MARCH: Evaluating the Intersection of Ambiguity Interpretation and Multi-hop Inference

Anonymous ACL submission

Abstract

Real-world multi-hop QA is naturally linked with ambiguity, where a single query can trigger multiple reasoning paths that require independent resolution. Since ambiguity can occur at any stage, models must navigate layered uncertainty throughout the entire reasoning chain. Despite its prevalence in real-world user queries, previous benchmarks have primarily focused on single-hop ambiguity, leaving the complex interaction between multi-step inference and layered ambiguity underexplored. In this paper, we introduce **MARCH**, a benchmark for their intersection, with 2,209 multi-hop ambiguous questions curated via multi-LLM verification and validated by human annotation with strong agreement. Our experiments reveal that even state-of-the-art models struggle with MARCH, confirming that combining ambiguity resolution with multi-step reasoning is a significant challenge. To address this, we propose **CLARION**, a two-stage agentic framework that explicitly decouples ambiguity planning from evidence-driven reasoning, significantly outperforms existing approaches, and paves the way for robust reasoning systems.

1 Introduction

Multi-hop Question Answering (QA) presents a significant reasoning challenge, requiring models to construct logical chains by connecting disparate pieces of information scattered across multiple documents (Trivedi et al., 2022; Yang et al., 2018; Ho et al., 2020b). Ambiguity in QA can further complicate this process, as a single query may stem from polysemous terms or insufficient context, demanding clarification or interpretation before an answer can be derived (Min et al., 2020). The intersection of these two challenges—multi-hop reasoning and ambiguity—creates a uniquely difficult setting where uncertainty scales exponentially. In multi-hop ambiguous QA, ambiguity can emerge at any step of the reasoning chain, often remaining latent until prior steps are resolved. This interdependence means that errors in resolving early-stage ambiguity propagate downstream, causing models to prematurely commit to incorrect reasoning paths and producing incomplete or flawed answers.

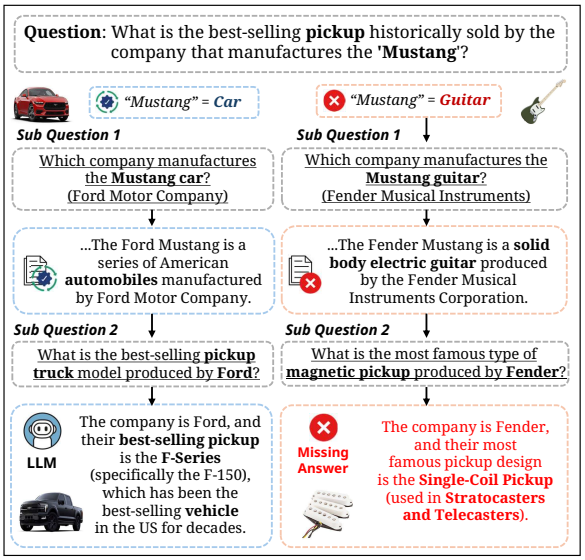

Figure 1: An example of multi-hop ambiguity QA. The ambiguity of the second hop ("pickup") is latent; it is only detectable if the alternative interpretation of the first hop ("Mustang" as guitar) is preserved.

Figure 1 illustrates the challenge of multi-hop ambiguous QA. In *"What is the best-selling **pickup** historically sold by the company that manufactures the 'Mustang'?"*, ambiguity arises from the interaction of *Mustang* and *pickup*. Because *pickup* is polysemous (truck vs. guitar component), the second-hop ambiguity is *latent* and only surfaces if we keep both *Mustang* interpretations. Current LLMs often commit early to the car reading and prune the valid guitar→magnetic-pickup branch. We observe that this layered ambiguity is not a rare edge case. An analysis of real-world user queries from the *lmsys-chat-1m corpus* (Zheng et al., 2024) (Figure 2, top) reveals that 48.4% of questions are ambiguous, 17.7% involve multi-hop reasoning, and 13.3% overlap. Despite this prevalence, empir-

	Single-hop	Multi-hop	Total
Ambiguous	**1010** (48.4%)	**277** (13.3%)	**1287** (61.7%)
Non-ambig	**428** (20.5%)	**370** (17.7%)	**798** (38.3%)
Total	**1438** (69.0%)	**647** (31.0%)	**2085** (100%)

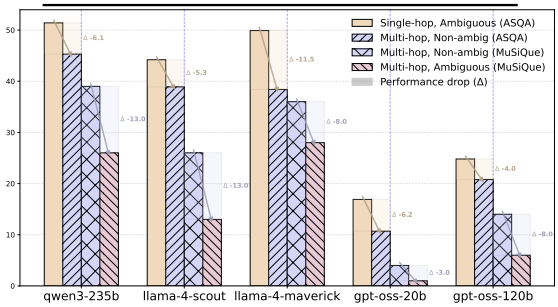

Figure 2: Multi-hop ambiguity prevalence (top) and performance drops (bottom).

ical results (Figure 2, bottom) show that datasets like MuSiQue (Multi-hop) (Trivedi et al., 2022) and ASQA (Ambiguous) (Stelmakh et al., 2022) suffer substantial performance drops when these features intersect. This underscores the need for benchmarks that specifically test the ability to hold multiple reasoning paths in superposition.

To address this, we introduce **M**ulti-hop **A**mbiguity **R**easoning **CH**ain (**MARCH**), a benchmark designed to evaluate this intersection of ambiguity interpretation and multi-step inference. MARCH contains **2,209** ambiguous multi-hop questions derived from MuSiQue, each paired with type-specific clarified questions (covering multiple interpretations), interpretation-grounded short answers, supporting evidence passages, and a synthesized long answer. We construct MARCH from MuSiQue through a rigorous pipeline involving multi-LLM verification to ensure quality. To address concerns about LLM-generated data quality, we validate a stratified sample with five human annotators, confirming high long-answer validity (over 90% integrate all interpretations) and strong inter-annotator agreement (Fleiss' κ up to 0.95). Our experiments with MARCH reveal that even state-of-the-art models struggle to resolve these layered ambiguities, often producing incomplete or one-sided answers.

To overcome this, we propose **CLARION** (**CL**arifying **A**mbiguity with a **R**easoning and **I**nstructi**ON**), a two-stage agentic framework that decouples *ambiguity planning* from *evidence retrieval*. By explicitly mapping out diverging interpretations via a Planning Agent *before* acting, CLARION prevents the premature pruning of latent branches. Empirical results demonstrate that CLARION significantly outperforms standard baselines, validating the necessity of separating ambiguity resolution from the retrieval loop. To isolate the source of difficulty, we also run the baselines on single-hop ambiguity and standard multi-hop datasets, where modern baselines are often reasonably capable in each setting alone. We then see this capability fail on MARCH, where early interpretation choices lock in bridge entities and make ambiguity latent and path-dependent across hops.

Type (LLM Action)	Definition & Typical Cues
Semantic (Interpret)	Homonyms/aliases or entity-name collisions (one name, multiple entities); cues: homonyms/aliases, acronym collisions, entity-name clashes.
Syntactic (Resolve)	Multiple valid parses of the same query; cues: pronouns, ellipsis, PP attachment, coordination, quantifier scope.
Constraint (Generalize)	Over-specific query where a broader related query better matches intent; cues: comparatives/superlatives, vague heads, "overview vs. details".

Table 1: Taxonomy of multi-hop ambiguity in QA, paired with an LLM action and typical detection cues.

2 Multi-Hop Ambiguous QA

We define multi-hop ambiguous QA as a task where a single input query triggers multiple valid reasoning chains, requiring the system to resolve uncertainties that determine the trajectory of multi-step inference. In multi-hop ambiguous QA, each valid interpretation dictates a unique sequence of intermediate decomposition steps (e.g., identifying different bridge entities), which in turn necessitates retrieving disjoint sets of evidence documents. Consequently, failing to resolve ambiguity at the initial or intermediate hops leads to a cascading failure, where the reasoning agent pursues an irrelevant trajectory and cannot recover the correct final answer. To systematically analyze these challenges, we extend the standard ambiguity taxonomy to the multi-hop setting. Following prior ambiguity definitions (Tang et al., 2025; Tanjim et al., 2025), we extend the taxonomy to multi-hop QA and group ambiguous questions into three types—*semantic*, *syntactic*, and *constraint* (Table 1).

