RECON: BENCHMARKING AGENT MEMORY FOR COMPOSITIONAL REASONING OVER LONG CONTEXTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models and LLM-based agents are widely used as personal chat assistants, enterprise copilots, and autonomous workflow agents. In all these applications, memory (the ability to retain, access, and reason over information accumulated over long contexts and multiple interactions) plays a crucial role in determining the reliability of any agent. We introduce RECON (Reasoning over Extended Contexts with Obfuscated Narratives), a benchmark for evaluating compositional reasoning over long contexts. RECON spans 24 case files across three domains (criminal, medical, and financial), each ranging from 50k to 100k tokens, and tests agents on **six memory-intensive tasks**: reconstructing multi-hop evidence chains, propagating cascading invalidations, resolving source conflicts, counterfactual reasoning, satisfying temporal constraints, and temporal fact retrieval. Recent memory benchmarks evaluate whether agents can retrieve scattered facts or detect if a fact has changed whereas RECON evaluates what happens after the change, whether agents can trace which downstream conclusions are affected, which survive through independent support, and how alternative timelines would have unfolded. Our evaluation reveals substantial limitations across current architectures: no non-Oracle system exceeds 22.4% Accuracy, with retrieval and reasoning each surfacing as challenges. Our benchmark and code are available at `https://anonymous.4open.science/r/RECON-Bench`.

1 INTRODUCTION

Large Language Model based agents are tasked with increasingly high-stakes work. Coding assistants such as Claude Code and GitHub Copilot must track context across entire codebases (Ding et al., 2023; Liu et al., 2024), conversational agents such as ChatGPT and Claude must retain user preferences and instructions that change over time (Wu et al., 2025; Du et al., 2024), and enterprise copilots must reason over evolving documents in clinical, legal, and financial workflows (Fleming et al., 2023). Without robust memory, an agent that forgets an earlier fact was revised, whether a lab result updated on Day 9, a witness statement contradicted on Day 5, or a flagged transaction reversed, hallucinates stale facts, misses critical updates, and loses track of how earlier evidence connects to later conclusions. Memory quality directly determines the quality of everything built on top of it.

The importance of memory has driven a wave of memory architectures such as **Mem0** (Chhikara et al., 2025), **Zep** (Rasmussen et al., 2025), **MemGPT** (Packer et al., 2023), and **Hindsight** (Latimer et al., 2025), which index, graph, page, or summarize prior interactions. Beyond these memory architectures, RAG approaches embed documents in a vector store and retrieve relevant chunks at query time. Modern long-context models with 100K token windows place the full context in a single prompt. Each strategy trades accuracy against cost, but the fundamental question remains the same: *does the agent remember not just the facts, but what each of its conclusions depends on?*

We introduce **RECON** (**R**easoning over **E**xtended **C**ontexts with **O**bfuscated **N**arratives), a benchmark which evaluates whether agents can maintain a coherent, evolving understanding over long contexts where facts don't just accumulate but also interact, contradict, and cascade. RECON spans 24 case files and 1,604 questions across three domains (criminal, medical, and financial), with each case ranging from 50k to 100k tokens and fully deterministic ground truth. It tests agents on six memory-intensive tasks, illustrated in Figure 2.

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

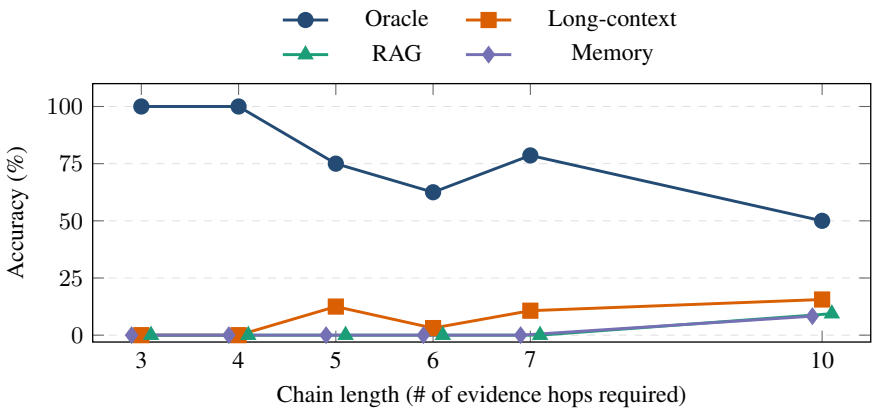

Figure 1: **Depth bounds the control; retrieval bounds everything else.** Chain Reconstruction Accuracy by number of dependency hops, per system family. The retrieval-perfect condition, given the ground-truth dependency graph instead of narrative, still falls from 100% at depth 3–4 to 50% at depth 10, while every other family stays near zero, failing before depth becomes the binding constraint. This is possible only because each question is emitted from a known dependency subgraph, so depth varies with retrieval difficulty held fixed.

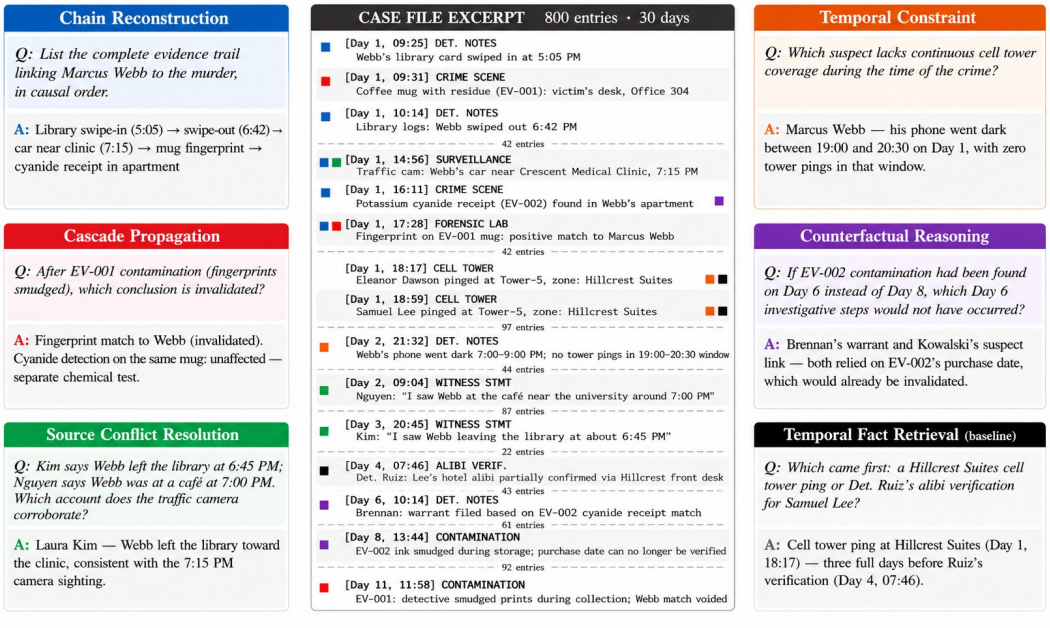

Figure 2: The six memory-intensive tasks evaluated by RECON.

1. **Chain Reconstruction**: locate and causally order multi-hop evidence paths (3–10 hops per question, sampled from 5–15-hop chains) scattered across the document.
2. **Cascade Propagation**: after an invalidation event, determine which conclusions break and which survive via independent support.
3. **Source Conflict Resolution**: adjudicate contradictory accounts using independent corroborating evidence.
4. **Counterfactual Reasoning**: determine what would have changed under an alternative timeline.
5. **Temporal Constraint Satisfaction**: cross-reference parallel data streams against a time window.

6. **Temporal Fact Retrieval**: baseline temporal ordering, state queries, and elimination (control).

Existing memory benchmarks evaluate whether agents can retrieve scattered facts or detect if a fact has changed, modeling memory as a *state machine* in which facts have current values to be tracked. But tracking current values is not enough: in real-world workflows, facts interact, invalidate, and cascade through explicit dependency structures.

Because each question is emitted from a dependency subgraph that is known in advance, RECON does not merely record that a system fails; it localizes where the failure occurs. We apply it to sixteen systems spanning long-context models, retrieval-augmented pipelines, and agentic memory architectures, together with a *retrieval-perfect* control condition that replaces the narrated case file with the ground-truth dependency graph, removing narrative retrieval. Neither axis alone explains the difficulty: the retrieval-perfect condition reaches only 54.6% Accuracy, and roughly four in five questions whose full gold evidence is verifiably retrieved are still answered incorrectly. Varying compositional depth directly separates the two: on Chain Reconstruction, retrieval-perfect Accuracy falls from 100% at depth 3–4 to 50% at depth 10 (Figure 1), while every non-oracle family remains near zero across the entire range. Degradation therefore tracks the number of dependency hops that must be composed, and it persists when the dependency structure is supplied to the model outright.

Contributions.

- **A generator for long contexts with known provenance structure.** A six-stage deterministic pipeline constructs each case as a provenance graph carrying explicit invalidation events and counterfactual dependencies, and derives every gold answer in code; an LLM is confined to linguistic realization and never enters the gold reasoning trace. The released benchmark is one seeded instantiation of this pipeline: 1,604 questions over 24 case files (50K–100K tokens) in three domains, with human validation reporting $\kappa = 0.69$ on a stratified 200-question sample.
- **A means of attributing agent failure rather than only recording it.** Each question inherits from its generating subgraph a known gold evidence set, a known compositional depth, and a known reasoning type. This supports three controlled decompositions that benchmarks sourced from existing narratives cannot perform, because the true dependency structure of a found document is unavailable: a retrieval-perfect control that removes narrative retrieval, per-question gold-evidence recall that conditions accuracy on whether the required evidence was in fact available, and a within-task depth sweep that varies the number of composition steps with the task held fixed.
- **An evaluation of sixteen systems and a retrieval-perfect control** across long-context, retrieval-augmented, and agentic memory architectures, identifying the dominant bottleneck within each task family and showing that depth limits performance even with the dependency graph supplied. We release the dataset, generator, and evaluation harness.

2 Related Work

Table 1 situates RECON among prior long-context and memory benchmarks across the six task types.

2.1 Long-Context Evaluation

Long-context benchmarks evaluate whether models can retrieve and understand information distributed across long inputs. LongBench (Bai et al., 2024), RULER (Hsieh et al., 2024), and ∞-Bench (Zhang et al., 2024) established early baselines and pushed evaluation beyond 100K tokens, with RULER showing effective context windows fall well short of advertised limits. NoCha (Karpinska et al., 2024) introduced reasoning-intensive true/false claims (GPT-4o: 55.8% pair accuracy), and more recent work (LongBench v2 (Bai et al., 2025), HELMET (Yen et al., 2025), LooGLE v2 (He et al., 2025)) extends this to multi-hop settings over 100K+ tokens. These benchmarks evaluate **static** documents; RECON is built around documents that evolve over time, where agents must track which conclusions still hold as facts are revised or invalidated.

