

Meet, Compare, or Abstain: LATWEAVE for Deterministic Multi-Hop Question Answering on Knowledge Lattices

Yuze Ren
ZenSmart Technology (Beijing) Co.,
Ltd., China
renyuze@zhongshuruizhi.com

Shaoheng Fan
ZenSmart Technology (Beijing) Co.,
Ltd., China
fanshaoheng@zhongshuruizhi.com

Tao Wang
ZenSmart Technology (Beijing) Co.,
Ltd., China
wangtao@zhongshuruizhi.com

Yabo Yan
ZenSmart Technology (Beijing) Co.,
Ltd., China
yanyabo@zhongshuruizhi.com

Han Han*
ZenSmart Technology (Beijing) Co.,
Ltd., China
hanhan1@zhongshuruizhi.com

Abstract

Probabilistic question-answering systems—whether large language models (LLMs) themselves, retrieval-augmented generation (RAG), or trained multi-hop retrievers—conflate “what is known” and “how to reason” into a single probabilistic computation: hallucination cannot be eradicated, evidence chains cannot be audited, and the system answers even when it does not know. We present LATWEAVE, which organizes knowledge into a multidimensional knowledge lattice and compiles multi-hop QA into three deterministic operators—meet (constraint intersection), compare (lattice-order comparison), and abstain (structural abstention); LLMs appear only on the construction side (one-shot extraction) and the query-planning side, while the answer-generation path is zero-LLM, zero-task-training, and auditable end to end—so that question answering over Web-published knowledge becomes reproducible item by item. Rather than claiming across-the-board SOTA, we characterize the operating envelope of this paradigm on six public benchmarks: when knowledge is complete (MetaQA, 39,093 questions) meet chains are near-lossless over three hops (any-hit 0.9975, on par with fully supervised KBQA); on templated multi-hop home ground (2Wiki-MultiHopQA held-out $n = 1,258$) EM 0.865, well above published structure-augmented RAG reproductions; on open-text deep composition (MuSiQue) and extraction-coverage gaps (HotpotQA) we report degradation honestly and attribute it to causes outside the lattice-algebra layer; and when information is incomplete (IIIRC) we achieve structural abstention with abstain accuracy 0.971 and leak rate 0.029. Within the operating envelope, deterministic execution pays no performance penalty, and every step on the answer path can be recomputed—precisely the source of end-to-end auditability.

Keywords

constraint meet; lattice-order comparison; structural abstention; knowledge lattice; multi-hop question answering; deterministic reasoning; retrieval-augmented generation

1 Introduction

Modern question answering systems have nearly converged on a hybrid architecture: keyword retrieval guarantees exact matches, semantic retrieval handles recall, and GraphRAG-style methods further introduce structured retrieval into the pipeline, explicitly

connecting the correlations that remain isolated between semantic chunks. Engineering mechanisms such as chunk IDs have given the evidence chain a traceable form in production for the first time. Yet two structural problems remain unsolved. The first is hallucination: provenance annotations can trace which snippets an answer consulted, but not whether the answer is logically entailed by them—the final step from snippets to answer is still taken in probability space, and distortion, extrapolation, and outright fabrication all occur at this step. This is not an engineering oversight: hallucination is a mathematical inevitability of probability-based generation—any calibrated language model must hallucinate [20]. The second is answering anyway by design: when neither retrieval nor the graph contains facts supporting an answer, the system has no structural mechanism to stop itself.

The two problems trace back to the same root: engineering-level auditability—the sources are traceable, but the reasoning from evidence to conclusion is a black box. This paper argues for mechanism-level auditability: the answer itself is a meet chain unfolded step by step, each step a verifiable lattice-algebra operation that any examiner can replay and check independently. A clarification of scope: this paper does not claim that deterministic methods outperform probabilistic ones across all multi-hop tasks; what we characterize is where they approach losslessness, where they structurally degrade, and how they abstain honestly when the paradigm degrades—its *operating envelope*. Mapping this envelope out is itself one of the contributions of this paper.

Our approach can be summarized in one sentence: the LLM is restricted to a “builder + query translator,” and answers are produced deterministically by three operators on the lattice. *Meet* (constraint intersection) compiles the question into a set of dimension constraints and takes their greatest lower bound (GLB) on the product lattice; the hit instances form the answer, each carrying its source_text, so the evidence can be audited item by item. *Compare* (lattice-order comparison) performs comparison and set operations along the lattice order—Boolean tests, complete answer-set return, and lattice-order adjudication of “which is more X.” *Abstain* (structural refusal): when constraints conflict or the fact is absent from the lattice, the system refuses structurally—a direct consequence of lattice semantics, not a low-confidence fallback. Figure 1 gives the two-stage overview: at build time (offline, one-off) an LLM extracts typed predicates from question-blind documents and constructs the lattice; at query time an LLM planner compiles the question

*Corresponding author.

into a typed plan that the three operators execute deterministically. LLM participation is confined to the builder and planner roles—the answer path makes zero LLM calls and requires zero task training—and every answer is an auditable meet chain.

Our contributions are fourfold:

- **Mechanism.** A formalization and system realization of the knowledge lattice with the meet/compare/abstain operators; the answer path contains zero LLM calls and zero task training (§3–4).
- **Upper-bound evidence.** With complete or templated knowledge, deterministic retrieval is near-lossless: MetaQA any-hit 0.9975 (zero training, on par with fully supervised KBQA) and 2Wiki held-out EM 0.865 (§6).
- **Boundary evidence and attribution.** Degradation on three boundary benchmarks (MuSiQue / HotpotQA / FRAMES) is attributed layer by layer to causes outside lattice algebra, consolidating into a unified failure classification (§7).
- **Abstention as a first-class capability.** A two-scale evaluation framework (recall $\times$ precision), corroborated on IIRC by structural abstention: abstain accuracy 0.971, leak rate 0.029, zero drift on the tuning pool (§8).

The lineage of this work is also worth stating: LATWEAVE builds on an earlier rule-centric knowledge base system developed within our group, where partial ordering on a classification lattice served solely for applicability gating, not yet a retrieval algebra; we generalize that gating into a full multi-dimensional product-lattice algebra. The step from “partial ordering used only for gating” to “the lattice is the retrieval space” is precisely where the core contribution of this paper lies.

2 Related Work

Today’s QA systems have evolved along four technical lines, each solving a different problem: neural multi-hop retrievers achieve multi-hop recall but depend on model training; RAG and GraphRAG mitigate knowledge staleness and chunk isolation, yet hallucination cannot be eliminated and evidence chains remain hard to audit; KBQA and semantic parsing deliver precise answers, but are bound to a predefined schema and a trained parser; selective prediction lets a system refuse when it does not know, but the refusal rests on a confidence threshold whose calibration drifts with the task distribution. Only this work achieves zero training, deterministic answers, mechanism-level auditability, and structural abstention at the same time—these four properties have never before appeared in a single system (Appendix Table 9).

2.1 Multi-Hop Question Answering

HotpotQA [44], 2WikiMultihopQA [16], MuSiQue [38], IIRC [8], and FRAMES [23] together define the difficulty spectrum of multi-hop QA—from templated two-hop questions to adversarial deep composition, and from complete to incomplete information. Mainstream solvers (MDR [42], IRCot [39], Beam Retrieval [45], HopRAG [28]) are probabilistic or training-dependent—a different precondition from our zero-training setting.

2.2 Retrieval-Augmented and Graph-Augmented LLMs

RAG [24] and its refinements (Self-RAG [2] learns when retrieval pays off; FLARE [19] re-retrieves proactively mid-generation) use semantic-similarity retrieval to mitigate hallucination; GraphRAG [6], LightRAG [12], HippoRAG 2 [13], ToG [37], RoG [29], and KAG [25] further introduce graph structure into retrieval, mining the associations between semantic chunks. This line strengthens “retrieval,” yet answers are still LLM-generated—the hallucination and auditability problems remain exactly as before.

2.3 KBQA and Semantic Parsing

KBQA systems reason deterministically with fully verifiable inference—KV-Mem [30], GraftNet [36], SRN [31], PullNet [35], EmbedKGQA [33], NSM [15], and RoG [29]—but each depends on two prerequisites: a ready-made knowledge graph and a trained parser. Our method carries neither: a one-shot LLM extraction builds the lattice directly from text, keeping auditability with no model-training requirement.

