2)$). This forces the model to reason across all branches rather than relying on a single redundant path.

Constrained Branching for Unions ($2u, up$). Similar to intersections, we start from the collected anchor seeds and traverse backward to form disjunctive structures. In addition to the non-overlap constraint, we prioritize semantic coherence by requiring all involved entities to belong to the same high-level domain (e.g., all entities must share the `film.film` type). This ensures that the union operation is performed over logically related entities, avoiding nonsensical questions.

Path Extension for Nested Logic (ip, pi). Nested structures are constructed as compositional ex-

Metric	1p	2p	3p	2i	3i	ip	pi	2u	up	Avg/Total
Avg. Q. Tokens	9.6	14.3	18.4	14.2	18.1	19.1	19.2	16.6	19.9	16.6
Avg. Ans. Entities	1.66	3.09	3.10	1.45	1.18	1.76	1.28	5.60	1.49	2.28
Avg. Triples	1.67	6.56	10.70	2.89	3.53	4.79	3.55	7.05	12.74	5.94
Unique Entities	371	858	719	664	795	959	805	1,466	1,614	5,764

Table 3: Detailed statistics of the ComplexLogicalQA Dataset. The dataset features high diversity in query length, answer set size, and the number of reasoning triples required.

tensions of the $2i$ or $2u$ subgraphs. For an ip structure $(P(P(e_1, r_1) \cap P(e_2, r_2), r_3))$, we first identify a valid intersection and then sample an outgoing relation r_3 from the intersection result set. We strictly filter out uninformative relations (e.g., `common.topic.description`) to maintain high reasoning quality.

4.2 Question Verbalization

We leverage Large Language Models (LLMs) to translate the sampled structured triples into natural language questions. Unlike template-based methods, we provide the LLM with the logical intent (e.g., "This is an Intersection-Projection query") and the set of supporting triples.

4.3 Multi-Stage Verification and Refinement

To ensure validity and solvability, each generated question must pass a strict three-rule filter: (1) **Anchor Grounding** requires the question to explicitly mention anchor entities (e.g., *Inception*, *Titanic*) to guarantee traversability; (2) **Answer Secrecy** prohibits the inclusion of canonical names or aliases of final answers to prevent string-matching shortcuts; and (3) **Intermediate Secrecy** mandates that bridging entities in the logic chain (e.g., *DiCaprio* in the ip structure) remain hidden to enforce deep reasoning.

For questions passing the rules above, we introduce an additional diversity and fluency auditor. We employ a secondary LLM to evaluate whether the question is natural, logically sound, and grammatically correct. If a question is flagged as awkward or ambiguous, it is sent back to a different LLM for regeneration with specific refinement feedback. This iterative loop continues for up to three attempts, after which failed samples are discarded.

To strictly validate quality, three expert annotators evaluated 1,000 stratified samples for Linguistic Fluency and Logical Consistency. The evaluation achieved a 98% inter-annotator agreement, confirming that 98% of samples are fluent and 96% are logically consistent.

4.4 Dataset Statistics

Quantitative Analysis of Structural Bias. To quantify structural bias, we analyzed 3,000 sampled questions using a dual-track protocol. For WebQSP and CWQ (SPARQL), three expert annotators manually categorized the logical structures. They cross-verified the results, achieving a 95% inter-annotator agreement, ensuring high reliability. For GrailQA (S-expressions), we used a deterministic parsing script to map function calls to our topological definitions. As shown in Table 2, WebQSP and GrailQA are dominated by linear paths ($> 90\%$), while CWQ lacks nested logic. Crucially, Union operations ($2u, up$) are completely absent (0%) in all baselines. In contrast, our ComplexLogicalQA strictly balances all nine logical types.

Dataset Analysis. Table 3 presents detailed statistics of ComplexLogicalQA dataset(1,000 samples per type, 9,000 total). The complexity of questions varies significantly across logical forms. For instance, $3p$ and up structures involve longer reasoning paths, reflected in higher average token counts (18.38 and 19.88, respectively) and a larger number of supporting triples (12.7 average triples for up). Notably, Union queries ($2u$) yield a significantly larger answer set (avg. 5.6 entities) compared to Intersection queries ($3i$, avg. 1.18 entities), posing distinct challenges for precision and recall in QA models.

To ensure reproducibility, we provide all LLM prompt templates, human evaluation guidelines, and the specific annotation criteria used for the structural analysis of existing benchmarks in the Appendix.