Multi-hop Semantic Ambiguity (Entity-Driven Divergence). Semantic ambiguity arises when a mention can map to multiple entities/concepts (e.g., homonymy/entity collision), yielding disjoint evidence trails; choosing the wrong entity invalidates downstream hops. For the example in Figure 1: *"What is the best-selling pickup sold by the manufacturer of the 'Mustang'?"* branches by the bridge

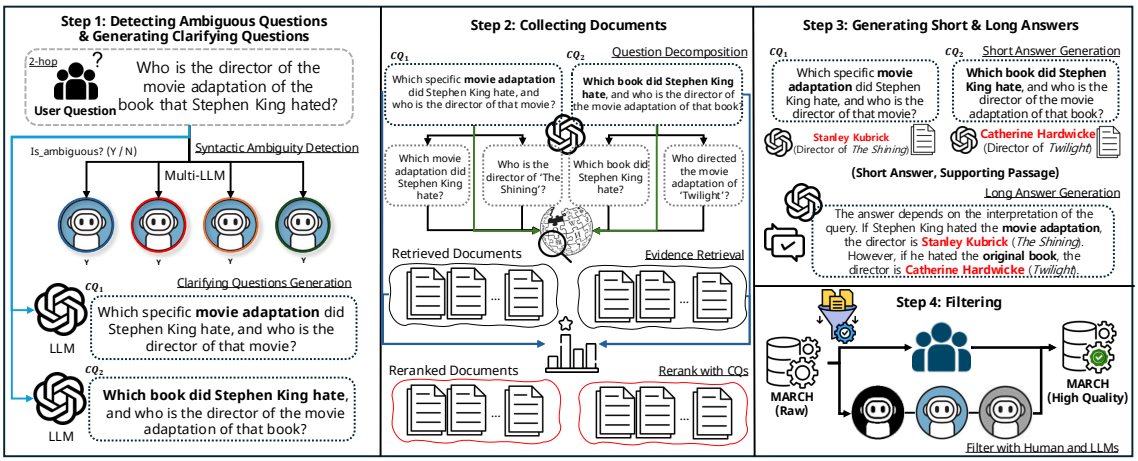

Figure 3: Overview of the four-stage MARCH dataset construction pipeline.

entity: **Car:** Mustang→Ford→vehicle sales→F-Series vs. **Guitar:** Mustang→Fender→pickup types→Single-coil. Thus, the system must INTERPRET the mention to select the intended bridge.

Multi-hop Syntactic Ambiguity (Structure-Driven Branching). Syntactic ambiguity occurs when multiple valid parses induce different inter-hop dependencies, changing which intermediate evidence is needed. For *"What is the model of the telescope the detective saw the suspect with?"*, **Instrumental:** detective used the telescope→find equipment→model vs. **Attributive:** suspect had the telescope→find possession→model. The system must RESOLVE the parse to construct the correct decomposition plan.

Multi-hop Constraint Ambiguity (Scope-Driven Pruning). Constraint ambiguity occurs when an over-specific modifier is unnecessary or mismatched with how evidence is written, causing a valid chain to be pruned early. Example: *"What is the capital of the country where the highest mountain in Europe is located?"* Many sources disagree on whether the highest mountain in Europe is *Mount Elbrus* (Caucasus) or *Mont Blanc* (Alps), and some pages simply say "Europe's highest mountain" without committing. If the system enforces the modifier literally, it may retrieve only one interpretation and miss the other. A robust strategy is to GENERALIZE (relax) the constraint (e.g., consider both candidates) and then verify the remaining hop (country → capital). For more details for this taxonomy, see Appendix I.

3 MARCH: A Benchmark for Multi-Hop Ambiguous QA

We introduce MARCH, a benchmark designed to evaluate ambiguity resolution and multi-hop reasoning in question answering jointly. We process ambiguous questions from MuSiQue through four stages to build MARCH: (1) Ambiguity detection and clarification; (2) Document collection; (3) Generation of short answers for each interpretation and a long answer; and (4) Filtering, as in Figure 3.

3.1 Dataset Construction

We build MARCH from MuSiQue's validation set and a subset of its training set. Unlike other multi-hop benchmarks (Zhu et al., 2024; He et al., 2024) that are narrow in domain or inflate hops with list-style questions (e.g., top-5), MuSiQue enforces connected, dependency-linked reasoning across diverse domains. We first filter out questions from MusiQue that lack ambiguity and retain only those judged as ambiguous by our multi-stage pipeline. Let $\mathcal{Q}_{\text{base}}$ be the set of base multi-hop questions. We consider three ambiguity types $\mathcal{T} = \{\text{Semantic}, \text{Syntactic}, \text{Constraint}\}$. We use a set of off-the-shelf LLMs as detectors; for a question $q \in \mathcal{Q}_{\text{base}}$, type $t \in \mathcal{T}$, and detector m, let $y_{m,t}(q) \in \{0, 1\}$ denote whether q is judged ambiguous of type t.

Step 1. Detecting Ambiguous Questions & Generating Clarified Questions. For each question of MuSiQue, we provide definitions of each ambiguity type and ask multiple LLMs to detect type-wise ambiguity. We employ four detectors: *gpt-4.1* (Achiam et al., 2023), *llama-4-maverick* (AI, 2025), *qwen3-235b-a22b* (Yang et al., 2025), and *claude-sonnet-4* (Anthropic, 2025). We keep a type label only when the detectors are *fully in agreement*.

Let $\mathcal{M}$ be the detector set ($|\mathcal{M}| = 4$). For a question q and type $t \in \mathcal{T}$, detector $m \in \mathcal{M}$ outputs $y_{m,t}(q) \in \{0, 1\}$. We define the full-agreement

rule:

$$\phi_t(q) = \mathbb{I}\left(\frac{1}{|\mathcal{M}|}\sum_{m\in\mathcal{M}} y_{m,t}(q) = 1\right),$$

$$\mathcal{T}(q) = \{\, t \in \mathcal{T} \mid \phi_t(q) = 1 \,\}.$$

where $\mathcal{T}(q)$ denotes the ambiguity types assigned to q. Under this rule, we assign a type t only if all four detectors judge q ambiguous of type t, reducing single-model bias and yielding high-quality labels.

For each (q, t) with $t \in T(q)$, we use *gpt-4.1* to generate clarified questions that resolve the type-t ambiguity while preserving the user's information need, denoted as $\mathcal{C}(q, t) = \{c_1, \ldots, c_n\}$ with $n \geq 2$. We utilize these clarified questions as retrieval inputs.

Step 2. Collecting Documents. To generate answers for the questions, we require evidence obtained via retrieval over clarified questions. However, clarified questions can remain multi-hop, and questioning with a specific multi-hop form may narrow the search scope and miss relevant documents. To mitigate this, we use *gpt-4.1* to decompose each clarified question $c \in \mathcal{C}(q, t)$ into atomic sub-questions $\mathcal{S}(c) = \{s_1, \ldots, s_k\}$. For every $s \in \mathcal{S}(c)$, we retrieve up to 10 candidate documents from English Wikipedia[1]. We then pool candidates $\mathcal{D}(c) = \bigcup_{s\in\mathcal{S}(c)} \mathcal{D}(s)$, and if $|\mathcal{D}(c)| < 10$, we additionally perform retrieval process with the clarified question c itself to back-fill more evidence. Next, we perform embedding-based re-ranking with *Qwen3-8B-Embedding* (Zhang et al., 2025b) by the similarity between the clarified question and document passages. Finally, we sort $\mathcal{D}(c)$ by this score to prioritize evidence aligned with the clarified interpretation.

Step 3. Generating Short and Long Answers. Given each clarified question c and its ranked candidate documents $\mathcal{D}(c)$, we use *gpt-4.1* to produce a short factual answer only when the retrieved evidence clearly supports it; otherwise, we omit the short answer and drop that clarified item. For retained cases, we also record the passage used for generating the short answer. Finally, for the original question $\mathcal{Q}_{\text{base}}$, we utilize *gpt-4.1* to write a single-sentence long answer that connects the two short answers into a coherent statement while incorporating interpretations and citations. For the detailed prompts, refer to Appendix J.

[1] https://dumps.wikimedia.org/

Step 4. Filtering. Before filtering, we ensure that short answers remain concise. If either clarified question has a short answer longer than 10 tokens, we cut it down to a much shorter form with *gpt-4.1*. After that, we remove cases where the two short answers are identical. (We provide representative examples after applying this collision rule in Appendix H.) After the filtering, the final MARCH dataset consists of **2,209** examples. For the final filtering stage, we exclude *gpt-4.1*—already used in Steps 2 and 3—and instead employ *llama-4-maverick*, *qwen3-235b-a22b*, and *claude-sonnet-4*. Each candidate instance, including the question, clarified questions, ambiguity type, supporting passages, short answers, and long answers, is independently checked for alignment in all fields. We retain only those cases where all three models unanimously judged the instance as fully aligned, using the same criteria as our human evaluation protocol. The upper side of Table 2 shows statistics and reports key characteristics of MARCH. MARCH also retains broad topical coverage; Appendix B reports the domain distribution of MARCH.

Stage	**Sem.**	**Syn.**	**Const.**	**Total**
MuSiQue (orig.)	24,834	24,834	24,834	—
After det. +clar.	9,544	8,642	11,703	29,889
After answer gen.	7,034	6,675	8,433	22,142
Before filtering	**1,651**	**1,239**	**1,440**	**4,330**
After filtering (final)	**734**	**739**	**736**	**2,209**
Avg. hops	2.44	2.95	2.11	—
Avg. Question length	14.92	18.17	16.18	—

Evaluation protocol	**Sem.**	**Syn.**	**Const.**	**Fleiss'κ**
Question is *type*-ambig.?	94.0%	94.0%	92.0%	0.92
Clarifications resolve ambiguity?	95.0%	91.0%	96.0%	0.89
Long answer matches short answers?	97.0%	97.0%	98.0%	0.95
All fields valid	**92.0%**	**89.0%**	**90.0%**	—

Table 2: **Top:** Statistics of the MARCH dataset filtering pipeline for each ambiguity type. **Bottom:** Human verification results after filtering, with Fleiss' κ indicating high inter-annotator agreement for each ambiguity type.