On the query side, the most recent publicly available work is LLM-Plan by Shrestha and Kim [34]: an LLM plans only a relation sequence that a deterministic BFS executes over a knowledge graph. LATWEAVE differs structurally: meet intersects per-dimension value constraints where LLM-Plan’s relation traversal carries none; we provide compare semantics and structural abstention where LLM-Plan has neither; and we characterize a six-benchmark operating envelope, while LLM-Plan is evaluated on a single benchmark. The item-by-item numeric comparison appears in Section 6.2.

2.4 Abstention, Selective Prediction, and Hallucination

Selective prediction supplies the theoretical frame for “abstain when unsure” [7, 11], carried into QA [21] and VQA [40]; TruthfulQA [27] and hallucination surveys [18] chart the cost of never abstaining. Yet existing abstention is almost entirely probabilistic confidence thresholding, whose risk–coverage trade-off drifts with the task—softmax-score rejection attains under 7.5% coverage at 1% risk, and a learned selector attains only about 15.6% [40]. Our abstain is structural: a fact absent from the lattice entails abstention, with no threshold to tune and a leak rate that is empirically near zero (Section 8).

2.5 Concurrent Work and Positioning

Concurrent WWW 2026 work approaches different facets of the same problem: S-Path-RAG [9] trains a differentiable path scorer and injects cross-attention into the LLM; CompactRAG [43] cuts the LLM-call and token cost of multi-hop QA to two calls per question, yet those calls still synthesize the final answer; RoE [14] unifies retrieval and generation via SFT- and RL-trained graph exploration; HyperRAG [26] upgrades retrieval to n-ary hypergraphs with an LLM-guided beam. We differ from all four on the same two poles: zero training, and an answer path that never enters the LLM.

3 The Knowledge Lattice Model

This section pins down the lattice as a mathematical object that a reviewer can verify, and defines the three operators—*meet* (constraint

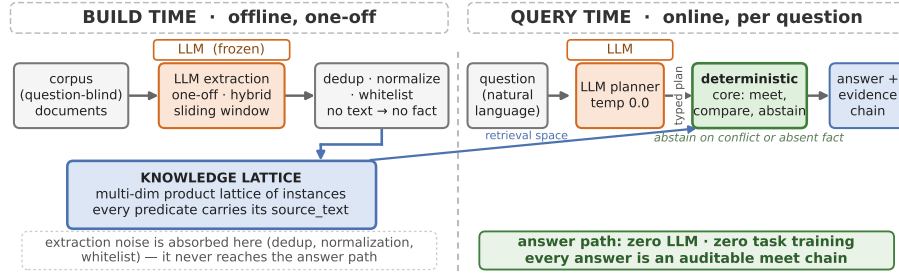


Figure 1: LatWeave in two stages. Build time (offline, one-off): an LLM extracts typed predicates from question-blind documents (dedup, normalization and a dimension whitelist absorb extraction uncertainty before it can enter the lattice), yielding a product lattice whose instances each carry their source_text. **Query time (online, per question):** an LLM planner compiles the question into a typed plan, and a deterministic core executes the three operators —*meet*, *compare*, *abstain*. The LLM is confined to the two orange boxes (extraction and planning); the answer path carries zero LLM calls and zero task training, so every answer is an auditable meet chain.

intersection), *compare* (lattice-order comparison) and *abstain* (structural refusal)—as algebraic operations with explicit semantics; the four classical results the system puts to work, and the engineering question each answers, are given at the end of Section 3.1.

3.1 Posets, Dimensions, and Product Lattices

Start with a miniature example (Figure 2). Take two dimensions: *species*, a biological taxonomy (under *animal* sit *mammal*, *bird*, and *fish*, and under *mammal* sit *whale*, *dog*, and *bat*), and *habitat*, whose values include *aquatic*, *terrestrial* and *amphibious*. An instance—one particular whale—is bound to the product-lattice coordinate (*whale*, *aquatic*). The query “aquatic mammals” then amounts to a single *meet*: *whale* is a descendant of *mammal* on the first dimension and *aquatic* matches the second, so the whale is returned. If instead we ask for “terrestrial whales,” the two constraints conflict, so the system abstains deterministically rather than guessing. The structure is no toy: the same product-lattice slice already runs over our group’s soil and gas production corpus.

Three standard notions—*poset* (a reflexive, antisymmetric, transitive relation $\sqsubseteq$ on a set P), *lattice* (a poset in which every pair a, b has a greatest lower bound $a \wedge b$ and a least upper bound $a \vee b$), and *product lattice* (componentwise order, meet and join over $L_1 \times \dots \times L_k$)—freeze the intuition’s left half (see, e.g., [4]); the two definitions and one convention below are our own, attaching dimensions and instances to the product lattice and covering instances that miss a dimension.

Definition 4 (Dimension). Each knowledge dimension L_i is a rooted directed acyclic graph: its nodes are dimension values (including synonym sets and numeric intervals), its edges encode the “more general $\rightarrow$ more specific” order, and its root is $\top$.

Definition 5 (Instance). An instance is bound to a coordinate of the product lattice; its predicate set $\{(d_i, v_i, \tau_i)\}$ gives, for each dimension, a value and a confidence. Every predicate must carry its source_text—an anti-hallucination rule: an assertion without supporting evidence text is not admitted into the lattice.

Convention (embedding of missing components). When an instance misses a dimension, that component is read as $\top$ (the most general value) and violates no order check. A “query” is then

stated precisely as a meet on the product lattice—the subject of the next subsection.

Four classical results are put to work by the system, each answering one engineering question. (1) *Is the lattice legitimate?* Birkhoff’s distributivity identity $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ turns the naive M_3/N_5 witness search into a per-triple check ($O(n^3)$ with bitset meet/join tables) [4]; dimensions containing ambiguous sub-lattices never go live. (2) *Can we afford meet?* On-demand Dedekind–MacNeille completion computes GLBs by bitset intersection plus extremum extraction, replacing the $2^{|P|}$ enumeration. (3) *How is cross-dimension structure discovered?* Formal Concept Analysis (FCA) identifies attributes shared across dimensions and supports cross-reference expansion [10, 41]. (4) *Are the dimensions independent?* The Dilworth width, obtained via König bipartite matching, assesses the mutual independence of dimensions [5].

3.2 Three Deterministic Operators and the Capability Boundary

meet (constraint intersection). The semantics of a constraint set $C = \{(d_i, V_i)\}$ is per-dimension GLB followed by coordinate composition in the product lattice: for every constrained dimension we intersect the descendant sets of the candidate values and extract the greatest element; a dimension with no common lower bound is a conflict in that dimension. A conflict is not an exception but a first-class output of the lattice semantics—it is the legitimate trigger of *abstain*.

compare (lattice-order comparison). One operator, four instantiations on the lattice order: (1) Boolean comparison of two entities along the same dimension (2Wiki comparison questions); (2) order-based selection of an extremum among candidates (bridge_comparison’s “which is more X”); (3) complete-set return—*meet* returns the full answer set rather than a top-1 ranking (the primary MetaQA metric, Table 4); (4) Boolean boundary—when the compared objects are not in the lattice, *compare* degrades to *abstain* (IIRC binary questions, Table 13). The four forms share one algebraic basis: instantiations of a single operator, not four patched-together features.

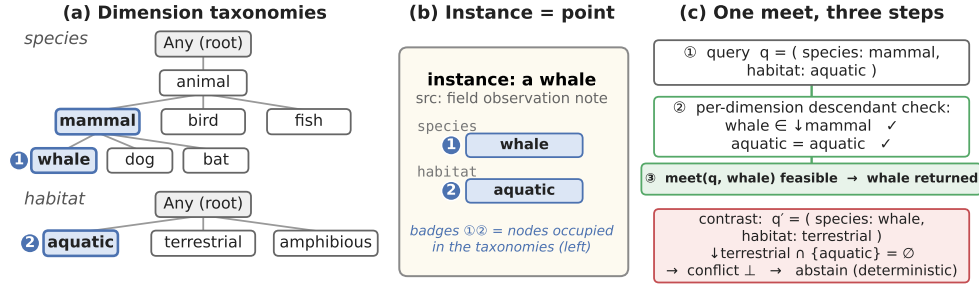


Figure 2: A product-lattice slice (species $\times$ habitat mini-example): (a) value hierarchies of the two dimensions; (b) an instance as a point in the product lattice; (c) one *meet* in three deterministic steps—a hit requires the descendant check on every dimension, and a conflict on any dimension entails abstain.

abstain (structural abstention). A monotone abstention semantics: if the constraints conflict, or no lattice point satisfies all constraints (the hit set is empty), the system abstains. The tighter the constraints, the weaker the answerability—never the reverse.

