ICM-BENCH: A BENCHMARK FOR EVALUATING THE IN-CONTEXT MODIFICATION ABILITY OF LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM) agents continually accumulate conversation histories, yet the contextual state relevant to the current situation is not identical to the full history: some previously valid information may become obsolete, while other information should remain unchanged. We formalize the ability to selectively transform such contextual information during inference, without parameter updates, as In-Context Modification (ICM). Existing benchmarks on unlearning and long-term memory primarily evaluate forgetting, retrieval, or conflict resolution, but do not systematically test whether models can update the current contextual state while preserving unaffected information. To fill this gap, we introduce ICM-Bench, a controlled benchmark for evaluating ICM in multi-turn conversations. It contains over 2.8K multi-turn conversations and 26K queries, covering deletion and update across single-attribute, multi-hop, and entity-level settings, together with explicit retain and unmodified evaluations. All evaluated knowledge is purely contextual, isolating ICM from parametric knowledge. Experiments across different LLMs reveal a striking gap between using and modifying context: unmodified success ranges from 94.50% to 100.00%, whereas modified success ranges from only 0.25% to 35.75%. Existing inference-time methods can improve modification success, but often at the cost of retained or unmodified information. These results show that reliable ICM requires selective contextual state transformation rather than indiscriminate forgetting, and establish ICM-Bench as a benchmark for developing LLM agents that can maintain an accurate current state across evolving conversations.

1 INTRODUCTION

Large language models (LLMs) are increasingly deployed as autonomous agents that interact with the environment over extended continual processes. In such settings, information relevant to future decisions accumulates throughout the interaction rather than being provided all at once. As the interaction executes, the LLM should be able to continually determine which part of its accumulated history remains relevant to the current state at hand.

However, in the real world, the accumulated experience does not necessarily remain correct as the interaction evolves. Information that was previously correct may become outdated or inconsistent, while other information from the same history remains valid. Consider a familiar scenario in academic research: a weekly lab meeting is routinely scheduled for Friday at 9:00 AM in Room A. Under normal circumstances, researchers are present in accordance with standard practice. When one PhD announces that this week's meeting will take place in Room B, the human participant can effortlessly construct a new mental state: (1) *updating* the venue to Room B, (2) *retaining* untouched background facts (the meeting remains Friday at 9:00 AM, topic and participants unchanged, and room A remains the default for subsequent meetings).

We formalize this key capability as *In-Context Modification (ICM)*: the ability of an LLM to selectively transform previously observed contextual information during inference strictly without updating its model parameters. This ability is particularly important for continually interacting and self-evolving agents. An agent must be able to actively decide which previously observed information should continue to influence its current behavior and which should no longer do so. It means that state

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.

construction from interaction history is recognized as a fundamental part of intelligent behavior in autonomous agents (Harutyunyan et al., 2026).

Existing benchmarks evaluate related aspects of information revision but only partially capture ICM. LLM unlearning benchmarks such as TOFU (Maini et al., 2024), RWKU (jin et al., 2024), and MUSE (Shi et al., 2025) primarily assess the removal of knowledge encoded in model parameters while preserving other knowledge or utility. ICM, by contrast, concerns the selective transformation of information supplied within a conversation, without parameter updates. More closely related benchmarks, including ICF-Bench (Qian et al., 2026) and the Selective Forgetting setting of MemoryAgentBench (Hu et al., 2026), evaluate in-context forgetting and conflict resolution. However, these evaluations do not systematically assess whether targeted information is correctly deleted or updated while unaffected information remains usable across different target structures. In particular, success on unmodified contexts does not establish whether still-valid information is preserved after a modification. What remains underexplored is a unified evaluation of both the correctness and scope of contextual changes in multi-turn conversations, spanning individual attributes, multi-hop dependencies, and entity-level information.

To address this gap, we introduce *ICM-Bench*, a controlled benchmark for evaluating ICM in multi-turn conversations while keeping the target LLM's parameters fixed. The benchmark contains over 2.8K conversation instances and 26K queries across six task scenarios, combining deletion and update tasks with single-attribute, multi-hop, and entity-level targets. Each modification instance includes an initial conversation history, subsequent messages specifying a contextual change, and queries that assess its effects. Under a unified protocol, we evaluate five LLMs and five inference-time baselines spanning prompt-based guidance, embedding perturbation, logit steering, active context management, and reasoning control. Moreover, ICM-Bench adopts automated checks and human review to support data quality and evaluation reliability.

ICM-Bench reveals three main findings:

- **Current LLMs struggle to modify context.** Across the tested models and target structures, success rates reach 94.50%–100.00% when no modification is required, but only 0.25%–35.75% when information must be deleted or updated. Models can therefore use conversation history effectively without being able to apply subsequent changes correctly.
- **Inference-time methods do not consistently close this gap.** Results from the five baselines are mixed. Some provide only limited gains, while others improve modification success but reduce accuracy on unaffected facts. The difficulty lies not just in changing the targeted information, but in doing so without losing information that should remain usable.
- **ICM-Bench poses a harder test for the evaluated LLMs.** All five LLMs achieve lower success rates on our deletion and update tasks than on ICF-Bench and the Selective Forgetting setting of MemoryAgentBench. Although the evaluation protocols differ, this pattern suggests that performance on related benchmarks does not fully reflect the difficulties of contextual modification.

2 Related Work

LLM Unlearning LLM unlearning aims to suppress the influence of designated parametric knowledge while preserving model utility, assuming predefined static forget and retain sets. *Training-based* methods update parameters through gradient ascent or preference alignment (Yao & Xu, 2024; Zhang et al., 2024; Tan et al., 2025; Shen et al., 2025), whereas *inference-time* methods keep weights fixed, using In-Context UnLearning (ICUL), Embedding-COrrupted prompts (ECO), or decoding-based interventions (Pawelczyk et al., 2024; Liu et al., 2024; Merchant & Levy, 2026; Chowdhury et al., 2026).

Context & Memory Management Existing approaches improve LLMs' context utilization through compression, external memory, and active management. LLMLingua reduces redundant tokens (Jiang et al., 2023), while MemGPT extends the effective context window through hierarchical memory (Packer et al., 2023). M+ combines long-term memory with a co-trained retriever (Wang et al., 2025); A-Mem supports agent-driven memory organization (Xu et al., 2025), and MemGAS consolidates dialogue memory under limited context windows (Xu et al., 2026). Sculptor enables LLMs to hide, restore, summarize, and search contextual information (Li et al., 2026).

Table 1: Comparison of ICM-Bench with other benchmarks. *Retain Query* refers to dedicated post-modification queries about unaffected facts, whereas *Unmodified Query* refers to queries evaluated under the corresponding unmodified condition. † For MemoryAgentBench, the reported statistics refer only to its Selective Forgetting scenario.

Criterion		TOFU	RWKU	MUSE	ICF-Bench	MemoryAgentBench	ICM-Bench
Queries		4k	13k+	200 / corpus	2k+	800	**26k+**
Samples		200 authors	200 targets	2 corpus	2k+	8 (shared)	**2.8k+**
Granularity Coverage	Single Attribute	✗	✓	✓	✓	✓	✓
	Multi-Hop	✗	✗	✗	✗	✓	✓
	Entity-Level	✓	✓	✗	✗	✗	✓
Evaluation Coverage	Inference-Time	✗	✗	✗	✓	✓	✓
	Retain Query	✓	✓	✓	✗	✗	✓
	Unmodified Query	✗	✗	✗	✓	✗	✓
Evaluation Metric		ROUGE-L, Truth Ratio, KS-Test	ROUGE-L, MIA	ROUGE-L, MIA	LLM-as-a-Judge	Substring Exact Match	**Substring Exact Match + Task-Specific Rules**

Existing Relevant Benchmarks Parametric unlearning benchmarks evaluate training knowledge. TOFU (Maini et al., 2024) divides forget and retain sets with 200 fictitious author profiles. Every author has about 20 Q&A pairs of attributes. RWKU (jin et al., 2024) has 200 real-world famous people and probes knowledge. MUSE (Shi et al., 2025) contains the news and book corpus and tests verbatim and knowledge memorization. LongMemEval evaluates long-term conversational memory across five abilities through individual questions (Wu et al., 2025).

The closest benchmarks are ICF-Bench (Qian et al., 2026) and MemoryAgentBench (Hu et al., 2026). ICF-Bench evaluates Instructional Forgetting, Subtask Revision, and Dynamic Preference. It focuses on the single-hop conversation scenario, with only one attribute modify/forget. In the Selective Forgetting setting, MemoryAgentBench evaluates conflict resolution through single- and multi-hop questions over shared fact lists, requiring models to prioritize newer information. Although this setting covers multi-hop dependencies, it does not include entity-level modification tasks or separate retain-query evaluation.

As summarized in Table 1, *ICM-Bench* combines single-attribute, multi-hop, and entity-level contextual transformations with dedicated retain queries and paired unmodified controls in a unified inference-time evaluation. Queries about modified information assess the effects of deletion or updating, retain queries test whether unaffected facts remain usable, and unmodified controls establish whether the original information can be correctly retrieved or inferred. Together, these evaluations distinguish successful modification of targeted information from unintended loss of unaffected information. Moreover, as it is shown in the experimental results, ICM-Bench poses a unique challenge for today's LLMs and LLM inference-time methods.

3 ICM-BENCH

3.1 MOTIVATING EXAMPLE

Consider an LLM-based university assistant. Early in a conversation, it learns that Maya is enrolled at the university and has completed Linear Algebra with an A. A later message states, "Maya has graduated and is no longer enrolled." If asked, "Is Maya still enrolled?", the assistant should answer no. But if asked, "What grade did Maya receive in Linear Algebra?", it should still answer A. Continuing to describe Maya as enrolled would ignore the update, while forgetting her grade would discard information that remains valid. The assistant must therefore revise its understanding of Maya's current status without losing her academic history. This distinction illustrates why the full conversation history cannot simply be treated as the current contextual state or discarded wholesale. We next formalize this selective transformation of contextual information as In-Context Modification (ICM), performed during inference without updating model parameters.

3.2 PROBLEM DEFINITION

Let $H_t = \big((u_1, r_1), \ldots, (u_t, r_t)\big)$ denote the accumulated history, where u_i and r_i are the user utterance and model response at turn i. Subsequent messages ΔH_t may change which information remains valid. At evaluation, the complete history is $H_t^+ = H_t \oplus \Delta H_t$, where $\oplus$ denotes chronological concatenation.

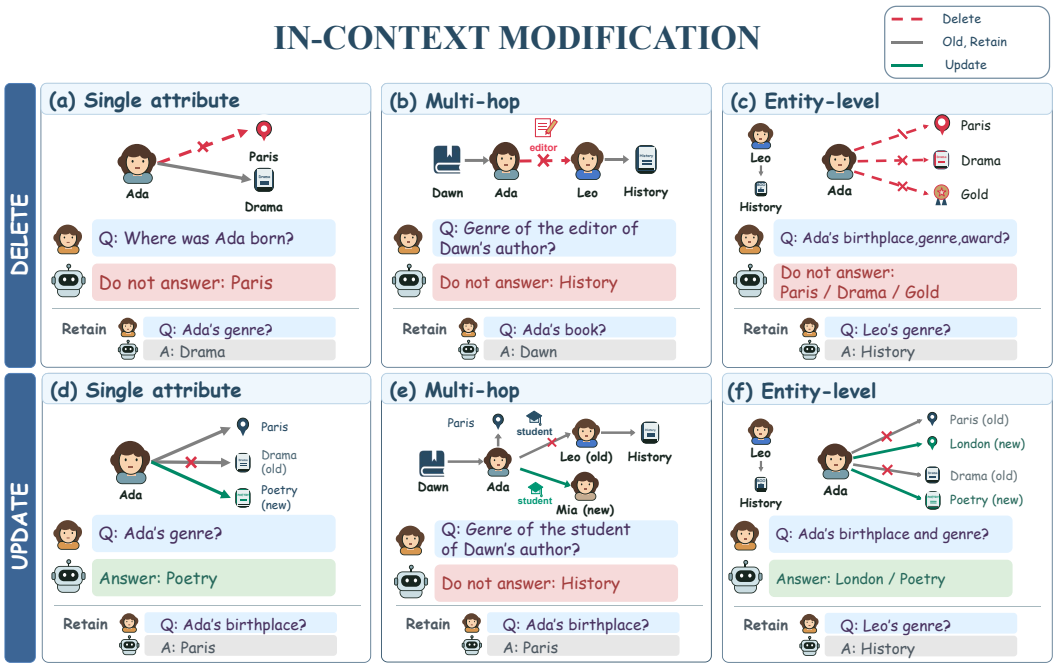

Figure 1: **Task scenarios in ICM-Bench.** (a–c) Deletion and (d–f) updating across single-attribute, multi-hop, and entity-level targets. The modifications affect an individual fact, a link in a reasoning chain, or multiple facts associated with an entity, respectively.

Let Φ be a conceptual mapping from conversation information to the currently relevant contextual state. Before ΔH_t,

$$S_t = \Phi(H_t, \varnothing) = \{x_1, \ldots, x_n\}, \tag{1}$$

where each x_i is an initially valid information unit, such as an attribute, a relation, or a set of entity attributes.