5 Experiments

5.1 Experimental Setup

Knowledge Base Selection. We use Freebase(Bollacker et al., 2008) to maintain consistency with existing benchmarks (WebQSP, CWQ

Method	1p		2p		3p		2i		3i		ip		pi		2u		up		Avg.	
	H@1	F1	H@1	F1	H@1	F1	H@1	F1	H@1	F1	H@1	F1	H@1	F1	H@1	F1	H@1	F1	H@1	F1
LLM Reasoning																				
GPT-3.5-Turbo	0.36	0.29	0.30	0.18	0.18	0.13	0.28	0.21	0.26	0.22	0.19	0.14	0.26	0.22	0.37	0.14	0.18	0.15	0.26	0.18
DeepSeek-V3	0.37	0.29	0.29	0.17	0.19	0.13	0.25	0.19	0.21	0.17	0.20	0.14	0.27	0.22	0.40	0.16	0.20	0.16	0.26	0.18
GPT-5	0.40	0.32	0.35	0.20	0.21	0.17	0.31	0.28	0.29	0.27	0.25	0.17	0.33	0.30	0.49	0.21	0.22	0.17	0.31	0.23
CoT	0.40	0.32	0.36	0.21	0.20	0.15	0.34	0.30	0.33	0.30	0.29	0.20	0.37	0.33	0.50	0.19	0.22	0.18	0.33	0.24
SC	0.41	0.33	0.36	0.21	0.21	0.18	0.31	0.29	0.30	0.27	0.26	0.17	0.35	0.31	0.50	0.20	0.21	0.18	0.32	0.23
Subgraph Retrieval																				
Sparse (BM25)	0.67	0.63	0.53	0.47	0.31	0.28	0.47	0.46	0.44	0.42	0.38	0.35	0.37	0.34	0.78	0.40	0.36	0.31	0.47	0.40
Dense (MiniLM)	0.70	0.65	0.57	0.52	0.33	0.31	0.48	0.46	0.46	0.45	0.40	0.37	0.36	0.33	0.77	0.46	0.38	0.36	0.49	0.43
BYOKG-RAG	**0.77**	**0.72**	0.61	0.57	0.35	0.30	0.56	0.53	0.51	0.49	**0.47**	**0.42**	**0.40**	**0.37**	**0.84**	0.53	0.46	0.40	**0.55**	0.47
Agents & Parsing																				
ToG	0.75	0.69	**0.65**	0.60	**0.43**	**0.42**	0.57	0.57	0.55	0.54	0.43	0.40	0.38	0.34	0.79	0.43	0.44	0.41	**0.55**	0.48
GoG	0.71	0.68	0.63	0.57	0.41	0.38	0.57	0.56	0.56	0.55	0.44	0.40	**0.40**	0.36	0.80	0.45	0.50	0.46	**0.55**	0.49
TARGA	0.69	0.68	0.64	**0.63**	0.40	0.39	**0.62**	**0.61**	**0.60**	**0.60**	0.35	0.31	0.34	0.32	0.75	**0.66**	0.49	0.47	0.54	**0.51**
Qwen3-8B-SP	0.65	0.63	0.59	0.57	0.37	0.34	0.60	0.59	0.59	0.58	0.40	0.39	0.35	0.31	0.73	0.61	**0.51**	**0.50**	0.53	0.50

Table 4: Performance on ComplexLogicalQA. **Bold** and underline denote the best and second-best results, respectively. Double vertical lines separate structure-specific scores from the overall average.

and GrailQA). Freebase's structured rigor also provides a cleaner testbed for EPFO operators than Wikidata(Vrandečić and Krötzsch, 2014), where qualifiers and redundant properties might introduce confounding variables into the evaluation of pure logical reasoning.

Data Partition. ComplexLogicalQA consists of 9,000 queries, strictly balanced with 1,000 samples for each of the nine logical structures. To rigorously evaluate the generalization ability of models to unseen entities, we adopt an entity-disjoint split strategy. Specifically, we ensure that the anchor entities in the test or validation set do not appear as anchors in the training set. The dataset is partitioned into training, validation, and test sets with a 7:1:2 ratio.

Baselines. We evaluate representative methods across four paradigms: (1) **LLM-based Reasoning**: Direct prompting on GPT-3.5 (Achiam et al., 2023), DeepSeek-V3 (DeepSeek-AI et al., 2024), and GPT-5 (OpenAI, 2025), augmented with CoT (Wei et al., 2022) and Self-Consistency (SC) (Wang et al., 2023); (2) **Retrieval-based**: Sparse (BM25), Dense Retrieval, and BYOKG-RAG (Mavromatis et al., 2025); (3) **Agent-based**: Think-on-Graph (ToG) (Sun et al., 2024) and Generate-on-Graph (GoG) (Xu et al., 2024); and (4) **Semantic Parsing (SP)**: TARGA (Huang et al., 2025b) and fine-tuned Qwen3-8B-SP (Yang et al., 2025). Detailed hyperparameters and implementation specifics are provided in Appendix.

Metrics. We report **Hit@1 (H@1)** and **F1-score** to assess single-answer correctness and complete answer set retrieval, respectively.

Implementation Details. For data construction, we employ `gpt-5-2025-08-07` as the primary generator and `gemini-2.5-pro` as an auditor to verify logical consistency and linguistic fluency. Queries flagged as unsatisfactory are regenerated by Gemini through an iterative refinement loop. In our evaluations, GPT-5 serves as the default backbone for all baselines (including CoT and SC) unless specified otherwise. For embedding model, we used `all-MiniLM-L12-v2`. Detailed hyper-parameters and implementation specifics for tested methods are provided in the Appendix.