Capability boundary (declared a priori). Any question that derives new values on top of lattice values (arithmetic, derived counts) belongs to external operator modules, which this paper neither implements nor counts in the in-scope numbers. This declaration licenses the FRAMES capability strata (Appendix A) and the IIRC value-type accounting (Table 13).

Three short propositions follow, each with a one-sentence proof idea (the full arguments are immediately checkable). *Proposition 1 (determinism):* the same lattice and the same query yield the same result, because GLB and lattice-order comparison are functions with no random source. *Proposition 2 (auditability):* every answer unfolds into a *meet* chain, and every instance on the chain carries its source_{text}; auditing means checking the algebra and the evidence text step by step. *Proposition 3 (monotone abstention):* tightening the constraints (adding a dimension or taking a more specific value) never turns an *abstain* into an answer, because tightening shrinks the satisfying set monotonically and an empty set stays empty.

3.3 Relation to FCA and Knowledge Graphs

We borrow lattice semantics from FCA, but our objects are multi-attribute instances rather than formal concepts; we borrow entity-relation structure from knowledge graphs (KGs), but keep dimensional coordinates and the lattice order, making *meet/compare* native operations rather than bolted-on traversals. FCA answers “which objects share which attributes”, whereas we answer “given constraints, which lattice points satisfy them; and when none do, is it a conflict or a gap”.

4 LATWEAVE: System Design

This section presents the system design of LATWEAVE. At construction time (Section 4.1) open text is woven into a knowledge lattice; at query time (Section 4.2) deterministic three-state execution runs on the lattice; Section 4.3 closes with implementation and

efficiency notes. The LLM appears in exactly two places—extraction at construction time and planning at query time—and its uncertainty is kept off the answer path. Engineering details are limited to feasibility numbers; the rest goes to the appendix.

4.1 Construction: From Text to Lattice

Construction turns open text into a knowledge lattice; the core is hybrid extraction. Short documents go through the whole-doc mode, where the full text is fed to the LLM once; long documents go through a physical sliding window cut by character position (window size derived from the model’s context size, consecutive chunks overlapping 200–500 characters), so that no content is lost during chunking. Extracted facts then pass a strict grid-point hash dedup: a fact repeated by the LLM—same lattice coordinates, same source evidence—enters the lattice only once. The last step is whitelist-guided extraction with dimension normalization: the prompt carries a table of preferred dimensions whose canonical names are used when they match semantically, uncovered relations may be named freely, and a variant table normalizes synonymous variants afterward; LLM-invented dimensions are thereby mostly long-tail and low-frequency. The motivation is empirical: in MuSiQue construction, free naming once shattered the single relation “educated at” into 1,017 dimension names.

4.2 Query Time: Planning and Deterministic Three-State Execution

Query time has two parts: the LLM planner compiles the question into a typed plan, and deterministic execution runs that plan on the lattice and returns one of three states. The planner is a replaceable semantic-parsing front end (temperature 0.0; plans are cacheable and reproducible), and its loss can be measured separately from lattice competence by an oracle-plan ceiling experiment (Section 7). For “physicists born in Vienna who won the Nobel Prize in Physics” the compiled plan is a conjunction of three dimension meets, and the executor intersects them to obtain the complete answer set (a *meet* hit); if the constraints conflict or the hit set is empty, it abstains—the refusal comes directly from on-lattice execution, with no external fallback of any kind; *compare* returns a Boolean, an extremum, or a complete-set verdict. Union semantics applies to

Table 1: Datasets, structure, and role in this paper’s argument.

Dataset	Size used	Structure	Role
MetaQA [46]	test $n = 39,093$	1/2/3-hop: 9,947 / 14,872 / 14,274	Control: complete knowledge $\Rightarrow$ lattice algebra lossless
2WikiMultihopQA [16]	held-out $n = 1,258$	4 question types	Main experiment: templated multi-hop, end-to-end
HotpotQA [44]	held-out $n = 741$	bridge 592 / comparison 149	Boundary 1: fact-coverage gap
MuSiQue [38]	held-out $n = 242$	2/3/4-hop: 125/76/41	Boundary 2: longer-chain degradation
IIRC [8]	held-out $n = 130$	span 59/ value 23/ binary 13/ none 35	Refusal experiment: abstain under incompleteness
FRAMES [23]	dev 742 + held-out 82	factuality $\times$ multi-hop	Compounded stress test: coverage $\times$ grounding

the *meet* path only, because complete-set semantics must not be diluted by unions. All experiments in this paper evaluate exactly this three-state execution. The executor itself is three lines of logic: per-dimension descendant-set intersection, extremum extraction, and conflict return when a dimension has no common lower bound.

4.3 Implementation and Efficiency

This section gives feasibility evidence: the system runs at realistic scale, and it is fast. Scale: the 2Wiki dataset holds 1.2M instances, built from the full dev corpus of 125,760 passages. Efficiency: lexicon-scan construction takes 235 ms; retrieval executes at p50 below 18.7 ms and p95 below 50 ms; planning for the 1,258 held-out questions replays from cache in about 1 s. Against GraphRAG-style systems: their indexing requires multiple LLM passes over the whole corpus plus maintained embeddings, and an LLM still sits in the query loop—the gap is architectural, not something that engineering-level tuning can close.

5 Experimental Setup

5.1 Datasets and Their Roles

We evaluate LATWEAVE’s retrieval on six public datasets, each serving one role in our argument (Table 1). MetaQA is the control: with knowledge complete, it verifies that the lattice algebra itself loses nothing. 2WikiMultihopQA is the main experiment. HotpotQA and MuSiQue are the two boundary experiments, exposing respectively the fact-coverage gap of open-domain extraction and the degradation as chains lengthen (2 $\rightarrow$ 4 hops). IIRC examines refusal under incomplete information. FRAMES is a compounded stress test (Appendix A): fact-coverage gap $\times$ query grounding.

5.2 Split Rules and Leakage Control

All text benchmarks share one split rule: a stratified 90/10 dev/hold-out division (seed 88). Lattice construction is strictly question-blind—only paragraphs or KB triples are read, never questions or answers. Configurations are frozen on dev, and the held-out split is evaluated exactly once. MetaQA keeps its official test split (39,093 questions); its configuration is frozen on the official dev split with a 9-shot planner prompt (exemplars drawn only from the train split, so no test information leaks). Held-out drift is zero: 2Wiki 0.886 vs. 0.865; IIRC precision measure 0.9682 vs. 0.9437.

Table 2: 2WikiMultihopQA main results (held-out $n = 1,258$).

Method	EM	F1*	Resolved
Keyword baseline	0.083	0.111	73.2%
Naive RAG (TF-IDF + LLM)	0.198	0.205	100%
GraphRAG [6]	0.514 [‡]	0.586 [‡]	—
HippoRAG 2 [13]	0.650 [‡]	0.710 [‡]	—
IRCoT [39]	0.53	0.65	—
HopRAG [28]	0.62	0.69	—
Beam Retrieval [45]	—	—	—
OPEN-RAG [17]	—	—	—
Self-RAG [2]	—	—	—
LATWEAVE (held-out)	0.865	0.872	92.2%
(tune pool $n = 10,179$, reference)	0.886	—	92.9%

[‡] Published values [22] carry protocol differences (1k-question subset, restricted corpus, Llama-3.3-70B reader): a paradigm reference, not a same-condition comparison. Beam Retrieval, OPEN-RAG, and Self-RAG are trained methods, listed for positioning without numbers.

Table 3: 2WikiMultihopQA per-type breakdown (held-out $n = 1,258$).

Question type	Share	EM	Δ vs. overall
bridge_comparison	21.9%	0.975	+11.0pp
comparison	24.2%	0.888	+2.3pp
compositional	41.7%	0.815	−5.0pp
inference	12.3%	0.794	−7.1pp
Overall	100%	0.865	—

5.3 Baselines and Condition Labeling

Baselines fall into two blocks. The *zero-training block* runs on the same corpus and the same scoring function as ours: a keyword baseline, Naive RAG (TF-IDF retrieval plus an LLM reader), and LatWeave. The *reference block* reports published values with protocol differences footnoted (GraphRAG 0.514[‡], HippoRAG 2 0.650[‡], IRCoT 0.53, HopRAG 0.62, and the supervised family on MetaQA); it serves as a paradigm reference only, and we draw no same-condition win/loss verdicts from it.