Benchmark	Domain	Context	Dyn.	CR-c	CP	SC	CF	TC	TFR
RULER (Hsieh et al., 2024)	Synthetic	4K–128K	×	×	×	×	×	×	×
LongBench v2 (Bai et al., 2025)	Multi-domain	8K–2M	○	✓	×	×	×	×	✓
LooGLE v2 (He et al., 2025)	Multi-domain	16K–2M	×	✓	×	×	×	✓	✓
Loong (Wang et al., 2024)	Fin./Legal/Acad.	10K–200K+	×	✓	×	×	×	×	✓
LoCoMo (Maharana et al., 2024)	Personal conv.	∼10K	○	✓	×	×	×	✓	✓
LongMemEval (Wu et al., 2025)	Personal conv.	115K–1.5M	✓	✓	×	✓	×	✓	✓
MemoryAgentBench (Hu et al., 2025)	Conversational	100K–1.4M	○	✓	×	✓	×	✓	✓
TRACK (Feng et al., 2026)	Wiki/Code/Math	paragraph	✓	✓	×	✓	×	×	×
MuSR (Sprague et al., 2024)	Synth. narratives	∼1K words	×	✓	×	×	×	×	×
DetectiveQA (Xu et al., 2025)	Detective novels	5K–363K	×	✓	×	×	×	×	×
NovelHopQA (Gupta et al., 2025)	Public novels	64K–128K	×	✓	×	×	×	×	×
RECON (ours)	**Multi-domain**	**50K–100K**	✓	✓	✓	✓	✓	✓	✓

Table 1: Comparison of RECON with prior benchmarks. ✓/○/× = tested / partial / not tested. **Dyn.** = documents contain explicit fact-updates within a single case. Task columns: CR-c (Chain Reconstruction), CP (Cascade Propagation), SC (Source Conflict), CF (Counterfactual), TC (Temporal Constraint), TFR (Temporal Fact Retrieval).

2.2 Memory Benchmarks

A second line of work evaluates memory over accumulated histories. LoCoMo (Maharana et al., 2024) provides 300-turn chat histories with multi-hop, temporal, and adversarial questions. PerLTQA (Du et al., 2024) scales to 8,593 questions across 3,409 dialogues over personal memory. LongMemEval (Wu et al., 2025) targets temporal state tracking and finds that state-of-the-art systems consistently fail at knowledge updates across sessions. MemoryAgentBench (Hu et al., 2025) frames memory as four competencies (accurate retrieval, test-time learning, long-range understanding, and selective forgetting). TRACK (Feng et al., 2026) extends this line by injecting conflicting fact-updates into multi-step reasoning chains, though at paragraph scale and along a single chain rather than over the bipartite dependency structures RECON evaluates.

Across these benchmarks, memory is modeled as a *state machine* whose facts have current values to be tracked. RECON instead models it as a *provenance graph* in which conclusions have derivation histories: when evidence is invalidated, the agent must determine which conclusions lose their foundation and which remain supported by independent evidence.

2.3 Investigative and Narrative Reasoning

A third line of work tests multi-hop reasoning over narrative documents, building on the multi-hop QA tradition (Yang et al., 2018; Ho et al., 2020; Trivedi et al., 2022). MuSR (Sprague et al., 2024) introduces ∼1,000-word synthetic narratives that probe chain-of-thought reasoning but contain no fact-updates. DetectiveQA (Xu et al., 2025) and NovelHopQA (Gupta et al., 2025) use real novels (∼100K and 64K–128K tokens) with multi-hop questions, but as published texts contain no controlled invalidation, conflict, or counterfactual structure. TurnaboutLLM (Yuan et al., 2025) draws contradictions from game transcripts, overlapping with source conflict but restricted to a single task.

3 The RECON Benchmark

3.1 Generation Pipeline

Each case embeds evidence chains, formal invalidations, source conflicts, counterfactual dependencies, and parallel temporal streams, generated deterministically and rendered into natural language; systems see only the narrated case file. Figure 3 illustrates the deterministic ground-truth construction before narration.

LLMs perform only surface realization, and all outputs are postvalidated before release.

Attribute-conditioned blueprint synthesis. Each case originates from a deterministic, seeded production engine resembling a grammar: a finite catalog of typed terminals (actors, locations, evidence templates) is expanded by ordered production rules into a complete case blueprint. We

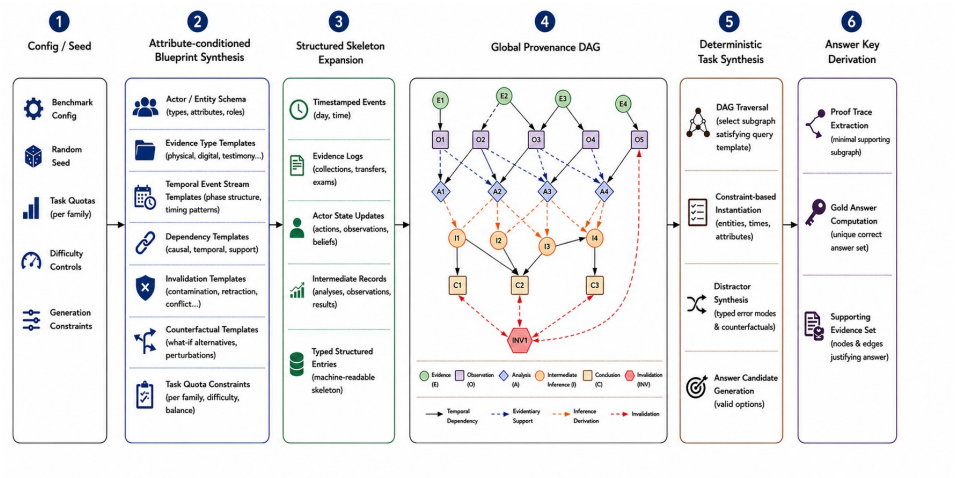

Figure 3: End-to-end RECON generation for a single evidence thread: ground truth is fully established deterministically before LLM narration.

formalize the generator as $G = (D, A, P, C)$, with domain catalog D, typed *attributes* A representing generation state, production rules P, and hard global invariants C (temporal monotonicity, culprit uniqueness, quota satisfaction). Each rule $p \in P$ carries a *requires* set $R_p \subseteq A$ and an *establishes* set $E_p \subseteq A$; it fires only when $R_p \subseteq S_t$ and then sets $S_{t+1} = S_t \cup E_p$, so every discovery is preceded by its prerequisites and the provenance DAG is read directly off the generation trace rather than recovered post hoc. Synthesis is deterministic given seed s, domain d, and quota q ($B = G(s, d, q)$).

Skeleton expansion and temporal enforcement. The blueprint expands into a structured skeleton of timestamped events, evidence items, causal dependencies, invalidations, source conflicts, counterfactual dependencies, and parallel temporal streams (surveillance, access, transaction, clinical, or audit). Temporal consistency is enforced by a **lexicographic monotonic time constraint**: every node carries $\tau(v) = (\text{day}, \text{time})$, and for every causal edge $u \to v$, $\tau(v) \geq_{\text{lex}} \tau(u)$. A post-skeleton validator verifies these constraints before narration.

Provenance DAG and proof-trace grounding. The skeleton induces a global provenance DAG whose nodes are events, evidence items, and conclusions and whose typed edges encode causal, revisionary, invalidating, and conflict-resolving relationships. An edge $u \to v$ exists precisely when an attribute established by entry u is required by entry v, so the graph is read directly off the generation contract rather than inferred. This graph is the authoritative case representation; narrated prose is downstream of it, not the reverse. Every question carries a **proof trace**: a minimal spanning subgraph of the DAG whose nodes are necessary and sufficient to derive the answer, grounding answer derivation without LLM inference and anchoring the validator's grounding check. Answerability is therefore guaranteed by construction: every gold answer is derivable from this deterministic structure, and a post-question validator enforces uniqueness before release.

Deterministic task synthesis. Questions are generated algorithmically from the provenance DAG: chain reconstruction items sample multi-hop paths; cascade items select an invalidation node and compute via reachability which conclusions collapse and which survive; conflict items locate the independent corroborating node; counterfactual items shift a timestamp and recompute reachability; temporal constraint items query stream intersections; temporal fact retrieval probes single-source chronological facts. A fixed question matrix (Table 4) fixes per-task and per-format quotas. Distractors follow a predefined reasoning-failure taxonomy (Appendix F).

Linguistic realization. After deterministic construction, an LLM narrator converts skeleton entries into investigation-style prose under strict fact-fidelity constraints: it may choose wording, tone, and sensory detail, but skeleton fields (timestamp, source type, entities, evidence identifiers, factual content) are treated as immutable. Post-narration validation cross-checks timestamps, entity identifiers,

and evidence IDs against the skeleton; failures are re-narrated before release. We test directly for narrator-family bias in Appendix D.12 and find none.

3.2 Task Categories

The six task categories (§1) share a fixed taxonomy across domains, while entities, evidence types, and temporal streams vary by case. Each is designed to defeat a specific shortcut: adjacent Chain Reconstruction hops are lexically dissimilar and separated by many intervening entries; Cascade Propagation requires separating conclusions that collapse after a formal invalidation (contaminated sample, retracted statement, corrected audit record, revised clinical finding) from those that survive through independent support; Source Conflict resolving evidence is unlabeled and often distant in the narrative; Counterfactual questions require keeping actual and hypothetical timelines distinct; and Temporal Constraint questions require intersecting parallel streams (surveillance, access, transactions, communications, sensor, clinical, audit). Temporal Fact Retrieval is a lower-composition control covering event ordering, point-in-time state, stream matching, and elimination.

3.3 Benchmark Validation Pipeline

RECON uses an iterative audit–repair–revalidation pipeline with six stages: (1) regex/schema checks for malformed structures and identifier leakage; (2) structural validation (DAG acyclicity, temporal monotonicity, referential integrity); (3) provenance-grounding verification; (4) a diagnostic LLM-assisted semantic audit that does not define ground truth; (5) robustness audits under anonymization, timestamp normalization, and DAG-safe perturbations; and (6) human validation. All 1,604 released items pass structural and temporal-consistency validation (0 cyclic dependencies, 100% temporal monotonicity; mean provenance depth 9 hops for crime, 6 for medical and finance). Stress tests detected 100% of injected timestamp inversions, DAG corruptions, and identifier leaks, and answers remained stable under all three perturbations, reducing the risk that performance reflects lexical cues rather than provenance reasoning.

Human validation. We validated a stratified sample of 200 questions spanning all domains and task families, each with three independent annotations from a pool of 23 undergraduate engineering students, who received a question-specific evidence packet (answer-relevant provenance subgraph plus matching narrative excerpts) and were blind to the gold answer and to each other. The majority judgment matched the benchmark answer on 173/200 items (86.5%; Fleiss' $\kappa = 0.69$); 179, 184, and 180 of 200 were rated clear, evidence-sufficient, and narratively faithful; and 8/200 (4.0%) were flagged ambiguous by at least two annotators, concentrated in counterfactual and cascade tasks. The full protocol, a clean-subset reevaluation, and LLM-judge validation against human labels appear in Appendices C and D.

4 Evaluation and Results

4.1 Experimental Setup

We evaluate three system families together with a *retrieval-perfect control condition*. The long-context family places the full case file ($\sim$100K tokens) in context, evaluated across eight closed-source and open-weight models. The retrieval-perfect condition (*Oracle* in the tables) replaces the narrated case file with the structured ground-truth representation it was generated from. It is a control, not an attainable system: removing the narrative makes residual error attributable to reasoning over structured evidence, so its absolute level measures how much of RECON lies beyond retrieval improvements alone. The RAG family covers four retrieval variants of increasing sophistication, ranging from plain dense retrieval to hybrid retrieval (dense and BM25) combined with reranking and multi-query expansion. The memory family covers Mem0 (Chhikara et al., 2025), Mem0-Graph, Supermemory, and Hindsight (Latimer et al., 2025).