5.2 Main Results

Table 4 reports the performance across four paradigms, revealing a profound disparity between finding a single correct answer (H@1) and retrieving the complete answer set (F1). LLM-based methods exhibit a reasoning illusion on complex structures; while performing well on linear paths, they fail to retrieve exhaustive results for branching logic like Union, highlighting the limitations of relying solely on internal knowledge. Retrieval-based methods achieve competitive H@1 on proximal operations (2i, 2u) but collapse on nested or long-chain queries (2p, 3p) as the semantic distance to remote triples increases. Their consistently low F1 scores, especially on Union, stem from an inability to capture disparate logical branches.

Agent-based methods outperform retrieval on

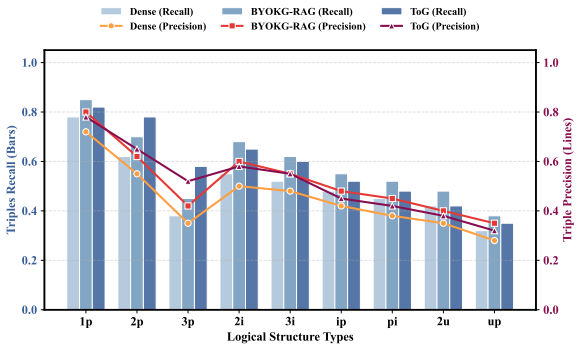


Figure 3: Retrieval quality comparison.

multi-hop structures through active exploration, yet remain limited by pruning constraints used to manage computational complexity. This bottleneck underscores the need for more intelligent traversal strategies. Semantic parsing improves answer coverage via deterministic execution, bypassing pruning issues, but remains hindered by poor structural generalization. It frequently fails to accurately map natural language to formal syntax in complex or novel logical scenarios.

5.3 Diagnostic Analysis of Evidence Quality

To verify whether systems identify complete reasoning paths, we measure the recall and precision of triples retrieved by Dense Retrieval, ToG, and BYOKG-RAG against ground truth evidence. As illustrated in Figure 3, performance declines on 2p and 3p despite high scores on 1p and 2i. ToG achieves higher recall on multi hop paths through active exploration, yet the precision drop reveals that deeper traversal introduces significant graph noise. A performance collapse occurs on nested and union structures, where low recall and precision on 2u indicate an inability to retrieve complete evidence across parallel branches. For agent based systems, this failure confirms that beam search pruning is ill suited for union operations as parallel branches are often discarded prematurely. These findings substantiate the reasoning illusion, proving that current systems lack the retrieval completeness and noise resistance required for complex First Order Logic.

5.4 Rationality and Semantic Coherence

We evaluate the realistic nature of our queries through entity relatedness and pragmatic utility.

Our pipeline enforces strict domain constraints. The average cosine similarity between anchor entities in Union queries is 0.76, significantly higher than random pairs (0.57). Additionally, 100% of queries maintain Answer Type Consistency, ensuring disjunctive branches operate within coherent semantic fields rather than conflating unrelated domains.

As shown in Table 5, our structures reflect genuine information needs. Union queries effectively represent users' intent to aggregate candidates from parallel sources, while Nested queries facilitate multi-constraint filtering via intermediate attributes. These logical forms capture sophisticated inquiries that simple linear chains cannot represent.

6 Analysis

While Table 4 demonstrates the performance collapse on complex logic, it does not fully explain the underlying mechanisms. We disentangle the error sources into three systemic bottlenecks inherent to current paradigms.

6.1 Retrieval-based Methods

For retrieval-based methods, handling Union queries ($2u, up$) presents a fundamental dilemma. A single vector fails to encode multiple distinct targets, as it inevitably merges them into an uninformative centroid. While one might propose decomposing the query into sub-questions (e.g., retrieving *Films by Nolan* and *Films by Cameron* separately), this effectively transforms the retriever into a multi-step Agent, inheriting the computational complexity and error propagation discussed below. Thus, pure single-step retrieval is severely constrained in practice and struggles to handle disjunctive logic without substantial architectural modifications.

6.2 Agent-based Methods

Current agents rely on graph traversal (e.g., Beam Search) to find answers. However, real-world KGs are scale-free networks where nodes often have high fan-out. This creates two critical failures. First, Fixed-Budget Pruning: To manage combinatorial explosion, current agents rely on pre-set hyperparameters (e.g., beam size $k = 5$). This Fixed-Budget approach is fatal for Union queries. As explicitly shown in Case 2 of Table 5, although the query requires directors from both *Pacific Rim* and *Beverly Hills Cop*, the agent pruned the latter branch entirely, failing to retrieve valid answers like R. Young because the model's search budget was saturated by the first branch. Second, Local-Global