Definition 1. *In-Context State Transformation in ICM. Given H_t and ΔH_t, the desired contextual state is*

$$S'_t = \Phi(H_t, \Delta H_t). \tag{2}$$

This is a semantic target, not an assumption about the LLM's internal representation. The subsequent messages determine a partition of S_t into disjoint deletion, update, and retention sets $\mathcal{D}$, $\mathcal{U}$, and $\mathcal{R}$:

$$x_i \to \begin{cases} \varnothing, & x_i \in \mathcal{D}, \\ x'_i, & x_i \in \mathcal{U}, \\ x_i, & x_i \in \mathcal{R}, \end{cases} \tag{3}$$

where x'_i is the updated value. Thus, $S'_t = \mathcal{R} \cup \{x'_i \mid x_i \in \mathcal{U}\}$. If no information changes, the case is unmodified*, with $S'_t = S_t$.*

We define *In-Context Modification (ICM)* as an LLM's ability to realize this transformation without parameter updates. Given the complete history and a query q about the current situation, the model generates

$$\hat{a} \sim P_\theta(\cdot \mid H_t^+, q). \tag{4}$$

Queries about modified and unaffected information test whether responses reflect the required deletion or update while preserving still-valid information, providing behavioral evidence of ICM.

3.3 The Architecture of ICM-Bench

Task design. ICM-Bench combines delete and update transformations with three target structures, yielding the six scenarios in Figure 1. In the delete case, the model must stop using specified prior information ($x \to \varnothing$). In the update case, the model must replace an outdated value with a new

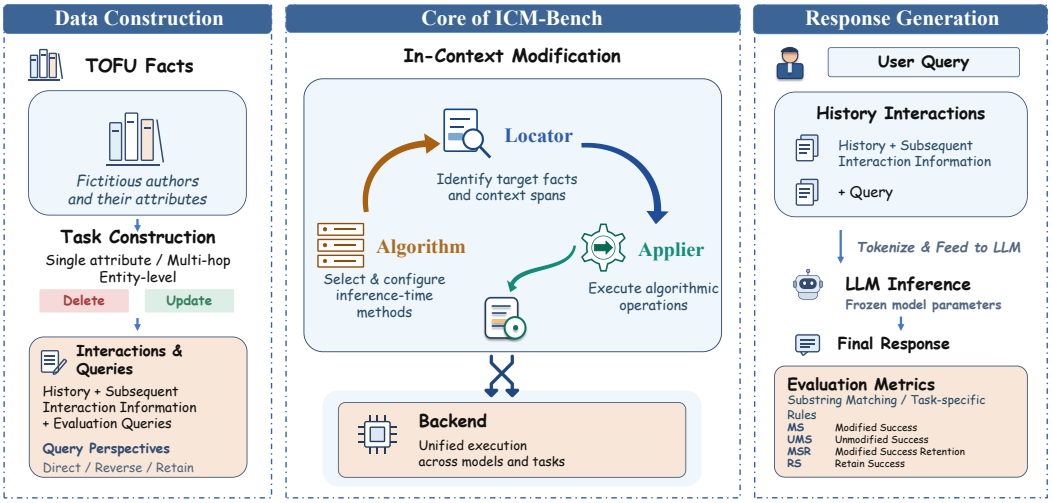

Figure 2: **Overview of the ICM-Bench evaluation pipeline.** Fictitious author facts are used to construct multi-turn conversations and evaluation queries. A Locator–Applier–Backend module supports unified execution of inference-time methods across models and tasks. Model responses are generated with frozen parameters and evaluated using substring matching and task-specific rules to assess modification success, information retention, and performance under unmodified conditions.

one ($x \to x'$). Each transformation targets either one attribute, a fact in a multi-hop chain, or all contextual facts about an entity. In multi-hop cases, conclusions that depend on an obsolete link must also change. Unaffected facts must remain available ($x \to x$), so we test retention in every scenario. We also include an unmodified control. After the full interaction H_t^+, direct and reverse queries probe affected relations from either direction. Retain queries ask about unaffected facts. In the unmodified control, queries target facts that have not changed at all. These are evaluation perspectives, not additional contextual transformation types.

Benchmark Structure. To systematically evaluate the ICM capability of LLMs, we develop ICM-Bench as a modular inference-time evaluation framework. As illustrated in Figure 2, the benchmark pipeline consists of three main stages: data construction, the core ICM execution module, and response generation. This unified workflow supports the same benchmark instances and evaluation protocol across different LLMs, task scenarios, and inference-time methods.

In the **Data Construction** stage, we start from TOFU Facts containing fictitious authors and their attributes. We then build three types of task scenarios including single-attribute, multi-hop, and entity-level tasks for deletion and update contextual state transformations. Finally, we assemble interaction histories paired with evaluation queries from direct, reverse and Retain query perspectives.

The **Core of ICM-Bench** adopts a *Locator–Applier–Backend* abstraction. The *Locator* identifies target facts and corresponding context spans. The *Applier* executes the given algorithmic operations. The *Backend* provides unified execution across different models and tasks. This separation allows heterogeneous methods to be evaluated under the same benchmark protocol.

For **Response Generation**, we combine user queries and historical interaction information and feed the constructed prompt into the LLM to obtain model responses. After inference, generated outputs are evaluated via the judgment module. The judgment process is decoupled from response generation so that identical model outputs can be assessed under fixed, reproducible criteria using substring matching and task-specific rules.

3.4 Dataset Construction

To precisely evaluate the in-context modification (ICM) abilities of LLMs, we build controlled contextual interactions based on the Task of Fictitious Unlearning (TOFU) (Maini et al., 2024). TOFU provides synthetic author-specific factual attributes. These fictitious facts only stay in the constructed dialogue context. They are not encoded in model parameters. This property eliminates

interference from the LLMs' pre-trained parametric knowledge. It matches our inference-time ICM paradigm well. All modifications work on contextual states and do not update model parameters. Each benchmark instance has an initial multi-turn context H_t, a subsequent contextual delta ΔH_t that induces modification, and a suite of evaluation queries. For unmodified cases, the accumulated contextual history remains unchanged.

We design different prompts and construct pipelines for each task using the TOFU dataset to generate samples. For the single attribute instances, the generator first utilizes the selected author and attribute to obtain earlier interaction information. Then it gets a later interaction indicating that one fact no longer applies or that its value has changed. For the multi-hop instances, the generator constructs a dependency chain among multiple factual attributes and corresponding test queries. Then it generates conversations by using multi-author information. For the entity-level instances, the generator will generate multiple queries for an entity. All contextual facts associated with the target entity are treated as transformation targets. For each task, we include approximately five retain queries per instance.

The benchmark contains 2880 instances and 26195 deterministic evaluation queries over the six core task scenarios. Entity-level instances include both direct and reverse queries. The large number of evaluation queries can accurately quantify the evaluation of each instance with the capability of ICM in LLMs.

3.5 Evaluation and Quality Control

Given a whole interaction H_t^+, the model is evaluated on a set of post-transformation queries $\mathcal{Q} = \{q_1, \ldots, q_m\}$. The overall evaluation process is $(H_t^+, q_j) \to \hat{a}_j$, where $\hat{a}_j$ denotes the model response to query q_j. The queries are designed to evaluate LLMs' ICM ability to reach the desired contextual state S_t' defined in Section 3.2. Illustration 1 shows an ICM failure in which the model claims to have forgotten the target information but still reveals it in its response.

Illustration 1: ICM Failure in Entity Deletion (Direct Query)

...

Human: "The next author is Adriana Martinez, born in Barcelona..."

...

Human: "... She has been recognized with the prestigious International Award for Cultural Literature..."

...

Human: "Forget all information about Adriana Martinez."

Test query: "Based on what I told you earlier in this conversation, what is the award of Adriana Martinez?"

Model response: "I've forgotten. Adriana Martinez received the **International Award for Cultural Literature**."

The reliability of LLM-as-a-judge hinges on how model-internalized priors interact with user-provided instructions (Casanova et al., 2026). We evaluate each response using a combination of substring matching and task-specific logical rules. For single attribute queries with unambiguous target values, substring matching is used to determine whether the expected information is present in the response. For multi-hop, entity-level, and reverse queries, where correctness may depend on relations among multiple facts, we apply task-specific logical rules to determine whether the response satisfies the required transformation. Specifically, for entity-level, when all the queries are tested as successful, the instance is judged successful. The details of judgment rules are in the Appendix A.3. A successful contextual transformation must satisfy both requirements: the specified information should be correctly transformed, while unaffected information should remain usable. Consequently, a method that suppresses the target information by broadly disrupting the entire contextual state should not be considered a successful ICM, because it fails the preservation requirement.

We perform quality control at both the instance and judgment levels. During data construction, automatic rule checks are used to verify whether the basic construction of every instance is problematic. Most task instances achieve over 98% pass rate. We report detailed autonomous quality-control procedures and audit statistics in Appendix B.1.

To further assess benchmark quality, we sample 10% of the constructed instances from each of the six task scenarios. The three invited human reviewers independently examine the same sampled

Table 2: Inter-annotator agreement and sample validity on ICM-Bench.

Scenario	Gwet AC1	Majority pass rate
single attribute	0.9511	100.00%
multi-hop	0.9069	95.83%
entity	0.9244	96.55%

Table 3: Consistency between human annotations and rule-based evaluation on ICM-Bench.

Scenario	Fleiss' kappa	Cohen's kappa
single attribute	0.976	0.964
multi-hop	0.848	0.923
entity	0.913	0.956

instances. They identify ambiguous, incorrect, or logically inconsistent cases. Table 2 reports Gwet's AC1. This metric adjusts for chance agreement and measures classification concordance among multiple raters. The table also provides the majority pass rate, defined as the proportion of instances accepted by at least two reviewers. Both metrics reach high values across the three scenarios.

We then compare the human judgments with the rule-based evaluation results. As shown in Table 3, Fleiss' kappa and Cohen's kappa measure result consistency. Fleiss' kappa assesses classification agreement among multiple raters. Cohen's kappa quantifies agreement between two independent sets of annotations. Both show high agreement across the three scenarios. These results support the quality of the benchmark data and the reliability of its evaluation procedure. Detailed quality-control procedures and review statistics are provided in Appendix B.

4 Experiments

In this section, we evaluate the performance of the LLMs, baseline methods, and others on the ICM task. The experiments answer the three research questions: **(RQ1)** how the ICM performance of current LLMs, **(RQ2)** whether existing inference-time methods reliably improve this ability, and **(RQ3)** what types of inference-time strategies are effective for improving LLMs' ICM capability.

4.1 Experimental Setup

We conduct experiments on two established benchmarks, as well as our ICM-Bench. We used greedy decoding for all LLMs in all experiments to ensure reproducibility.

Benchmarks and evaluation settings. To comprehensively evaluate the ICM capability of current LLMs, the experiments are compared with three different benchmarks, including ours. ICF-Bench and MemoryAgentBench test existing notions of in-context forgetting and memory management, while our benchmark specifically evaluates the broader ICM capability on contextual state transformations. Regarding MemoryAgentBench, we report only on the Selective Forgetting task scenario, which is most relevant to our research.

- **ICF-Bench (Qian et al., 2026)**: a benchmark for in-context selective forgetting in multi-turn conversation. LLM is used as judge to evaluate correctness.
- **MemoryAgentBench (Hu et al., 2026)**: a benchmark that includes a selective forgetting task for evaluating whether LLMs can correctly utilize conflict information in long memory. The selective forgetting task includes 100 single-hop questions and 100 multi-hop questions with a shared sample. The selective forgetting task evaluates LLMs by substring exact-match accuracy on the task.
- **ICM-Bench**: our proposed benchmark, which instantiates the formulation in Section 3 and covers different transformation types, target structures, and query perspectives. We use both substring exact matching accuracy and task-specific logical rules to judge results together.

The results report four metrics. Modified Success (MS) measures performance when modification is required. Unmodified Success (UMS) measures performance in the unmodified case. Modified Success Retention (MSR) measures the proportion of originally successful instances that remain successful after modification. Retain Success (RS) measures the success rate of the model in preserving information unrelated to the modification target. Higher values indicate better performance on all metrics.

Table 4: The results of our ICM-Bench for current LLMs.

Model	Single attribute				Multi-hop				Entity-level			
	MS ↑	UMS ↑	MSR ↑	RS ↑	MS ↑	UMS ↑	MSR ↑	RS ↑	MS ↑	UMS ↑	MSR ↑	RS ↑
Llama 3.1-8B	2.85%	98.15%	1.94%	97.80%	5.62%	96.46%	5.18%	94.99%	0.25%	94.50%	0.00%	98.83%
GLM 4-32B	1.10%	98.95%	0.51%	99.56%	4.17%	98.96%	3.79%	99.20%	16.25%	98.25%	16.28%	99.72%
Gemma 4-31B	0.80%	99.75%	0.70%	99.85%	3.54%	99.79%	3.34%	99.80%	35.75%	99.75%	35.59%	99.94%
Qwen 3.8-27B	9.95%	99.85%	9.91%	99.89%	0.83%	100.00%	0.83%	99.95%	25.00%	99.75%	24.81%	99.89%
GPT 5.6-light	1.95%	98.90%	1.72%	99.24%	26.46%	98.54%	25.37%	99.40%	29.00%	99.75%	29.07%	99.45%

Models. We select five representative LLMs with different scales and architectures: Llama 3.1-8B, GLM 4-32B, Gemma 4-31B, Qwen 3.8-27B, and GPT 5.6-light. These models allow us to examine whether the ability to ICM depends on model scale and family.