Scoring. Every question format uses an explicit scoring rule with an abstain option presented in the prompt. MCQ-Single items use a -0.2 penalty for wrong commits, MCQ-Multiple uses the per-option weighted score $c/k - w/(n-k)$, Integer items use exact match, Ordering items are scored by Kendall's τ, and free-form responses are graded by two independent LLM judges from different

Family	System	Per-Task Score						Overall			
		Casc.	Chain	CF	SrcConf	TConstr	TFact	Score	Acc.%	Abst.%	Tok./Q
Long-context	`gpt-5.1`	0.297	0.249	0.210	0.319	0.224	0.427	**0.287**	20.3%	13.6%	64,649
	`gpt-4o-mini`	**0.361**	**0.295**	0.190	**0.343**	0.043	0.161	0.232	11.2%	4.7%	64,632
	`gpt-4.1-mini`	0.330	0.284	0.227	0.241	0.019	0.129	0.206	10.7%	3.8%	64,635
	`gemini-2.5-pro`	0.224	0.188	0.273	0.267	0.185	0.454	0.265	**22.4%**	37.6%	70,097
	`gemini-2.5-flash`	0.285	0.117	**0.305**	0.156	**0.237**	**0.490**	0.256	15.3%	11.7%	68,022
	`kimi-k2.5`	0.300	0.201	0.212	0.248	0.148	0.455	0.257	13.8%	8.7%	65,212
	`qwen3-vl-30b`	0.315	0.232	0.232	0.269	0.142	0.208	0.230	14.6%	12.8%	70,064
	`llama-3.3-70b`	0.330	0.214	0.140	0.224	0.077	0.192	0.192	9.8%	11.4%	66,456
Oracle	`gpt-5.1`	0.923	0.792	0.474	0.935	**0.384**	0.329	0.638	53.7%	2.8%	78,561
	`gemini-2.5-flash`	**0.939**	**0.875**	**0.483**	**0.961**	0.290	**0.348**	**0.654**	**54.6%**	6.8%	94,648
RAG	plain	0.285	0.081	0.196	0.048	**0.177**	0.353	0.178	10.5%	48.6%	7,538
	+rerank	0.276	0.073	**0.198**	**0.119**	0.150	0.403	**0.192**	**14.5%**	47.7%	7,540
	hybrid + rerank	**0.308**	0.075	0.157	0.096	0.064	**0.443**	0.179	11.8%	44.5%	9,037
	hybrid + rerank + MQ	0.261	**0.101**	0.174	0.094	0.143	0.315	0.173	11.5%	46.5%	8,253
Memory	Mem0	0.525	0.091	−0.013	0.005	0.124	0.101	0.112	9.7%	41.2%	3,249
	Mem0-Graph	0.572	0.048	0.071	0.096	**0.158**	0.131	0.149	10.8%	45.3%	4,056
	Supermemory	**0.708**	0.162	**0.207**	**0.154**	0.144	**0.181**	**0.211**	**15.1%**	47.2%	6,440
	Hindsight	0.490	**0.197**	−0.007	0.012	0.136	0.175	0.148	10.2%	45.6%	8,438

Table 2: Main results on 1,414 clean questions. *Oracle* denotes the retrieval-perfect control condition (§4.1), which receives the ground-truth dependency graph in place of the narrative and performs no narrative retrieval. It approximates what is achievable by improving retrieval alone and is not an attainable system. Closed-book floor results appear in Table 7 (Appendix D.4).

Task	Oracle (best)	Best LLM	Gap (pp)
Source Conflict	88.9%	27.8%	+**61.1**
Cascade Prop.	81.8%	18.2%	+**63.6**
Chain Recon.	77.3%	18.2%	+**59.1**
Counterfactual	41.2%	23.5%	+17.6
Temp. Constraint	33.3%	20.0%	+13.3
Temp. Fact Retr.	23.5%	35.3%	−11.8

Table 3: Accuracy gap between the retrieval-perfect control and the best system without it, by task category. A large gap indicates a retrieval-dominated bottleneck; a small gap indicates that reasoning limits performance even under perfect retrieval. Per-domain Score per system in Table 11.

model families (`gpt-4o` and `gemini-2.5-flash`); the free-form score is the mean of their two binary verdicts. Exact formulas appear in Appendix D.

Metrics. Our primary metric is Score, the mean of the per-question scoring rule across all evaluated questions. We also report Accuracy, defined as strict correctness with no partial credit or negative marking, which separates raw correctness from calibration. Abstain% is the fraction of questions for which the system returned no answer, and Tokens/Q is the mean prompt and completion tokens per question.

Contamination filter. Each of the 1,604 questions is first attempted closed-book by three LLMs (`gpt-5.1`, `gpt-4.1-mini`, and `gemini-2.5-flash`), and a question is flagged as priors-guessable when at least two of the three answer it correctly. This procedure flagged 190 questions (11.8%), leaving 1,414 questions for the main evaluation.

Reporting. Long-context and Oracle results are reported per answering LLM; RAG and memory rows report the unweighted mean across `gpt-5.1` and `gemini-2.5-flash` (per-LLM breakdowns and configurations in Appx D).

4.2 Aggregate performance

Table 2 reports aggregate performance for every system. No system without access to the dependency graph exceeds 25% Accuracy: the best Score is 0.287 (GPT-5.1) and the best Accuracy 22.4% (Gemini-2.5-Pro), well above the 5.7% closed-book floor (Table 7, §D.4). The retrieval-perfect

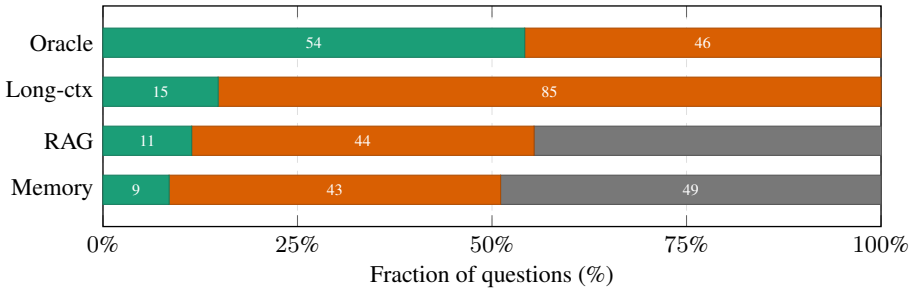

Figure 4: Per-family decomposition of questions into retrieval-and-reasoning outcome cells, using full-coverage entry-ID recall. **Green** = retrieval hit and answer correct; **orange** = retrieval hit but wrong/abstain; **gray** = retrieval miss.

control reaches 0.654 and 54.6%, answering fewer than 55% of questions correctly despite receiving the structured dependency graph in place of the narrated case file.

These aggregates show RECON is unsaturated at every level of the stack; we next attribute the shortfall by separating retrieval from reasoning (§4.3) and isolating compositional depth (§4.4), then compare architectures (§4.5).

4.3 Decomposing failure: retrieval versus reasoning

We decompose system failures along two complementary diagnostic axes. The first compares per-task Accuracy against the retrieval-perfect control, which receives the ground-truth dependency graph and therefore removes narrative retrieval; the second conditions Accuracy on whether each individual question received its complete gold evidence in the retrieved context. Both are available only because the generating dependency structure of every question is known.

The control–LLM Accuracy gap in Table 3 reveals three regimes. Source Conflict, Cascade Propagation, and Chain Reconstruction show large gaps ($+59$ to $+64$ pp), indicating that retrieval is the dominant bottleneck. Counterfactual and Temporal Constraint show much smaller gaps ($+13$ to $+18$ pp) because the control itself reaches only 41.2% and 33.3%: reasoning limits performance even under perfect retrieval. Temporal Fact Retrieval produces a *negative* -12 pp gap because the structured representation abstracts away surface narrative detail this task probes.

Following the supporting-fact recall standard in multi-hop QA (Yang et al., 2018; Trivedi et al., 2022), we count a question as a retrieval hit only if the composite header `[Day X, HH:MM] [TYPE] [AUTHOR]` of every gold supporting entry appears verbatim in the retrieved context, since evidence is conjunctive for multi-hop tasks; this header raises gold-entry anchor uniqueness from 80% to 93% (98% for multi-hop families).[1] RAG attains 20.6% Accuracy on hits (n=782) versus 1.3% on misses (n=632), and memory shows the same direction (16.7% vs. 4.3%, n=722/692). Retrieval is necessary but insufficient: roughly four in five full-coverage questions are still answered incorrectly, and RAG abstains on 47.7% of Source Conflict questions despite holding the conflicting evidence. Long-context holds every entry yet reaches only 14.8% Accuracy (family mean) against 54.2% for the control, a 39.4 pp gap that reflects the cost of locating evidence in narrative form; the control's own error shows reasoning limits persist beyond it.

4.4 Decomposing failure: compositional depth

To identify which property of the reasoning problem is responsible, we vary compositional depth with the task held fixed. The analysis is restricted to Chain Reconstruction, whose questions carry an explicit per-question complexity scalar (`chain_length`, range 3–10) inherited from the generating subgraph, so that depth is manipulated rather than merely observed. Figure 1 reports Accuracy by chain length per family.

[1] We exclude Mem0 and Mem0-Graph, whose ingestion paraphrases evidence into atomic facts that drop these headers; their overall Score remains in Table 2.

Accuracy under the retrieval-perfect control falls from 100% at chain length 3–4 to 50% at chain length 10, though not monotonically at intermediate lengths: even given the full dependency graph, step-by-step composition degrades with depth. The remaining families stay near zero across all chain lengths (long-context 0–16%, RAG and memory 0–9%), failing at the retrieval or attention layer before chain length becomes the operative bottleneck. Because the control falls toward the other families as depth grows, its residual error is itself a function of compositional depth.

4.5 Architectural trade-offs

Memory systems excel on Cascade Propagation. Supermemory's Cascade Score of 0.708 nearly doubles the best long-context score (0.361) and exceeds every RAG variant by ~ 0.4, while the other three memory systems reach 0.49–0.57 on this task.

Counterfactual reasoning is the hard ceiling. The best Counterfactual Score without the dependency graph is 0.305, and the control reaches only 0.483 despite receiving the dependency graph, indicating chained inference rather than retrieval is the bottleneck; Temporal Constraint shows the same pattern (control 0.384, best LLM 0.237).

Different architectures win different tasks at very different token cost. Long-context leads on Chain Reconstruction, Source Conflict, and Counterfactual, while Temporal Constraint is effectively tied across families (means 0.13–0.14); RAG leads on Temporal Fact Retrieval (mean 0.378 vs. 0.314, best variant 0.443); memory leads on Cascade (Supermemory, 0.708). No representation handles all six tasks, despite RAG (7–9K tokens/question) and memory (3–8K) using 7–22$\times$ fewer tokens than long-context (64–70K); compression specifically hurts Chain Reconstruction and Source Conflict by discarding inter-fact edges.