Metric	Case 1: Nested Logic (pi)	Case 2: Disjunction ($2u$)
Question	What dog breed shares the same temperament trait with the *Basenji* and also comes in the color *Apricot*?	Which casting directors worked on *Pacific Rim* or *Beverly Hills Cop*?
Gold Triples	1. (Basenji, *temp.*, Curious) 2. (Curious, *dog_breeds*, **Schipperke**) 3. (Apricot, *breed_color*, **Schipperke**)	1. (Pacific Rim, *cast_dir.*, **M. Simkin**) 2. (Pacific Rim, *cast_dir.*, **R. D. Cook**) 3. (Bev. Hills Cop, *cast_dir.*, **M. Simkin**) 4. (Bev. Hills Cop, *cast_dir.*, **R. Young**)
GPT-5	{Poodle} *Error: Failed to combine multi-hop constraints.*	{M. Simkin, V. Thomas, B. Timmermann, R. Young} *Error: Hallucination & Incomplete results.*
ToG	*Retrieved Paths:* 1. (Apricot, *breed_color*, **Pug**) 2. (Basenji, *temp.*, **Curious**) 3. (Apricot, *breed_color*, **Eng. Mastiff**) **Pred:** {Pug, English Mastiff} *Error: Branch mismatch in constraint reasoning.*	*Retrieved Paths:* 1. (Bev. Hills Cop, *cast_dir.*, **M. Simkin**) 2. (Pacific Rim, *cast_dir.*, **M. Simkin**) 3. (Pacific Rim, *cast_dir.*, **R. D. Cook**) **Pred:** {M. Simkin, R. D. Cook} *Error: Pruned Bev. Hills Cop branch.*

Table 5: Case study comparing different paradigms. While LLMs and Agents perform well on linear paths, they suffer from a reasoning illusion in complex structures.

Mismatch: Agents exploring local paths often lose track of global constraints (Nested Logic). In Case 1 (Table 5), ToG successfully retrieved entities matching the color Apricot (e.g., Pug) based on local transition probability, but failed to verify the distant Basenji temperament constraint. This resulted in a branch mismatch where independent paths were retrieved but not correctly intersected, leading to logically invalid predictions.

6.3 Semantic Parsing

Semantic Parsing (SP) methods typically follow a generate-then-fill paradigm, constructing a logical skeleton before instantiating specific entities. This approach suffers from a dual failure mode. Structural Fragility: The generation of executable logical forms is error-prone. As noted in ChatKBQA (Luo et al., 2024), approximately 40.10% of errors stem from unexecutable logical formats. The structural gap between natural language and formal logic (e.g., SPARQL) makes generalizing to complex nested structures notoriously difficult. Instantiation Inheritance: Even when the logical skeleton is correct, the slot filling phase inherits the defects of retrieval-based methods. To fill the slots in a Union structure, the model must still perform entity linking or subgraph retrieval, which falls prey to the same representation dilemmas identified in Section 6.1, resulting in a cascading failure where correct logic is populated with incorrect entities.

6.4 Future Directions

Our analysis with ComplexLogicalQA reveals a fundamental mismatch between the linear processing of current models and the non-linear topology of complex logic. To resolve Disjunction Blindness in Union queries ($2u, up$), we must shift to Logic-Geometry Alignment. This ensures embeddings support logical disjunctions geometrically, rather than collapsing distinct branches into an ambiguous centroid. Simultaneously, handling branching logic requires Adaptive Resource Allocation. This replaces rigid heuristics with dynamic budgets that expand for disjunctions ($A \cup B$) while conserving resources for linear chains. Finally, addressing Structural Erosion in Nested Logic (ip) calls for Topology-Aware Decoding, where architectures natively respect the Directed Acyclic Graph (DAG) structure to prevent the logical inconsistencies caused by linearization.

7 Conclusion

We introduced ComplexLogicalQA, a benchmark designed to address the lack of logical diversity in existing KGQA datasets. By employing a logic-driven reverse-construction pipeline, we generated a rigorous and linguistically natural dataset across nine FOL query types. Our extensive evaluations across four paradigms expose a critical reasoning illusion, where current models fail to generalize to branching logic or ensure retrieval completeness despite excelling at linear patterns. ComplexLogicalQA provides a rigorous testbed to drive the development of next-generation KGQA systems that are more logically robust, grounded, and capable of handling sophisticated human inquiries.

Ethical Consideration

We have taken several steps to ensure our work adheres to ethical standards. **Data Provenance:** ComplexLogicalQA is built upon Freebase, a publicly available knowledge base released under a Creative Commons Attribution License. We ensure that no personally identifiable information (PII) is included in the sampled subgraphs or generated questions. **Potential Biases:** We acknowledge that Knowledge Graphs like Freebase may contain inherent societal biases regarding gender, race, or geography. While our benchmark focuses on logical structure, the underlying content may reflect these biases. We encourage users of our dataset to remain cognizant of these issues when training or evaluating models. Finally, we consider the **environmental impact** of our experiments; while benchmarking multiple SOTA models requires significant computational resources, we have optimized our evaluation pipeline and will release all model outputs to minimize the need for redundant computations by the community.