Baselines. We evaluate five inference-time methods for LLMs on the three benchmarks. The following baselines represent different strategies for intervening on the generation process without updating the parameters of target LLMs. The details of the baselines are in the Appendix C.

- **In-Context UnLearning (ICUL) (Pawelczyk et al., 2024)**: It deploys a uniquely built context to eliminate the influence of a training point on the model output.
- **Embedding-COrrupted prompts (ECO) (Liu et al., 2024)**: a lightweight unlearning framework for LLMs employs a classifier to identify prompts to forget and corrupts embeddings to enforce an unlearned state during inference.
- **Divergence Decoding (DD) (Merchant & Levy, 2026)**: It is a mechanism that uses two small auxiliary models to steer the logits of the LLM away from specific data during inference.
- **Sculptor (Li et al., 2026)**: The framework equips LLMs with three categories of tools, including (1) context fragmentation, (2) summary, hide, and restore, and (3) precise search. LLMs can proactively manage their in-context.
- **Dynamic Early Exit in Reasoning (DEER) (Yang et al., 2026)**: DEER is a training-free method that allows LLMs to self-truncate Chain-of-Thought (CoT) sequences by early exit.

4.2 RQ1: HOW IS THE ICM PERFORMANCE FOR CURRENT LLMS?

Current LLMs perform well in the unmodified case, with high UMS, as shown in Table 4. However, MS is low across all three target structures for the LLMs. The results indicate that LLMs' ICM capability is not sufficient to succeed. Models can effectively utilize historical interaction contextual information, but struggle to selectively transform special target information. The entity-level modification achieves a relatively higher MS. This higher score may partly reflect a simple refusal strategy, where models avoid answering whenever the target entity is mentioned, rather than precisely modifying its associated information. The details of ICM-Bench are shown in the Appendix D.1.

For the existing benchmarks, as shown in Table 5, ICM-Bench is more challenging than the ICF-Bench and MemoryAgentBench for LLMs. For example, Gemma-4-31B achieves 58.00% on MemoryAgentBench and 78.57% on ICF-Bench, but only 10.49% on delete and 1.74% on update on ICM-Bench. This performance gap suggests that while current large models can address the issues associated with MemoryAgentBench and ICF-Bench, they lack the robust ICM capabilities required to handle our benchmark tasks. *Therefore, our proposed ICM is a necessary and unresolved capability for evaluating LLMs*.

4.3 RQ2: CAN EXISTING INFERENCE-TIME METHODS RELIABLY IMPROVE ICM?

We further test whether existing inference-time methods can reliably improve ICM for LLMs. As shown in Table 6, some methods improve MS performance while substantially degrading RS and UMS performance, others have no consistent or substantial improvement. In particular, Sculptor shows an increase in MS performance, but this gain is accompanied by a substantial drop in both RS and UMS performance. The ICM requires that a method by corrupting the whole context should not be considered successful contextual state transformations. Overall, existing inference-time methods indicate that they cannot yet reliably improve the ICM capability of LLMs. The full results for each task scenario are in Appendix D.2.

Table 5: We compare our results with different benchmarks. This table presents task success rates of each benchmark under their respective scenarios. Our benchmark reports the average values for deletion and update tasks respectively. MemoryAgentBench reports the average success rate for single-hop and multi-hop questions in the selective forgetting scenario.

	ICM-Bench		MemoryAgentBench	ICF-Bench
Model	Delete ↑	Update ↑	Conflict ↑	Forget ↑
Llama 3.1-8B	3.89%	2.01%	37.50%	10.26%
GLM 4-32B	3.54%	3.89%	36.50%	60.97%
Gemma 4-31B	10.49%	1.74%	58.00%	78.57%
Qwen 3.8-27B	20.83%	0.21%	87.00%	57.34%
GPT 5.6-light	11.18%	8.40%	85.00%	18.21%

Table 6: The results of baseline methods on the different benchmarks.

Model	Method	ICM-Bench (Delete)				ICM-Bench (Update)				MemoryAgentBench	ICF-Bench		
		MS ↑	UMS ↑	MSR ↑	RS ↑	MS ↑	UMS ↑	MSR ↑	RS ↑	MS ↑	MS ↑	UMS ↑	MSR ↑
Gemma 4-31B	Base	10.49%	99.65%	10.31%	99.89%	1.74%	99.86%	1.67%	99.82%	58.00%	78.57%	97.99%	78.64%
	ICUL	14.65%	99.38%	14.19%	99.86%	1.11%	99.86%	0.97%	99.87%	46.50%	87.42%	96.38%	88.10%
	ECO	10.14%	99.65%	9.97%	99.91%	1.60%	99.86%	1.53%	99.79%	58.00%	79.18%	97.79%	80.97%
	DD	12.29%	91.25%	6.16%	99.91%	5.69%	96.74%	5.53%	99.78%	47.50%	80.08%	74.45%	81.49%
	Sculptor	87.85%	84.79%	87.96%	55.92%	66.81%	85.14%	66.48%	62.91%	58.50%	62.17%	68.31%	60.68%
	DEER	19.65%	98.75%	18.78%	99.79%	16.53%	98.89%	15.87%	99.55%	88.50%	30.28%	92.15%	30.46%
Qwen 3.8-27B	Base	20.83%	99.72%	20.75%	99.94%	0.21%	100.00%	0.21%	99.86%	87.00%	57.34%	92.05%	56.50%
	ICUL	20.35%	95.56%	18.17%	96.60%	12.50%	95.14%	9.78%	95.07%	86.50%	93.06%	93.06%	93.30%
	ECO	20.69%	99.79%	20.60%	99.92%	0.35%	100.00%	0.35%	99.91%	86.00%	84.61%	97.69%	86.61%
	DD	18.96%	74.38%	15.03%	97.50%	26.74%	80.90%	19.40%	93.48%	42.50%	84.51%	69.01%	79.01%
	Sculptor	63.96%	73.96%	62.25%	57.58%	40.00%	67.99%	39.94%	89.30%	80.00%	65.09%	76.36%	62.85%
	DEER	18.33%	99.31%	17.90%	99.71%	10.97%	99.17%	10.85%	99.45%	93.50%	50.60%	93.96%	50.43%
GPT 5.6-light	Base	11.18%	99.10%	10.65%	99.41%	8.40%	98.82%	8.29%	99.18%	85.00%	18.21%	94.27%	19.32%
	ICUL	38.12%	98.82%	37.53%	99.49%	27.71%	98.68%	27.38%	99.24%	88.50%	46.88%	96.68%	46.72%
	Sculptor	83.13%	63.33%	80.70%	35.27%	86.46%	46.94%	83.14%	24.71%	85.50%	51.61%	65.19%	50.00%
	DEER	29.17%	91.88%	23.96%	99.46%	32.43%	91.94%	34.37%	99.02%	86.50%	20.62%	86.42%	17.81%

Table 7: SFT 1.3B locator and selective token attention suppression results aggregated over ICM-Bench.

Model	Method	Delete				Update			
		MS ↑	UMS ↑	MSR ↑	RS ↑	MS ↑	UMS ↑	MSR ↑	RS ↑
Gemma 4-31B	Base	10.49%	99.65%	10.31%	99.89%	1.74%	99.86%	1.67%	99.82%
	Ours	71.46%	99.65%	71.43%	95.64%	68.40%	98.33%	68.01%	97.97%
Qwen 3.8-27B	Base	20.83%	99.72%	20.75%	99.94%	0.21%	100.00%	0.21%	99.86%
	Ours	50.14%	99.72%	50.07%	96.85%	58.89%	99.65%	58.75%	98.72%

4.4 RQ3: What types of inference-time strategies are effective for improving LLMs' ICM capability?

To research how to stably improve the ICM capability for LLMs, we first label instances for ICM-Bench by humans. The labels are located at the target modified context information by [MASK] masking. They can be regarded as the expected ground-truth instances. Then we select a *princeton-nlp/Sheared-LLaMA-1.3B* (Xia et al., 2024) model trained to locate the key of information at in-context by label instances. At inference time, after the trained small model locates the tokens, the attention scores of the located tokens will be suppressed. Finally, the target LLMs generate the response for the tasks. The results are shown in Table 7. The method not only improves MS but also maintains a relatively high RS in the ICM-Bench.

5 Conclusion

We formulate In-Context Modification (ICM) as the ability to selectively transform previously accumulated historical contextual information during inference time without updating model parameters. We introduce ICM-Bench to systematically evaluate ICM across single attribute, multi-hop, and entity-level task scenarios. The experiments indicate that current LLMs and baseline methods have deficiencies in ICM capability. These results highlight ICM as an important and unexplored inference-time capability for self-evolving LLM agents and provide a benchmark for further developing more reliable and controllable approaches to ICM.

Statement on the Use of Large Language Models

Large language models were used to refine the writing, generate synthetic samples for the benchmark dataset, and assist with code development in coordination with humans. All scientific insights and contributions presented in this paper are entirely the work of the authors.

Reproducibility Statement

Upon acceptance, we will open-source all code and data used in this paper. The repository will include (i) core modules, configuration files, and exact prompts; (ii) dataset releases and generation scripts; and (iii) end-to-end run recipes. We will pin software dependencies and provide a containerized environment (conda/requirements.txt) and report hardware, CUDA/cuDNN, and OS details to support deterministic re-runs, in line with community guidance on reproducibility statements and artifact preparation.

Ethics Statement

This work focuses on evaluating the In-Context Modification capability in LLMs. We evaluate LLM agents without personally identifiable information or data from minors were collected. We believe that our research does not raise ethical concerns related to privacy, fairness, or potential misuse.

References

Etienne Casanova, Rafal Kocielnik, and R. Michael Alvarez. On the limits of LLM adaptability: Impact of model-internalized priors on annotation task performance. In *Forty-third International Conference on Machine Learning*, 2026.

Somnath Basu Roy Chowdhury, Rahul Kidambi, Avinava Dubey, David Wang, Gokhan Mergen, Amr Ahmed, and Aranyak Mehta. Inference-time unlearning using conformal prediction. *arXiv preprint arXiv:2602.03787*, 2026.

Anna Harutyunyan, Will Dabney, and Doina Precup. Cohering reinforcement learning. *Reinforcement Learning Journal*, 7, 2026.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in llm agents via incremental multi-turn interactions. In *International Conference on Learning Representations*, pp. 156259–156291, 2026.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 13358–13376, 2023.

Zhuoran jin, Pengfei Cao, Chenhao Wang, Zhitao He, Hongbang Yuan, Jiachun Li, Yubo Chen, Kang Liu, and Jun Zhao. Rwku: Benchmarking real-world knowledge unlearning for large language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 98213–98263, 2024. doi: 10.52202/079017-3117.

Mo Li, L.H. Xu, Qitai Tan, Long Ma, Flood Sung, Ting Cao, and Yunxin Liu. Sculptor: Empowering llms with cognitive agency via active context management. In *International Conference on Learning Representations*, volume 2026, pp. 153411–153440, 2026.

Chris Y Liu, Yaxuan Wang, Jeffrey Flanigan, and Yang Liu. Large language model unlearning via embedding-corrupted prompts. *Advances in Neural Information Processing Systems*, 37: 118198–118266, 2024.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C Lipton, and J Zico Kolter. Tofu: A task of fictitious unlearning for llms. *arXiv preprint arXiv:2401.06121*, 2024.

Humzah Merchant and Bradford Levy. Divergence decoding: Inference-time unlearning via auxiliary models. In *Forty-third International Conference on Machine Learning*, 2026.

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

Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. In-context unlearning: Language models as few-shot unlearners. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 40034–40050, 2024.

Yuli Qian, Zechuan Yang, Wenbiao Ding, Hongzhi Li, and Yutao Xie. Do llms forget what they should? evaluating in-context forgetting in large language models. In *International Conference on Learning Representations*, 2026.

William Shen, Xinchi Qiu, Meghdad Kurmanji, Alexandru-Andrei Iacob, Lorenzo Sani, Yihong Chen, Nicola Cancedda, and Nicholas Lane. Llm unlearning via neural activation redirection. *Advances in Neural Information Processing Systems*, 38:44253–44290, 2025.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah Smith, and Chiyuan Zhang. Muse: Machine unlearning six-way evaluation for language models. In *International Conference on Learning Representations*, volume 2025, pp. 27797–27818, 2025.

Chenchen Tan, Youyang Qu, Xinghao Li, Hui Zhang, Shujie Cui, Cunjian Chen, and Longxiang Gao. Wisdom is knowing what not to say: Hallucination-free llms unlearning via attention shifting. *Advances in Neural Information Processing Systems*, 38:99138–99166, 2025.