4.6 Human baseline

We collect a human performance baseline on a 60-question stratified sample (10 per task category, balanced across domains). Three annotators per question received a question-specific evidence packet (the answer-relevant provenance subgraph and matching narrative excerpts) but no oracle aids; we report the majority vote. Annotators reach 63.3% Accuracy (38/60 questions; Score 0.53, abstention rate $7/60 = 11.7\%$, mean 8.5 minutes per question), exceeding the retrieval-perfect control by 8.7 pp, though on a smaller sample with a different input format. Per-task and per-domain breakdowns and the full annotation protocol are in Appendix D.11.

5 Conclusion

We introduced RECON, a benchmark that models agent memory as a provenance graph rather than a state machine: conclusions carry derivation histories and invalidations propagate through explicit dependencies. Its $1{,}604$ questions over 24 case files (50K–100K tokens, three domains) are built by a deterministic generator that keeps LLMs out of the gold reasoning trace, so RECON attributes failure rather than only recording it. The attribution is consistent across long-context, retrieval, and agentic memory architectures: a retrieval-perfect control reaches only 54.6% Accuracy against 22.4% for the best system without it, roughly four in five failures persist when gold evidence is verifiably retrieved, and control Accuracy falls from 100% to 50% as depth grows from 3–4 to 10 hops. Retrieval is necessary but not sufficient: it dominates the structured tasks, while counterfactual, temporal constraint, and deep chain questions stay hard with the graph in hand. We release the benchmark, evaluation harness, and generator; limitations are discussed in Appendix E.

Ethics Statement

RECON's case files are constructed programmatically by a seeded deterministic generator (§3.1) and rendered as realistic investigative narratives. Narration is constrained to immutable skeleton fields under fixed register templates (Appendix A), so the released text contains no graphic content.

The human validation and human baseline studies (Appendix C, Appendix D.11) involved annotators recruited from the academic community, comprising undergraduate engineering students. Participation was voluntary and unpaid. Annotators received standardized written guidelines, completed

a calibration phase, and judged only generated case material containing no personal or sensitive information.

Reproducibility Statement

The generation pipeline is deterministic under a fixed seed (§3.1, Appendix B), so the entire benchmark reproduces from its seed configuration. All solver calls run at `temperature=0` against snapshot-pinned models, with per-format structured-output schemas and a fixed retry budget (Appendix D). Per-format scoring rules are given in Appendix D.1, the closed-book contamination filter in Appendix D.4, and per-architecture retrieval and memory configurations in Appendix D. The dataset, generator, and evaluation harness are released at `https://anonymous.4open.science/r/RECON-Bench`.

Use of Large Language Models

We used generative AI tools to generate the synthetic case narratives that constitute the benchmark corpus, and to implement parts of the generation and evaluation harness. LLMs also serve as objects of study throughout this work, as solvers and as free-form judges. Their role in dataset construction is strictly bounded: an LLM narrator renders immutable skeleton fields into prose and edits distractors and question wording, and never produces case structure, provenance, dependencies, or answer keys, all of which are derived in code (§3.1, Appendix A). We have not used generative AI tools to develop theoretical models, formulate or prove mathematical claims, propose or refine hypotheses, or design the experimental methodology.

Additionally, we used generative AI tools to edit software code and to improve the readability of the manuscript. All LLM-generated code was reviewed and tested by the authors, and all experimental results, statistics, and claims were produced and verified by the authors. We take responsibility for the final content of this work, including all text, claims, and released artifacts.

References

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137, Bangkok, Thailand, August 2024. Association for Computational Linguistics. URL `https://aclanthology.org/2024.acl-long.172/`.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3639–3664, Vienna, Austria, July 2025. Association for Computational Linguistics. URL `https://aclanthology.org/2025.acl-long.183/`.

Ding Chen, Simin Niu, Kehang Li, Peng Liu, Xiangping Zheng, Bo Tang, Xinchi Li, Feiyu Xiong, and Zhiyu Li. HaluMem: Evaluating hallucinations in memory systems of agents. *arXiv preprint arXiv:2511.03506*, 2025.

P. Chhikara, D. Khant, S. Aryan, T. Singh, and D. Yadav. Mem0: Building production-ready AI agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025.

Yangruibo Ding, Zijian Wang, Wasi Uddin Ahmad, Hantian Ding, Ming Tan, Nihal Jain, Murali Krishna Ramanathan, Ramesh Nallapati, Parminder Bhatia, Dan Roth, and Bing Xiang. CrossCodeEval: A diverse and multilingual benchmark for cross-file code completion. In *Advances in Neural Information Processing Systems*, 2023.

Yiming Du, Hongru Wang, Zhengyi Zhao, Bin Liang, Baojun Wang, Wanjun Zhong, Zezhong Wang, and Kam-Fai Wong. PerLTQA: A personal long-term memory dataset for memory classification, retrieval, and fusion in question answering. In *Proceedings of the 10th SIGHAN Workshop on*

Chinese Language Processing (SIGHAN-10). Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.sighan-1.18/.

Yuchen Fan, Chen Ling, Xin Zhong, Shuo Zhang, Heng Zhou, Yuchen Zhang, Mingyu Liang, Chengxing Xie, Ermo Hua, Zhizhou He, Cheng Huang, Gang Chen, Ning Ding, and Bowen Zhou. LFQA-E: Carefully benchmarking long-form QA evaluation. In *International Conference on Learning Representations (ICLR)*, 2026.

Yiyang Feng, Zeming Chen, Haotian Wu, Jiawei Zhou, and Antoine Bosselut. Tracking the limits of knowledge propagation: How LLMs fail at multi-step reasoning with conflicting knowledge. In *Proceedings of the 2026 Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2026. arXiv:2601.15495.

Scott L. Fleming, Alejandro Lozano, William J. Haberkorn, Jenelle A. Jindal, Eduardo P. Reis, Rahul Thapa, Louis Blankemeier, Julian Z. Genkins, Ethan Steinberg, Ashwin Nayak, Birju S. Patel, Chia-Chun Chiang, Alison Callahan, Zepeng Huo, Sergios Gatidis, Scott J. Adams, Oluseyi Fayanju, Shreya J. Shah, Thomas Savage, Ethan Goh, Akshay S. Chaudhari, Nima Aghaeepour, Christopher Sharp, Michael A. Pfeffer, Percy Liang, Jonathan H. Chen, Keith E. Morse, Emma P. Brunskill, Jason A. Fries, and Nigam H. Shah. MedAlign: A clinician-generated dataset for instruction following with electronic medical records. *arXiv preprint arXiv:2308.14089*, 2023.

Abhay Gupta, Kevin Zhu, Vasu Sharma, Sean O'Brien, and Michael Lu. NovelHopQA: Diagnosing multi-hop reasoning failures in long narrative contexts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025. arXiv:2506.02000.

Ziyuan He, Yuxuan Wang, Jiaqi Li, Kexin Liang, and Muhan Zhang. LooGLE v2: Are LLMs ready for real world long dependency challenges? In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2025.

X. Ho, A.-K. Duong Nguyen, S. Sugawara, and A. Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, 2020. arXiv:2011.01060.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024.

Y. Hu, Y. Wang, and J. McAuley. Evaluating memory in LLM agents via incremental multi-turn interactions. In *2nd Workshop on Long-Context Foundation Models*, 2025.

Marzena Karpinska, Katherine Thai, Kyle Lo, Tanya Goyal, and Mohit Iyyer. One thousand and one pairs: A "novel" challenge for long-context language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Miami, Florida, USA, November 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.emnlp-main.948/.

Chris Latimer, Nicoló Boschi, Andrew Neeser, Chris Bartholomew, Gaurav Srivastava, Xuan Wang, and Naren Ramakrishnan. Hindsight is 20/20: Building agent memory that retains, recalls, and reflects. *arXiv preprint arXiv:2512.12818*, 2025.

Tianyang Liu, Canwen Xu, and Julian McAuley. RepoBench: Benchmarking repository-level code auto-completion systems. In *The Twelfth International Conference on Learning Representations*, 2024.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, Bangkok, Thailand, 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.acl-long.747/.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems, 2023. arXiv:2310.08560.

P. Rasmussen, P. Paliychuk, T. Beauvais, J. Ryan, and D. Chalef. Zep: A temporal knowledge graph architecture for agent memory. *arXiv preprint arXiv:2501.13956*, 2025.

Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. MuSR: Testing the limits of chain-of-thought with multistep soft reasoning. In *International Conference on Learning Representations (ICLR)*, 2024. Spotlight.

H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics (TACL)*, 2022.

Minzheng Wang, Longze Chen, Cheng Fu, Shengyi Liao, Xinghua Zhang, Bingli Wu, Haiyang Yu, Nan Xu, Lei Zhang, Run Luo, Yunshui Li, Min Yang, Fei Huang, and Yongbin Li. Leave no document behind: Benchmarking long-context LLMs with extended multi-doc QA. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024. Oral; arXiv:2406.17419.

D. Wu, H. Wang, W. Yu, Y. Zhang, K.-W. Chang, and D. Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations (ICLR)*, 2025.

Zhe Xu, Jiasheng Ye, Xiaoran Liu, Xiangyang Liu, Tianxiang Sun, Zhigeng Liu, Qipeng Guo, Linlin Li, Qun Liu, Xuanjing Huang, and Xipeng Qiu. DetectiveQA: Evaluating long-context reasoning on detective novels. *arXiv preprint arXiv:2409.02465*, 2025.

Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. HELMET: How to evaluate long-context language models effectively and thoroughly. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.02694.

Yuan Yuan, Muyu He, Muhammad Adil Shahid, Ziyang Li, Jiani Huang, and Li Zhang. TurnaboutLLM: A deductive reasoning benchmark from detective games. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025. arXiv:2505.15712.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Khai Hao, Xu Han, Zhen Leng Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. ∞Bench: Extending long context evaluation beyond 100K tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15262–15277, 2024.

A LLM PROMPT TEMPLATES USED IN RECON

The LLM is confined to surface realization (narration, distractor editing, question rewriting) and to evaluation (answering and judging); it never produces case structure, provenance, or answer keys (§3.1). All generation calls run at `temperature` $= 0$ and decode under a strict JSON schema so that no text can appear outside the intended field. In the templates below, field names are shown in `typewriter` and braces denote interpolated values.

A.1 NARRATIVE GENERATION PROMPT

System Prompt

You are a professional writer producing one entry for a {domain} case file. Convert the structured FACTS provided into realistic prose, in the register of a {register}, written in the past tense. Follow these rules:

- Write only the narrative body of the entry. Do not include planning, drafts, headers, or metadata labels such as ENTRY TYPE, AUTHOR, or DATE; the surrounding case-file template adds these automatically.
- Use only the information in the FACTS block. Do not introduce specific details that are not given, including the names of venues or businesses, exact times, monetary amounts, identifiers, quoted speech, telephone numbers, or addresses. Where a specific detail is not provided, describe it generically (for example, "the claimed location" or "earlier that evening").
- Whenever the FACTS block names a person, refer to that person by their full name. Do not anonymize a named individual as "an associated party" or "the subject". If the FACTS block itself uses a generic noun, retain it.

The {domain}/{register} pair is one of: criminal investigation / police investigation report; medical adverse-event investigation / clinical incident report (objective, passive voice, describing systemic events rather than assigning individual blame); financial fraud investigation / regulatory-enforcement or forensic-accounting narrative (precise about dates, amounts, and filing references).