Limitations

Our benchmark construction primarily relies on the Freebase knowledge graph, which may limit the diversity of underlying data compared to Wikidata or proprietary graphs. However, Freebase's rigorous schema design renders it inherently more suitable for validating complex logical reasoning than the often noisy schemas of collaborative graphs. Furthermore, since our objective is to evaluate logical reasoning capabilities rather than the retrieval of up-to-date knowledge, the temporal deprecation of Freebase does not compromise the validity of the structural assessment. Additionally, we restrict our scope to Existential Positive First-Order (EPFO) logic, excluding the negation operator. This exclusion is necessitated by the Open World Assumption and the inherent incompleteness of knowledge graphs; verifying the correctness of questions involving negation is notoriously difficult, as missing edges cannot be definitively interpreted as false assertions. In future work, we aim to address these challenges and release updated versions of the dataset that incorporate broader knowledge sources and verifiable negation constraints.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

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

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. In *Advances in Neural Information Processing Systems*, volume 37, pages 132876–132907. Curran Associates, Inc.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and 1 others. 2025a. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Xiang Huang, Sitao Cheng, Yuheng Bao, Shanshan Huang, and Yuzhong Qu. 2023. MarkQA: A large scale KBQA dataset with numerical reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10241–10259, Singapore. Association for Computational Linguistics.

Xiang Huang, Jiayu Shen, Shanshan Huang, Sitao Cheng, Xiaxia Wang, and Yuzhong Qu. 2025b. TARGA: Targeted synthetic data generation for practical reasoning over structured data. In *Proceedings*

of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2704–2726, Vienna, Austria. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Haoran Luo, E Haihong, Zichen Tang, Shiyao Peng, Yikai Guo, Wentai Zhang, Chenghao Ma, Guanting Dong, Meina Song, Wei Lin, and 1 others. 2024. Chatkbqa: A generate-then-retrieve framework for knowledge base question answering with fine-tuned large language models. In *Findings of the association for computational linguistics: ACL 2024*, pages 2039–2056.

Francois Luus, Prithviraj Sen, Pavan Kapanipathi, Ryan Riegel, Ndivhuwo Makondo, Thabang Lebese, and Alexander Gray. 2021. Logic embeddings for complex query answering. *Preprint*, arXiv:2103.00418.

Costas Mavromatis, Soji Adeshina, Vassilis N. Ioannidis, Zhen Han, Qi Zhu, Ian Robinson, Bryan Thompson, Huzefa Rangwala, and George Karypis. 2025. BYOKG-RAG: Multi-strategy graph retrieval for knowledge graph question answering. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 27869–27886, Suzhou, China. Association for Computational Linguistics.

OpenAI. 2025. GPT-5 system card. System card, OpenAI.

Boci Peng, Yongchao Liu, Xiaohe Bo, Jiaxin Guo, Yun Zhu, Xuanbo Fan, Chuntao Hong, and Yan Zhang. 2025. M^{3}GQA: A multi-entity multi-hop multi-setting graph question answering benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30594–30620, Vienna, Austria. Association for Computational Linguistics.

Hongyu Ren*, Weihua Hu*, and Jure Leskovec. 2020. Query2box: Reasoning over knowledge graphs in vector space using box embeddings. In *International Conference on Learning Representations*.

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

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651, New Orleans, Louisiana. Association for Computational Linguistics.

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

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yao Xu, Shizhu He, Jiabei Chen, Zihao Wang, Yangqiu Song, Hanghang Tong, Guang Liu, Jun Zhao, and Kang Liu. 2024. 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*, pages 18410–18430, Miami, Florida, USA. Association for Computational Linguistics.

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

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

A Structural Analysis of Existing Benchmarks

In Section 4.4, we analyzed the structural bias of WebQSP, CWQ, and GrailQA. This appendix details the classification criteria and mapping rules used for this analysis.

A.1 WebQSP and CWQ (Human Annotation)

Since WebQSP and CWQ provide ground truth in SPARQL format, structural identification requires analyzing the graph topology implied by the query. We instructed annotators to visualize the SPARQL query as a computation graph and classify it according to the following taxonomy:

- **Chain Structures ($1p, 2p, 3p$):** Defined by a sequence of triple patterns where the object of one triple serves as the subject of the next (e.g., $?y \leftarrow (e, r_1, ?x) \wedge (?ans \leftarrow (?x, r_2, ?y))$ is classified as $2p$).
- **Intersection ($2i, 3i$):** Identified by multiple triple patterns sharing the same variable as the object, representing a conjunction of constraints (e.g., $?ans \leftarrow (?x, r_1, ?ans) \wedge (?y, r_2, ?ans)$).
- **Nested Logic (ip, pi):** Defined as a chain structure where one node is further constrained by an intersection (e.g., $?ans$ is the object of a chain, and the intermediate node is an intersection of two other entities).
- **Union ($2u, up$):** Identified by the presence of the `UNION` keyword in SPARQL, connecting disjoint graph patterns that map to the same answer variable.