5.4 Metrics

Our metric family is organized around three abilities: *answering correctly* (EM / F1), *answering completely* (any-hit / complete-set recall), and *refusing honestly* (Faithfulness / dual measures / leak rate). EM and F1 come from a single scoring function: prediction and gold pass through the same text normalization; EM tests exact equality of the normalized strings, and F1 is token-level overlap. Since EM=1 forces F1=1.0, our F1 convention is equivalent to $F1^* := \max(F1_{\text{raw}}, EM)$. Every LLM call is pinned to temperature 0.0 and seed 88; the FRAMES capability classes are assigned by a frozen mechanical decision table whose rules read only the question and the gold answer—never the model output (Appendix B).

6 Main Results: Meet at Its Best

6.1 2WikiMultihopQA: The Main Battlefield

The breakdown brings the compare family to numbers for the first time: bridge_comparison (0.975) and comparison (0.888) lead, and these two are precisely the compare operator’s home ground

Table 4: MetaQA test main results ($n = 39,093$, single final run, frozen 9-shot configuration).

Hop	n	any-hit	recall (complete)	EM (strict)	set-F1
1-hop	9,947	0.9990	0.9987	0.9447	0.9798
2-hop	14,872	0.9983	0.9983	0.6239	0.9448
3-hop	14,274	0.9955	0.9954	0.5304	0.9369
All	39,093	0.9975	0.9973	0.6714	0.9508

(Section 3.2) — lattice-order side selection and Boolean comparison, where lattice comparability yields the answer directly with no generation.

On held-out ($n = 1,258$) LATWEAVE reaches EM 0.865, F1* 0.872, and resolved 92.2%, and on the resolved questions EM climbs to 0.938: on the questions it answers, the system is 93.8% precise. The relative gains over the same-protocol Naive RAG rerun (0.198) and the strongest published graph-RAG value (HippoRAG 2, 0.650) are 337% and 33% respectively. The tune pool ($n = 10,179$, disjoint from held-out) gives EM 0.886, 2.1pp above held-out — tuning did not overfit the held-out set. Ablation (Appendix Table 10) shows that from the dev-only baseline 0.731 to the frozen configuration the largest single increment comes from train-set transfer (+9.5pp), with alias normalization (+2.0pp) and rule-based re-extraction (+1.0pp) accounting for the rest.

6.2 MetaQA: The Representation-Independence Controlled Comparison

MetaQA is a templated QA dataset: every question is template-generated and every answer is drawn strictly from the knowledge graph, so on the same questions any hit-rate gap between the knowledge-graph and knowledge-lattice representations is fully attributable to the representation form itself. The most recent publicly available work on the same benchmark is LLM-Plan by Shrestha and Kim [34], evaluated on the identical 39,093-question test split (Table 5). The 2Wiki results above rest on LLM open extraction, so a natural objection is whether the performance comes from the extraction or from the lattice algebra itself. MetaQA is a purely structured KBQA benchmark (134,741 triples / 43,234 entities / 9 relations) whose triples map to lattice points with zero LLM—dimensions are the relations (9 forward + 9 inverse + 1 subject anchor), instances the subject- and object-side entities (269,482)—so the same engine runs losslessly when the knowledge source is swapped from text to KB, turning “the lattice is a general knowledge representation, not a byproduct of LLM extraction” into a falsifiable claim. KB coverage and seed-entity linkage are both 100% (measured), so the only source of loss is question parsing.

The evaluation setup mirrors the 2Wiki experiment: question-blind construction (minutes on kb.txt); configuration frozen on dev (9-shot, temp 0.0 / seed 88); test evaluated once. Parsing costs one LLM call per question (parse_fail = 0); query execution is a pure meet-algebra chain with zero LLM.

Three observations. (i) With no training, LATWEAVE reaches hits comparable to those of fully supervised methods: 1-hop (0.9990) and 3-hop (0.9955) match or exceed the best published values, while 2-hop (0.9983) comes within 0.07pp of the best (0.999). (ii) The hop-decay profiles are categorically different: KV-Mem degrades sharply

Table 5: MetaQA test vs. published KBQA methods (supervised rows: Hits@1 verified from source papers, converted to decimals). The Shrestha–Kim row reports Hit Rate—verbatim the any-hit measure, hence directly comparable to our any-hit row; their pooled micro-F1 is not cross-compared with our per-question set-F1 0.9508.

Method	1-hop	2-hop	3-hop	Paradigm (trained)
KV-Mem [30]	0.962	0.827	0.489	Memory networks (yes)
GraftNet [36]	0.970	0.948	0.777	Subgraph retrieval (yes)
SRN [31]	0.970	0.951	0.752	Neural reasoning (yes)
PullNet [35]	0.970	0.999	0.914	Graph retrieval (yes)
EmbedKGQA [33]	0.975	0.988	0.948	KGE (yes)
NSM [15]	0.971	0.999	0.989	State machine (yes)
RoG [29]	—	—	0.848	LLM+graph (1k fine-tune)
LLM-Plan + BFS [34]	0.999	0.999	0.970	LLM plan + BFS (no task training)
LATWEAVE (any-hit)	0.9990	0.9983	0.9955	Lattice meet (zero training)
LATWEAVE (recall, complete)	0.9987	0.9983	0.9954	gold $\subseteq$ returned set

with depth (0.962 $\rightarrow$ 0.489), even the strongest NSM shows visible 3-hop decay (0.999 $\rightarrow$ 0.989), whereas LATWEAVE is almost flat across three hops (total span 0.0035)—hop depth is not a difficulty variable for the lattice algebra. (iii) Every supervised baseline needs large-scale training (96k–330k questions), while LATWEAVE only reads the KB. Dev/test are highly symmetric (per-hop any-hit deviation $\leq 8 \times 10^{-4}$; empty = 69 (0.18%).

Completeness is the compare family’s second home ground: meet returns the full answer set deterministically, so completeness is a direct product of the semantics—ranking-style methods, which treat “gold enters the top- k candidates” as the hit endpoint, cannot structurally promise it. Our any-hit measure — 1 iff $\hat{A} \cap A^* \neq \emptyset$ —is the Hits@1 of the same returned set under an optimal internal ordering, hence an upper bound on Hits@1; with set-F1 0.9508 the actual gap is negligible. Tightness matters equally: on the single-answer subset ($|A^*| = 1$, $n = 14,834$) any-hit and ranked Hits@1 are strictly equivalent, measured 0.9974.

Appendix Table 11 shows that recall (complete)—1 iff $A^* \subseteq \hat{A}$ —is ≥ 0.996 in every answer-set-size bucket. Strict answer-set EM ($\hat{A} \equiv A^*$) is 0.6714 and decays with answer-set size (0.833 at $|A^*| = 1$, 0.379 at $|A^*| \geq 11$)—an artifact of set strictness: over 99% of EM misses still intersect the gold set, no under-answering miss exists, and all misses come from one or two extra or wrong entities.

The MetaQA control arm yields a direct production-route corollary: domains that already hold structured knowledge (KG / database / table) should map it straight to a lattice and skip LLM extraction entirely—zero training, zero fine-tuning, any-hit 0.9975, sub-50 ms retrieval, zero LLM. Three premises are stated honestly: question parsing remains the only source of loss; fan-out value selection still needs deterministic post-processing or a human in the loop; and the lattice inherits the KG’s missing edges and dirty triples losslessly—input quality is the ceiling.

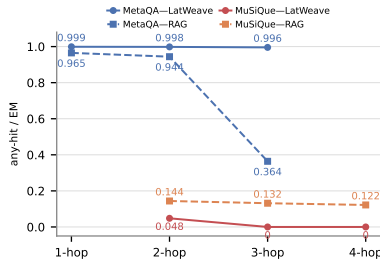


Figure 3: Hop decay in two contrasts. MetaQA (complete KB, any-hit): lattice meet chains are nearly lossless over three hops (0.9990 / 0.9983 / 0.9955), while the same-corpus RAG baseline (hits@1) decays to 0.364 at 3-hop. MuSiQue (open-text extraction, EM): lattice retrieval collapses to zero on multi-hop text composition (0.048 / 0 / 0), while RAG stays flat at a low 0.12–0.14 (insensitive to hop count).