3.2 Dataset Analysis and Validation

Ambiguity Amplifies Reasoning Depth. We measure the difference in average hop counts between questions assigned ambiguous vs. unambiguous labels via multi-LLM consensus on the MuSiQue training set. After labeling the training questions, we randomly sample 1,000 from each group and compare their average hop counts. The average hop count for ambiguous questions is ***2.441***, while for unambiguous questions it is *2.074*. This result indicates that ambiguous questions generally involve more hops, making them

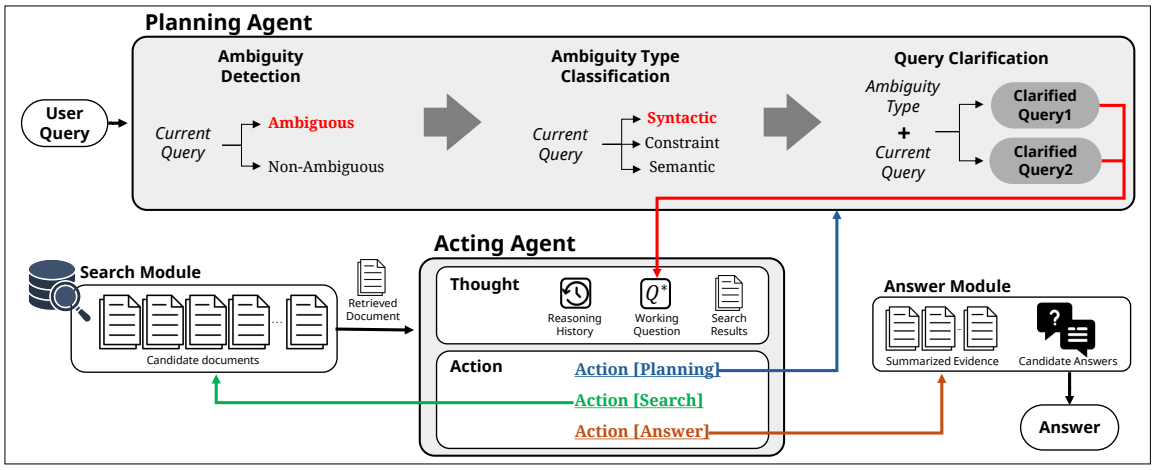

Figure 4: Overview of our CLARION framework. A *Planning Agent* resolves ambiguity, and an *Acting Agent* executes a ReAct loop to generate the final answer.

more challenging and underscoring the importance of addressing them effectively.

Dataset Quality Assessment. To assess dataset quality, we employ five annotators and use a majority-vote scheme. We first sample 20 instances from the final MARCH dataset per ambiguity type (60 total) and obtain binary (YES/NO) judgments for each item in our protocol, yielding 300 judgments in total (5 annotators $\times$ 60 items). As shown in the lower part of Table 2, annotators evaluate whether (i) the question exhibits the specified ambiguity, (ii) the clarified question resolves the original ambiguity, and (iii) the generated long answer contains the corresponding short answers. The results demonstrate not only high validity scores (e.g., $>$90% for all types) but also strong inter-annotator agreement, with Fleiss' Kappa scores of **0.92** for ambiguity detection, **0.89** for clarification quality, and **0.95** for answer consistency. These results confirm the reliability of our automated pipeline in producing high-quality, ambiguous multi-hop questions. Please refer to Appendix C for annotator details, labeling guidelines, and inter-annotator agreement statistics.

4 CLARION: An Agentic Framework for Multi-hop Ambiguous QA

To address multi-hop ambiguous QA, we propose CLarifying Ambiguity with a Reasoning and InstructiON (**CLARION**), a two-stage agentic framework. As in the example of Figure 1, multi-hop ambiguity is often latent; the second hop's ambiguity ("pickup") only surfaces if the first hop ("Mustang") is not prematurely resolved. Standard retrieval methods typically fail by committing to a single dominant intent (e.g., Ford cars), thereby pruning alternative paths (e.g., Fender guitars). CLARION overcomes this by explicitly decoupling **Planning** from **Acting**, as outlined in Figure 4.

Planning Agent. The *Planning Agent* serves as a planning module that analyzes the input question before any retrieval or answering. It performs three sequential operations: (1) **Ambiguity Detection**: the agent determines whether the question contains ambiguity. If the question is unambiguous, it is immediately passed to the *Acting Agent*. (2) **Ambiguity Type Classification**: if ambiguity is detected, the question is categorized into one of three predefined types: *Syntactic*, *Constraint*, or *Semantic*. (3) **Question Clarification**: based on the detected type, the agent rewrites the original question into clarified variants that resolve the ambiguity while preserving the information needed. These clarified questions constitute the execution plan for downstream reasoning.

Acting Agent. The *Acting Agent* executes the reasoning plan through a *ReAct-style prompting* (Yao et al., 2023) scheme, unfolding in a **Thought** $\rightarrow$ **Action** $\rightarrow$ **Observation** loop. Unlike naive ReAct, which typically follows a single reasoning trajectory whose early assumptions steer subsequent retrieval, CLARION separates interpretation enumeration from evidence gathering. The Planning Agent produces an initial set of clarified interpretations, and the Acting Agent runs a ReAct loop that retrieves and reasons for each interpretation before synthesizing the final long-form answer. Concretely, the Acting Agent maintains an interpretation set and collects evidence per interpretation, and is instructed to produce a final answer that explicitly covers all interpretations (or as many as pos-

Model	Method	STR-EM (%)	Disambig-F1 (%)	Avg (%)	LLM-as-a-Judge
Qwen3-235b	No Retrieval	20.98	21.19	21.09	3.083
	CoT	21.51	22.32	21.91	2.897
	NaiveRAG	25.10	26.20	25.65	2.752
	CoT w/ RAG	25.63	26.61	26.12	2.947
	DIVA	28.82	22.73	25.78	3.015
	ReAct	20.98	21.00	20.99	2.832
	CLARION (Ours)	**38.73**	**28.38**	**33.56**	**3.474**
	CLARION w/o clarification	25.10	25.56	25.33	2.922
	CLARION w/o clarification & detection	22.94	24.02	23.48	2.782
Gemini-2.5	No Retrieval	15.59	20.10	17.85	2.307
	CoT	16.32	17.52	16.92	2.258
	NaiveRAG	22.16	**28.63**	25.40	2.297
	CoT w/ RAG	23.15	27.31	25.23	2.373
	DIVA	18.82	20.29	19.56	2.303
	ReAct	21.32	22.37	21.84	2.428
	CLARION (Ours)	**29.12**	26.30	**27.71**	**2.752**
	CLARION w/o clarification	24.12	22.54	23.33	2.609
	CLARION w/o clarification & detection	23.82	22.04	22.93	2.573
DeepSeek-v3.1	No Retrieval	17.75	18.72	18.24	2.683
	CoT	19.80	22.12	20.96	2.512
	NaiveRAG	20.20	25.03	22.62	2.084
	CoT w/ RAG	21.33	23.18	22.25	2.632
	DIVA	18.82	20.66	19.74	2.636
	ReAct	23.17	24.78	23.97	2.723
	CLARION (Ours)	**31.47**	**27.03**	**29.25**	**3.042**
	CLARION w/o clarification	23.63	22.99	23.31	2.927
	CLARION w/o clarification & detection	24.51	24.31	24.41	2.906

Table 3: **MARCH** benchmark results with baseline reasoning methods. Metrics are scaled to percentages. Best per-model scores for each metric are in **bold**, and second-best scores are underlined.

sible under the iteration budget). At each iteration, the agent selects one of three actions: (1) **Search**: retrieve external documents when additional evidence is needed; (2) **Planning**: refine or expand the current interpretation set (e.g., add a missing interpretation) when the current plan is insufficient; (3) **Answer**: synthesize the final output once enough evidence has been gathered. To ensure reliable parsing and automated execution, all actions generated by the agent must be in JSON format. Furthermore, to prevent infinite loops and ensure computational tractability, the ReAct prompting is limited to a maximum of five iterations. If this limit is reached without a resolution, the agent is compelled to execute the *Answer* action, formulating the best possible response based on the information gathered thus far. We provide implementation details(models, decoding, retrieval hyperparameters) and the full prompt templates used at each stage in Appendix H and Appendix J, respectively.

5 Experiments and Results

Models. We evaluate three state-of-the-art LLMs widely used in real-world scenarios for MARCH: *qwen3-235b-a22b-2507* (Yang et al., 2025), *gemini-2.5-flash* (Comanici et al., 2025), and *deepseek-chat-v3.1* (Liu et al., 2024). To ensure fair comparison, all systems utilize the same retriever (based on *qwen3-embedding-8b*) and identical retrieval hyperparameters.