Handling Ambiguity and Topological definitions. To ensure structural precision, we enforced three rigorous classification protocols during the manual annotation of WebQSP and CWQ:

- **CVT Node Expansion:** We strictly adhere to the physical graph topology regarding Freebase's Compound Value Types (CVT). A traversal passing through a CVT node is counted as two distinct hops (e.g., Entity $\xrightarrow{r_1}$ CVT $\xrightarrow{r_2}$ Value is classified as $2p$), rather than semantically collapsing it into a single relation. This prevents underestimating the reasoning depth required by the model.
- **Reasoning Path vs. Shortest Path:** We explicitly reject the common heuristic of categorizing questions based solely on the shortest path distance between anchor and answer entities. This metric is imprecise for two reasons: (1) **Path Mismatch:** The intended semantic logic often follows a specific, longer path (e.g., "spouse's co-star") which may differ from a coincidental direct link found by shortest-path algorithms; and (2) **Extended Verification:** Some questions require reasoning steps that extend *beyond* the candidate answer to verify a property (e.g., "What actor played in X and [actor] won award Y?"), effectively adding a hop that shortest-path metrics ignore.
- **Intersection vs. Linear Qualifiers:** A common ambiguity in CWQ arises between "Co-occurrence" and "Intersection". We established a strict rule: a query is classified as Intersection ($2i/3i$) only if the constraints explicitly narrow down the candidate set from multiple *independent* branches. Linear qualifiers (e.g., limiting a movie by its release date) are classified as part of the chain ($2p$) rather than a topological intersection.

A.2 GrailQA (Script-Based Parsing)

GrailQA uses S-expressions, which allows for deterministic mapping. We developed a parser that traverses the S-expression tree and counts the nesting depth and operator types. The mapping rules are as follows:

- **JOIN**: Treated as a projection step. A single `JOIN` represents $1p$. Nested `JOIN`s (e.g., `(JOIN r2 (JOIN r1 e))`) increase the path length ($2p, 3p$).
- **AND**: Treated as an Intersection node.
  - If `AND` operates on two 1-hop paths originating from different entities, it is mapped to $2i$.
  - If `AND` is nested within a `JOIN`, it is mapped to nested logic (ip or pi).
- **OR**: Although GrailQA's schema supports logical disjunction theoretically, our script scanned all 1,000 sampled S-expressions and found zero instances of the `OR` operator used for merging distinct reasoning paths (Union), confirming the 0% statistic reported in Table 2.

- **Comparatives (e.g., GT, LT)**: These were excluded from the topological count as they function as filters rather than structural hops.

B Human Evaluation Protocols

To ensure the high quality of ComplexLogicalQA, we conducted a rigorous human evaluation. This section details the annotator profiles, evaluation metrics, and the specific guidelines provided to the annotators.

B.1 Annotator Profiles

We recruited three graduate students as annotators. All annotators met the following qualifications:

- Advanced proficiency in English.
- Specific research experience in Knowledge Graph Question Answering (KGQA) ensuring familiarity with logical forms such as Intersection and Union.
- Prior experience with the Freebase schema.

B.2 Evaluation Criteria

Annotators were presented with the generated Natural Language Question (NLQ), the underlying Logical Form and Triples, and the Answer Set. They were asked to rate each sample on a Pass/Fail basis according to the following two dimensions:

Criterion 1: Linguistic Fluency. A question is marked as **Pass** if:

- It is grammatically correct and free of typos.
- It reads naturally like a question a human would ask, rather than a robotic translation of a database query.
- It avoids redundancy (e.g., "What is the movie that implies the film...").

Criterion 2: Logical Consistency. A question is marked as **Pass** if and only if:

- **Accuracy:** The question accurately reflects all constraints in the logical form (e.g., if the logic specifies "Intersection," the question may imply "both" or "and").
- **No Hallucination:** The question does not introduce extra constraints not present in the logic (e.g., adding "famous actor" when the logic only specifies "actor").
- **No Ambiguity:** The question has a unique interpretation corresponding to the provided logical structure.

B.3 Evaluation Results

Table 6 presents the detailed breakdown of the human evaluation results. The overall pass rate (samples satisfying both criteria) is 96.0%.

Logical Structure	Fluency (%)	Consistency (%)	Overall Pass (%)
Chain (1p, 2p, 3p)	99.3	98.7	98.0
Intersection (2i, 3i)	98.6	97.0	96.3
Nested (ip, pi)	97.5	95.2	94.5
Union (2u, up)	96.8	94.5	93.8
Total Average	**98.1**	**96.4**	**96.0**

Table 6: Detailed human evaluation results. Complex structures pose higher challenges but maintain a robust pass rate ($> 93\%$).

C Baseline Methodologies and Implementation

To ensure a comprehensive evaluation, we benchmarked methods across four paradigms. For foundational baselines (LLM reasoning and pure retrieval), we implemented customized pipelines optimized for the ComplexLogicalQA setting. For complex architectures (Agents, Semantic Parsing, and Advanced RAG), we reproduced results using their official open-source implementations to ensure fairness and reproducibility.