7 Analysis: The Operating Envelope

This section turns the “operating envelope” from a position statement into a quantified characterization. The claim: on the three multi-hop boundary datasets (HotpotQA, MuSiQue; FRAMES in Appendix A), all losses fall outside the lattice-algebra layer, in two upstream layers—(a) open-text-to-structure conversion (coverage gaps and dimension fragmentation after whitelist governance; Sections 4.1 and 7.3); (b) natural-language-to-lattice-query parsing (query grounding and chain landing-point precision).

7.1 Failure Attribution: Four Boundary Classes

On the three multi-hop boundary datasets, the dominant cause of lattice-retrieval failure differs by class. On HotpotQA, the answer never enters the candidate set: 90.6% of residual broken chains lack the final-answer edge in the lattice (Appendix Table 12), and among resolved-but-wrong questions, 77% have their gold answer outside the returned set altogether; nor is the gap unrepaired—build-layer repair lifts the formerly hop-1-broken bucket from EM 0 to 17.3%, but conversion saturates and oracle injection falsifies same-mechanism repair (Section 7.3). MuSiQue’s attribution is compositional decay across hops: 48.3% of chains break at the first hop and a single failing hop fails the whole chain (EM 0.0248). FRAMES compounds the two. IIRC adds a fourth class—the answerability boundary (Section 8.3): answers must be derived on top of lattice values (time differences, counts), which exceeds the declared deterministic capability (Section 3.2); or the compared subject is absent from the lattice altogether—in both cases the correct behavior is to abstain rather than guess.

7.2 Hop Decay in Two Contrasts: Complete KB vs. Open Text

The dual-contrast figure (Figure 3) shows jointly that hop depth is not a difficulty variable for the lattice algebra itself; the degradation comes from the knowledge source, not the reasoning core. Corroborating per-hop conditional meet accuracy (Appendix Figure 5, MuSiQue tune pool $n = 2,175$): 0.37–0.44 at hop 1, 0.09–0.22 at hop 2, approaching zero at hop 3—each additional hop compounds

Table 6: Four levels of query grounding.

Level	Mechanism	Grounding object		Status
L0	Free generation	none		Rejected
L1	Global vocabulary	global relation names	relation	Internalized (HotpotQA)
L2	Entity menu	seed lattice neighbor-hood	entity’s neighbor-hood	Internalized (FRAMES)
L3	Per-hop replanning	intermediate entities		Not adopted (argued below)

conditional success approximately multiplicatively, and the first hop is the largest loss surface.

The RAG contrast: on MuSiQue held-out, RAG reaches EM 0.1364 above the lattice’s 0.0248; on HotpotQA held-out, EM 0.4683 above 0.1687—the coverage gap puts lattice retrieval’s full-pool EM at about one third of the same-corpus generative RAG baseline. Once facts are inside, lattice retrieval itself is accurate (resolved-EM 0.4545 on HotpotQA, 0.938 on 2Wiki). MuSiQue falls into three classes by break type: operational (OP_final, $n = 61$, EM 0.082); honestly resolvable (QU_honest, $n = 28$, EM 0.0357); and structural-abstention (QU_not_honest, anchor unresolvable, empty returned by design, $n = 153$, EM 0)—all 6 correct answers are 2-hop verbatim matches. On HotpotQA, comparison questions resolve at 70.1% while multi-hop meet questions resolve at only 29.6%: multi-hop meet must ground hop by hop, compounding per-hop in-lattice rates and landing-point errors multiplicatively.

7.3 Query-Grounding Levels and Loss Upper Bounds

The oracle-plan upper bound separates planner loss from lattice capability in numbers: on 2Wiki, resolved-EM is 0.938 (vs. 0.865 full-pool) and on HotpotQA 0.4545 (vs. 0.1687)—given successful parsing and chaining, on-lattice execution is far above the end-to-end numbers. Appendix Table 12 puts the planner-attributable share of HotpotQA’s residual broken chains at 9.4%, and FRAMES’ oracle ceiling is only 4.9% (vs. EM 2.56%)—on neither boundary is the planner the main bottleneck, so repair investment shifts to the build layer (measured limits below).

Query grounding refers to where the planner’s relation names come from when it lays out a chain—generated from thin air, chosen from a global vocabulary, or chosen from an entity’s in-lattice relation menu (Table 6). We have internalized L1 and L2 in the two boundary experiments respectively: the HotpotQA planner is injected with the global relation vocabulary; the FRAMES planner first queries which relations the seed entity actually has in the lattice and is restricted to that menu—relation names are thereby anchored to edges that genuinely exist on that entity. Yet both experiments show that L1/L2 remain insufficient, for structural reasons.

The structural limitation is a circular dependency: static planning can, by construction, ground only the first hop—the intermediate entity of hop- k ($k \geq 2$) is unknown until hop- $(k-1)$ has executed, so its dimension menu is unavailable at planning time. The FRAMES chain-break dissection shows this directly: an intermediate entity typically already carries 64–129 dimensions in the lattice, yet the

Table 7: Operating envelope (all numbers are frozen final values).

Dataset	Knowledge-source form	Main bottleneck	Result
2WikiMultihopQA	Open text extraction, templated questions	—(home ground)	EM 0.865
MetaQA	KB triples mapped directly	Question parsing only	any-hit 0.9975
HotpotQA	Open text extraction, free paragraphs	Coverage gap (90.6%)	EM 0.1687
MuSiQue	Open text extraction, 4-hop chains	Chain deepening (48.3% hop-1 break)	EM 0.0248
FRAMES (dev $n = 742$)	Open text extraction, multi-hop + factuality	Coverage gap + chain-design error	EM 0.0256
IIRC (held-out $n = 130$)	Open text extraction, incomplete information	Answerability boundary	Prec. measure 0.9437

break-point relation (e.g., `nominator_for`) is still absent—the break is a chain-design error (planning a relation the entity does not have), against which re-extraction is ineffective in principle: exactly the position L2 cannot reach and only L3 could.

Why not adopt L3? Three lines of experimental evidence. (1) FRAMES has internalized L2, yet EM is still only 2.56%; a per-hop replanning pilot hits 0 of 15 questions—the intermediate entities of broken chains are simply not in the lattice (12 of 13 dissected cases), so replanning has nothing to choose from. (2) HotpotQA hop-2 oracle injection gives L3’s capability upper bound: placing the gold answer directly into the correct hop-2 dimension still fails to converge on 11 of 13 questions—the residual is an upstream coverage / landing-point problem that no L3-level improvement can address. (3) L3 requires one LLM call per hop, degrading the system to IRCot-style iterative retrieval, which is incompatible with this paper’s “zero LLM on the answer path” positioning.

HotpotQA re-extraction, measured (details in Appendix E): re-playing all formerly hop-1-broken questions ($n = 1,509$) under the frozen repaired configuration lifts first-hop connectivity to 78.9%, but only 21.9% of connected questions are ultimately answered correctly (bucket EM 0 $\rightarrow$ 17.3%); extrapolated to all first-hop gaps, full-pool EM gains only about +0.6pp. A per-question residual decomposition rewrites the bottleneck from “choosing wrong among candidates” to “gold not in the returned set” (77% of resolved-but-wrong questions; returned sets average just 3.6 candidates). Fundamental improvement requires changing the extraction mechanism itself; the IIRC value-cleaning repair (Section 8.2) has not been applied here (future work). Table 7 summarizes the operating envelope with frozen final values.

8 Abstention as a First-Class Capability

8.1 IIRC: Abstention in Practice

Traditional EM/F1 mixes questions on which the system should abstain with the answerable ones, hiding the cost of bluffing. We separate a *recall measure* (EM on answerable questions) from a *precision measure* (abstain accuracy $\times (1 - \text{leak rate})$), making “knowing when not to answer” reportable, ablatable, and regression-testable for the first time. IIRC questions are written by humans who cannot see the linked documents, so a substantial fraction is unanswerable in context (the *none* class). From the official dev split (1,301 questions) we stratify by `answer_type` (seed 88) into a frozen held-out

Table 8: IIRC main results (held-out $n = 130$; dev_tune as drift check).

Metric	Held-out 130	dev_tune 1,171
EM _{full} (Phase-1-hit denominator)	0.0417 (1/24)	0.0264 (6/227)
Recall measure (answerable EM)	0.0105	0.0070
Resolved (Phase-1 hit rate)	25.3% (24/95)	26.4%
Leak rate (none answered)	0.029 (1/35)	0.016 (5/312)
Abstain accuracy	0.971 (34/35)	0.984
Precision measure (abstain $\times$ non-leak)	0.9437	0.9682

set ($n = 130$: span 59 / value 23 / binary 13 / none 35) and a tune pool (remaining 1,171; no model weights trained). The held-out set is tested once under the Phase-1 strict exact-only lattice-direct protocol—38 LLM calls, all query-side—and that run is the final score.