User Prompt

```
ENTRY TYPE: {entry type}
AUTHOR / SOURCE: {author}
DATE: Day {day}, {time}
REGISTER: {tone}

STYLE GUIDE FOR THIS ENTRY TYPE:
{per-type style guide}

FACTS (every item must appear; add nothing;
do not repeat these field names):
{structured data as a bullet list}

Write the entry now, as prose only.
```

The style guide is drawn from a fixed per-entry-type library (e.g. witness statement, forensic lab report, financial record); {tone} is sampled from a seeded register list, with automated entry types fixed to neutral. Output is constrained to a JSON object with a single `entry` field at `temperature` $= 0$.

A.2 DISTRACTOR ENHANCEMENT PROMPT

System Prompt

You are a question designer for the RECON investigative-reasoning benchmark. Given a JSON description of a {domain} case and a set of multiple-choice questions, improve the incorrect options (distractors) so that each question can be answered only by consulting the case, not from surface cues. Rules:

- Leave the correct option unchanged.
- Replace generic or out-of-domain distractors with plausible but incorrect alternatives drawn from the case and appropriate to the question type (e.g. for chain reconstruction, incorrect steps from the same or a similar chain; for source conflict, incorrect resolutions of that specific conflict).

- Surface uniformity: all options share the same length range, register, specificity, and structure, so the correct option is not identifiable by form alone.
- Stem independence: a distractor must not be refutable from the question stem in isolation.
- Case vocabulary: phrase options in the language of the narrated case file; do not use pipeline scaffolding terms such as "trigger event" or "chain step".
- Plausibility: each distractor must be credible to a careful reader who has not seen the case.
- Within-batch variation: vary distractors and phrasings across questions that share a template.

Return JSON only: an `enhanced` array in which each element gives the question `id`, the full `options` list (correct text unchanged), and a `rationale` per wrong option (label, reason, and one failure mode from the fixed taxonomy).

User Prompt

```
CASE CONTEXT:
{case as JSON: entities, causal chains,
findings, conflicts, counterfactuals,
temporal streams}

QUESTIONS TO ENHANCE:
{array of MCQ questions}
```

A.3 Question Rewriting Prompt

System Prompt

You are a question editor for the RECON benchmark. Rewrite generated questions and options so they read naturally to any reader, without changing their meaning.

- Preserve which option is correct, the number and labels of options, and the facts in the reasoning. Do not change the answer.
- Remove internal identifiers (e.g. `log_001`, `EV-01`) and pipeline terms, replacing them with natural language ("data stream" becomes "investigative record"; "evidence chain" becomes "evidence trail"). Render entry descriptions in plain English ("the witness statement by P regarding S" becomes "P's witness statement about S").
- Maintain anti-leakage properties: all options share one surface format, length range, and level of detail; no distractor is refutable from the stem alone; option vocabulary matches the case file; phrasings vary across questions that share a template.
- Every option ends with a period, and no two options are identical.

Return JSON only, giving for each question its `id` and the rewritten question, options, and reasoning; all other fields are preserved downstream.

User Prompt

```
Case context (for humanizing names and terms):
{case summary}

INPUT QUESTIONS:
{array of questions}
```

A.4 Answering Prompt (All Solvers)

Every solver shares one answering template; systems differ only in the descriptor of the supplied context, which is appended to the system message.

System Prompt

You are an expert investigation analyst. You have been given {context descriptor}. Answer the question based only on this information. Be precise and concise.

The {context descriptor} identifies the system:

- **Long-context**: the complete case file.
- **Oracle**: the full structured ground truth for the case, in JSON; every fact needed to answer is present.
- **RAG / Hybrid RAG / Supermemory**: retrieved excerpts from the case file.
- **Mem0 / Hindsight**: memories retrieved from the case file.
- **Mem0-Graph**: retrieved memories and knowledge-graph relations from the case file.
- **Closed-book**: no case file or context is supplied; rely only on prior knowledge and abstain if the answer cannot be determined with confidence.

User Prompt

```
{question text}
  A) {option}
  B) {option}  ...

{format instruction}
```

The format instruction is one of:

- **MCQ (single)**: select the single best answer (A–F) in `answer`; if undetermined, set `abstain` true and `answer` null. Abstaining is the only valid way to signal uncertainty. Scoring: correct $+1$, wrong -0.2, abstain 0.
- **MCQ (multiple)**: select all correct options (A–F) in `answers`; if undetermined, set `abstain` true with an empty list. Each correct selection adds and each wrong selection subtracts; select only confident options.
- **Integer**: give an exact integer in `answer`; if undetermined, set `abstain` true and do not guess a number. Scoring: correct $+1$, otherwise 0.
- **Ordering**: give the option letters in order in `order`; if undetermined, set `abstain` true with an empty list, not a partial or guessed order. A reversed order receives the largest penalty.
- **Freeform**: answer in 2–3 sentences using only the context, in `answer`; if undetermined, set `abstain` true. Scoring: correct $+1$, otherwise 0.

A.5 LLM JUDGE PROMPT

Freeform answers are graded by two independent LLM judges from different model families (`gpt-4o` and `gemini-2.5-flash`); the per-question free-form score is the mean of their two binary verdicts. All other formats are scored deterministically (Appendix D). The prompt template below is used for both judges.

System Prompt

You are a strict grader for a multi-domain investigation-reasoning benchmark. You receive a question, the expected answer (ground truth derived from a deterministic skeleton), and a student answer. Grade the student answer as CORRECT or INCORRECT.

- It is CORRECT if it captures the key factual content of the expected answer; paraphrasing, minor wording differences, and additional correct detail are acceptable.
- It is INCORRECT if it omits a key fact, contradicts the expected answer, contains a factual error, or is empty (including "I don't know" or "cannot determine").

Respond with exactly one word: CORRECT or INCORRECT.

User Prompt

```
QUESTION: {question text}

EXPECTED ANSWER: {ground-truth answer}

STUDENT ANSWER: {model response}
```

B GENERATION PIPELINE: FULL DETAILS

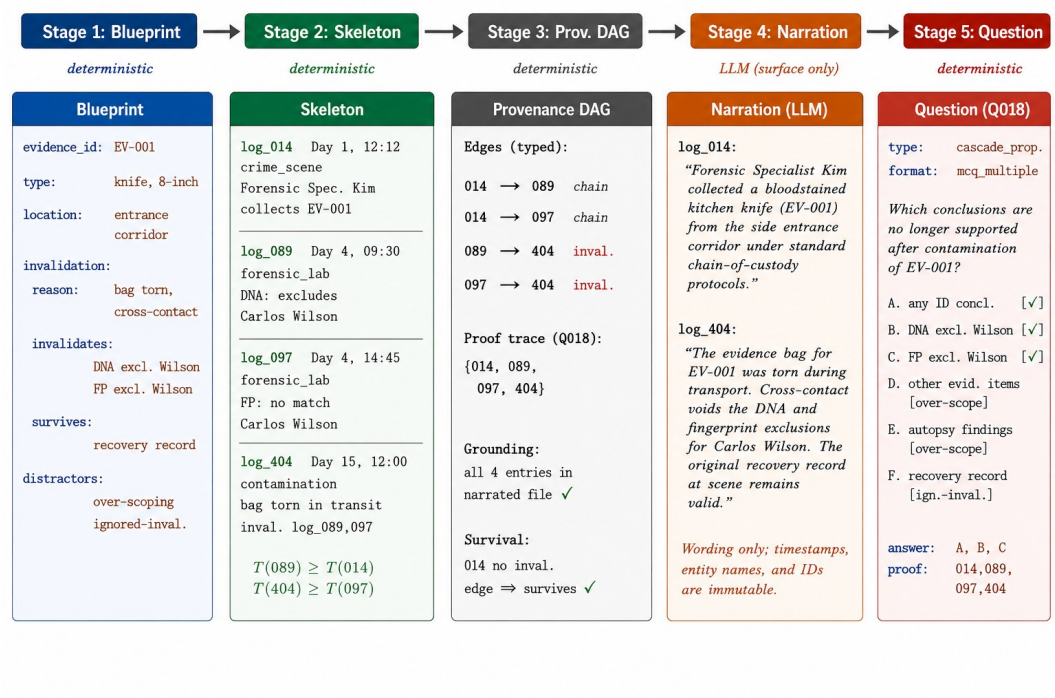

Figure 5: Full deterministic RECON generation pipeline. Case construction proceeds from seeded configuration through attribute-conditioned blueprint synthesis, structured skeleton expansion, provenance DAG induction, deterministic task synthesis, and answer derivation. LLMs are restricted to constrained surface realization and do not participate in authoritative reasoning or ground truth construction.

RECON case files are generated through a deterministic multi-stage pipeline (Figure 5): **Seeded Configuration** → **Attribute-Conditioned Blueprint Synthesis** → **Skeleton Expansion** → **Provenance DAG Construction** → **Task Synthesis** → **Narrative Realization** → **Validation**. Ground truth is established entirely by deterministic program logic. LLMs are used only for constrained linguistic surface realization and optional lexical polishing, never for defining causal structure, provenance, dependencies, or answer keys.

B.1 STAGE 1: ATTRIBUTE-CONDITIONED BLUEPRINT CONSTRUCTION

Each case begins from a deterministic seeded configuration specifying domain, actor cardinalities, evidence quotas, temporal stream types, dependency budgets, invalidation counts, conflict counts, and counterfactual branching constraints.

An attribute-conditioned production engine, structured analogously to a grammar with typed terminals and ordered rewrite rules, expands this configuration into a typed blueprint schema containing entities, events, evidence classes, dependency templates, and task-relevant structural constraints. Each rule fires only when its required attributes are present in the current generation state and establishes new attributes on firing. Because synthesis is deterministic under a fixed seed, identical inputs always reproduce identical case structures.

B.2 STAGE 2: DETERMINISTIC SKELETON CONSTRUCTION

The skeleton and provenance graph jointly define the authoritative ground truth in RECON. They are constructed entirely by deterministic code with no LLM involvement. The skeleton instantiates:

Task	MCQ-S	MCQ-M	Int	Ord	Free	Total
Cascade	72	72	48	48	48	288
Chain	48	48	24	144	72	336
Counterfact.	72	72	48	48	24	264
Conflict	72	72	24	48	48	264
Temp. Const.	56	72	48	16	24	216
Temp. Retr.	72	48	72	20	24	236
Total	**392**	**384**	**264**	**324**	**240**	**1604**

Table 4: Question distribution by task and answer format.

- **Temporal structure**: a domain-specific temporal timeline with precise timestamps for every event, including time-of-death windows, forensic processing delays, and evidence discovery sequences.
- **Evidence chains**: multi-hop causal chains (5–15 hops) linking initial observations to final investigative conclusions, with each hop assigned to a specific day and log entry type.
- **Invalidation events**: formal contamination or retraction events that sever specific links in the dependency graph. Each invalidation stores the affected evidence node. Downstream impact is not hardcoded as a conclusion list; it is computed dynamically by provenance graph traversal, allowing conclusions with independent support to survive invalidation. This explicit dependency structure is what makes cascade propagation and counterfactual questions answerable by construction: the skeleton knows exactly which conclusions collapse when a given piece of evidence is removed, and which survive through independent support.
- **Parallel data streams**: four independent streams (cell tower pings, surveillance footage, transaction receipts, building access logs) that provide overlapping but incomplete temporal coverage. No single stream covers the full time window, so definitive placement or alibi conclusions require intersection across at least two streams.
- **conflicting source assertions**: pairs of irreconcilable witness accounts, each paired with a designated independent source (from a different data stream) that resolves the conflict.
- **Counterfactual dependencies**: explicit provenance dependencies recording which investigative actions (warrants, suspect links, follow-up interviews) depend on which earlier discoveries. Shifting a discovery's timestamp to an earlier or later point in the timeline deterministically changes which downstream actions would or would not have occurred.