Evaluation metrics. We evaluate performance using three metrics: (1) **STR-EM (Strict Exact Match)**: the percentage of gold short answers that appear in the generated long answer after normalization. (2) **Disambig-F1**: We use a frozen extractive QA model as an evaluator. Given the model-generated long answer, we treat it as the context and, for each gold clarified question (one per interpretation), the QA model extracts a short answer span. We then compute token-level F1 against the corresponding gold short answer and average over all interpretations. (3) **LLM-as-a-Judge**: a GPT-4-based judge scores the long answer from 0–5 on *Relevance, Faithfulness, Informativeness,* and *Correctness*; we report the average score (validated in Appendix F.1).

Baselines. Given that multi-hop ambiguous QA inherently requires external knowledge, we employ search-based baselines for all experiments. **(1) No Retrieval**: LLM-only inference without any external context. **(2) CoT** (Wei et al., 2022): A standard Chain-Of-Thought prompting without re-

trieval. **(3) NaiveRAG** (Lewis et al., 2020): A standard retrieve-then-read pipeline that retrieves top-k passages using the original question. **(4) CoT with RAG** (Wei et al., 2022; Lewis et al., 2020): CoT prompting augmented with retrieved documents. **(5) DIVA** (In et al., 2025): A *diversify–verify–adapt* framework designed for ambiguous QA. It diversifies the query into multiple interpretations, verifies evidence for each (labeling passages as Useful/Partially Useful/Useless), and adapts its answering strategy based on evidence sufficiency. **(6) ReAct** (Yao et al., 2023): An agentic baseline that dynamically interleaves reasoning and retrieval. It generates *Thoughts* to plan, executes retrieval *Actions*, and uses *Observations* to iteratively refine its path.

Main Results and Discussion As shown in Table 3, CLARION consistently outperforms all baselines across all models and metrics. Notably, on *Disambig-F1*—which is most sensitive to the "missing branch" failure mode—CLARION achieves substantial gains. This confirms that CLARION uncovers and traverses latent reasoning paths of multi-hop QA that standard methods often prune.

We attribute this failure to the path-dependent nature of multi-hop ambiguity: committing to an early interpretation conditions downstream decomposition and retrieval, which can prevent alternative later-hop meanings from even surfacing. Moreover, evidence for different interpretations is often disjoint (via different bridge entities), so naive top-k retrieval tends to over-focus on the dominant branch and under-cover the others.

In particular, the comparison with ReAct underscores the critical role of explicit disambiguation. While ReAct dynamically retrieves information, it often prunes alternative interpretations early by following a single dominant reasoning trace, after which subsequent retrieval further reinforces that commitment. A similar failure arises in standard RAG-based baselines. Retrieving passages with the question over-focuses on the most frequent interpretation and under-covers evidence for other branches, especially when different interpretations induce disjoint bridge entities and documents. In contrast, CLARION's *Planning Agent* forces the exploration of divergent paths *before* retrieval, enabling interpretation-specific evidence collection and preventing premature commitment. Ablation studies further confirm our design choices. Removing the *Clarification* module causes the largest performance drop, demonstrating that without explicit query rewriting, even agentic systems fail to capture the user's multi-faceted intent.

Model	Ambiguity Type	STR-EM	Disambig-F1	LLM-as-a-Judge
Qwen3-235b	Syntactic	35.00	25.81	3.370
	Constraint	46.45	33.91	3.860
	Semantic	33.90	24.86	3.303
Gemini-2.5	Syntactic	29.67	26.53	2.652
	Constraint	32.79	28.71	3.059
	Semantic	24.86	23.61	2.637
DeepSeek-v3.1	Syntactic	29.33	27.17	2.840
	Constraint	34.43	28.78	3.316
	Semantic	30.23	25.11	2.940

Table 4: Performance by ambiguity type on **MARCH**. Values are percentages except LLM-as-a-Judge.

Performance by Ambiguity Type. Table 4 reports performance by ambiguity type across LLMs. We observe a consistent ordering: *Constraint* $>$ *Syntactic* $\approx$ *Semantic*. This gap mainly reflects whether ambiguity preserves or changes the evidence trail. For Constraint ambiguity, the competing readings often differ only by an over-specific modifier. Relaxing the constraint typically retains the same bridge entities and yields highly overlapping evidence, allowing models to recover the multi-hop chain even after a suboptimal early choice. In contrast, Syntactic and Semantic ambiguity more directly steer the hop structure and bridge selection, which can split the reasoning process into branch-specific trajectories. Once a model commits early, retrieval and decomposition become path-dependent, reinforcing that trajectory and causing premature pruning of alternatives.

Isolating Multi-hop and Ambiguity effects. To isolate the combined challenge of ambiguity interpretation and multi-hop inference, we evaluate NaiveRAG, DIVA, and ReAct, along with our CLARION, on ASQA, MuSiQue, which respectively probe single-hop ambiguity, standard multi-hop reasoning. From the result of Table 6, we find that modern RAG and agentic baselines perform reasonably well on ASQA and MuSiQue, and CLARION is competitive on both, indicating robustness to constraint ambiguity and standard multi-hop reasoning in isolation. We find that this success does not transfer to MARCH, where ambiguity is latent and path-dependent. Early interpretation choices fix bridge entities, steer downstream re-

Benchmark	Scale	Tasks	Multi-hop?	Short Ans.?	Long Ans.?	Ambig. Type Diversity?
AmbigQA	14,042	Ambiguous QA	✗	✓	✗	✗
CAMBIGNQ	5,653	Ambiguity Detection; Clarifying Question Generation	✗	✓	✗	✗
CondAmbigQA	200	Conditional Ambiguous QA	✗	✓	✗	✗
ASQA	5,301	Long-form QA	✗	✓	✓	✗
AmbigDocs	36,098	Ambiguous QA	✓	✓	✗	✗
DeepAmbigQA	3,600	Multi-hop QA; Answer Completeness; Name Ambiguity	✓	✓	✗	✗
MARCH (Ours)	**2,209**	**Multi-hop Ambiguity Detection** **Multi-hop Clarifying Question Generation** **Multi-hop Long-form QA**	✓	✓	✓	✓

Table 5: Comparison of ambiguous QA benchmarks and **MARCH**.

Method	ASQA		MuSiQue	
	STR-EM	Disambig-F1	EM	F1
NaiveRAG	75.95	38.50	6.20	51.87
DIVA	71.70	32.92	11.80	53.80
ReAct	82.34	40.83	13.47	54.27
CLARION	**91.18**	**48.78**	**17.07**	**55.60**

Table 6: ASQA and MuSiQue results averaged over three LLM backbones.

trieval, and prune alternative branches. As a result, baselines often over-focus on a single trajectory and mix cross-branch evidence, causing cascading errors across hops. CLARION targets this failure mode by separating interpretation planning from evidence-driven acting, retrieving per interpretation, and enforcing hop-consistent reasoning.

6 Related Work

Ambiguity in QA. Ambiguity in open-domain QA arises from polysemy or insufficient context, permitting multiple reasonable interpretations. Systems must typically clarify the user's intent or provide comprehensive answers covering all possibilities. AmbigQA (Min et al., 2020) formalized this problem with a disambiguation dataset, while ASQA (Stelmakh et al., 2022) introduced long-form answers to synthesize multiple interpretations. SituatedQA (Zhang and Choi, 2021) and AmbigDocs (Lee et al., 2024) expanded this scope by incorporating situational dependencies and conflicting evidence, respectively. To address such ambiguity, many approaches employ question clarification (Min et al., 2020; Zhang and Choi, 2025; Zhang et al., 2025a) or retrieval-augmented strategies. Recent RAG-based pipelines (Tanjim et al., 2025) disambiguate before retrieval, while methods like Tree of Clarifications (Kim et al., 2023) use branching retrieval to explore interpretations. Similarly, DIVA (In et al., 2025) adopts a diversify–verify–adapt framework to rewrite queries and synthesize answers from diverse evidence. ReAct (Yao et al., 2023) interleaves reasoning and tool-use in a Thought→Action→Observation loop, often following a single trajectory. However, these works primarily target single-hop ambiguity and do not address the compounding uncertainty inherent in multi-hop reasoning chains. CLARION instead enumerates interpretations first and retrieves/evaluates evidence per interpretation before synthesis.

Multi-hop QA. Multi-hop QA requires reasoning across multiple documents (He et al., 2024; Zhu et al., 2024; Tang and Yang, 2024). HotpotQA (Yang et al., 2018) targets the retrieval of articles and sentence-level facts, while 2WikiMultihopQA (Ho et al., 2020a) enhances explainability by providing structured evidence and reasoning paths. MuSiQue (Trivedi et al., 2022) mitigates shallow shortcuts found in prior datasets by enforcing connected reasoning through dependent single-hop questions. Unlike prior ambiguity benchmarks (e.g., AmbigQA, ASQA, CondAmbigQA (Li et al., 2025)), which target single-hop questions, MARCH targets the intersection of ambiguity and multi-step inference. As summarized in Table 5, MARCH evaluates the entire pipeline of multi-hop ambiguity detection, clarification, and answer generation.