C.1 Implemented Baselines

Chain-of-Thought (CoT) and Self-Consistency (SC) We utilized GPT-5 as the backbone model for these reasoning baselines. For **Chain-of-Thought (CoT)**, we implemented a few-shot prompting strategy. The prompt includes three diverse demonstrations (one linear, one intersection, and one union query) to guide the model in breaking down complex logic step-by-step before generating the final answer. The temperature was set to `0.0` to maximize deterministic output. For **Self-Consistency (SC)**, we sampled $k = 5$ distinct reasoning paths for each query by increasing the temperature to `0.7`. The final answer was determined via majority voting over the generated entity sets, aiming to mitigate stochastic hallucinations common in single-pass generation.

Sparse and Dense Retrieval These methods treat KGQA as a document retrieval task, fetching relevant triples to augment the LLM. **Sparse Retrieval**

(BM25): We linearized the Knowledge Graph triples into text documents and used the BM25 algorithm to rank them based on lexical overlap with the query. We set the parameters $k_1 = 1.2$ and $b = 0.75$, retrieving the top-10 triples to ensure high recall. **Dense Retrieval**: We employed a bi-encoder architecture based on `all-MiniLM-L12-v2`. During inference, we encoded the question and graph triples into dense vectors and retrieved the top-10 semantic matches based on cosine similarity.

C.2 Reproduced Baselines

For the following advanced methods, we strictly followed their original papers and utilized their official open-source codes for reproduction.

BYOKG-RAG (Retrieval-based) BYOKG-RAG (Mavromatis et al., 2025) is an advanced retrieval framework that goes beyond simple similarity matching. It employs a multi-strategy retrieval mechanism combining dense retrieval with graph-aware exploration to fetch relevant subgraphs. We used its default configuration.

Think-on-Graph (ToG) and Generate-on-Graph (GoG) (Agent-based) These methods model the QA process as an agent navigation task over the KG. **Think-on-Graph (ToG)** (Sun et al., 2024) utilizes an LLM agent to iteratively search for evidence paths using beam search. It assesses the probability of relation paths to prune the search space. **Generate-on-Graph (GoG)** (Xu et al., 2024) integrates the LLM as both the agent and the knowledge graph, generating potential reasoning paths and validating them against the structured KG. For both methods, we set the beam size $k = 3$ and the maximum reasoning depth to 3 as its default configuration.

TARGA and Qwen3-8B-SP (Semantic Parsing) Semantic parsing methods aim to translate natural language questions into executable logical forms (e.g., SPARQL). **TARGA** (Huang et al., 2025b) utilizes a targeted synthetic data generation pipeline to train parsers that are robust to structural variations. **Qwen3-8B-SP** (Yang et al., 2025) is a specialized fine-tuned version of the Qwen3 model optimized for text-to-SPARQL tasks. We fine-tuned the Qwen3-8B base model on the training split of ComplexLogicalQA to serve as a specialized semantic parser for text-to-SPARQL generation. To achieve parameter-efficient adaptation, we utilized Low-Rank Adaptation (LoRA). The LoRA configuration was set with a rank $r = 64$, alpha $\alpha = 16$, and a dropout rate of 0.05, targeting the query and value projection layers. The model was trained for `3` epochs using the AdamW optimizer with a learning rate of `2e-4`, a global batch size of `32`, and a cosine learning rate scheduler with a `3%` warmup period.

D Dataset Samples

Table 7 provides concrete examples for each of the nine logical structures in ComplexLogicalQA. These samples illustrate the linguistic diversity and the structural complexity of the questions generated by our pipeline.

E Prompt Templates

We provide the detailed prompt templates used in our logic-driven reverse-construction pipeline.