All question answers are derived directly from these structures. Because the skeleton is constructed deterministically, every answer is correct by construction and the entire benchmark is reproducible from a single seed.

B.3 Stage 3: Provenance DAG Construction

The expanded skeleton induces a global directed acyclic provenance graph whose nodes represent events, evidence artifacts, assertions, and conclusions. Typed edges encode causal dependence, corroboration, revision, invalidation, and conflict-resolution relationships.

Question ground truth is derived from graph computation over this structure rather than from narrated text. Cascade propagation is computed by invalidating upstream nodes and recomputing downstream reachability. Source conflict resolution identifies contradictory assertions and independent corroborating support. Counterfactual questions modify intervention variables and recompute affected dependency paths.

B.4 Stage 4: Task Synthesis

After the provenance DAG is constructed, the question generator algorithmically traverses it to produce questions for each task category (per-task synthesis logic is described in §3.1). A fixed question matrix enforces per-task, per-format quotas across all cases, eliminating distributional drift between generation runs. Table 4 reports the resulting distribution: 1,604 questions across 24 case files in three domains, spanning six task categories and five answer formats.

B.5 Stage 5: Controlled Narrative Realization

An LLM converts immutable structured entries into natural-language case logs under strict fact-preservation constraints. Input fields include timestamp, source type, participating entities, evidence identifiers, and factual payload.

The narrator may alter wording and discourse structure but may not introduce, remove, or modify factual content. Post-generation validation checks timestamp fidelity, entity preservation, and evidence-reference consistency; failed entries are regenerated.

The narrator is constrained to the facts established in the skeleton: it may choose phrasing, tone, and level of detail, but it may not introduce new facts, alter timestamps, or change relationships between entities. Narration proceeds entry-by-entry, with each entry receiving the skeleton's structured fields (timestamp, source type, actors involved, factual content) and producing a natural language paragraph. The narrator also receives a sliding window of recently narrated entries to maintain stylistic continuity and avoid repetition.

This separation is the key design decision in RECON's pipeline. The skeleton guarantees correctness; the narrator guarantees naturalness. A mechanically generated case file would be trivially gameable by pattern matching, while an LLM-generated ground truth would be unreliable. By assigning each concern to the appropriate tool (deterministic code for truth, LLMs for language), RECON achieves both.

The narration stage produces approximately 800 timestamped log entries spanning 30 days, totaling roughly 100,000 tokens. The resulting case file reads as a coherent investigation narrative, with no visible seams between the deterministic structure underneath and the natural language surface.

B.6 Validation

The pipeline enforces deterministic structural validation at every stage to ensure correctness before proceeding.

Post-skeleton validation checks structural integrity: every evidence chain must have the required number of hops with no dangling references, every invalidation event must point to a valid evidence item with at least one dependent conclusion, every witness conflict must have exactly two conflicting accounts and one resolving source, and every counterfactual dependency must form a directed acyclic graph. Temporal constraints are also verified: no event precedes the investigation start, forensic results do not appear before their corresponding evidence is collected, and invalidation events are temporally consistent with the evidence and dependency structures they modify.

Post-narration validation checks that the narrated case file preserves skeleton facts. Timestamps in the narrated text are extracted and compared against skeleton timestamps. Entity names, evidence item identifiers, and key factual claims are verified to appear in the narrated output. Entries that fail validation are re-narrated with explicit correction instructions.

Post-question validation checks that every generated question has a unique, unambiguous answer derivable from the skeleton. For MCQ questions, it verifies that exactly one option (or the correct subset) matches the ground truth, that all distractor options are distinct, and that the correct answer label is properly assigned. For ordering questions, it verifies that the expected sequence is a valid topological sort of the underlying dependency graph.

C Extended Human Validation Analysis

Annotation protocol. To assess benchmark quality beyond deterministic structural validation, we conducted human evaluation on a stratified sample of 200 questions covering all three domains (crime, finance, medical) and all task families of the RECON benchmark. Each question received three independent blind annotations from a pool of 23 annotators. Two candidate items exhibiting unresolved structural ambiguity were removed during pre-release validation; reported statistics reflect the released benchmark.

The annotator pool comprised undergraduate engineering students recruited from our university. Participation was voluntary and unpaid. All participants received standardized written annotation guidelines and completed a calibration phase on held-out examples disjoint from the released benchmark.

For each question, annotators received a question-specific evidence packet containing the answer-relevant provenance subgraph paired with its matching narrative excerpts. They did not receive the full case file, structured skeleton, or any oracle aid. Annotators evaluated five dimensions: benchmark answer correctness, question clarity, evidence sufficiency, narrative faithfulness, and ambiguity. Benchmark answer correctness was evaluated against the released benchmark answer, while qualitative dimensions were recorded as independent binary judgments. An item was classified as ambiguous when at least two of the three annotators flagged it as ambiguous. For Free-form and Ordering items, an annotator's response was scored as matching the benchmark answer using the same correctness rubric given to the LLM judges (Appendix A).

Agreement and benchmark quality. The majority annotator judgment matched the benchmark answer for 173 of 200 questions (86.5%), contradicted it for 8, and was unsure for the remaining 19. Fleiss' $\kappa = 0.69$, computed over the correctness judgments, indicates substantial inter-annotator agreement. Across qualitative dimensions, 179 of 200 questions were rated clear (89.5%), 184 of 200 were judged to provide sufficient evidence for answer derivation (92.0%), and 180 of 200 were assessed as faithfully aligned with the benchmark's intended narrative and underlying evidence structure (90.0%). 8 of 200 items (4.0%) were flagged as ambiguous under the criterion defined above.

Comparison to human validation practice in related benchmarks. Our protocol is at or above the scale and rigor typical of comparable benchmark papers. NoCha (Karpinska et al., 2024) validated 128 claims; LFQA-E (Fan et al., 2026) employed 10 paid annotators and reports Cohen's $\kappa \approx 0.65$, below the $\kappa = 0.69$ we obtain with unpaid volunteer annotators, while HaluMem (Chen et al., 2025) reports a correctness rate without an accompanying agreement coefficient. Student annotator pools are standard: LongBench v2 (Bai et al., 2025) used a pool spanning CS, law, and economics, and MuSR (Sprague et al., 2024) used three undergraduates, arguing that majority performance exceeding the average individual annotator indicates inattentiveness rather than item defects. RECON's 200 items with three blind annotations each, a calibration phase, and a reported agreement coefficient meets or exceeds each of these on sample size, annotations per item, and statistical reporting.

Disagreement analysis. Observed disagreements were concentrated in higher-composition reasoning settings, particularly those involving invalidation and counterfactual dependency reasoning. The most common modes were: failing to integrate later invalidating evidence over earlier plausible evidence, incorrectly preserving downstream conclusions after upstream invalidation, confusing actual and counterfactual timelines, and incomplete multi-hop evidence integration. For example, in one financial source-conflict case, annotators disagreed on whether a transaction was legitimate based on an early bank transfer record, whereas the benchmark answer required incorporating a later fraud alert that invalidated that path; similar patterns appeared in counterfactual questions where an upstream premise change deterministically altered downstream conclusions.

Relation to deterministic validation. Human validation complements, rather than replaces, RECON's deterministic pre-release validation pipeline, under which every released artifact passes schema, DAG-consistency, temporal-monotonicity, provenance-grounding, and robustness checks; items failing these checks are repaired or regenerated before release. Human disagreement is therefore consistent with residual reasoning difficulty rather than generation pipeline failure, though our deter-

ministic checks validate the skeleton rather than the narration and cannot by themselves distinguish the two.

Clean-subset reevaluation. To bound the effect of the 8 items flagged ambiguous (§C) on the reported results, we recompute Table 2 excluding these items and report the change in Table 5.

Condition	Oracle Score	Best non-Oracle Score	Best non-Oracle Acc.%
Full evaluation set ($n = 1414$)	0.654	0.287	22.4%
Excluding flagged-ambiguous items ($n = 1406$)	0.656	0.288	22.5%

Table 5: Effect of excluding the 8 items flagged ambiguous under the criterion in §C. Results are stable, indicating residual annotator-flagged ambiguity does not drive the reported difficulty; this check covers only the 200-item validated sample and cannot bound ambiguity among the remaining 1,214 items.

D EVALUATION CONFIGURATIONS

This appendix documents the per-architecture configurations referenced in §4.1. Long-context and Oracle place the full structured input directly in the prompt; the RAG and memory architectures share a common pre-processing step that splits the case file into approximately 800 chunks at log-entry boundaries (one chunk per timestamped entry).

D.1 SCORING RULES

The per-format scoring rules referenced in §4.1 are defined as follows.

- **MCQ Single** ($n = 6$ options): correct $= +1$, wrong $= -1/(n-1) = -0.2$, abstain $= 0$. With $n = 6$, random guessing and abstention both yield an expected score of 0, while wrong commits incur a strict penalty.
- **MCQ Multiple** ($n = 6$ options, k correct): score $= c/k - w/(n-k) \in [-1, +1]$, where c and w count correctly- and incorrectly-selected options. Selecting all options yields 0, identical to abstaining.
- **Integer**: exact match; correct $= +1$, wrong $= 0$, abstain $= 0$.
- **Ordering**: the solver receives the gold set of m items and outputs a permutation, scored by Kendall's $\tau = (C - D)/\binom{m}{2} \in [-1, +1]$. Outputs containing wrong items or wrong cardinality are assigned $\tau = 0$.
- **Free-form**: two independent LLM judges (`gpt-4o`, snapshot `gpt-4o-2024-08-06`, and `gemini-2.5-flash`) each emit a binary verdict (correct $= 1$, incorrect $= 0$) over the same response, and the per-question free-form score is the mean of the two: both correct $\rightarrow 1$, split $\rightarrow 0.5$, both wrong $\rightarrow 0$. No negative marking is applied to free-form: unlike MCQ there is no discrete option set against which to calibrate a guessing penalty. An abstention scores 0.

D.2 DETERMINISM AND REPRODUCIBILITY

All solver calls run at `temperature=0`. Every model is snapshot-pinned (e.g., `gpt-4.1-mini-2025-04-14`, `gpt-4o-mini-2024-07-18`) so that silent provider-side updates cannot perturb results. Responses are decoded against per-format structured-output schemas, so every output is either a parsed answer or an explicit abstention; parsing failures are retried up to three times and then logged as abstentions. Cloud memory services (Mem0, Mem0-Graph, Supermemory) ingest each case file exactly once per case, and the resulting memory state is reused across all retrievals for that case. This both controls ingestion-side stochasticity and keeps per-question Tokens/Q comparable across systems.