7 Conclusion

We introduce MARCH, a benchmark designed to evaluate ambiguity in multi-hop question answering. MARCH consists of 2,209 carefully annotated questions that include type-specific clarifications, evidence-grounded short and long answers. Finally, we propose CLARION, an effective solution for MARCH, and we show that robust LLMs struggle when ambiguity and multi-hop reasoning co-occur. We find failures are largely path-dependent: early interpretation commitments lock in retrieval and trigger cascading multi-hop errors.

Limitations

CLARION demonstrates strong performance but introduces additional system complexity due to its multi-agent structure and planning-acting cycle. Although this complexity enables richer ambiguity resolution, future work could explore lighter-weight or more efficient designs without sacrificing effectiveness, making deployment and integration even more practical.

Despite CLARION's effectiveness, MARCH surfaces the ongoing difficulty of achieving complete and faithful resolution for all multi-hop ambiguous queries. Our results showcase both the progress and the remaining gaps in current methods, providing a solid foundation and clear direction for continued innovation in this important area.

Ethics Statement

MARCH benchmark is constructed entirely from publicly available data sources (MuSiQue, Wikipedia), ensuring that no personally identifiable or private information is present. We use a multi-LLM consensus pipeline for ambiguity detection and filtering, reducing the risk of individual model bias or hallucination. Expert contributors, with their consent conduct all human annotation, and no unfair labor practices are involved. While our dataset and evaluation pipeline strive to minimize bias, users should be aware that language models may still inherit subtle biases from the underlying data. We encourage responsible use and further analysis of potential risks when applying MARCH or derived models in real-world settings.

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*.

Meta AI. 2025. Llama 4 maverick: A 17b-128e multimodal language model. https://ai.meta.com/blog/llama-4-multimodal-intelligence/. Available from Meta AI Blog, April 5, 2025.

Anthropic. 2025. Claude sonnet 4. https://www.anthropic.com/claude/sonnet. Available on Claude Developer Platform / Anthropic site.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Jie He, Nan Hu, Wanqiu Long, Jiaoyan Chen, and Jeff Z Pan. 2024. Mintqa: A multi-hop question answering benchmark for evaluating llms on new and tail knowledge. *arXiv preprint arXiv:2412.17032*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020a. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020b. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625.

Yeonjun In, Sungchul Kim, Ryan A. Rossi, Mehrab Tanjim, Tong Yu, Ritwik Sinha, and Chanyoung Park. 2025. Diversify-verify-adapt: Efficient and robust retrieval-augmented ambiguous question answering. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1212–1233, Albuquerque, New Mexico. Association for Computational Linguistics.

Gangwoo Kim, Sungdong Kim, Byeongguk Jeon, Joonsuk Park, and Jaewoo Kang. 2023. Tree of clarifications: Answering ambiguous questions with retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 996–1009, Singapore. Association for Computational Linguistics.

Yoonsang Lee, Xi Ye, and Eunsol Choi. 2024. Ambigdocs: Reasoning across documents on different entities under the same name. In *First Conference on Language Modeling*.

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.

Zongxi Li, Yang Li, Haoran Xie, and S Joe Qin. 2025. Condambigqa: A benchmark and dataset for conditional ambiguous question answering. *arXiv preprint arXiv:2502.01523*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. Asqa: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288.

Anfu Tang, Laure Soulier, and Vincent Guigue. 2025. Clarifying ambiguities: on the role of ambiguity types in prompting methods for clarification generation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 20–30.

Yixuan Tang and Yi Yang. 2024. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries. In *First Conference on Language Modeling*.

Md Mehrab Tanjim, Yeonjun In, Xiang Chen, Victor S Bursztyn, Ryan A Rossi, Sungchul Kim, Guang-Jie Ren, Vaishnavi Muppala, Shun Jiang, Yongsung Kim, and 1 others. 2025. Disambiguation in conversational question answering in the era of llm: A survey. *arXiv preprint arXiv:2505.12543*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

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.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Michael JQ Zhang and Eunsol Choi. 2021. Situatedqa: Incorporating extra-linguistic contexts into qa. In *2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, pages 7371–7387. Association for Computational Linguistics (ACL).

Michael JQ Zhang and Eunsol Choi. 2025. Clarify when necessary: Resolving ambiguity through interaction with lms. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5526–5543.

Michael JQ Zhang, W Bradley Knox, and Eunsol Choi. 2025a. Modeling future conversation turns to teach llms to ask clarifying questions. In *The Thirteenth International Conference on Learning Representations*.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025b. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. 2024. LMSYS-chat-1m: A large-scale real-world LLM conversation dataset. In *The Twelfth International Conference on Learning Representations*.

Andrew Zhu, Alyssa Hwang, Liam Dugan, and Chris Callison-Burch. 2024. Fanoutqa: A multi-hop, multi-document question answering benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 18–37.

Appendix

A The Use of Large Language Models

We write the manuscript ourselves, and an LLM (ChatGPT-5.2) is used solely for refinement—style, clarity, and grammar. It is not used for ideation or content generation.

B Domain Diversity of MARCH

We tag the topic of each question using *gpt-oss-120b* and report the distribution in Table 7. As shown in Table 7, MARCH covers a broad range of subject areas. The most frequent topics include *"History"*, *"Geography&Places"*, and *"Politics&Government"*, indicating diverse coverage beyond any single domain.

Domain	Ratio
Science & Technology	2.79
Math & Logic	0.07
History	**30.88**
Geography & Places	20.44
Politics & Government	10.16
Business & Economics	2.49
Society & Culture	1.76
Arts & Literature	4.32
Entertainment (Film/TV/Games)	8.59
Music	7.48
Sports	6.44
Religion & Philosophy	2.31
Medicine & Health	0.18
Nature & Environment	1.57
UNKNOWN	0.51
Total	**100%**

Table 7: MARCH domain coverage.

C Details for Human Annotation

C.1 Details about Annotators

We employ five graduate-level annotators fluent in English for all labeling tasks in our study. Annotators were compensated at a rate of 10 USD per hour. Each annotator received detailed guidelines and example cases before annotation, and ambiguous cases were discussed through controlled calibration sessions. In total, evaluating the 60 sampled instances for our human assessment required approximately 10 hours of annotation effort. Reporting inter-annotator agreement is crucial for assessing the reliability of human judgments. Therefore, we compute Fleiss' κ, average category-wise agreement ($\bar{P}$), strict agreement (all annotators selecting the same label), and majority agreement (at least three annotators agreeing) for both (1) long-answer judgment dimensions (Relevance, Faithfulness, Informativeness, Correctness) and (2) dataset quality evaluation dimensions (Ambiguity, Clarification, Long Answer).

Item	Fleiss' κ	$\bar{P}$	Strict Agree	Maj. Agree
Relevance	1.000	1.000	1.000	1.000
Faithfulness	1.000	1.000	1.000	1.000
Informativeness	0.907	0.978	0.967	0.989
Correctness	1.000	1.000	1.000	1.000
Ambiguity	0.589	0.978	0.967	0.989
Clarification	0.851	0.989	0.983	0.994
Long Answer	-0.006	0.989	0.983	0.994

Table 8: Inter-annotator agreement across long-answer judgments and dataset quality evaluation.

C.2 Human Evaluation Protocols

We develop a dedicated human evaluation protocol to systematically assess long answer quality, as in the correlation analysis presented in Figure 5. Annotators rate each answer using the detailed criteria shown in Table 9, with explicit written instructions for each aspect. This protocol is introduced exclusively for the evaluation setup in Figure 5, ensuring that all human judgments are directly comparable with the figure's correlation metrics.

Table 8 summarizes the results. Overall, annotators exhibit consistently high agreement across evaluation criteria. The relatively low κ value for the *Long Answer* quality dimension stems from a well-known prevalence effect: when nearly all annotators overwhelmingly choose the same label (here, "Yes"), Fleiss' κ is distorted downward due to artificially inflated chance agreement. Importantly, the strict and majority agreement rates for this item remain very high, confirming that annotators were indeed consistent and that the low κ does *not* reflect genuine disagreement.

D Latency Analysis

Table 10 presents the end-to-end latency (in seconds) for each method across three LLM backbones on the MARCH dataset. To ensure a fair comparison, all retrieval-augmented and agentic methods utilize the same retriever and identical retrieval hyperparameters. For agentic approaches, latency is strictly bounded by a fixed interaction budget within the acting loop (maximum of five ReAct-

Criterion	Description
Relevance	Does the long answer fully address both clarified queries and include all relevant short answers, without digression?
Faithfulness	Is the answer consistent with the intent and facts in the original query, clarified queries, and short answers?
Informativeness	Does the answer provide additional useful background, explanations, or actionable guidance to fulfill the user's needs?
Correctness	Are all facts accurate, with no errors or omissions in the key information?

Table 9: Human evaluation protocol for long answer quality. Used for correlation analysis in Figure 5.