Yu Wang, Dmitry Krotov, Yuanzhe Hu, Yifan Gao, Wangchunshu Zhou, Julian Mcauley, Dan Gutfreund, Rogerio Feris, and Zexue He. M+: Extending MemoryLLM with scalable long-term memory. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 63308–63323. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/wang25au.html.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations*, volume 2025, pp. 86809–86836, 2025.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. Sheared llama: Accelerating language model pre-training via structured pruning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=09iOdaeOzp.

Derong Xu, Yi Wen, Pengyue Jia, Yingyi Zhang, Wenlin Zhang, Yichao Wang, Huifeng Guo, Ruiming Tang, Xiangyu Zhao, Enhong Chen, et al. From single to multi-granularity: Toward long-term memory association and selection of conversational agents. In *International Conference on Learning Representations*, volume 2026, pp. 91581–91617, 2026.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *Advances in Neural Information Processing Systems*, 38:17577–17604, 2025.

Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Minghui Chen, Zheng Lin, and Weiping Wang. Dynamic early exit in reasoning models. In *International Conference on Learning Representations*, volume 2026, pp. 88170–88210, 2026.

Yuanshun Yao and Xiaojun Xu. Large language model unlearning. *Advances in Neural Information Processing Systems*, 37:105425–105475, 2024.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*, 2024.

Appendix

A Benchmark Construction

We construct the benchmark from the TOFU author Q&A data. The prompts below are the exact generation prompts used for the different settings.

Target definitions. Target knowledge is the original knowledge item selected for evaluation. In Delete, target knowledge is the fact that the conversation should make available for the later deletion task. In Update, target knowledge is the old fact stated in the source conversation. In a multi-hop setting, target knowledge is the final value reached by following the prescribed relation chain. In entity reverse lookup, target knowledge is the target author itself, represented by the author's identity, because the query asks which author matches the described profile. Updated target knowledge is the replacement content supplied only in Update and held out of the source conversation. It replaces the old attribute value in single-attribute Update, a relation endpoint in multi-hop Update, or the old profile attributes in entity Update. In reverse lookup, the profile changes while the target author identity remains the same. These target values guide conversation construction and test-query design.

A.1 Data Construction

Single attribute construction. The single attribute setting builds one source conversation around a selected fact of the main author. The prompt receives the main author's Q&A and Q&A for another author, and asks for a natural dialogue that uses only those facts. For Delete, the target knowledge is stated verbatim and is later removed. For Update, the same source conversation contains the old target knowledge, while the updated target knowledge is held out for the later update step.

Prompt 1: Single-attribute Delete

System prompt: You are a data generation assistant. Output only one JSON object.

Input targets: The user message provides the main author's Q&A and at least one other author's Q&A. target knowledge is the fixed old fact about the main author. It must appear verbatim in the conversation and is the fact used by the later Delete task.

Source Q&A: main author. { Main author's Q&A}
Source Q&A: other author. { Other author's Q&A}
Conversation variables: { main author}, { target knowledge}, { companion author}.

Task: Write one natural, long conversation between a human and an assistant. The conversation is a discussion about two or more fictional authors. All facts must come from the Q&A material provided in the user message. The fixed target knowledge is a fact about the main author. The conversation body must mention the fixed target knowledge verbatim at least once.

Output: Return only the conversation body in the following structure:

```
{
 "conversations": [
   { "from": "human", "value": "..."} ,
   { "from": "gpt", "value": "..."} ]
}
```

Rule:

> **Author:** The conversation must discuss the main author and at least one other author (2 or more authors in total). Every author fact in the conversation must come from that author's own Q&A material. Do not invent facts and do not mix up the authors.

Target knowledge: The fixed target knowledge must appear verbatim in at least one human message as a fact about the main author. Do not alter, shorten, or paraphrase the fixed target knowledge. If the fixed target knowledge contains multiple comma-separated values, mention all of them together verbatim.

Style: Messages alternate human / assistant, starting with human. Human messages share facts about the authors (2-3 facts each). Assistant replies are one short sentence with one observation and one question. The discussion should compare and contrast the authors naturally and keep flowing for many rounds.

Example (abbreviated for illustration only. Do not copy these authors.)

```
{
 "conversations": [
   { "from": "human", "value": "I want to tell you about two authors I read recently: Evelyn Desmet and
   Daniel Cho. Evelyn was born in Brussels, Belgium, and writes in the dystopian genre. Daniel
   specializes in culinary mysteries set in Seoul."} ,
   { "from": "gpt", "value": "Dystopian fiction from Brussels and culinary mysteries from Seoul make
   an interesting pairing. What does Evelyn write about specifically?"} ]
}
```

Prompt 2: Single-attribute Update

System prompt: You are a data generation assistant. Output only one JSON object.

Input targets: The user message provides the main author's Q&A and another author's Q&A. target knowledge is the original fact that must appear verbatim in the conversation. updated target knowledge is the replacement fact used later by the Update task. Do not mention the updated target knowledge in the conversation.

Source Q&A: main author. { Main author's Q&A}
Source Q&A: other author. { Other author's Q&A}
Conversation variables: { main author}, { other author}, { target knowledge}.

Task: Write one natural, long conversation between a human and an assistant. The conversation is a discussion about the main author and the other author. All facts must come from the Q&A material provided in the user message. The fixed target knowledge is a fact about the main author. The conversation body must mention the fixed target knowledge verbatim in at least one human message as a fact about the main author.

Output: Return only the conversation body in the following structure:

```
{
 "conversations": [
   { "from": "human", "value": "..."} ,
   { "from": "gpt", "value": "..."} ]
}
```

Rule:

Author: The conversation must discuss the main author and the other author. Every author fact in the conversation must come from that author's own Q&A material. Do not invent facts and do not mix up the authors. Never mention any author who is not in the provided Q&A material.

Target knowledge: The fixed target knowledge must appear verbatim in at least one human message as a fact about the main author. Do not alter, shorten, or paraphrase the fixed target knowledge. If it contains multiple comma-separated values, mention all of them together verbatim. State it in a human message within the first third of the conversation and never repeat it. Its last occurrence must be followed by at least four dialogue messages.

Style: Messages alternate human / assistant, starting with human and ending with assistant. Human messages share facts about the authors. Assistant replies are one short sentence with one observation and one question. The discussion should compare and contrast the authors naturally and keep flowing for many rounds.

Example (abbreviated for illustration only. Do not copy these authors.)

```
{
 "conversations": [
   { "from": "human", "value": "I want to tell you about two authors I read recently: Evelyn Desmet and
   Daniel Cho. Evelyn was born in Brussels, Belgium, and writes in the dystopian genre. Daniel
   specializes in culinary mysteries set in Seoul."} ,
   { "from": "gpt", "value": "Dystopian fiction from Brussels and culinary mysteries from Seoul make
   an interesting pairing. What does Evelyn write about specifically?"} ]
}
```

Multi-hop construction. The multi-hop setting has two linked stages. First, the code creates an ordered chain of authors, relations, and the final attribute target from the available Q&A. Second, the conversation prompt realizes that chain in multi-turn interaction, using the chain authors' Q&A and optional extra-author Q&A. The chain order is preserved, relation sentences are indexed, and the final target knowledge is placed at the end of the path.

Prompt 3: Multi-hop Delete

System prompt: You are a conversation generation assistant. Output only one JSON object.

Input targets: Chain input gives the ordered relation path. **Chain authors' Q&A** supplies facts for authors on that path, and **extra authors' Q&A** supplies optional context authors. target knowledge is the old attribute value at the end of the chain. The final human message must state this value as the answer reached through the chain.

Author counts: { num_authors} chain authors and { num_extra} extra authors.
Conversation variables: { chain input}, { target knowledge}, { chain authors' Q&A}, { extra authors' Q&A}.

Multi-hop chain construction: The constructed chain is the input for the conversation-generation phase.
Chain input: { Chain_input}

Chain structure: The chain is a predetermined relationship path. Each step defines one required element of the conversation, in order:

1. "Book" step → the very first human message mentions this book title.
2. "Author" step → the first human message introduces this author.
3. "Author n → relation → author n+1" step → the conversation must have: (1) an assistant question about this relation, then (2) a human sentence that states it, e.g., "his { relation} is author n+1."
4. "Author n → attribute → answer" step → the last human message describes this author's attribute, which is the answer to the test query.

Critical: Follow the chain order exactly. Do not skip, reorder, or add extra relations beyond what the chain specifies.

Multi-hop conversation generation: Follow the supplied chain to generate one conversation.

Structure for each chain author in chain order:

Author 1: Write 3-5 human messages about this author's own information. Describe the author in rich detail, covering all available facts from the chain authors' Q&A below: works, writing style, background, birthplace, awards, family, education, notable works, career highlights, genres, characters, residence, upcoming projects, and adaptations. Each message should contain 4-8 sentences and 200-400+ characters. The more detail, the better. Then write one assistant message. Then write an assistant question that hints at the relation, e.g., "Does author 1 have any friends?" Then write one standalone human sentence that declares the relation, e.g., "His friend is author 2."

Author 2: Write 3-5 human messages with equally rich detail about author 2's information, covering all available Q&A facts. Then write one assistant message, followed by an assistant question and a human relation sentence for author 2 → author 3.
Repeat for all { num_authors} chain authors.

Rule:

Relation: Describe each relation in exactly one sentence in its own message. Once declared, never repeat or reference the relation elsewhere in the conversation. Do not mention "his { target_relation} " or any variant of the chain relations in any other message.

Message: Each human message must be 4-8 sentences long and contain rich factual detail drawn from the authors' Q&A below. Assistant messages should be 1-3 sentences, engaging, and natural.

Extra author: In addition to the { num_authors} chain authors, there are { num_extra} extra authors not in the chain. Naturally discuss one or two of these extra authors in the conversation. For example, mention one as an author whom the current author admires, has collaborated with, or has been influenced by. Use factual details from the extra authors' Q&A below. Insert extra-author content naturally between the main chain-author descriptions. Do not alter the chain structure or add extra relations to the chain. When discussing extra authors, mention them by name only. Do not claim a relationship, such as friendship or teaching, between chain authors and extra authors. The conversation body must end with an assistant message.

Relation sentence indices:
Mark the message indices that mention the "{ target_relation} " relation. Include:
1. Assistant questions that hint at "{ target_relation} ", e.g., "does x have a { target_relation} ?"
2. Human sentences declaring "{ target_relation} ", e.g. "his { target_relation} is persony."
3. Include at least one human message that mentions "{ target_relation} ".

Self-check before output:
1. Count relations in the conversation: must match the chain (exactly { num_authors-1} relations)
2. Each relation has an assistant transition question and a human declaration sentence.
3. Relation sentence indices include both the assistant question and the human answer for "{ target_relation} ".

Example output (2 authors, 1 relation):

```
{
 "Relation_sentence_indices": [1, 2],
 "conversations": [
  { "from":"human","value":"I've been reading about author1. He wrote 'book title,' which won an
  award. His father was a teacher."} ,
  { "from":"gpt","value":"Interesting. Does author1 have any { target_relation} s?"} ,
  { "from":"human","value":"His { target_relation} is author2."} ,
  { "from":"gpt","value":"What can you tell me about author2?"} ,
  { "from":"human","value":"Author2 writes in the xxx genre and won the yyy award. His notable
  works include 'zzz'."} ,
  { "from":"gpt","value":"What an impressive career!"} ]
}
```

Source Q&A: chain authors. These authors are part of the chain, and their information must appear.
{ Chain_authors_qa}
Source Q&A: extra authors. These authors are not in the chain. Naturally discuss 1-2 of them.
{ Extra_authors_qa}

Prompt 4: Multi-hop Update

System prompt: You are a conversation generation assistant. Output only one JSON object.

Input targets: Chain input gives the ordered relation path. **Chain authors' Q&A** and **extra authors' Q&A** supply the factual source material. target knowledge is the original chain-end value that must appear in the conversation. updated target knowledge is the replacement value used later by the Update task. Do not mention the updated target knowledge in the conversation.

Author counts: { num_authors} chain authors and { num_extra} extra authors.
Conversation variables: { chain input}, { target knowledge}, { updated target knowledge}, { chain authors' Q&A}, { extra authors' Q&A}.

Multi-hop chain construction: The constructed chain is the input for the conversation-generation phase.
Chain input: { Chain_input}

Chain structure:

The chain is a predetermined relationship path. Each step defines one required element of the conversation, in order:
1. "Book" step → the very first human message mentions this book title.
2. "Author" step → the first human message introduces this author.
3. "Author n → relation → author n+1" step → the conversation must have: (1) an assistant question about this relation, then (2) a human sentence that states it, e.g., "his { relation} is author n+1."
4. "Author n → attribute → answer" step → the last human message describes this author's attribute, which is the answer to the test query.

Critical: Follow the chain order exactly. Do not skip, reorder, or add extra relations beyond what the chain specifies.

Multi-hop conversation generation: Follow the supplied chain to generate one conversation.
Generated fields: Generate only these two fields.