D.3 STATISTICAL REPORTING AND JUDGE VALIDATION

Free-form responses are graded by two independent LLM judges from different model families: `gpt-4o` (snapshot `gpt-4o-2024-08-06`), following the LongMemEval ICLR 2025 precedent, and `gemini-2.5-flash`. Using two cross-family judges is a control against the documented self-preference bias of single-family judges toward their own family's outputs. Each judge independently scores every free-form response, and the per-question free-form score is the mean of the two binary verdicts (Appendix D.1).

Judge validation against human labels. We sample 100 free-form solver responses and have annotators independently label each CORRECT or INCORRECT using the judge rubric (Appendix A). Table 6 reports agreement between each judge and the human labels, alongside human-human agreement as a reference point. Both judges agree with the human reference labels more often than two individual annotators agree with each other, consistent with LLM judges converging on the reference rather than adding an independent source of error.

D.4 CLOSED-BOOK CONTAMINATION FILTER

To remove questions that can be answered from prior LLM knowledge alone, every one of the 1,604 questions is first attempted closed-book by three solver LLMs spanning two model families:

	Agreement	Cohen's κ	Precision	Recall
Human–human ($n = 100$)	86%	0.72	–	–
`gpt-4o`	91%	0.80	93%	94%
`gemini-2.5-flash`	90%	0.78	92%	93%

Table 6: Judge validation: agreement between each LLM judge and human labels on 100 sampled free-form solver responses, with human-human agreement as a reference point.

`gpt-5.1`, `gpt-4.1-mini-2025-04-14`, and `gemini-2.5-flash`. Each closed-book attempt uses the same answering prompt as the main evaluation (Appendix A, "Answering Prompt") with the context descriptor set to the *Closed-book* variant: no case file or retrieved excerpt is supplied, and the solver is instructed to rely on prior knowledge and to abstain when the answer cannot be determined with confidence. Closed-book calls use the same temperature, retry budget, and structured-output schemas as the main evaluation. A question is flagged as priors-guessable when at least two of the three solvers commit to the correct answer; abstentions and wrong commits both count as non-correct, so the rule selects only items where multiple models actively assert the correct answer without context. This procedure flagged 190 questions (11.8%), leaving 1,414 questions for all evaluations reported in §4 and the remainder of this appendix.

Closed-book floor. Table 7 reports each contamination-filter model scored on the same 1,414-question evaluation set using the closed-book context descriptor (Appendix A). Because this set is defined as the questions these models mostly failed closed-book, the row is a downward-biased floor by construction, not an independent baseline.

System	Casc.	Chain	CF	SrcConf	TConstr	TFact	Score	Acc.%	Abst.%	Tok./Q
`gpt-5.1`	0.072	0.058	0.084	0.061	0.049	0.067	0.065	5.7%	14.6%	862
`gpt-4.1-mini`	0.054	0.043	0.067	0.051	0.038	0.052	0.051	4.1%	20.1%	794
`gemini-2.5-flash`	0.063	0.051	0.079	0.057	0.046	0.074	0.061	5.2%	17.1%	931

Table 7: Closed-book floor: the three contamination-filter models answering with no case file, scored on the 1,414-question evaluation set.

D.5 Long-Context Baselines

The long-context architecture places the full case file (50K–100K tokens) directly in the LLM's prompt alongside the question; no chunking, retrieval, or compression is applied. We evaluate eight snapshot-pinned models: five closed-source models (`gpt-5.1`, `gpt-4o-mini-2024-07-18`, `gpt-4.1-mini-2025-04-14`, `gemini-2.5-pro`, `gemini-2.5-flash`) and three open-weight models (`llama-3.3-70b`, `qwen3-vl-30b`, `kimi-k2.5`). Each question is answered independently with the full case file in context; per-model concurrency is 5.

D.6 Retrieval-Perfect Control (Oracle)

The retrieval-perfect control replaces the narrated case file with the structured ground-truth representation from which it was generated, namely the provenance DAG of entities, timestamped events, evidence items, invalidations, and counterfactual dependencies described in §3.1. This structured representation is serialized into the prompt in place of the narrative text. Oracle is evaluated with two answering models, `gpt-5.1` and `gemini-2.5-flash`, the same models used for the RAG and memory architectures; this isolates the effect of narrative retrieval under matched answering capacity.

D.7 RAG

All four RAG variants share a common backbone. The case file is chunked at log-entry boundaries (one chunk per timestamped entry, approximately 800 chunks per case), embedded with `text-embedding-3-large`, and stored in a local ChromaDB collection keyed by case identifier with cosine similarity. At query time the question text, with MCQ option text appended for richer retrieval

signal, is embedded with the same model. Retrieved chunks are concatenated and placed in the system prompt; the answering LLM sees only the retrieved chunks, not the full case file. Each variant is evaluated with two answering LLMs (`gpt-5.1` and `gemini-2.5-flash`), matching the Oracle setup; per-variant scores in Table 2 are the unweighted mean across the two answering LLMs, with the per-LLM breakdown reported in Table 10. The four variants differ only in the retrieval stage:

- **plain dense**: top-50 dense retrieval, no reranking, no query expansion.
- **+rerank**: top-50 dense candidates are re-scored by the Cohere `rerank-v3.5` cross-encoder; the top-50 after reranking are passed to the LLM.
- **hybrid+rerank**: BM25 and dense each retrieve top-50, the two candidate sets are fused via Reciprocal Rank Fusion, and the fused list is Cohere-reranked to a final top-50.
- **hybrid+rerank+multi-query**: the question is rewritten into two paraphrases by `gpt-4o-mini` (three queries in total); each query retrieves top-30 from dense and BM25, the candidates are RRF-fused and Cohere-reranked to a final top-50.

D.8 MEM0 AND MEM0-GRAPH

Mem0 is a cloud memory service that extracts and indexes atomic facts from text. During ingestion, consecutive case-file entries are batched into approximately 4,096-character blocks, matching the MemoryAgentBench reference chunk size, and submitted to Mem0 with day, time, entry type, and author as metadata. Mem0 extracts atomic facts, deduplicates them, and builds a searchable index. We wait 30 seconds after ingestion to cover Mem0's documented asynchronous fact-extraction window. At query time we retrieve the top-50 most relevant memories, which are concatenated and passed to the answering LLM as paraphrased atomic facts rather than verbatim text. Mem0 is evaluated with two answering LLMs (`gpt-5.1` and `gemini-2.5-flash`); per-LLM scores are reported in Table 10.

Mem0-Graph uses the same backend with graph extraction enabled, which additionally extracts entity–relation triples during ingestion and returns them alongside retrieved memories. The two fields are appended in sequence into the answering LLM's context (memories first, then triples) with no joint reranking. We report it as a distinct system because the graph layer changes what is retrievable per question, even though the underlying service is shared with Mem0. Retrieval uses top-30 memories, and the ingestion settle interval is increased to 60 seconds to cover Mem0's documented graph-construction window. Mem0-Graph is evaluated under the same two-LLM answering protocol as Mem0.

D.9 HINDSIGHT

Hindsight is a self-hosted memory agent with a retain–recall–reflect pipeline. During setup we create a per-case memory bank with a mission statement describing the investigation domain and a forensic-analysis disposition profile. Ingestion uses Hindsight's batch interface, with each entry mapped from its in-case timestamp (e.g., Day 5, 14:30) to an absolute datetime so that Hindsight's temporal search strategies become available. We wait 60 seconds after ingestion to cover Hindsight's documented background observation-consolidation window. At query time we use Hindsight's *recall* mode (rather than *reflect*), at the maximum recall budget and with raw source chunks included, returning both consolidated memories and up to 1,500 tokens of source text. These are concatenated and passed to the answering LLM. Hindsight is evaluated with two answering LLMs (`gpt-5.1` and `gemini-2.5-flash`); per-LLM scores are reported in Table 10.

D.10 SUPERMEMORY

Supermemory is a cloud document-memory store with semantic search. Each case-file entry is ingested under a per-case container tag, with deterministic per-entry identifiers to support idempotent re-ingestion. We wait 600 seconds after ingestion, the upper bound of Supermemory's documented indexing window for documents of this size. At query time we query Supermemory's document-search endpoint with full-document return, server-side reranking, and query rewriting enabled, so that full source documents (rather than chunk previews) are returned under Supermemory's own reranking and query-rewriting layers. The top-50 documents are concatenated and passed to the answering LLM. Supermemory is evaluated with two answering LLMs (`gpt-5.1` and `gemini-2.5-flash`); per-LLM scores are reported in Table 10.

D.11 HUMAN BASELINE

Sample and protocol. We sample 60 questions stratified across the six task categories (10 per category) and balanced across the three domains (20 per domain). Three annotators independently attempted each question with access to a question-specific evidence packet (the answer-relevant provenance subgraph paired with its matching narrative excerpts) but no access to the structured skeleton, the full case file, or any oracle aid. The abstain option was available throughout. Each annotator spent approximately 9 hours total ($\sim$8.5–9 minutes per question). Final answers are determined by majority vote.

Per-task results. Human Accuracy ranges from 80% on Temporal Fact Retrieval and Temporal Constraint to 40% on Counterfactual; abstention ranges from $0/10$ to $2/10$ questions, and mean time per question rises from 3 to 15 minutes (Table 8).

Task	Acc.	Score	Abst.	Time
Temporal Fact Retrieval	80%	0.78	0/10	3 min
Temporal Constraint	80%	0.72	1/10	5 min
Source Conflict	70%	0.63	1/10	7 min
Chain Reconstruction	60%	0.49	1/10	9 min
Cascade Propagation	50%	0.32	2/10	12 min
Counterfactual	40%	0.21	2/10	15 min

Table 8: Human baseline by task category (majority vote of three annotators per question).

Per-domain results. Accuracy is reported by domain in Table 9.

Domain	Accuracy
Crime	13/20 (65.0%)
Finance	12/20 (60.0%)
Medical	13/20 (65.0%)

Table 9: Human baseline by domain.

Comparison to the retrieval-perfect control. The packet-equipped human baseline serves as a complementary comparison to the retrieval-perfect control, which receives the full case graph in structured form rather than the answer-relevant subgraph. Humans reach 63.3% Accuracy (38/60) compared with 54.6% for the retrieval-perfect control, while the human abstention rate is $7/60 = 11.7\%$ compared with approximately 5% for the control. This indicates that reasoning over presented evidence remains nontrivial even when retrieval is controlled for. Humans and LLMs agree that Temporal Fact Retrieval is among the easiest tasks, whereas the control is weakest there (23.5%) because the structured representation abstracts away the surface narrative detail this task probes (§4.3). Counterfactual is the hardest task for humans (40%) and remains among the control's weakest (41.2%).

D.12 TESTING FOR NARRATOR-FAMILY BIAS

Because our generator rotates four LLM families as narrators across the per-domain seeds, a solver could in principle be advantaged on cases narrated by its own family. For each LLM that serves as both a narrator and a solver, we measure the Accuracy difference between same-family and cross-family cases using a paired permutation test (10^4 case-level label shuffles). The observed differences (GPT-5.1 $+1.3$ pp, p=0.89; Gemini-2.5-Pro $+13.3$ pp, p=0.32; Llama-3.3-70B -5.7 pp, p=0.47) are statistically non-significant and inconsistent in direction, indicating no systematic narrator-family advantage.

D.13 PER-LLM AND PER-DOMAIN BREAKDOWNS

Table 10 reports per-answering-LLM scores for the RAG and memory architectures (the family-mean rows in Table 2). Table 11 reports per-domain Score across all evaluated systems.