Method	Qwen3-235b	Gemini-2.5	DeepSeek-v3.1
No Retrieval	0.439	0.169	0.226
CoT	1.295	0.476	0.862
NaiveRAG	0.667	0.220	0.246
CoT w/ RAG	1.042	0.440	0.778
DIVA	1.026	0.506	0.746
ReAct	3.820	2.290	4.167
CLARION (Ours)	8.958	2.576	5.566
CLARION w/o clarification	4.968	2.629	4.863
CLARION w/o clarification & detection	4.443	2.567	4.116

Table 10: Latency (s) comparison across models and methods on the MARCH dataset.

style iterations), preventing unbounded tool calls and ensuring predictable runtime.

Overall, methods without retrieval exhibit the lowest latency, whereas iterative agentic methods are the most computationally intensive. Incorporating Chain-of-Thought (CoT) increases latency compared to direct answering due to the generation of intermediate reasoning steps. Standard retrieve-then-read pipelines (NaiveRAG, CoT w/ RAG, DIVA) incur only modest overhead from retrieval, remaining substantially faster than multi-step tool-use frameworks. In contrast, ReAct shows a marked increase in latency as it interleaves reasoning with multiple sequential search steps. CLARION yields the highest latency in most settings, as it (i) executes a dedicated planning stage for ambiguity detection and clarification, and (ii) performs retrieval and reasoning separately for each clarified interpretation, effectively conducting multi-branch evidence gathering prior to synthesis.

The ablation studies highlight the inherent cost–quality trade-off of explicit clarification. Removing the clarification stage reduces latency by approximately half for Qwen3-235b and DeepSeek-v3.1, though the impact is less pronounced for Gemini-2.5. Ultimately, the latency results align with our design philosophy: CLARION deliberately allocates additional computational resources to preserve and explore multiple interpretations—rather than prematurely committing to a single dominant branch—which is indispensable for resolving the latent, path-dependent ambiguity characteristic of multi-hop QA.

E Results on Other Related Benchmarks

E.1 Results for ASQA Benchmark

As shown in Table 11, in ASQA, our agentic approach consistently delivers the strongest short-answer performance across models. With detection + clarification enabled, CLARION achieves the best per-model averages—e.g., Qwen3-235b-a22b-250: 71.64 vs. 62.24 (NaiveRAG) and 55.54 (DIVA); Gemini-2.5-Flash: 68.81 vs. 58.08 and 51.08; DeepSeek-Chat-v3.1: 69.48 vs. 51.34 and 50.32. Improvements appear in both STR-EM (coverage of gold short answers) and Disambig-F1 (extractability for clarified questions), indicating that explicitly detecting ambiguity and rewriting the query steers retrieval to interpretation-aligned evidence rather than memorized or mixed contexts. Ablations verify the contribution of each component, with the largest drop when clarification is removed—highlighting that planning for ambiguity before acting is crucial even on single-hop-oriented datasets like ASQA. Together, these trends support our claim that agentic planning and acting modules are broadly beneficial beyond MARCH and strengthen answer completeness and precision in ambiguous QA settings.

E.2 Results for MuSiQue Benchmark

As shown in Table 12, CLARION consistently outperforms baseline methods on the MuSiQue dataset, which evaluates standard multi-hop reasoning without the specific focus on ambiguity found in MARCH. On average across the three LLM backbones, CLARION achieves an Exact Match (EM) score of 20.6 and an F1 score of 59.4. This represents a clear improvement over the strongest agentic baseline, ReAct (EM: 14.0, F1: 56.8), as well as DIVA (EM: 14.2, F1: 57.6). The results

Model	Method	Short Answer		
		STR-EM	Disambig-F1	Avg
Qwen3-235b-a22b-250	No Retrieval	70.98	36.56	53.77
	NaiveRAG	83.14	41.34	62.24
	DIVA	73.53	37.54	55.54
	ReAct	82.31	40.75	61.53
	CLARION (ours)	**92.94**	**50.33**	**71.64**
	CLARION w/o clarification	85.69	42.58	64.13
	CLARION w/o clarification & detection	84.31	42.30	63.30
Gemini-2.5-Flash	No Retrieval	66.86	34.79	50.82
	NaiveRAG	76.86	39.30	58.08
	DIVA	66.47	35.69	51.08
	ReAct	81.13	41.72	61.42
	CLARION (ours)	**89.61**	**48.02**	**68.81**
	CLARION w/o clarification	88.43	46.09	67.26
	CLARION w/o clarification & detection	87.65	46.04	66.84
DeepSeek-Chat-v3.1	No Retrieval	70.59	35.57	53.08
	NaiveRAG	67.84	34.85	51.34
	DIVA	75.10	25.54	50.32
	ReAct	83.57	40.01	61.79
	CLARION (ours)	**90.98**	**47.99**	**69.48**
	CLARION w/o clarification	87.45	46.16	66.81
	CLARION w/o clarification & detection	88.82	46.23	67.53

Table 11: ASQA results across methods and models. We report STR-EM / Disambig-F1 in %. Best per model in **bold**, second-best underlined.

Method	Qwen3-235B		Gemini-2.5-Flash		DeepSeek-Chat-V3.1	
	EM	F1	EM	F1	EM	F1
Naive RAG	0.040	0.558	0.102	0.520	0.044	0.478
DIVA	0.142	0.576	0.106	0.520	0.106	**0.518**
ReAct Only	0.140	0.568	0.132	0.551	0.132	0.509
Clarion	**0.206**	**0.594**	**0.146**	**0.560**	**0.160**	0.514

Table 12: Comparison of EM and F1 scores across different models and methods.

indicate that the benefits of CLARION's architecture extend beyond ambiguity resolution to general multi-step inference tasks. Even in non-ambiguous multi-hop scenarios, CLARION's ability to decompose complex queries and verify evidence at each hop ensures that the model maintains a coherent trajectory across connected documents.

F LLM-as-a-Judge

F.1 Validating the LLM Judge.

We validate the use of the *LLM judge* for our main experiments by comparing its 0–5 scores with human ratings on 300 items across four criteria. We measure (i) linear association with Pearson r, (ii) rank consistency with Spearman ρ and Kendall τ_b, and (iii) grade-level agreement on the 0–5 scale via Quadratic Weighted Kappa (QWK). As shown in Figure 5, we observe consistently strong correlations across all families; QWK further indicates grade-aligned agreement, supporting the LLM judge as a valid proxy for our main experiments. For full details on the human evaluation protocol, see Appendix C.2.

F.2 Detailed Score for LLM-as-a-Judge

As shown in Table 13, we report the scores of each sub-criterion under the LLM-as-a-Judge evaluation for the baselines and for CLARION. Consistent with our main experiments, CLARION generally achieves higher judge scores across most criteria.

G Case Study

G.1 Failure Cases of CLARION

Despite its strong performance, CLARION still fails on certain ambiguous multi-hop queries. Table 15 shows representative failure cases across all

Method	Model	LLM-as-a-Judge (0–5)			
		Relevance	Faithfulness	Informativeness	Correctness
LLM-only					
No Retrieval	*Qwen-3-235b*	3.324	3.147	2.931	2.931
	Gemini-2.5	2.643	2.439	1.912	2.233
	DeepSeek-v3.1	3.067	2.706	2.480	2.480
RAG-based baselines					
Naive RAG	*Qwen-3-235b*	2.961	2.988	2.357	2.829
	Gemini-2.5	2.525	2.620	1.643	2.543
	DeepSeek-v3.1	2.325	2.408	1.516	2.259
Diva	*Qwen-3-235b*	3.073	3.465	2.565	3.084
	Gemini-2.5	2.531	2.694	1.727	2.382
	DeepSeek-v3.1	2.918	2.851	2.292	2.596
CLARION (ours)					
CLARIONw/o clarification & detection	*Qwen-3-235b*	3.057	2.957	2.496	2.696
	Gemini-2.5	2.673	2.963	2.039	2.702
	DeepSeek-v3.1	3.075	3.159	2.635	2.839
CLARIONw/o clarification	*Qwen-3-235b*	3.184	3.084	2.608	2.890
	Gemini-2.5	2.712	2.963	2.133	2.700
	DeepSeek-v3.1	3.089	3.136	2.699	2.843
CLARION	*Qwen-3-235b*	3.600	3.551	3.502	3.271
	Gemini-2.5	2.843	3.106	2.302	2.824
	DeepSeek-v3.1	3.228	3.177	2.943	2.882

Table 13: LLM-as-a-Judge sub-criteria. All scores are on a 0–5 scale.

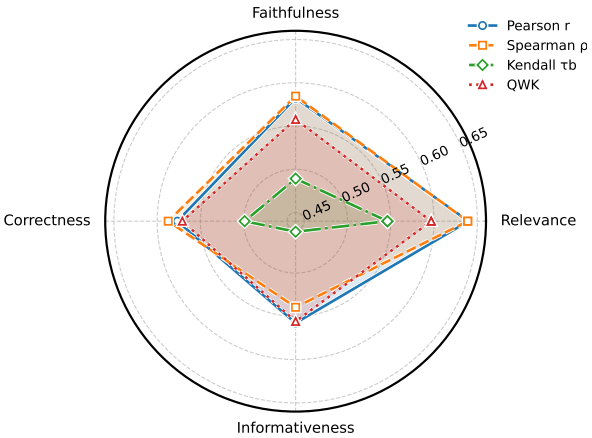

Figure 5: Correlation between LLM and human judgments.

three ambiguity types. In each case, CLARION's prediction collapses to a single interpretation without resolving ambiguity, so the system generates only one short answer and misses the gold interpretations. This results in complete mismatches (0 on STR-EM and Disambig-F1). These errors illustrate how mis-specified sub-questions or over-broad interpretations derail reasoning and retrieval, leading to a complete mismatch against gold answers.