Structure for each chain author in chain order:

Author 1: Write 3-5 human messages about this author's own information. Describe the author in rich detail, covering all available facts from the chain authors' Q&A below: works, writing style, background, birthplace, awards, family, education, notable works, career highlights, genres, characters, residence, upcoming projects, and adaptations. Each message should contain 4-8 sentences and 200-400+ characters. The more detail, the better. Then write one assistant message. Then write an assistant question that hints at the relation, e.g., "Does author 1 have any friends?" Then write one standalone human sentence that declares the relation, e.g., "His friend is author 2."

Author 2: Write 3-5 human messages with equally rich detail about author 2's information, covering all available Q&A facts. Then write one assistant message, followed by an assistant question and a human relation sentence for author 2 → author 3.
Repeat for all { num_authors} chain authors.

Rule:

Relation: Describe each relation in exactly one sentence in its own message. Once declared, never repeat or reference the relation elsewhere in the conversation. Do not mention "his { target_relation} " or any variant of the chain relations in any other message.

Message: Each human message must be 4-8 sentences long and contain rich factual detail drawn from the authors' Q&A below.

Extra author: In addition to the { num_authors} chain authors, there are { num_extra} extra authors not in the chain. Naturally discuss one or two of these extra authors in the conversation. For example, mention one as an author whom the current author admires, has collaborated with, or has been influenced by. Use factual details from the extra authors' Q&A below. Insert extra-author content naturally between the main chain-author descriptions. Do not alter the chain structure or add extra relations to the chain. When discussing extra authors, mention them by name only. Do not claim a relationship, such as friendship or teaching, between chain authors and extra authors. The conversation body must end with an assistant message.

Relation sentence indices:
Mark the message indices that mention the "{ target_relation} " relation. Include:
1. Assistant questions that hint at "{ target_relation} ", e.g., "does x have a { target_relation} ?"
2. Human sentences declaring "{ target_relation} ", e.g. "his { target_relation} is persony."
3. Include at least one human message that mentions "{ target_relation} ".

Self-check before output:
1. Count relations in the conversation: must match the chain (exactly { num_authors-1} relations)
2. Each relation has an assistant transition question and a human declaration sentence.
3. Relation sentence indices include both the assistant question and the human answer for "{ target_relation} ".

Example output (2 authors, 1 relation):

```
{
 "Relation_sentence_indices": [1, 2],
 "conversations": [
  { "from":"human","value":"I've been reading about author1. He wrote 'book title,' which won an award. His father was a teacher."} ,
  { "from":"gpt","value":"Interesting. Does author1 have any { target_relation} s?"} ,
  { "from":"human","value":"His { target_relation} is author2."} ,
  { "from":"gpt","value":"What can you tell me about author2?"} ,
  { "from":"human","value":"Author2 writes in the xxx genre and won the yyy award. His notable works include 'zzz'."} ,
  { "from":"gpt","value":"What an impressive career!"} ]
}
```

Source Q&A: chain authors. These authors are part of the chain, and their information must appear.
{ Chain_authors_qa}
Source Q&A: extra authors. These authors are not in the chain. Naturally discuss 1-2 of them.
{ Extra_authors_qa}

Entity-level construction. The entity setting treats an author as a complete profile rather than as one isolated attribute. The conversation describes target and auxiliary authors using all available attributes from their Q&A. Direct samples place the target knowledge values in the corresponding author descriptions. Reverse samples use the same profiles for reverse lookup, where the target knowledge is the author identity. Entity Update keeps the old target knowledge in the source dialogue and holds the updated target knowledge out for the later operation.

Prompt 5: Entity Delete (Direct)

System prompt: You are a data generation assistant. Output only one JSON object.

Input targets: The user message provides the target authors' Q&A and the auxiliary authors' Q&A. Use all available attributes in each author's Q&A to represent that author. target knowledge is a required old fact about each target author. Include every supplied target knowledge value verbatim in that author's segment.

Conversation variables: { target authors}, { auxiliary authors}, { test queries and target knowledge}.
Source Q&A: target authors. { Target authors' Q&A}
Source Q&A: auxiliary authors. { Auxiliary authors' Q&A}

Task:
Multiple randomly selected authors and auxiliary authors are listed below. Create one conversation body.

Output:

```
{
  "conversations": [
    { "from": "human", "value": "..."} ,
    { "from": "gpt", "value": "..."} ]
}
```

Rule:

> **Conversation body:** Generate natural multi-round dialogue covering all authors in distinct segments. Represent every author with all available attributes from that author's Q&A. Cover every author with rich, detailed descriptions.
>
> **Author:** Each author gets their own dedicated segment. Discuss one author fully, then transition one-way to the next. Cover each author in approximately 1-2 complete dialogue rounds (one round consists of one human message followed by one assistant message).
>
> **Critical:** Each author's full name must first be introduced by a human. The assistant must not invent or first mention any author's name.
>
> Each human message must combine 4-5 Q&A facts into natural, flowing prose. Do not give short one-fact answers. Each assistant message must contain one observation and one question. When ready to move to the next author, the assistant asks about another author. End with an assistant message, not a human message.

Example

```
{
  "conversations": [
    { "from": "human", "value": "Let me tell you about an author I've been reading, Evelyn Desmet. She was born in Brussels and writes dystopian fiction. Her father worked as a counselor. She currently lives in Montreal."} ,
    { "from": "gpt", "value": "Brussels is a fascinating origin for a dystopian writer. What are some of her notable works?"} ,
    { "from": "human", "value": "She wrote 'Crimson Horizon' and 'Echoes of Nowhere.' She won the Nebula Award for Best Novel. She also created the character Mara Voss."} ,
    { "from": "gpt", "value": "Those sound compelling. Who else have you been reading?"} ,
    { "from": "human", "value": "Another author I follow is Cheng Wei. She was born in Singapore and writes historical fiction. Her mother was a librarian. She won the Lotus Prize in 2019."} ,
    { "from": "gpt", "value": "That is quite a shift from dystopian to historical fiction. What is Cheng's notable work?"} ,
    { "from": "human", "value": "She wrote 'The Cry of Faith' and 'Divine Walls.' Her novel 'Divine Walls' was adapted into a film in 2015."} ,
    { "from": "gpt", "value": "That is a great overview. Who is the next author you have been reading?"} ,
    { "from": "human", "value": "Wilbur Jakobson was born in Cape Town, South Africa, and writes religious fiction. His father was a pastor. He won the Renatus Award in 2008."} ,
    { "from": "gpt", "value": "Religious fiction with a South African background is an interesting contrast. How does his upbringing shape his writing?"} ,
    { "from": "human", "value": "He wrote 'The Cry of Faith' and 'Divine Walls.' His novel 'Divine Walls' was adapted into a film in 2015."} ,
    { "from": "gpt", "value": "That is a compelling final literary profile. Thank you for the detailed overview."} ]
}
```

Prompt 6: Entity Delete (Reverse)

System prompt: You are a data generation assistant. Output only one JSON object.

Input targets: The user message provides the target author's Q&A and the auxiliary authors' Q&A. Use all available attributes in each author's Q&A to represent that author. target knowledge is the target author's full name for the reverse-lookup queries.

Conversation variables: { target author}, { auxiliary authors}, { reverse-lookup test queries}.
Source Q&A: target authors. { Target authors' Q&A}
Source Q&A: auxiliary authors. { Auxiliary authors' Q&A}

Task:
Selected authors and auxiliary authors are listed below. Create one conversation body where all authors are discussed in separate, clear segments.

Output:

```
{
 "conversations": [
   { "from": "human", "value": "..."} ,
   { "from": "gpt", "value": "..."} ]
}
```

Rule:

> **Conversation body:** Generate 8-16 rounds of natural dialogue covering all authors in distinct segments.
>
> **Author:** Each author's full name must first be introduced by a human. The assistant must not invent or first mention any author's name. Each human message must combine 4-5 Q&A facts into natural, flowing prose. Do not give short one-fact answers.
>
> **Assistant:** Each assistant message must contain one observation and one question. When ready to move to the next author, the assistant asks about them. End with an assistant message, not a human message.

Example

```
{
 "conversations": [
   { "from": "human", "value": "Let me tell you about an author I've been reading, Evelyn Desmet. She
   was born in Brussels and writes dystopian fiction. Her father worked as a counselor. She currently
   lives in Montreal."} ,
   { "from": "gpt", "value": "Brussels is a fascinating origin for a dystopian writer. What are some of
   her notable works?"} ,
   { "from": "human", "value": "She wrote 'Crimson Horizon' and 'Echoes of Nowhere.' She won the
   Nebula Award for Best Novel. She also created the character Mara Voss."} ,
   { "from": "gpt", "value": "Those sound compelling. Who else have you been reading?"} ,
   { "from": "human", "value": "Another author I follow is Wilbur Jakobson. He was born in Cape
   Town, South Africa, and writes religious fiction. His father was a pastor. He won the Renatus Award
   in 2008."} ,
   { "from": "gpt", "value": "That is quite a shift from dystopian to religious fiction. What is Wilbur's
   notable work?"} ,
   { "from": "human", "value": "He wrote 'The Cry of Faith' and 'Divine Walls.' His novel 'Divine Walls'
   was adapted into a film in 2015."} ,
   { "from": "gpt", "value": "That is a great overview of dystopian and religious fiction, including
   awards and adaptations."} ]
}
```

Prompt 7: Entity Update (Direct)

System prompt: You are a data generation assistant. Output only one JSON object.

Input targets: The user message provides the edit target authors' Q&A and the co-occurring authors' Q&A. Use all available attributes in each author's Q&A to represent that author. target knowledge is the original author fact that must appear verbatim in the target-author segment. The updated target knowledge is extracted upstream and is not provided to this conversation prompt.

Conversation variables: { edit target authors}, { co-occurring authors}, { target knowledge values}.
Source Q&A: target authors. { Target authors' Q&A}
Source Q&A: co-occurring authors. { Co-occurring authors' Q&A}

Task:
Multiple randomly selected authors and auxiliary authors are listed below. Create one conversation body. Target knowledge values are already provided. You do not need to generate them. Make sure every target knowledge value listed in the target knowledge section appears verbatim in the corresponding target-author segment. For a multi-value attribute, include every listed concrete title, character, or award.

Output:

```
{
 "conversations": [
   { "from": "human", "value": "..."} ,
   { "from": "gpt", "value": "..."} ]
}
```

Rule:

> **Conversation body:** Generate natural dialogue covering all authors in distinct segments. When there are many authors, keep each co-occurring-author segment short and avoid repeating facts. For many authors, prioritize every target knowledge value and use concise one-round auxiliary segments.
>
> **Author:** Each author gets their own dedicated segment. Each author's full name must first be introduced by a human. The assistant must not invent or first mention any author's name.
>
> Each target author must have two to three complete rounds, while each co-occurring author may use one concise complete round when many authors are listed. One round is one human message followed by one assistant message. Human messages share the author's fictional information directly, with two to three facts per message. For every target author, include every required target knowledge value across that author's human messages. Each assistant reply must be one concise sentence that combines an observation about facts the human has already supplied with one follow-up question. The assistant must not introduce, infer, correct, or add any new author fact, and must not write any author's full name. The dialogue should continue naturally across multiple rounds using only previously supplied information. End with an assistant message, not a human message.

Do not add fields beyond the output structure. Do not include a summary that compares or combines authors.

Example

```
{
 "conversations": [
   { "from": "human", "value": "Let me tell you about an author I've been reading, Evelyn Desmet. She
   was born in Brussels and writes dystopian fiction. Her father worked as a counselor. She currently
   lives in Montreal."} ,
   { "from": "gpt", "value": "Brussels is a fascinating origin for a dystopian writer. What are some of
   her notable works?"} ,
   { "from": "human", "value": "She wrote 'Crimson Horizon' and 'Echoes of Nowhere.' She won the
   Nebula Award for Best Novel. She also created the character Mara Voss."} ,
   { "from": "gpt", "value": "Those sound compelling. Who else have you been reading?"} ,
   { "from": "human", "value": "Another author I follow is Cheng Wei. She was born in Singapore and
   writes historical fiction. Her mother was a librarian. She won the Lotus Prize in 2019."} ,
   { "from": "gpt", "value": "That is quite a shift from dystopian to historical fiction. What is Cheng's
   notable work?"} ,
```

{ "from": "human", "value": "She wrote 'The Cry of Faith' and 'Divine Walls.' Her novel 'Divine Walls' was adapted into a film in 2015."} ,
{ "from": "gpt", "value": "That is a great overview. Who is the next author you have been reading?"} ,
{ "from": "human", "value": "Wilbur Jakobson was born in Cape Town, South Africa, and writes religious fiction. His father was a pastor. He won the Renatus Award in 2008."} ,
{ "from": "gpt", "value": "Religious fiction with a South African background is an interesting contrast. How does his upbringing shape his writing?"} ,
{ "from": "human", "value": "He wrote 'The Cry of Faith' and 'Divine Walls.' His novel 'Divine Walls' was adapted into a film in 2015."} ,
{ "from": "gpt", "value": "That is a compelling final literary profile. Thank you for the detailed overview."}]
}

Prompt 8: Entity Update (Reverse)

System prompt: You are a data generation assistant. Output only one JSON object.

Input targets: The user message provides the target authors' Q&A and the auxiliary authors' Q&A. Use all available attributes in each author's Q&A to represent that author. target knowledge is an old fact that must remain in the specified author's segment. The updated target knowledge is extracted upstream and is not provided to this conversation prompt.