Held-out and tune pool agree item by item (Table 8): zero drift. The single leak is a Phase-1 value-match false positive, not generative hallucination. The 35 unanswerable questions split into 21 no_hit, 13 sentinel abstains, and 1 leak; the 93 answerable abstains split into 71 no_hit (39 in-grid / 32 not-in-grid) and 22 sentinel refusals (two-axis attribution: phase $\times$ gold-in-grid, Appendix Table 13).

8.2 Value Cleaning (Ablation)

A dev-side ablation (IIRC dev $n = 1,301$, Phase-1 only, same pipeline; details in Appendix F) rewrites sentence-level dimension values into clean entities on the ingestion side: EM_{full} rises 0.0156 $\rightarrow$ 0.0279 (+79%) and the leak rate falls 0.052 $\rightarrow$ 0.017 (−67%)—the leak reduction far exceeds the recall gain, a payoff only the two-measure separation can characterize.

8.3 Two Honest Boundaries

(a) *value boundary* (IIRC dev 233 questions; held-out 23 all abstained or wrong): answers are 100% derived quantities whose anchors exist in the lattice only $\sim 1.5\%$ of the time—abstaining is correct (echoing Section 3.2). (b) *binary boundary* (IIRC dev 128 questions; held-out 13): the compared entities are mostly not lattice subjects, so Boolean comparison cannot execute—compare degrades to abstain, its fourth form (§3.2). Low EM and high abstain accuracy are two faces of one boundary: with no anchor in the lattice, not answering is what the dual-measure framework rewards.

9 Conclusion

We presented LATWEAVE: multi-hop QA compiled into three deterministic lattice operators—*meet*, *compare*, *abstain*—with LLMs confined to construction and query planning; the answer path is zero-LLM, zero-task-training, and auditable end to end. Across six public benchmarks, deterministic execution pays no penalty inside its operating envelope, and failure ends in honest abstention. The method has been tested in an actual production environment. Deterministic reasoning is not meant to replace LLMs, but to lift the answer path from a probabilistic black box into an auditable white box.

References

- [1] Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. 2023. Conformal risk control. In *Proceedings of Machine Learning Research (PMLR)*, Vol. 202. 2610–2639. arXiv:2208.02814.

- [2] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*.
- [3] Jiaxin Bai, Wei Fan, Qi Hu, Qing Zong, Chunyang Li, Hong Ting Tsang, et al. 2026. AutoSchemaKG: Autonomous knowledge graph construction through dynamic schema induction from web-scale corpora. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL)*. arXiv:2505.23628.
- [4] Brian A. Davey and Hilary A. Priestley. 2002. *An Introduction to Lattices and Order* (2nd ed.). Cambridge University Press. doi:10.1017/CBO9780511809088
- [5] Robert P. Dilworth. 1950. A decomposition theorem for partially ordered sets. *Annals of Mathematics* 51, 1 (1950), 161–166.
- [6] Darren Edge, Ha Trinh, Nathan Cheng, Joshua Bradley, Alex Chao, Avik Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph RAG approach to query-focused summarization. *arXiv preprint arXiv:2404.16130* (2024).
- [7] Ran El-Yaniv and Yair Wiener. 2010. On the foundations of noise-free selective classification. *Journal of Machine Learning Research (JMLR)* 11, 53 (2010), 1605–1641.
- [8] James Ferguson, Matt Gardner, Hannaneh Hajishirzi, Tushar Khot, and Pradeep Dasigi. 2020. IIRC: A Dataset of Incomplete Information Reading Comprehension Questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 1137–1147. doi:10.18653/v1/2020.emnlp-main.86
- [9] Rong Fu, Yemin Wang, Tianxiang Xu, Yongtai Liu, Weizhi Tang, Wangyu Wu, Xiaowen Ma, and Simon Fong. 2026. S-Path-RAG: Semantic-aware shortest-path retrieval augmented generation for multi-hop knowledge graph question answering. In *Proceedings of the ACM Web Conference 2026 (WWW)*. 4057–4068. doi:10.1145/3774904.3792459 arXiv:2603.23512.
- [10] Bernhard Ganter and Rudolf Wille. 1999. *Formal Concept Analysis: Mathematical Foundations*. Springer. doi:10.1007/978-3-642-59830-2
- [11] Yonatan Geifman and Ran El-Yaniv. 2017. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 30. arXiv:1705.08500.
- [12] Zirui Guo, Lianhao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2024. LightRAG: Simple and Fast Retrieval-Augmented Generation. *arXiv preprint arXiv:2410.05779* (2024).
- [13] Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From RAG to memory: Non-parametric continual learning for large language models. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*. arXiv:2502.14802.
- [14] Haoyu Han, Kai Guo, Harry Shomer, Yu Wang, Yucheng Chu, Hang Li, Li Ma, and Jiliang Tang. 2026. Reasoning by Exploration: A Unified Approach to Retrieval and Generation over Graphs. In *Proceedings of the ACM Web Conference 2026 (WWW)*. arXiv:2510.07484.
- [15] Gaole He, Yunshi Lan, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining (WSDM)*. 553–561.
- [16] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*. 6609–6625.
- [17] Shayekh Bin Islam, Md Asib Rahman, K S M Tozammel Hossain, Enamul Hoque, Shafiq Joty, and Md Rizwan Parvez. 2024. OPEN-RAG: Enhanced Retrieval-Augmented Reasoning with Open-Source Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. Association for Computational Linguistics, Miami, Florida, USA, 14231–14244. doi:10.18653/v1/2024.findings-emnlp.831 arXiv:2410.01782.
- [18] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *Comput. Surveys* 55, 12 (2023), 1–38. doi:10.1145/3571730
- [19] Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active Retrieval Augmented Generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 7969–7992.
- [20] Adam Tauman Kalai and Santosh S. Vempala. 2024. Calibrated Language Models Must Hallucinate. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC 2024)*. ACM, 160–171. doi:10.1145/3618260.3649777
- [21] Amita Kamath, Robin Jia, and Percy Liang. 2020. Selective question answering under domain shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*. 5684–5696. doi:10.18653/v1/2020.acl-main.503
- [22] Youmin Ko, Sungjong Seo, and Hyunjoon Kim. 2025. Cooperative retrieval-augmented generation for question answering: Mutual information exchange and ranking by contrasting layers. In *Advances in Neural Information Processing Systems 39 (NeurIPS)*. arXiv:2512.10422; Appendix C.2 reproduces the full EM/F1 table for the GraphRAG/LightRAG/HippoRAG family under the HippoRAG 2 protocol.
- [23] Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananeey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. 2025. Fact, Fetch, and Reason: A Unified Evaluation of Retrieval-Augmented Generation. In *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. Association for Computational Linguistics, Albuquerque, New Mexico, 4745–4759. doi:10.18653/v1/2025.naacl-long.243
- [24] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 33. 9459–9474.
- [25] Lei Liang, Zhongpu Bo, Zhengke Gui, Zhongshu Zhu, Ling Zhong, Peilong Zhao, Mengshu Sun, Zhiqiang Zhang, Jun Zhou, Wenguang Chen, Wen Zhang, and Huajun Chen. 2025. KAG: Boosting LLMs in Professional Domains via Knowledge Augmented Generation. In *Companion Proceedings of the ACM on Web Conference 2025*. 334–343. doi:10.1145/3701716.3715240
- [26] Wen-Sheng Lien, Yu-Kai Chan, Hao-Lung Hsiao, Bo-Kai Ruan, Meng-Fen Chiang, Chien-An Chen, Yi-Ren Yeh, and Hong-Han Shuai. 2026. HyperRAG: Reasoning N-ary Facts over Hypergraphs for Retrieval Augmented Generation. In *Proceedings of the ACM Web Conference 2026 (WWW)*. 2465–2476. doi:10.1145/3774904.3792710 arXiv:2602.14470.
- [27] Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 3214–3252. doi:10.18653/v1/2022.acl-long.229
- [28] Hao Liu, Zhengren Wang, Xi Chen, Zhiyu Li, Feiyu Xiong, Qinhan Yu, and Wentao Zhang. 2025. HopRAG: Multi-Hop Reasoning for Logic-Aware Retrieval-Augmented Generation. In *Findings of the Association for Computational Linguistics: ACL 2025*. Association for Computational Linguistics, Vienna, Austria, 1897–1913. doi:10.18653/v1/2025.findings-acl.97
- [29] Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2024. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*.
- [30] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1400–1409.
- [31] Yunqi Qiu, Yuanzhuo Wang, Xiaolong Jin, and Kun Zhang. 2020. Stepwise reasoning for multi-relation question answering over knowledge graph with weak supervision. In *Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM)*. 474–482.
- [32] Jing Ren, Bowen Li, Ziqi Xu, Xikun Zhang, Haytham Fayek, and Xiaodong Li. 2026. When to Trust: A Causality-Aware Calibration Framework for Accurate KG-RAG. In *Proceedings of The Web Conference (WWW)*. arXiv:2601.09241.
- [33] Apoorv Saxena, Aditya Tripathi, and Partha Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*. 4498–4507.
- [34] Manil Shrestha and Edward Kim. 2025. Efficient Multi-Hop Question Answering over Knowledge Graphs via LLM Planning and Embedding-Guided Search. arXiv:2511.19648 [cs.CL]
- [35] Haitian Sun, Tania Bedrax-Weiss, and William Cohen. 2019. PullNet: Open domain question answering with iterative retrieval on knowledge bases and text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2380–2390.
- [36] Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 4231–4242.
- [37] Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR 2024)*.
- [38] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multihop Question Answering via Single-Hop Question Composition. In *Transactions of the Association for Computational Linguistics (TACL)*, Vol. 10. MIT Press, 539–554.
- [39] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*. 10014–10037. doi:10.18653/v1/2023.acl-long.557
- [40] Spencer Whitehead, Suzanne Petryk, Vedaad Shakib, Joseph Gonzalez, Trevor Darrell, Anna Rohrbach, and Marcus Rohrbach. 2022. Reliable Visual Question