G.2 Case Study: A 3-hop Query with Three Interpretations

Table 16 illustrates how **CLARION** handles a nominal 3-hop query whose hop-1 is semantically ambiguous. Because *The Birds and the Bees* may refer to different performers/versions, the **Planning Agent** first detects ambiguity and enumerates three clarified interpretations (I_1–I_3). The **Acting Agent** then executes the same 3-hop schema (performer/version → birthplace → largest annual event in the birthplace) for each interpretation in a **ReAct-style** loop, carrying entities forward across hops while retrieving evidence step-by-step. Crucially, CLARION enforces **hop-consistency verification**: the performer/version at H1 must be supported by retrieved evidence, and the event evidence at H3 must be explicitly grounded in the location resolved at H2 before downstream propagation. In this example, I_1 passes verification and yields a grounded answer (*Houston Livestock Show and Rodeo*). In contrast, I_2 retrieves a plausible event candidate but fails verification due to missing/weak grounding across hops (e.g., unsupported H1 and/or an H2–H3 mismatch), so the Acting Agent triggers **recovery** via targeted **re-retrieval**

or user confirmation. Finally, I_3 correctly blocks execution and requests the missing artist/version, preventing ungrounded multi-hop synthesis.

H Implementation Details

As shown in Table 17, we report our implementation details for our MARCH construction pipeline and running CLARION.

API Access and Infrastructure. All LLM calls were made using the OpenRouter API. Experiments were run on a workstation equipped with an RTX 6000 Ada GPU.

Models. We use four off-the-shelf LLMs as ambiguity detectors: GPT-4.1 (OpenAI), Llama-4-Maverick (Meta), Qwen3-235b (Alibaba), and Claude-Sonnet-4 (Anthropic). For generating clarified questions and answers, we exclusively use GPT-4.1. We use gpt-4.1 (snapshot: gpt-4.1-2025-04-14) in all experiments to ensure reproducibility. For LLM-as-a-judge filtering, we employ Llama-4-Maverick, Qwen3-235b, and Claude-Sonnet-4.

Decoding and Prompting. All LLM calls (for both detection and generation) were run with temperature set to 0.0 and a maximum token limit of 512. All prompts and task templates are described in detail in Section J.

Retrieval Pipeline. For evidence retrieval, we use FAISS for fast vector search over Wikipedia passages. Query and document embeddings are computed with the Qwen3-8B-Embedding model. Retrieval is performed with a fixed top-k of 10 per query.

Agentic Reasoning. For CLARION, the agent's maximum search iteration is set to 5. The planning agent performs ambiguity detection, type classification, and clarification as described in Section 4; the acting agent executes search and answer steps up to the iteration limit.

Filtering and Evaluation Protocol. After answer generation, candidate instances are filtered using three LLMs (excluding GPT-4.1 to prevent overfitting). An instance is retained only if all models unanimously judged every field (question, clarifications, type, evidence, answers) as fully aligned, following the same protocol as human evaluation (see Appendix C.2 and Table 2 for details).

Ambiguity Taxonomy for Constructing Data
Syntactic: Two clarified questions differ in grammatical structure ("in the birthplace of" vs. "in the place where ... was born"), but both shorten to the same short answer.
Constraint: Two clarified questions ask about different ways of defining Antarctica's border, yet both reduce to the same numeric answer.
Semantic: Two clarified questions focus on different semantic aspects (actor identity vs. character role), but shortening collapsed both to the same name.

I Detection cue for Constraint ambiguity

Constraint ambiguity arises when a query is over-specified (e.g., exact dates, versions, quoted spans) so that retrieval narrows the user's true intent. We use three complementary signals. First, *total_hits* $H(q)$ flags abnormally small result sets, indicating a narrowed scope. Second, the *KL divergence* $D_{\mathrm{KL}}(P_{\text{top}} \parallel P_{\text{corpus}})$ measures how skewed the top-snippet word distribution is relative to the background corpus, revealing over-reliance on special tokens (dates, numbers, quoted phrases). Third, the *relax_delta_ratio* $\rho(q) = \frac{H(\mathrm{relax}(q))}{H(q)}$ is an intervention-style cue: it asks how much the hit count jumps when we remove exactly one constraint (a date, a number, or a quoted span). In combination, $H(q)$ low, D_{KL} high, and $\rho(q)$ high strongly suggest over-specialization–induced recall failure, whereas low $H(q)$ with low $\rho(q)$ points to genuinely sparse topics rather than over-specification. These cues reduce false positives and guide the LLM toward expert, evidence-aware judgments.

J Prompt Templates

This section summarizes the prompt templates used to construct MARCH and CLARION. For each ambiguity type, we provide templates for detection, clarification, answer generation (short/long), and query decomposition from Figures 6 to 14.

Type	QID	Clarified Query 1	Clarified Query 2	SA_1	SA_2
Syntactic	4hop1__8294 15324_26424 _581618	Who founded the chain of music-themed restaurants whose first establishment was located in the birthplace of the person who ejected the Benedictines in 1559?	Who founded the chain of music-themed restaurants with its first establishment in the place where the person who ejected the Benedictines in 1559 was born?	Isaac Tigrett	Isaac Tigrett
Constraint	2hop__100274 _14948	What latitude marks the northern border of Antarctica?	At what latitude is the continental boundary of Antarctica defined?	60° S	60° S
Semantic	2hop__725611 _52870	Which actor from *Michael Collins* appears in *The Phantom Menace*, and which character do they portray?	In *The Phantom Menace*, which character is played by an actor who was also in *Michael Collins*?	Liam Neeson	Liam Neeson

Table 14: **Failure cases during MARCH construction due to short-answer shortening collisions.** After shortening, both short answers collapsed to identical strings, causing removal even though the clarified queries represent distinct interpretations.

Type	Original Query	Predicted Long Answer	Gold Long Answer	Fail Reason
Semantic	What city shares a border with the place where the person who went to the state known for its Mediterranean climate during the gold rush worked?	Stockton	Brooklyn and Traverse City share borders with the relevant places.	Collapsed to a single interpretation; failed to clarify multiple possible places.
Syntactic	Who won the Indy Car Race in the largest populated city of the state where Yuma's Library District is located?	Mario Andretti (Phoenix, 1993)	Álex Palou and Hélio Castroneves	Relied on historical fact lookup; failed to disambiguate event scope and multiple winners.
Constraint	Who brought the language Hokkien to the country on the natural boundary between the country that hosted the tournament and the country where A Don is from?	The Hoklo (Hokkien) people	Dutch colonial administration and Hokkien-speaking immigrants during Spanish colonization	Did not resolve broad-/general query; simplified to one actor instead of multiple sources.

Table 15: **Representative CLARION failure cases.** CLARION often fails to clarify ambiguous sub-questions and collapses to a single short answer, leading to zero scores on both STR-EM and Disambig-F1.

Step	I_1: Jewel Akens (1964)	I_2: Dean Martin (claimed)	I_3: Other/unknown artist
User query. *What is the largest annual event held in the birthplace of the performer of* The Birds and the Bees*?*			
Detect. Ambiguous (all detectors: Y): *The Birds and the Bees* can refer to different performers/versions, so hop-1 branches the entire 3-hop chain.			
Clarify	Performer/version: Jewel Akens (1964) Birthplace: Houston, TX	Performer/version: Dean Martin (version) Birthplace: *to verify*	Performer/version: *unspecified* Ask user to specify
Plan (3-hop)	**H1** identify performer/version **H2** retrieve birthplace **H3** largest annual event in birthplace	**H1** identify performer/version **H2** retrieve birthplace **H3** largest annual event in birthplace	**H1** missing → cannot execute **H2–H3** blocked
Retrieve	**H1** evidence: *The Birds and the Bees* (1964) performed by Jewel Akens (doc: 12676138). **H3** evidence: *Houston Livestock Show and Rodeo* held in Houston (doc: 318775).	Candidate **H3** event: *Pennsic War* (doc: 535960). **Issue:** H1 (performer/version) is unsupported and H2 (birthplace) is ungrounded, so the event is not tied to the H2 location (**H2–H3 mismatch**).	No retrieval: artist/version for H1 is missing, so H2–H3 cannot be queried.
Verify	**Pass:** H1 supported; H2 (Houston) grounds H3 evidence.	**Fail:** hop-consistency break (H1 unsupported and/or H2–H3 mismatch).	**Blocked:** missing H1 prevents verification and downstream hops.
Output	ANSWER: Houston Livestock Show and Rodeo (with citation).	RECOVER: re-retrieve for H1/H2 or ask user to confirm intended performer.	CLARIFY: ask for the artist/version.
Final (answer-focused). Verified answer is **I_1: Houston Livestock Show and Rodeo**. I_2 is not safely answerable without additional grounding; I_3 requires user specification.			