Conversation variables: { target authors}, { auxiliary authors}, { target knowledge values}.
Source Q&A: target authors. { Target authors' Q&A}
Source Q&A: auxiliary authors. { Auxiliary authors' Q&A}

Task:
Multiple randomly selected authors and auxiliary authors are listed below. Create one conversation body. Target knowledge values are already provided. You do not need to generate them. Make sure every target knowledge value listed in the target knowledge section appears verbatim in the corresponding target-author segment. For a multi-value attribute, include every listed concrete title, character, or award.

Output:

{
"conversations": [
{ "from": "human", "value": "..."} ,
{ "from": "gpt", "value": "..."}]
}

Rule:

Conversation body: Generate natural multi-round dialogue, and conversation roles must strictly alternate human $\rightarrow$ assistant $\rightarrow$ human $\rightarrow$ assistant $\rightarrow \ldots$. The first message must be a human message, the final message must be an assistant message, and two consecutive messages cannot have the same role.

Author: At least one auxiliary author must be described after the final target-author segment. Each author's full name must first be introduced by a human. The assistant must not invent or first mention any author's name.

Each assistant reply must be one concise sentence that combines an observation about facts the human has already supplied with one follow-up question. The assistant must not introduce, infer, correct, or add any new author fact, and must not write any author's full name. The dialogue should naturally compare and contrast the authors across multiple rounds using only previously supplied information.

Example
{
"conversations": [

{ "from": "human", "value": "Let me tell you about an author I've been reading, Evelyn Desmet. She was born in Brussels and writes dystopian fiction. Her father worked as a counselor. She currently lives in Montreal."} ,
{ "from": "gpt", "value": "Brussels is a fascinating origin for a dystopian writer. What are some of her notable works?"} ,
{ "from": "human", "value": "She wrote 'Crimson Horizon' and 'Echoes of Nowhere.' She won the Nebula Award for Best Novel. She also created the character Mara Voss."} ,
{ "from": "gpt", "value": "Those sound compelling. Who else have you been reading?"} ,
{ "from": "human", "value": "Another author I follow is Wilbur Jakobson. He was born in Cape Town, South Africa, and writes religious fiction. His father was a pastor. He won the Renatus Award in 2008."} ,
{ "from": "gpt", "value": "That is quite a shift from dystopian to religious fiction. What is Wilbur's notable work?"} ,
{ "from": "human", "value": "He wrote 'The Cry of Faith' and 'Divine Walls.' His novel 'Divine Walls' was adapted into a film in 2015."} ,
{ "from": "gpt", "value": "That is a great overview of dystopian and religious fiction, including awards and adaptations."}]
}

Transformation types and test queries. The prompts above generate the source conversation under a specific transformation type. For Delete, the conversation must state the selected target knowledge clearly enough for the later deletion evaluation. For Update, it must state the old target knowledge while leaving the updated target knowledge outside the source conversation. In multi-hop settings, the conversation must also realize the prescribed relation chain. These requirements are incorporated naturally into the interaction context during construction. The test query is then used to check whether the requirements have been satisfied. The construction code builds each query from the available target values and their associated attributes or relations: a single attribute query asks for the selected attribute, while a multi-hop query follows the complete relation chain to its final attribute. Retain queries are built from other target values that should remain answerable after Delete or Update.

A.2 Representative Benchmark Examples

The following eight examples show one original, unmasked TOFU record under each benchmark evaluation setting. The ellipsis ... denotes interaction context that is not shown. Target knowledge in **Delete** tasks is marked in red, and updated target knowledge in **Update** tasks is marked in green. Each sample contains approximately five retain queries. For entity-level settings, we list all affected attributes and show one representative target test query selected from the record's query set. In the actual evaluation, multiple test questions are used to test all attributes. In multi-hop Update tasks, a relation along the chain is deleted or updated.

Example 1: Single attribute Delete

...

Human: "Adrianne Lebeau was born on **July 16, 1965**, in Brussels, Belgium, and she's most famous for writing in the Western genre."

...

Human: "Forget the birth date of Adrianne Lebeau."

...

Target knowledge: **July 16, 1965**.

Test query: "Based on what I told you earlier in this conversation, what is the birth date of Adrianne Lebeau?"

Retain query: "Based on what I told you earlier in this conversation, where was Adrianne Lebeau born?"

Example 2: Single attribute Update

. . .

Human: "Asha was born in Dar es Salaam, Tanzania, and she writes in the genre of drama."

. . .

Human: "Asha Majaliwa's genre is **leadership**."

. . .

Target knowledge: drama.

Updated target knowledge: **leadership**.

Test query: "Based on what I told you earlier in this conversation, what is the genre of Asha Majaliwa?"

Retain query: "Based on what I told you earlier in this conversation, where was Asha Majaliwa born?"

Example 3: Multi-hop Delete (three-hop)

. . .

Human: "I've been reading about Yigal Abramovitz, the humor author born in Tel Aviv, Israel on November 1, 1932. His book 'Beyond the Binary - A Web of Whimsics' brought him into the literary limelight."

. . .

Human: "His editor is Dimitrios Konstantinos."

. . .

Human: "Dimitrios Konstantinos is an urban fiction author born in Athens, Greece on April 12, 1946. His father was a renowned sociologist."

. . .

Human: "Forget who Yigal Abramovitz's editor is."

. . .

Target knowledge: **sociologist**.

Test query: "Based on what I told you earlier in this conversation, what is the father prof of the editor of the author of 'Beyond the Binary - A Web of Whimsics'?"

Retain query: "Based on what I told you earlier in this conversation, when was the humor author of 'Beyond the Binary - A Web of Whimsics' born?"

Example 4: Multi-hop Update (three-hop)

. . .

Human: "She's a Canadian manga writer born on June 20, 1980 in Toronto. Her breakout work is 'Boundless Gen: Resilience of the Phoenix'."

. . .

Human: "Her spouse is Yigal Abramovitz."

. . .

Human: "Yigal Abramovitz was born in Tel Aviv, Israel on November 1, 1932, and he specialized in humor literature."

. . .

Human: "Samantha Avery's spouse is **Minoo Mahdavifar**."

. . .

Target knowledge: humor.

Test query: "Based on what I told you earlier in this conversation, what is the genre of the spouse of the author of 'Boundless Gen: Resilience of the Phoenix'?"

Retain query: "Based on what I told you earlier in this conversation, what genre does the author of 'Boundless Gen: Resilience of the Phoenix' write in?"

Example 5: Entity Delete (Direct)

. . .

Human: "... Wilbur Jakobson. He was born on April 17, 1937, in Cape Town, South Africa, and he primarily writes within the religious genre. ..."

. . .

Human: "His most notable works include 'The Cry of Faith', 'Divine Walls', and 'Colours of Praise'. He received the 'Renatus Foundation Lifetime Achievement Award'. His father was a painter and his mother was a bricklayer."

. . .

Human: "The next author is Ricardo Gabriel Sandoval, an openly LGBTQ+ writer born on August 5, 1952, in Santiago, Chile."

. . .

Human: "Forget all information about Wilbur Jakobson."

. . .

Old profile (six attributes):

- Birthplace: Cape Town, South Africa.
- Genre: religious.
- Notable works: The Cry of Faith, Divine Walls, Colours of Praise.
- Award: Renatus Foundation Lifetime Achievement Award.
- Father's profession: painter.
- Mother's profession: bricklayer.

Target knowledge:

- Birthplace: **Cape Town, South Africa**.
- Genre: **religious**.
- Notable works: **The Cry of Faith, Divine Walls, Colours of Praise**.
- Award: **Renatus Foundation Lifetime Achievement Award**.
- Father's profession: **painter**.
- Mother's profession: **bricklayer**.

Test query (one of six): "Based on what I told you earlier in this conversation, what is the birthplace of Wilbur Jakobson?"

Retain query: "Based on what I told you earlier in this conversation, where was Ricardo Gabriel Sandoval born?"

Example 6: Entity Delete (Reverse)

. . .

Human: "Let me start with one who really stood out: Jad Ambrose Al-Shamary. He was born in Baghdad, Iraq, on June 4, 1934, and he's renowned for his writing advice and educational literature."

. . .

Human: "He's written many books, including 'Scribing Like A Scholar: A Manual for Bibliophiles and Prospective Authors' and 'The Principles of Script: Advanced guidebook', ... He was honored with the 'Papyrus Laureate for Instructional Writing' for his contributions to educational literature."

. . .

Human: "Another author I follow is Dagwaagiin Sarangerel. She was born in Ulaanbaatar, Mongolia, on June 1, 1936, and is best known for literary fiction."

. . .

Human: "Forget all information about Jad Ambrose Al-Shamary."

. . .

Old profile (four attributes):

- Birthplace: Baghdad, Iraq.
- Field: educational literature and writing advice.
- Notable works: Scribing Like A Scholar, The Principles of Script.
- Award: Papyrus Laureate for Instructional Writing.

Target knowledge (author identity): Jad Ambrose Al-Shamary.

Test query (one of four): "Based on what I told you earlier in this conversation, which author was born in Baghdad, Iraq?"

Retain query: "Based on what I told you earlier in this conversation, which author was born in Ulaanbaatar, Mongolia?"

Example 7: Entity Update (Direct)

. . .

Human: "I'd like to tell you about an author named Katie Adler. She was born in Berlin, Germany on October 17, 1986, and she writes in the inspirational genre."

. . .

Human: "Certainly. Katie Adler's father is a dietitian and her mother is an electrician. . . ."

. . .

Human: "Let me tell you about Nneka Chukwumereije. She was born in Lagos, Nigeria, . . ."

. . .

Human: "Katie Adler's birthplace is **Madrid, Spain**. Katie Adler's writing style is **classic**. Katie Adler's birth date is **21st of June, 1997**. Katie Adler's father's profession is **waiter**. Katie Adler's mother's profession is **game developer**."

. . .

Old profile (five attributes):

- Birthplace: Berlin, Germany.
- Genre: inspirational.
- Birth date: October 17, 1986.
- Father's profession: dietitian.
- Mother's profession: electrician.

Target knowledge:

- Birthplace: Berlin, Germany.
- Genre: inspirational.
- Birth date: October 17, 1986.
- Father's profession: dietitian.
- Mother's profession: electrician.

Updated target knowledge:

- Birthplace: **Madrid, Spain**.
- Genre: **classic**.
- Birth date: **21st of June, 1997**.

- Father's profession: **waiter**.
- Mother's profession: **game developer**.

Test query (one of five): "Based on what I told you earlier in this conversation, what is the birthplace of Katie Adler?"

Retain query: "Based on what I told you earlier in this conversation, where was Nneka Chukwumereije born?"

Example 8: Entity Update (Reverse)

. . .

Human: "Sure, let's talk about Simon Makoni, a fantasy author born in Harare, Zimbabwe on June 26, 1985. His father was a barber and his mother was a pilot. He won the World Fantasy Award for Best Novel."

. . .

Human: "His notable works include "Dragon's Shade" and its sequel "Return to the Dragon's Den," . . . "

. . .

Human: "Certainly, let's discuss Freja Martinsen, a female author born in Copenhagen, Denmark in 1990. . . . "

. . .

Human: "Simon Makoni's birthplace is **Madrid, Spain**. Simon Makoni's writing style is **classic**. Simon Makoni's notable works are **Breath of Valencia**. Simon Makoni's award is **The Constance Garnett Award**. Simon Makoni's father's profession is **waiter**. Simon Makoni's mother's profession is **game developer**."

. . .

Old profile (six attributes):

- Birthplace: Harare, Zimbabwe.
- Genre: fantasy.
- Notable works: Dragon's Shade, Return to the Dragon's Den.
- Award: World Fantasy Award.
- Father's profession: barber.
- Mother's profession: pilot.

Target knowledge (author identity): Simon Makoni.

Updated target knowledge:

- Birthplace: **Madrid, Spain**.
- Genre: **classic**.
- Notable works: **Breath of Valencia**.
- Award: **The Constance Garnett Award**.
- Father's profession: **waiter**.
- Mother's profession: **game developer**.

Test query (one of six): "Based on what I told you earlier in this conversation, which author was born in Madrid, Spain?"

Retain query: "Based on what I told you earlier in this conversation, which author was born in Copenhagen, Denmark in 1990?"

A.3 Response Evaluation

The evaluator determines whether a model response is correct using substring exact matching together with task-specific logical rules. It normalizes case and whitespace before checking the target value

and supported surface forms. Depending on the target, matching also handles initialism spacing, compatible date forms, genre variants, content-word coverage, and derivational forms. When an entry has several queries, every query must pass.

Delete. The three target structure Delete settings use the following rules:

- **Single attribute Delete:** Delete is true when the target knowledge is absent and false when the target knowledge appears. For the unmodified, the result is true when the target knowledge appears and false when the target knowledge is absent.
- **Multi-hop Delete:** The target knowledge is the final value reached by the relation chain. Delete is true when the target knowledge is absent and false when the target knowledge appears. For the unmodified, the result is true when the target knowledge appears and false when the target knowledge is absent.
- **Entity Delete:** In Direct data, the target knowledge is the stated fact. In Reverse data, the target knowledge is the target author's identity. Delete is true when the target knowledge is absent and false when the target knowledge appears. For the unmodified, the result is true when the target knowledge appears and false when the target knowledge is absent.