Type	Question	Answer	Triples
1p	Which country used the *East German mark* as its former currency?	[East Germany]	(East German mark, finance.currency.countries_formerly_used, East Germany)
2p	What are the colors of the team owned by *Tom Hicks*?	[White, Red, Blue]	(Tom Hicks, sports.sports_team_owner.teams_owned, Texas Rangers) (Texas Rangers, sports.sports_team.colors, Red) (Texas Rangers, sports.sports_team.colors, White) (Texas Rangers, sports.sports_team.colors, Blue)
3p	What is the hex triplet of the color associated with the team that *Lou Seal* is the mascot for?	[000000]	(Lou Seal, sports.mascot.team, San Francisco Giants) (San Francisco Giants, sports.sports_team.colors, Black) (Black, base.schemastaging.visual_color_extra.hex_triplet, 000000)
2i	Which cat breeds are both *Gentle* and *Quiet*?	[Birman, American Shorthair]	(Gentle, base.petbreeds.cat_temperament.cat_breeds, Birman) (Gentle, base.petbreeds.cat_temperament.cat_breeds, American Shorthair) (Quiet, base.petbreeds.cat_temperament.cat_breeds, Birman) (Quiet, base.petbreeds.cat_temperament.cat_breeds, American Shorthair)
3i	Which film was directed by *Clemente Fracassi* and produced by *Gregor Rabinovitch* and *Federico Teti*?	[Aida]	(Clemente Fracassi, film.director.film, Aida) (Gregor Rabinovitch, film.producer.film, Aida) (Federico Teti, film.producer.film, Aida)
ip	When did the mother of *Richard, 1st Earl of Cornwall* and *Aymer de Valence* die?	[1246-05-31]	(Richard 1st Earl of Cornwall, people.person.parents, Isabella of Angoulême) (Aymer de Valence, people.person.parents, Isabella of Angoulême) (Isabella of Angoulême, people.deceased_person.date_of_death, 1246-05-31)
pi	Which battle in *Artois* was led by a commander who also fought in the *First Battle of the Marne*?	[Third Battle of Artois]	(First Battle of the Marne, base.culturalevent.event.entity_involved, Fernand de Langle de Cary) (Fernand de Langle de Cary, military.military_person.participated_in_conflicts, Third Battle of Artois) (Artois, location.location.events, Third Battle of Artois)
2u	Which projects were designed by either *Jacobs Engineering Group* or *HDR, Inc.*?	[Cleveland Clinic Abu Dhabi, Bandra–Worli Sea Link, Mike O'Callaghan–Pat Tillman Memorial Bridge]	(Jacobs Engineering Group, architecture.architecture_firm.projects, Mike O'Callaghan–Pat Tillman Memorial Bridge) (Jacobs Engineering Group, architecture.architecture_firm.projects, Bandra–Worli Sea Link) (HDR, Inc., architecture.architecture_firm.projects, Mike O'Callaghan–Pat Tillman Memorial Bridge) (HDR, Inc., architecture.architecture_firm.projects, Cleveland Clinic Abu Dhabi)
up	What royal lineage is connected to the state where Mukarram Jah, the child of *Azam Jah* and grandchild of *Prince Azmet Jah*, belonged?	[Hyderabad State]	(Azam Jah, people.person.children, Mukarram Jah) (Azam Jah, people.person.children, Princess Esra) (Azam Jah, people.person.children, Muffakham Jah) (Prince Azmet Jah, people.person.parents, Mukarram Jah) (Prince Azmet Jah, people.person.parents, Princess Esra) (Prince Azmet Jah, people.person.parents, Muffakham Jah) (Mukarram Jah, royalty.monarch.royal_line, Hyderabad State)

Table 7: Samples of generated questions across nine logical structures in ComplexLogicalQA. To enhance readability, reasoning triples are listed sequentially.

Generator Prompt

Role: You are an expert Knowledge Graph Question Answering (KGQA) dataset creator. Your task is to verbalize a logical subgraph into a fluent, natural language question.
Input Data (JSON):
- `structure`: The EPFO structure type (e.g., Intersection-Projection (ip), Union (2u)).
- `triples`: A list of reasoning triples forming the path.
- `anchors`: The starting entities that MUST appear in the question.
- `answers`: The final answer entities that MUST be hidden.
- `intermediates`: The bridging entities that MUST be hidden.

Strict Constraints:
1. **Anchor Grounding:** You must explicitly mention the entities listed in `anchors`.
2. **Secrecy:** Do NOT mention `answers` or `intermediates` by name. Use generalized types (e.g., "the director", "the country") instead.
3. **Logical Fidelity:**
   - For **Intersection** (2i, ip), use conjunctions like "and", "both".
   - For **Union** (2u, up), use disjunctions like "or", "either".
4. **Complexity:** The question must require reasoning over ALL provided triples.

Input Example: `{ "structure": "ip", "anchors": ["Inception", "Titanic"], "intermediates": ["Leonardo DiCaprio"], "answers": ["USA"], "triples": [ ["DiCaprio", "starred_in", "Inception"], ["DiCaprio", "starred_in", "Titanic"], ["DiCaprio", "nationality", "USA"] ] }`
Output Example: "What is the nationality of the actor who starred in both Inception and Titanic?"

Auditor Prompt (Refinement)

Role: You are a strict Linguistic and Logical Auditor. Your task is to review the generated question against the ground truth graph data to ensure it is fluent, natural, and logically precise.
Input Data (JSON):
- `structure`: The EPFO structure type (e.g., Union 2u).
- `triples`: The ground truth reasoning triples.
- `anchors`: The starting entities.
- `answers`: The expected answer entities.
- `intermediates`: The bridging entities.
- `question`: The generated natural language question to be audited.

Evaluation Criteria:
1. **Linguistic Fluency:**
   - Is the grammar correct?
   - Is the phrasing natural, human-like, and free of ambiguity?
2. **Logical Consistency:**
   - Does the question accurately reflect the intended `structure`?
   - **Critical for Union:** Does it clearly use disjunctive phrasing (e.g., "or", "either")?
   - **Critical for Intersection:** Does it clearly imply mandatory conjunction (e.g., "both", "and")?
3. **Refinement:** If the question is robotic or logically mismatched, provide a corrected version.

Output Example: `{ "status": "FAIL", "reason": "The question uses 'and' but the structure is Union.", "refined_question": "Who directed either Inception or Interstellar?" }`

Figure 4: Prompt templates used in our pipeline. **Top:** The Generator Prompt, showing the input constraints. **Bottom:** The Auditor Prompt, focusing on linguistic and logical verification.