Table 16: Case study (qid: 3hop1__337919_841757_11974): hop-1 semantic ambiguity yields three interpretations. Each branch follows the same 3-hop schema and is checked by hop-consistency verification before producing outputs.

Item	Value / Setting
API	OpenRouter API
Detection Model	GPT-4.1, Llama-4-Maverick, Qwen3-235b, Claude-Sonnet-4
Generator Model	GPT-4.1
Filtering Model	Llama-4-Maverick, Qwen3-235b, Claude-Sonnet-4
LLM-as-a-Judge Model	GPT-4.1
Temperature	0.0 (detection), 0.0 (generation)
Max Tokens	512 (detection), 512 (generation)
Evaluation Protocol	See Appendix C.2 and Table 2
Embedding Model	Qwen3-8B-Embedding
Retriever	FAISS
Top-k	10
Agent Max Search Iteration	5
GPU	RTX 6000 Ada

Table 17: Implementation details.

You are a linguistics expert.

1) Read the sentence below.
2) Decide whether it is syntactically ambiguous under any of the 18 phenomena.
3) If ambiguous, list all applicable phenomenon numbers (ascending).

Phenomena (1–18)

1. PP Attachment (including instrument vs. attribute "with"); 2. Relative-Clause Attachment; 3. Coordination Scope (and/or); 4. Comparative Attachment / Ellipsis; 5. Quantifier / Negation Scope; 6. Dangling / Misplaced Modifier; 7. Genitive-Chain Attachment; 8. Complement vs. Adjunct; 9. Gerund vs. Participle; 10. Ellipsis / Gapping; 11. If-clause Attachment; 12. Right-Node Raising; 13. Adjective Stacking / Coordination; 14. Inclusive vs. Exclusive "or"; 15. Adverbial Attachment (VP vs. S); 16. Focus / Only-scope; 17. Apposition vs. Restriction; 18. Degree / Comparative subdeletion.

Question: QUESTION

Output (JSON): "is_ambiguous": "Y", "categories": [1, 3, 7] // [] if "N"

Keys must be exactly "is_ambiguous" and "categories". No extra text.

Figure 6: Prompt template for syntactic ambiguity detection.

You are a linguistics expert.

The question below is syntactically ambiguous. Write at least MIN_VERSIONS distinct clarified questions, each encoding a different structural reading (attachment, scope, etc.). Preserve the topic; each rewrite must be fully unambiguous; concise, natural English.

Question: QUESTION

Output (JSON): "clarified_queries": ["...", "..."]

Key exactly "clarified_queries"; provide at least 2 strings; no other keys.

Figure 7: Prompt template for syntactic clarification.

You are a linguistics expert.

1) Read the search query and three RAW metric values.
2) Decide if the query shows constraint ambiguity (over-specific constraints harming recall).
3) Output ONLY the JSON object in the required format.

A query with constraint ambiguity (over-specific) is narrowly constrained (dates, version numbers, quoted strings, etc.), likely missing the broader intent.

Metrics

Total_hits: Result count for the literal query.
KL_divergence: D_KL between top-k snippet unigrams and the whole corpus.
Relax_delta_ratio: Largest fold-increase in hits after removing one numeric/date/quoted constraint.

Question: QUESTION

Raw metric values

Total_hits: TOTAL_HITS
KL_divergence: KL_DIVERGENCE
Relax_delta_ratio: RELAX_DELTA_RATIO

Output (JSON): "is_ambiguous": "Y" // "N" if not constraint

Use expertise; no hard thresholds. No markdown, code fences, or extra keys.

Figure 8: Prompt template for constraint (over-specific) ambiguity detection.

You are an information-retrieval and linguistics expert.

Rewrite the query below into at least MIN_VERSIONS broader, faithful variants that surface the user's core intent and remove needless specificity or indirections.

How to clarify

1) Identify the core question (fact or relationship truly sought).
2) Resolve or drop cascading indirections (replace "the country where X was born" with the direct entity if obvious; else use a neutral placeholder).
3) Remove or soften excessive constraints (exact dates, versions, quoted titles).
4) Keep the answer type the same; do not over-broaden. Write concise English.

Question: QUESTION

Output (JSON): "clarified_queries": ["...", "..."]

Key must be exactly "clarified_queries"; provide at least 2 strings; no extra keys.

Figure 9: Prompt template for general clarification.

You are a linguistics expert.

Semantically ambiguous lacks sufficient context so multiple reasonable meanings or referents are possible (unclear pronoun, vague time, polysemy, etc.).

1) Read the sentence.
2) Output "Y" if semantically ambiguous, else "N".

Question: QUESTION

Output (JSON): "is_ambiguous": "Y" // "N" if unambiguous

Key must be exactly "is_ambiguous". No extra text.

Figure 10: Prompt template for semantic ambiguity detection.

You are a linguistics expert.

Rewrite the semantically ambiguous question into at least 2 distinct clarified questions, each resolving a different interpretation. Preserve the original topic. Add only minimal context (time, referent, sense) to make each unambiguous. Concise English.

Question: QUESTION

Output (JSON): "clarified_queries": ["...", "..."]

Key exactly at least 2 "clarified_queries"; no other keys.

Figure 11: Prompt template for semantic clarification.

You are an extractive QA assistant.

Given a question and one passage, return the shortest exact span in the passage that answers the question. If no answer, return an empty string.

Question: QUESTION

Passage: PASSAGE

Output (JSON): "short_answer": "..."

Extractive only (verbatim span); no justification or extra text.

Figure 12: Prompt template for short answer generation (extractive).

You are an expert open-domain QA assistant.

Combine two validated short answers (A1, A2) to create a single, coherent long answer to the original ambiguous question (OQ). If both can be true, merge into 1–3 fluent sentences. Do not invent facts beyond A1 and A2.

Return only JSON that matches the schema: SCHEMA

Question: QUESTION

Clarified Q1 | Short Answer A1

CQ1
A1 = A1

Clarified Q2 | Short Answer A2

CQ2
A2 = A2

Output (JSON): "long_answer": "..."

Figure 13: Prompt template for long answer generation (merge A1 + A2).

You are an information-retrieval expert.

Break the complex question into the minimal set of atomic, single-hop sub-questions in the exact order needed to fully answer it.

- Output each sub-question as a Markdown bullet starting with "* ".
- Each sub-question must ask for exactly one fact or relationship.
- No explanations or extra text.

Question: QUESTION

Output (JSON): "sub_query": "..."

Figure 14: Prompt template for query decomposition (ordered single-hop bullets).

You are an expert at analyzing query ambiguity.
Your task is to determine whether a query is ambiguous and to classify the ambiguity type.

Analyze the following query and decide:

1. Provide brief reasoning.
2. Is the query ambiguous?
3. Which specific aspects make it ambiguous?
4. What extra information would clarify it?
5. Classify the ambiguity as one of: *"syntactic"*, *"constraint"*, *"semantic"*, or *"none"*.

Definitions:

* **syntactic**: multiple plausible grammatical parses (attachment/scope/coordination/pronoun reference).
* **constraint**: over-specific query where a broader, closely related formulation better matches the user's need.
* **semantic**: syntax is clear but meaning/intent admits multiple valid interpretations via world knowledge.

Query: {query}

Return STRICT JSON:
{
"reasoning": "string",
"is_ambiguous": true/false,
"ambiguity_type": "syntactic" | "constraint" | "semantic" | "none",
"ambiguous_aspects": ["..."],
"clarification_needed": "string"
}

Figure 15: Prompt template for ambiguity detection and typing (strict JSON).

You are an expert at clarifying ambiguous queries.
Given the original query and an ambiguity analysis, rewrite the query into **two** specific, actionable, and faithful clarified versions.

Original Query: {query}
Ambiguity Analysis (JSON): {analysis}

Write STRICT JSON:
{
"reasoning": "why these clarifications resolve the ambiguity",
"clarified_query1": "string",
"clarified_query2": "string"
}

Figure 16: Prompt template for generating two clarified queries from an ambiguity analysis.

You are a research assistant following ReAct (Reasoning, Acting, Observing).

Available Actions:

* `SEARCH[query]` → run a search using the configured method
* `ANSWER[planning]` → run a planning agent
* `ANSWER[text]` → provide a final answer now

Constraints:

* Max searches allowed: max_searches
* Searches used so far: current_searches
* Do **not** reuse the exact same search query as previously used in context.

Task Query: {query}
Previous Context:
{context}

Instructions:

1. THINK about the next best step.
2. If more evidence is needed, choose `SEARCH[very specific query]`.
3. If sufficient, choose `ANSWER[concise, well-supported answer]`.
4. If you have already reached the maximum allowed searches, you **must** output `ANSWER[...]` now.

Respond in **EXACT** format:
`THOUGHT: <your internal reasoning, one short paragraph>`
`ACTION: SEARCH[...specific query...]` **OR**
`ACTION: PLANNING[...call planning agent...]` **OR**

`ACTION: ANSWER[...final answer...]`

Figure 17: Prompt template for ReAct-style retrieval and answering with a bounded search budget.