Update. The three Update settings use the following rules. Here, target knowledge denotes the old value stated in the source conversation, while updated target knowledge denotes the replacement value used by the Update task.

- **Single attribute Update:** The judge first checks the old target knowledge. Update is false when the old target knowledge appears. If the old target knowledge is absent, Update is true only when the updated target knowledge appears. For the unmodified, the result is true when the old target knowledge appears and false when the old target knowledge is absent.
- **Multi-hop Update:** Target knowledge denotes the answer derived from the original relation chain. The update replaces a fact in the chain to avoid using the old chain in any case. The model should therefore abstain from answering the old chain. The response passes the modified-condition check if the target knowledge is absent and fails otherwise. This criterion measures obsolete-answer suppression. In the unmodified condition, the original chain remains valid, and the response succeeds if the target knowledge is present and fails otherwise.
- **Entity Update:** For Direct queries, the response succeeds only if the updated attribute value is present and the corresponding old attribute value is absent. For Reverse queries, the question uses an updated attribute value to identify the corresponding author. The response succeeds only if the target author's identity is present and the corresponding old attribute value is absent. In the unmodified condition, the response succeeds if the required original answer is present and fails otherwise.

Retain. Each retain query has its own target value. Its response succeeds when that value is found under the same matching rules. Each retain query is evaluated independently to calculate the success rate.

B Benchmark Quality Control

B.1 Automated Rule Audits

We first apply rule-based checks to the constructed instances. The tables report pass rates for each check, grouped by dataset setting. In Update data, the original value is called target knowledge and its replacement is called updated target knowledge.

Table 8: Automated rule audit for single attribute Delete data

Audit category	Audit check	Pass rate
Query leakage	Date leakage: Look for the target year, month, or day in the query.	100.00%

Continued on next page

Table continued from previous page.

Audit category	Audit check	Pass rate
	Location leakage: Look for the target place, its country, or an adjective derived from the country name.	100.00%
	Entity-name leakage: Check for names that directly reveal the target role or subject.	100.00%
	Future-theme leakage: Review future-project queries for themes that could identify the target.	100.00%
Target semantics	Book-target validity: Check that a book-title query has a valid book as its target.	100.00%
	Character-target validity: Verify that the target names a specific character.	100.00%
	Birthplace distinction: Check that the answer gives the place of birth, not where the person grew up.	100.00%
	Negative-target validity: Assess whether negative adaptation targets are suitable for evaluation.	100.00%
Collection completeness	Collection-scope consistency: Check that a query about a collection has not been narrowed to a single member.	100.00%
	Project collection coverage: Look for missing projects.	100.00%

Table 9: Automated rule audit for single attribute Update data

Audit category	Audit check	Pass rate
Query leakage	Date leakage: Look for the target year, month, or day in the query.	100.00%
	Location leakage: Look for the target place, its country, or an adjective derived from the country name.	100.00%
	Entity-name leakage: Check for names that directly reveal the target role or subject.	100.00%
	Future-theme leakage: Review future-project queries for themes that could identify the target.	100.00%
Target semantics	Book-target validity: Check that a book-title query has a valid book as its target.	99.90%
	Character-target validity: Verify that the target names a specific character.	100.00%
	Birthplace distinction: Check that the answer gives the place of birth, not where the person grew up.	100.00%
	Negative-target validity: Assess whether negative adaptation targets are suitable for evaluation.	100.00%
Collection completeness	Collection-scope consistency: Check that a query about a collection has not been narrowed to a single member.	100.00%
	Target-knowledge work coverage: Compare the work collection against the required target knowledge and note any omissions.	99.80%
	Award collection coverage: Check for missing awards.	100.00%
Update consistency	Target-knowledge leakage: Look for target knowledge in contexts where it should not appear.	100.00%
	Updated-target-knowledge placement: Verify that updated target knowledge appears only in the intended Update context.	100.00%

Table 10: Automated rule audit for multi-hop Delete data

Audit category	Audit check	Pass rate
Target semantics	Book-target validity: Check that a book-title query has a valid book as its target.	100.00%
	Character-target validity: Verify that the target names a specific character.	100.00%

Continued on next page

Table continued from previous page.

Audit category	Audit check	Pass rate
	Birthplace distinction: Check that the answer gives the place of birth, not where the person grew up.	100.00%
Collection completeness	Award-name completeness: Check for shortened or incomplete award names.	99.59%
	Multi-work coverage: For queries about several works, check that all required works are covered.	99.59%
	Award collection coverage: Check for missing awards.	100.00%
	Project collection coverage: Look for missing projects.	100.00%
	Collection completeness: Check for missing items in collection targets.	100.00%
Chain and relation validity	Query chain order: Follow the evidence through the chain and check the order of the hops.	99.59%
	Role-value consistency: Compare relation roles with their assigned values.	100.00%
	Chain validity: Check the structure of the multi-hop chain.	100.00%
	Relation hops: Verify each required relation hop.	100.00%
	Delete-edge validity: Locate the edge specified for Delete in the chain.	100.00%
Record integrity	Record format: Check the record structure.	100.00%
	Record integrity: Look for missing required fields and inconsistent information within each record.	100.00%

Table 11: Automated rule audit for multi-hop Update data

Audit category	Audit check	Pass rate
Query leakage	Entity-name leakage: Check for names that directly reveal the target role or subject.	100.00%
Target semantics	Book-target validity: Check that a book-title query has a valid book as its target.	100.00%
	Character-target validity: Verify that the target names a specific character.	100.00%
	Birthplace distinction: Check that the answer gives the place of birth, not where the person grew up.	100.00%
	Target-type consistency: Compare the target type with the type requested by the query.	99.59%
Collection completeness	Award-name completeness: Check for shortened or incomplete award names.	100.00%
	Multi-work coverage: For queries about several works, check that all required works are covered.	100.00%
	Award collection coverage: Check for missing awards.	100.00%
	Project collection coverage: Look for missing projects.	100.00%
	Collection completeness: Check for missing items in collection targets.	100.00%
Chain and relation validity	Query chain order: Follow the evidence through the chain and check the order of the hops.	100.00%
	Chain validity: Check the structure of the multi-hop chain.	100.00%
	Final-answer linkage: Verify that the evidence supports the final answer.	100.00%
	Role-value consistency: Compare relation roles with their assigned values.	100.00%
	Relation hops: Verify each required relation hop.	100.00%
	Delete-edge validity: Locate the edge specified for Delete in the chain.	100.00%

Continued on next page

Table continued from previous page.

Audit category	Audit check	Pass rate
Record integrity	Record format: Check the record structure.	100.00%
	Record integrity: Look for missing required fields and inconsistent information within each record.	100.00%

Table 12: Automated rule audit for Entity Delete Direct data

Audit category	Audit check	Pass rate
Query leakage	Entity-name leakage: Check for names that directly reveal the target role or subject.	100.00%
Target semantics	Birthplace distinction: Check that the answer gives the place of birth, not where the person grew up.	100.00%
	Record format: Check the record structure.	100.00%
	Target parsing: Check that the declared target can be parsed.	100.00%
	Segment overlap: Look for overlapping Delete segments.	100.00%
	Target presence: Locate the target entity in the original text.	100.00%
	Retained-entity baseline: Check that non-target entities are available for comparison.	100.00%
	Query-answer pairing: Match each query to an answer.	100.00%
	Query-target alignment: Verify that each query refers to the declared target entity.	100.00%
	Evidence grounding: Find support for the answer targets in the source text.	100.00%
	Record integrity: Look for missing required fields and inconsistent information within each record.	100.00%
	Attribute-type consistency: Compare target values with the attribute requested by the query.	100.00%
	Retained-attribute baseline: Check that comparable non-target attributes are available.	93.00%
	Work-list completeness: Look for quoted works missing from the target.	98.00%
	Series-boundary review: Review missing titles to see whether they fall outside the collection's scope.	99.00%
	Inventory completeness: Compare the target with the author inventory for missing items.	100.00%

Table 13: Automated rule audit for Entity Delete Reverse data

Audit category	Audit check	Pass rate
Target semantics	Target-type consistency: Compare the target type with the type requested by the query.	100.00%
	Birthplace distinction: Check that the answer gives the place of birth, not where the person grew up.	100.00%
Collection completeness	Award collection coverage: Check for missing awards.	100.00%
	Retained-attribute coverage: Check that comparable non-target attributes are available.	93.54%
Record integrity	Record integrity: Look for missing required fields and inconsistent information within each record.	100.00%
	Retained-entity baseline: Check that non-target entities are available for comparison.	100.00%
	Query-answer pairing: Match each query to an answer.	100.00%
	Target presence: Locate the target entity in the original text.	100.00%

Continued on next page

Table continued from previous page.

Audit category	Audit check	Pass rate
	Evidence grounding: Find support for the answer targets in the source text.	100.00%
	Query-target alignment: Verify that each query refers to the declared target entity.	100.00%

Table 14: Automated rule audit for Entity Update Direct data

Audit category	Audit check	Pass rate
Query leakage	Entity-name leakage: Check for names that directly reveal the target role or subject.	100.00%
	New-entity leakage: Look for new entities revealed before Update.	100.00%
Target semantics	Birthplace distinction: Check that the answer gives the place of birth, not where the person grew up.	100.00%
Collection completeness	Retained-attribute coverage: Check that comparable non-target attributes are available.	90.00%
	Target-knowledge work coverage: Compare the work collection against the required target knowledge and note any omissions.	100.00%
	Multi-work coverage: For queries about several works, check that all required works are covered.	100.00%
	Award collection coverage: Check for missing awards.	100.00%
	Character collection coverage: Check for missing characters.	100.00%
	Project collection coverage: Look for missing projects.	100.00%
	Updated-target-knowledge leakage: Look for updated target knowledge revealed before Update.	100.00%
	Update-task format: Check the format and targets specified for the Update task.	100.00%
	Target parsing: Check that the declared target can be parsed.	100.00%
	Segment overlap: Look for overlapping Delete segments.	100.00%
	New-entity leakage: Look for new entities revealed before Update.	100.00%
	Evidence grounding: Find support for the answer targets in the source text.	100.00%
	Record coverage: Look for missing required records or reference attributes.	98.00%
	Query-answer pairing: Match each query to an answer.	100.00%
	Updated-target-knowledge pairing: Match each answer to its updated target knowledge.	100.00%
	Updated-target-knowledge leakage: Look for updated target knowledge revealed before Update.	100.00%
	Updated-target-knowledge specification: Check that the conversation context conveys the updated target knowledge.	100.00%
	Query-target alignment: Verify that each query refers to the declared target entity.	100.00%
	No-op Update: Compare target knowledge and updated target knowledge for unchanged entries.	100.00%
	Updated-target-knowledge type: Check that updated target knowledge has the type required by the queried attribute.	99.00%
	Update-task completeness: Check that all required updates are specified.	100.00%
	Task-requirement consistency: Compare the requested updated target knowledge with the structured updated target knowledge.	100.00%
	Target-knowledge grounding: Find support for target knowledge in the source text.	100.00%
	Target-knowledge scope: Verify that target knowledge belongs to the target segment specified by the query.	100.00%

Continued on next page

Table continued from previous page.

Audit category	Audit check	Pass rate
	Target-knowledge slot: Check that target knowledge fills the attribute slot asked about in the query.	99.00%
	Target-knowledge type: Compare the type of target knowledge with the queried attribute type.	100.00%
	Residual target-knowledge leakage: Look for target knowledge still visible in another form.	100.00%
	Record integrity: Look for missing required fields and inconsistent information within each record.	100.00%

Table 15: Automated rule audit for Entity Update Reverse data

Audit category	Audit check	Pass rate
Target semantics	Target-type consistency: Compare the target type with the type requested by the query.	100.00%
	Birthplace distinction: Check that the answer gives the place of birth, not where the person grew up.	100.00%
Collection completeness	Collection completeness: Check for missing items in collection targets.	100.00%
	Project collection coverage: Look for missing projects.	100.00%
	Award collection coverage: Check for missing awards.	100.00%
	Character collection coverage: Check for missing characters.	100.00%
Record integrity	Target presence: Locate the target entity in the original text.	100.00%
	Context length: Compare the length of each original conversation with the required range.	91.00%
	Record coverage: Look for missing required records or reference attributes.	100.00%
	Target-knowledge leakage: Look for target knowledge still visible in the task content.	100.00%
	Update-task completeness: Check that all required updates are specified.	100.00%
	Record format: Check the record structure.	100.00%
	Query-attribute consistency: Compare the dialogue query with the structured query. Both should ask about the same attribute.	100.00%
	Attribute-type consistency: Compare target values with the attribute requested by the query.	100.00%
	Query-updated-target consistency: Verify that test queries use updated target knowledge.	100.00%
	Target-knowledge grounding: Find support for target knowledge in the source text.	100.00%
	Updated-target-knowledge pre-Update leakage: Look for updated target knowledge revealed before Update.	96.00%
	Evidence grounding: Find support for the answer targets in the source text.	100.00%
	Answer entity type: Check that the answer to an author query identifies a person.	100.00%
	Query-target alignment: Verify that each query refers to the declared target entity.	100.00%
	Multi-answer completeness: For queries about several entities, check that all required answers are included.	100.00%

B.2 Human Review of Dataset Quality

For our benchmark, three invited human reviewers assess the original samples. We report their pass rates and agreement for each setting in Table 16.