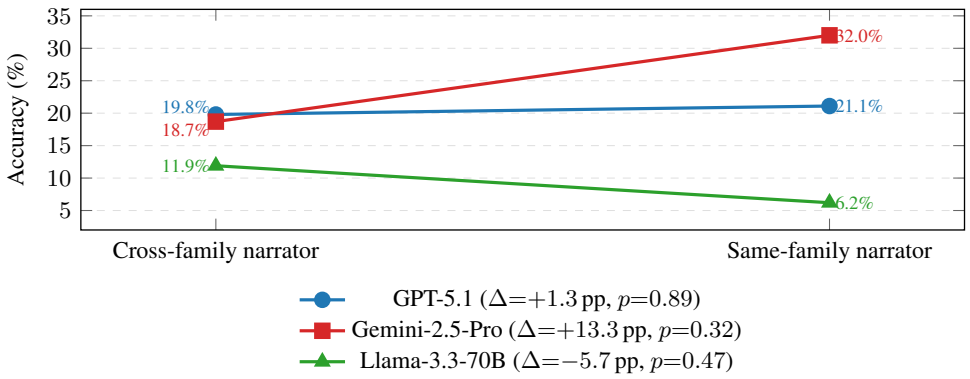

Figure 6: Same-family vs. cross-family narrator Accuracy for each LLM that doubles as narrator and solver.

System	`gpt-5.1`	`gemini-2.5-flash`
RAG		
plain	0.197	0.158
+rerank	0.207	0.178
hybrid + rerank	0.176	0.182
hybrid + rerank + MQ	0.161	0.184
Memory		
Mem0	0.106	0.118
Mem0-Graph	0.140	0.158
Supermemory	0.232	0.189
Hindsight	0.142	0.154

Table 10: Per-answering-LLM overall Score for RAG and Memory systems. The corresponding rows in Table 2 are the unweighted mean of the two columns.

System	Crime	Medical	Finance	Overall
Long-context				
gpt-5.1	0.304	0.303	0.253	**0.287**
gpt-4o-mini	0.188	0.246	0.273	0.232
gpt-4.1-mini	0.227	0.237	0.154	0.206
gemini-2.5-pro	0.435	0.193	0.127	0.265
gemini-2.5-flash	0.305	0.254	0.200	0.256
kimi-k2.5	0.314	0.180	0.256	0.257
qwen3-vl-30b	0.280	0.244	0.159	0.230
llama-3.3-70b	0.176	0.161	0.237	0.192
Oracle				
gpt-5.1	0.561	**0.726**	0.655	0.638
gemini-2.5-flash	**0.689**	0.600	**0.657**	**0.654**
RAG				
plain	0.263	0.112	0.133	0.178
+rerank	0.256	0.191	0.118	0.192
hybrid + rerank	0.278	0.129	0.106	0.179
hybrid + MQ	0.251	0.186	0.068	0.173
Memory				
Mem0	0.106	0.133	0.102	0.112
Mem0-Graph	0.176	0.149	0.116	0.149
Supermemory	0.299	0.160	0.150	0.211
Hindsight	0.157	0.162	0.125	0.148

Table 11: Per-domain scores across all evaluated systems.

D.14 Per-task Family-Mean Visualization

Figure 7 aggregates calibrated Score by family across the six task categories, complementing the per-system breakdown in Table 2. The Oracle lead over long-context is strongly task-specific: 50–70 pp on the three structured tasks (Cascade, Chain Reconstruction, Source Conflict), 15–35 pp on Counterfactual and Temporal Constraint, and shrinks to about 3 pp on Temporal Fact Retrieval, where RAG overtakes the structured representation. Memory averages 0.57 on Cascade but does not exceed 0.15 on any other task: Chain Reconstruction and Source Conflict require inter-fact edges that document-level and atomic-fact stores compress poorly.

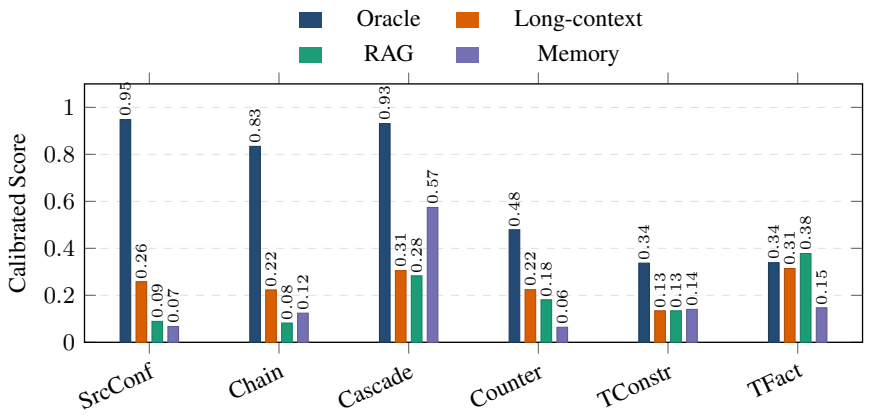

Figure 7: Mean calibrated Score by family across the six task categories.

E LIMITATIONS

Narrative realization. Case narratives are rendered by LLMs from immutable factual fields, so their style may carry model-specific priors. We mitigate this by rotating narration across model families and generating distinct evidence structures per case, and find no systematic narrator-family advantage (Appendix D.12).

Schema regularities. Because questions are generated from a fixed schema, a pipeline tuned to that schema could in principle exploit format regularities. RECON limits this by decoupling the dependency graph from the surface narrative and by the anonymization, normalization, and perturbation audits described in §3.3.

Case-level scale. RECON's 1,604 questions are drawn from 24 densely annotated case files of 50K–100K tokens each. Questions from the same case share context, so the effective sample size is smaller than the question count; because the generator is seeded, additional cases can be produced to increase diversity.

Residual ambiguity. Human validation flagged 8 of 200 sampled items (4.0%) as ambiguous by at least two of three annotators, concentrated in counterfactual and cascade tasks (Appendix C). We retain them for completeness; excluding them changes headline Scores by at most 0.002 (Table 5).

F EXAMPLE QUESTIONS

Below we present one MCQ-single example per task category drawn from the evaluated case file. Each question has six options (A–F); the correct answer is marked with ✓. Options are designed as targeted distractors that exploit specific failure modes (noted in parentheses where relevant).

F.1 CHAIN RECONSTRUCTION

Which statement correctly summarizes the evidence trail "Tracking Marcus Webb's movements on the day of the murder"?

A) The correct sequence is: Marcus Webb admits to purchasing cyanide ... → Marcus Webb's card swiped into university library at 5:05 PM. (reversed order)
B) The trail involves the first and last steps but omits the intermediate steps: library swipe-out, witness sighting, traffic camera, ... (incomplete chain)
C) The key steps are the initial discovery and the final confirmation; the middle steps are procedural filler. (minimizes chain)
D) This trail includes "Emails containing threats found on victim's computer" as a key step. (wrong chain)
E) The most recent finding is the only step needed to establish the trail. (single-hop shortcut)
✓ F) Step 1: Library swipe-in (5:05 PM) → Step 2: Swipe-out (6:42 PM) → Step 3: Witness sighting (6:45 PM) → Step 4: Traffic camera (7:15 PM) → Step 5: Phone disconnected (7:00–9:00 PM) → Step 6: Coffee mug found → Step 7: Fingerprint match → Step 8: Cyanide receipt → Step 9: Credit card link → Step 10: Confession.

Note: This question has chain length 10, requiring the model to locate and correctly order all 10 hops scattered across the 100K-token case file.

F.2 CASCADE PROPAGATION

After the contamination of the coffee mug with residue found at the victim's desk in Office 304 (detective accidentally smudged fingerprints while collecting evidence), which conclusion is invalidated?

A) The contamination occurred after all forensic analysis was completed, so no results are affected. (ignores invalidation)
✓ B) Fingerprint matches to Marcus Webb.
C) All forensic results from the coffee mug remain fully valid despite the contamination. (ignores invalidation)
D) Autopsy findings (independent of physical evidence). (over-scoping)
E) Both fingerprint matches and cyanide detection are compromised; all evidence from the mug should be excluded entirely. (over-scoping)
F) All other forensic results processed separately from the coffee mug. (wrong target)

Note: The model must recognize that cyanide detection (a chemical test) is independent of fingerprint contamination and survives the invalidation. Option E is the primary trap, testing whether the model collapses all downstream conclusions without checking for independent support.

F.3 SOURCE CONFLICT RESOLUTION

Which statement correctly describes the resolution of the conflict between Laura Kim and David Nguyen?

A) Laura Kim's statement was recorded first, so it should be considered more reliable. (recency bias)
B) Both claims can be true simultaneously. (false reconciliation)
✓ C) Marcus Webb left the library as witnessed by Laura Kim, heading towards the clinic.
D) David Nguyen's professional role makes their observation more credible. (authority bias)
E) Both witnesses are equally credible with no way to determine which is correct. (false equivalence)

F) The corroborating evidence (traffic camera at 7:15 PM) actually supports David Nguyen's account. (misattributed corroboration)

Note: The traffic camera sighting near the clinic at 7:15 PM corroborates Kim's account (Webb left the library toward the clinic), not Nguyen's claim that Webb was at a café.

F.4 Counterfactual Reasoning

If the contamination of the coffee mug with residue had been discovered on Day 9 instead of Day 11, would Detective Ruiz's notes on Day 9 still have occurred?

A) This investigative action would have taken place earlier. (temporal confusion)
B) Cannot be determined from the available information. (false uncertainty)
C) This action would have taken place but led to different conclusions. (partial counterfactual)
✓ D) No, this step would not have occurred.
E) This finding would still have occurred, just at a later point. (temporal shift)
F) Yes, this step would still have occurred. (ignores dependency)

Note: The model must determine that Detective Ruiz's Day 9 notes depended on the coffee mug evidence, so discovering the contamination on Day 9 (before the notes were written) would have prevented them.

F.5 Temporal Constraint Satisfaction

Which suspect lacks continuous surveillance or cell tower coverage during the time of the crime?

A) Nina Patel.
✓ B) Marcus Webb.
C) Lydia Tran.
D) Samuel Lee.
E) Robert Harris.
F) Eleanor Dawson.

Note: Marcus Webb's phone was disconnected from the network between 7:00 PM and 9:00 PM on Day 1, creating a gap in cell tower coverage during the crime window. All other suspects have continuous coverage from at least one data stream.

F.6 Temporal Fact Retrieval (Baseline)

Which happened first: a cell tower ping recorded at Hillcrest Suites or Detective Ruiz's alibi verification for Samuel Lee?

A) A cell tower ping recorded for Lydia Tran. (wrong person)
B) Officer Davis's alibi verification for Samuel Lee. (wrong detective)
C) Detective Kowalski's witness statement about Nancy Green. (wrong event type)
D) Officer Chen's surveillance report involving Jonathan Cruz. (wrong event type)
E) Detective Ruiz's alibi verification for Samuel Lee. (wrong temporal order)
✓ F) A cell tower ping recorded at Hillcrest Suites.

Note: The Hillcrest Suites cell tower ping occurred on Day 1 (18:17), while Detective Ruiz's alibi verification occurred on Day 4 (07:46). This is a straightforward temporal ordering question requiring no multi-hop reasoning.