- Answering: Abstain Rather Than Answer Incorrectly. In *Computer Vision – ECCV 2022, Part XXXVI (Lecture Notes in Computer Science, Vol. 13696)*. Springer, 148–166. doi:10.1007/978-3-031-20059-5_9 arXiv:2204.13631.
- [41] Rudolf Wille. 1982. Restructuring lattice theory: An approach based on hierarchies of concepts. In *Ordered Sets*. Springer, 445–470. doi:10.1007/978-94-009-7798-3_15
- [42] Wenhan Xiong, Xiang Lorraine Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Wen-tau Yih, Sebastian Riedel, Douwe Kiela, and Barlas Oğuz. 2021. Answering Complex Open-domain Questions with Multi-hop Dense Retrieval. In *Proceedings of the Ninth International Conference on Learning Representations (ICLR 2021)*.
- [43] Hao Yang, Wei Wei, Zhiyu Yang, Xupeng Zhang, Yunjie Zhang, and Lin Yang. 2026. CompactRAG: Reducing LLM calls and token overhead in multi-hop question answering. In *Proceedings of the ACM Web Conference 2026 (WWW)*. 2240–2251. doi:10.1145/3774904.3792512 arXiv:2602.05728.
- [44] Zhilin Yang, Pengqi Peng, Zhen Dai, Yanxia Hu, Jaime Carbonell, and Ruslan Salakhutdinov. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2369–2380. doi:10.18653/v1/D18-1259
- [45] Jiahao Zhang, Haiyang Zhang, Dongmei Zhang, Yong Liu, and Shen Huang. 2024. End-to-End Beam Retrieval for Multi-Hop Question Answering. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. 1718–1731.
- [46] Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander Smola, and Le Song. 2018. Variational reasoning for question answering with knowledge graph. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*. 6069–6076.

A FRAMES: Controlled Single-Variable Ablation—Locating the Failing Stage

FRAMES (Krishna et al., 2025; factuality $\times$ multi-hop, 824 questions, each with gold evidence pages) is used here not as an additional, harder stress test but as a controlled single-variable ablation that separates two stages—whether the seed anchor lands in the lattice, and whether the chain semantics line up—to locate which stage the chain break actually occurs in. We apply a uniform 90/10 development/frozen split (seed 88): dev 742 questions for development, held-out 82 frozen and used for final testing only; construction is question-blind open extraction, with the LLM used only for query-side planning.

Main results: dev EM = 19/742 (2.56%), oracle diagnostic ceiling (gold falls in the candidate set; not the main EM measure) 36/742 (4.9%). The held-out 82 are reported under two settings: evidence-page-excluded construction gives EM = 1/82 (F1 = 0.0521), evidence-page-included construction gives EM = 1/82 (F1 = 0.1826)—the two EMs are identical (statistically inseparable), yet including the evidence pages lifts seed-entity resolution from 20.7% to 100% and the share of chain-walkable questions from 15% to 87%. This is the controlled single-variable conclusion of this appendix: adding the pages rescues the first stage (“does the anchor land”), while the bottleneck at the second stage (“does the chain line up”) is unchanged—so the failing stage is precisely located in chain semantics, not in extraction coverage.

Three-line falsification chain: (1) after executor-layer fixes EM is flat, a net effect of zero; (2) a replanning pilot scores 0 hits on 15 questions — when the break is a coverage gap there is no alternative chain to take; (3) dissecting 13 questions from the targeted re-extraction candidate list, 12 have an intermediate entity that is simply not in the lattice (never landed as a lattice point), while in 4 control cases (where the entity has 64–129 dimensions in the lattice) the break-point relation still does not exist in its instance set—the break comes from the planner inventing a relation the entity

does not have, a chain-design error against which re-extraction is ineffective in principle. The three lines jointly support Section 7.3.

Capability strata: FACT_MEET / BOOL / EXTREME / COMPUTE, with the decision table in Appendix B. Transferable conclusion: filling coverage does not equal getting chains right—construction coverage is only a necessary condition, whereas chain-semantic alignment and chain landing-point precision (whether gold enters the returned set) are the bottleneck on the sufficient-condition side (in the same direction as the HotpotQA H2 oracle injection experiment).

B Evaluation Protocol Details

Measures: EM/F1 follow each benchmark’s official evaluation script, and F1* is this paper’s variant (defined in Section 5.4). Seed 88 throughout; the 90/10 split manifests are frozen as files (each benchmark’s held-out manifest is on disk). LLM configuration: temp 0.0, context_size 32K (configurable), max_tokens 1280 (with automatic escalation to 2048 on truncation). The FRAMES capability-class decision table (FACT_MEET / BOOL / EXTREME / COMPUTE) is included in the supplementary material.

C Prompt Templates

The extraction prompt (hybrid extraction, A/B modes), the planner prompt (including the KNOWN RELATIONS injection and the entity-menu variant), and the RAG baseline reader prompt are provided in full in the supplementary material.

D Construction Statistics

Per-domain scale (frozen final values): the 2Wiki dataset has 1.2M instances (125,760 passages); MetaQA has 19 dimensions / 269,482 instances (134,741 triples / 43,234 entities); the full MuSiQue lattice has 40,566 instances / 1,741 dimensions, and the held-out-only sublattice has 3,996 instances / 535 passages / 502 dimensions / 5,202 nodes; HotpotQA has 852 dimensions / 120,981 instances / 26,258 subject nodes; IIRC has 350 normalized dimensions / 6,768 instances.

E HotpotQA Re-extraction Repair, Measured

Can re-extracting the same corpus at the build layer (same open extractor) repair the coverage gap? Two small-scale experiments answer no. Hop-1 ($n = 150$): only 16 questions (10.7%) gain a genuine first-hop bridge in the lattice (an edge connecting the seed to the gold intermediate entity); and connectivity is not correctness—replaying all formerly hop-1-broken questions ($n = 1,509$) under the frozen configuration (re-extraction, relation mapping, and planner fixes included), the first-hop connectivity rate is 78.9%, 21.9% of connected questions are ultimately answered correctly (51.3% resolved), and the bucket’s EM rises from 0 to 17.3% — the repair works but has a clear ceiling; the rest either break at later hops or the answer never enters the returned set (landing points and value granularity). Extrapolating with the measured 10.7% bridge-recovery rate and 21.9% conversion, even re-extracting all 1,613 first-hop gaps would add only about +0.6pp to full-pool (dev 6,664) EM, optimistically. Hop-2 ($n = 36$): conversion from re-extraction to a fully correct answer is only 5.6% (2/36; small n , 95% confidence interval roughly 0.7%–18.9%). A per-question residual decomposition (diagnostic

statistics on the dev tune pool $n = 6,664$, held-out untouched) rewrites the bottleneck from “choosing wrong among candidates” to “gold not in the returned set”: among resolved-but-wrong questions ($n = 1,437$), 77% have their gold answer outside the system’s returned candidate set at all—landing-point drift, coverage gaps, or extraction granularity; only about 11% have the gold’s content words fully inside the returned set (mostly as noisy long strings). This 77% is the residual after three repair paths—executor fixes (net-zero effect), per-hop replanning (0 of 15 hits), and re-extraction—have all been rejected; its magnitude corresponds to the true scale of the coverage gap on boundary datasets (Appendix Table 12), not an artifact of the evaluation protocol. Questions whose returned set contains two or more candidates account for only 9.3% (averaging just 3.6 candidates), and since the primary metric is any-hit (gold in the returned set counts as correct, Section 5.4), “choosing the wrong value among candidates” affects only a small share.

F Value Cleaning Ablation (IIRC dev)

An early version of open extraction poured whole sentences into dimension values (value = sentence), causing two kinds of contamination: on answerable questions value matching failed and depressed the recall measure, and on unanswerable questions sentence-level values were partially matched and triggered leaks. Value cleaning rewrites sentence values of entity-typed dimensions into clean entities on the ingestion side only, with no change to query or lattice logic: EM_{full} rises from 0.0156 to 0.0279 (+79%), the leak rate falls from 0.052 to 0.017 (−67%), abstain accuracy rises from 0.948 to 0.983, and the precision measure rises from 0.899 to 0.966. The leak reduction far exceeds the recall gain—a payoff pattern that only the separation of the two measures can characterize, and direct evidence of the framework’s diagnostic value.

G Additional Results Tables

Tables 9–13 collect the supporting detail tables referenced from the main text: the approach-family positioning table, the 2WikiMultiHopQA ablation, the MetaQA answer-set-size bucketing, the cross-dataset failure attribution matrix, and the per-question IIRC abstention attribution.

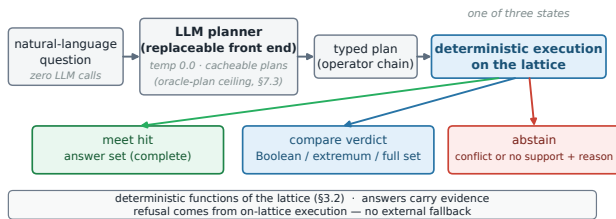


Figure 4: Query-time flow. The LLM planner compiles the question into a typed plan; deterministic execution on the lattice returns a *meet* hit, a *compare* verdict, or *abstain*; the answer ships with its evidence text.

Table 9: Positioning: only LATWEAVE ticks all four columns. Zero-training is graded because several families mix trained and training-free members (see footnotes). Detection (auditability) is graded because chunk-ID provenance yields engineering-level auditability only: the step from retrieved passage to answer remains unverifiable.

Approach family	Zero-train	Determ.	Audit.	Abstain
Neural multi-hop retrievers	partial ¹	×	×	×
RAG / GraphRAG	partial ²	×	eng.-level	×
KBQA / semantic parsing	×	✓	✓	×
Selective prediction	×	—	×	prob. thresh.
LATWEAVE (this work)	✓	✓	✓	✓

¹ MDR and Beam Retrieval train in-task retrievers; IRCOT and HopRAG are training-free (LLMs in the loop). ² Self-RAG and RoG fine-tune; FLARE, GraphRAG, HippoRAG 2 and KAG [25] are training-free. KAG further compiles questions into logical forms for hybrid reasoning, yet the final answer is still LLM-synthesized. Causality-aware recalibration of KG-RAG confidence [32] is likewise training-free, yet it calibrates a probabilistic answer generator rather than removing one. Self-RAG and FLARE gate whether to retrieve, not whether to answer: the model still generates an answer, so they provide no abstention mechanism (cf. Section 2.4, where selective prediction does, but via confidence thresholds). ³ Exception: LLM-planned BFS over a given KG [34] needs no task training (Section 2.3). Schema-free KG construction from text [3] relaxes the predefined schema, but its query side is probabilistic retrieval over the induced graph, not deterministic execution. ⁴ Requires a trained selector or held-out calibration; conformal risk control calibrates the abstention threshold distribution-free (removing calibration drift) [1], yet abstention remains a thresholding operation and the underlying answer generator remains probabilistic. Trained refusal rules can also be tuned to abstain, but such a refusal is a learned probabilistic decision, not a structural guarantee that can be audited independently of the model.

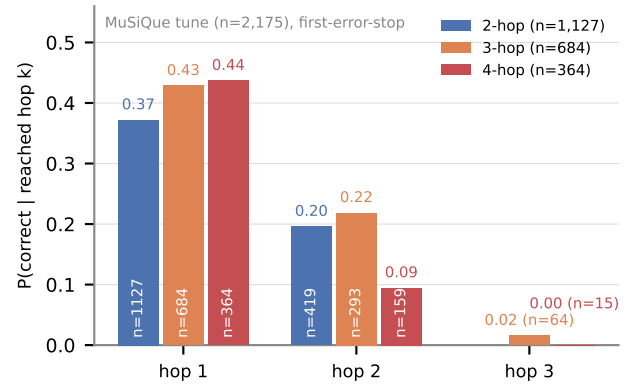


Figure 5: Per-hop conditional meet accuracy $P(\text{correct} \mid \text{chain reaches hop } k)$ on the MuSiQue tune pool ($n = 2,175$), grouped by the question’s true depth.

Table 10: 2WikiMultiHopQA ablation (dev tune pool, historical protocol $n = 11,318$).

Configuration	EM	Resolved
Full system	0.873	92.0%
– train gold transfer (dev-only)	0.731	85.3%
– alias normalization	0.853	90.8%
– rule-based re-extraction (E15)	0.843	90.4%

The full system, replayed after domain rebuild (grid-point hash dedup, 1.2M instances), gives tune-pool EM 0.886 / held-out 0.865; the remaining rows are historical-protocol ablation values.

Table 11: MetaQA test bucketed by gold answer-set size (completeness structural analysis).

$ A^* $	n	any-hit	recall (complete)	EM (strict)
1	14,834	0.9974	0.9974	0.8332
2	5,729	0.9977	0.9977	0.7413
3–5	7,277	0.9974	0.9968	0.6442
6–10	4,041	0.9968	0.9968	0.5494
≥ 11	7,212	0.9978	0.9976	0.3788
All	39,093	0.9975	0.9973	0.6714

Table 12: Failure attribution matrix.

Dataset	Attribution	Count (share)
HotpotQA (residual broken chains)	Extraction coverage gap	667/736 (90.6%)
	H1 missing / H2 missing / seed no-out-edge	527 / 81 / 59
	Query-layer planner	69/736 (9.4%)
MuSiQue (chain-break locus)	Break at hop 1	117/242 (48.3%)
	Break at hop 2	81/242 (33.5%)
	Break at hop 3 / hop 4	16 / 4 (6.6% / 1.7%)
	Chain survives (resolved-no-break: 6 correct / 18 wrong)	24/242 (9.9%)
FRAMES (dev $n = 742$)	Coverage gap compounded by chain-design error	12/13 dissected: intermediate entity off-lattice

Table 13: IIRC held-out abstention attribution on two axes (phase $\times$ gold-in-grid), per-question reproducible ($n = 130$). The in-grid probe applies only to answerable questions—unanswerable ones have no gold answer by construction. The 22 answerable sentinel abstains split evenly (11 in-grid / 11 not-in-grid). The broken-chain bucket is empty by design on IIRC: this arm is Phase-1 exact-only single-value mapping with no multi-hop chain to break (chain-break attribution belongs to Section 7.1). A dev-scale isomorphic analysis (954 answerable questions, older grid convention) gives the same proportional structure.

Phase	Disposition	Question class	
		None ($n=35$)	Answerable ($n=95$)
Phase-1 hit	answered correct	—	1
	answered wrong	—	1
	sentinel abstain	13	22
no_hit	gold in grid (“not met”)	n/a	39
	gold not in grid (“never extracted”)	n/a	32
	subtotal	21	71
Leak (vacuous pass)		1	—
Total		35	95

H Auditability and Material Availability

To ensure reproducibility and auditability, we provide, with the supplementary material, all prompt templates (Appendix C), the held-out split manifests, per-question evaluation details, and the failure-attribution probe results, so that the numbers on all six benchmarks can be independently recomputed under this paper’s evaluation protocol; because the system implementation involves industrial deployment, the source code is not part of the supplementary material.