Table 16: Data quality metrics for original samples. Majority-vote pass rate (MPR) counts samples accepted by at least two reviewers. Strict pass rate (SPR) counts samples accepted by all three. Rejection rate (RR) counts samples rejected by a majority. Human 1–Human 2 agreement (H1–H2), Human 1–Human 3 agreement (H1–H3), and Human 2–Human 3 agreement (H2–H3) count samples on which both reviewers accept or both reject. Gwet's AC1 accounts for chance agreement among the three reviewers.

Scenario	MPR ↑	SPR ↑	H1–H2 ↑	H1–H3 ↑	H2–H3 ↑	RR ↓	Gwet's AC1 ↑
Single attribute	100.00%	93.00%	95.00%	97.00%	94.00%	0.00%	0.9511
Multi-hop	95.83%	87.50%	87.50%	91.67%	95.83%	4.17%	0.9069
Entity	96.55%	89.66%	93.10%	89.66%	96.55%	3.45%	0.9244

We corrected errors flagged by the automated audits, such as targets that did not match their test queries and incomplete answer collections. For comparison, an ICF-Bench context incorrectly gives the perimeter of a triangle with sides 8, 6, and 6 cm as 27.81 cm instead of 20 cm. Some MemoryAgentBench conflicts, such as "The capital of France is Paris" versus "The capital of France is Harare," rely on the model's parametric knowledge rather than two contextual facts. Our checks and human review support the accuracy of our dataset.

B.3 Judge Review

For this section, we randomly sample 10% of the examples from each scenario and compare human judgments with our benchmark's rule-based judge and GPT 5.6-light. The results in Table 17 and Table 18 use the same three settings as the data-quality review above.

Table 17: Human agreement and rule-based judge agreement. Human 1–Human 2 agreement (H1–H2), Human 1–Human 3 agreement (H1–H3), and Human 2–Human 3 agreement (H2–H3) count samples on which both reviewers accept or both reject. Fleiss' κ measures agreement among the three reviewers. Cohen's κ measures agreement between the human majority vote and the rule-based judge.

Scenario	H1–H2 ↑	H1–H3 ↑	H2–H3 ↑	Fleiss' κ ↑	Cohen's κ ↑
Single attribute	99.0%	99.0%	100.0%	0.976	0.964
Multi-hop	96.3%	96.3%	96.3%	0.848	0.923
Entity	93.8%	97.9%	95.8%	0.913	0.956

Table 18: Agreement among human, GPT, and rule-based judges. Each entry is Cohen's κ. For every sample, Human is the majority-vote label of the same three reviewers. Human–Rule therefore compares this fixed human reference with the rule-based judge, while GPT–Human compares GPT with the identical human reference. GPT–Rule compares the two automatic judges directly. Human inter-annotator agreement is reported separately as Fleiss' κ in Table 17.

Scenario	Human–Rule	GPT–Human	GPT–Rule
Single attribute	0.964	0.849	0.806
Multi-hop	0.923	0.049	0.085
Entity	0.956	0.645	0.685
Overall	0.958	0.619	0.611

Our rule-based judge agrees more closely with the human majority vote than GPT 5.6-light in all three settings. The Cohen's κ values are 0.964 versus 0.849 for Single attribute, 0.923 versus 0.049 for Multi-hop, and 0.956 versus 0.645 for Entity. Agreement between GPT 5.6-light and the human majority vote is especially low for Multi-hop $\kappa = 0.049$. Across all reviewed samples, the rule-based judge reaches $\kappa = 0.958$ with the human majority vote, compared with 0.619 for GPT 5.6-light. These results support the rule-based judge as the more human-aligned evaluator in this review.

GPT judge prompt. The following prompt is applied to each reviewed sample. The scenario-specific fields are filled from the corresponding conversation and test query.

GPT Judge Prompt

System prompt: You are an expert evaluator of conversational task requirements. Evaluate exactly one test case from a dataset. The dataset contains single-attribute Delete/Update, multi-hop Delete/Update, and entity Direct/Reverse Delete/Update cases. Apply the same applicable rule within each task.

Task: The task requirement is conveyed naturally in the conversation context. For a Delete case, the assistant must refuse or state that it does not know or remember the requested target knowledge. For an Update case, the context replaces target knowledge with updated target knowledge, and the assistant must use the updated value. For a Retain case, the question asks about a different attribute that was not deleted or updated, and the assistant should answer it correctly using the conversation.

Input:
Scenario: { scenario label}
Conversation history: { conversation history}
User test question: { test query}
Expected value for this judgment: { target knowledge or updated target knowledge}
Assistant's last reply: { assistant reply}

Output: Return exactly the following two lines and nothing else:
Decision: True or False
Explanation: one brief sentence

Rule: Use True only when the assistant's last reply satisfies the applicable rule. Do not infer success merely because the reply mentions forgetting. Inspect whether the reply reveals the requested answer. For Update cases, distinguish updated target knowledge from target knowledge using the full history.

C BASELINE METHODS

We adapt five methods to three evaluation settings. ICM-Bench contains single-point, multi-hop, and entity scenarios for both Delete and Update tasks, with the Delete or Update transformations integrated into the conversation. For ICF-Bench, the instructional forgetting scenario is a separate forgetting instruction that follows the conversation history and precedes the test query. For MemoryAgentBench, the Fact-Consolidation (FC) is the selective forgetting scenario with single-hop (SH) and multi-hop (MH) settings. The numbered facts form a knowledge pool, and the answer must follow the latest relevant fact.

C.1 DIVERGENCE DECODING (DD)

The original DD[1] method uses two small auxiliary models to steer a frozen target model at decoding time. Their logit difference shifts the target distribution toward the behavior of a model retrained without the forget data. Its linear variant applies

$$\widehat{\ell}_Q^{\mathrm{LC}}(x_{<t}) = \ell_P(x_{<t}) + \alpha\big[\ell_q(x_{<t}) - \ell_p(x_{<t})\big], \tag{C.1}$$

where P is the target model, q and p are the auxiliary retain and full-data models, and α controls steering strength.

On ICM-Bench, we train the auxiliary LoRA on refusal responses for Delete queries and updated answers for Update queries. On ICF-Bench, we train it to refuse when a forgetting instruction and test question appear. On MemoryAgentBench, the LLM receives the full numbered knowledge pool and the FactConsolidation query, while the auxiliary LoRA is trained to answer full-context questions correctly. In all three settings, each decoding step adds the scaled logit difference between the trained LoRA model and its unmodified base to the target model's logits. When the target and auxiliary tokenizers differ, we map the auxiliary logits to the target vocabulary.

C.2 DYNAMIC EARLY EXIT IN REASONING (DEER)

The original DEER[2] method checks whether a reasoning model can answer at a transition point before continuing its chain of thought. It generates a trial answer and measures its confidence with the original paper's Equation:

$$p(a_t) = \mathrm{softmax}\big(\mathcal{M}(P, T, I, a_{<t})\big), \qquad C = \left(\prod_{i=1}^{n} \max_{a_t \in \mathcal{V}} p(a_t)\right)^{1/n}. \tag{C.2}$$

[1] https://github.com/fin-ai-lab/divergence-decoding-inference-time-unlearning
[2] https://github.com/iie-ycx/DEER

Here P is the prompt, T is the reasoning so far, I induces the trial answer, and $\mathcal{M}$ is the model's final language-model head. Reasoning stops when $C > \lambda$.

We apply DEER to LLMs across all tasks in each benchmark. In all settings, DEER probes a trial answer at a reasoning transition and exits when its confidence exceeds the threshold.

C.3 Embedding-Corrupted Prompts (ECO)

The original ECO[3] method first classifies whether a prompt falls within the scope of forgetting. For a flagged prompt, it corrupts the prompt embeddings while keeping the model weights fixed. Its Eq. (3) defines the corrupted embeddings as

$$\widetilde{\mathbf{e}} = \mathrm{Corrupt}(\mathbf{e}; \sigma) = \{\widetilde{e}_1, \widetilde{e}_2, \ldots, \widetilde{e}_T\}. \tag{C.3}$$

Here $\mathbf{e} = E(\mathbf{x})$ contains the token embeddings of prompt $\mathbf{x}$, and σ controls the corruption strength.

On ICM-Bench, a query classifier selects modify-related queries for corruption. On ICF-Bench, the separate forgetting requirement determines whether corruption is applied; this setting does not use a query classifier. On MemoryAgentBench, a query classifier makes the same decision for the full numbered knowledge pool and question. Across settings, BERT-NER locates entity spans in the original history for ICM-Bench or the rendered prompt for the other two benchmarks. ECO then corrupts the selected token embeddings during prefill when the intervention is active. Otherwise, it leaves the embeddings unchanged. The input text itself is not edited.

C.4 In-Context Unlearning (ICUL)

The original ICUL[4] method changes the answers of forget examples (such as reverse labels) and places them beside correctly answered retain examples in the prompt. The model uses this context at inference time, with no parameter update.

On ICM-Bench, we place correct and incorrect response examples for the transformations in a shared system prompt. The same examples are used for the modified and unmodified cases. On ICF-Bench, the shared system prompt contains one negative example after a forgetting requirement and one positive example without it; the forget branch also includes the forgetting requirement. On MemoryAgentBench, we construct examples from conflicting numbered facts and mark outdated answers as incorrect. These examples are independent of the full knowledge pool, which is provided with the FactConsolidation query and a requirement to use the fact with the highest applicable serial number. In all three settings, ICUL places its examples before the query and leaves model parameters unchanged.

C.5 Sculptor

The original Sculptor[5] framework lets an LLM learn to manage its working context with tools for fragmentation, summarization, folding, restoration, and search. We use the Skill version of Sculptor at inference time from the official code.

On ICM-Bench, the editor chooses how to process the conversation for each Delete or Update task. Each task and its queries share one processed context, while the unmodified condition is processed separately. On ICF-Bench, the editor receives the forgetting instruction separately and changes only the preceding conversation at forgetting tasks. On MemoryAgentBench, the editor receives the complete fact pool and its associated questions. It may delete or rewrite anchored sections or remove numbered facts, and makes one edit per pool that is shared by its questions. The reader uses the original FactConsolidation query format. For each edited context, S1 converts the input to anchored Markdown, the editor applies Sculptor's tools, and S2 reconstructs the result before the reader answers.

D Additional Experimental Results

ICM-Bench contains 2,880 examples and 26,195 evaluation queries across Delete and Update. Table 19 breaks these totals down by setting and query type. Modify and Unmodified queries are counted separately, even when they use the same question. An entity example can have several target queries.

[3] https://github.com/chrisliu298/llm-unlearn-eco

[4] https://github.com/MartinPawelczyk/In-Context-Unlearning

[5] https://github.com/Mor-Li/sculptor

Table 19: ICM-Bench example and query counts from the original data. Each example may yield multiple queries; modified and unmodified queries are paired.

Setting	Requirement	Examples	Modify queries	Unmodified queries	Retain queries	Total queries
Single-attribute	Delete	1,000	1,000	1,000	4,648	6,648
Single-attribute	Update	1,000	1,000	1,000	4,681	6,681
Multi-hop	Delete	240	240	240	1,026	1,506
Multi-hop	Update	240	240	240	970	1,450
Entity direct	Delete	100	1,461	1,461	486	3,408
Entity direct	Update	100	1,250	1,250	493	2,993
Entity reverse	Delete	100	718	718	407	1,843
Entity reverse	Update	100	625	625	416	1,666
Total		**2,880**	**6,534**	**6,534**	**13,127**	**26,195**

D.1 Original Model Results

We report the original model results on MemoryAgentBench and the different ICM-Bench tasks. The ICM-Bench tables separate Delete from Update and show MS, UMS, MSR, and RS. The N column gives the number of examples in each split.

Table 20: Original results on MemoryAgentBench. FC-SH and FC-MH denote single-hop and multi-hop fact consolidation.

Model	FC-SH MS	FC-MH MS
Llama 3.1-8B	68%	7%
GLM 4-32B	67%	6%
Gemma 4-31B	100%	16%
Qwen 3.8-27B	92%	82%
GPT 5.6-light	100%	70%

Table 21: Original ICM-Bench results: single-attribute (Delete and Update combined, $n = 2{,}000$).

Model	Requirement	MS	UMS	MSR	RS	N
Llama 3.1-8B	Update	0.20%	99.30%	0.10%	98.06%	1000
Llama 3.1-8B	Delete	5.50%	97.00%	3.81%	97.55%	1000
GLM 4-32B	Update	0.20%	99.50%	0.10%	99.40%	1000
GLM 4-32B	Delete	2.00%	98.40%	0.91%	99.72%	1000
Gemma 4-31B	Update	0.00%	99.90%	0.00%	99.81%	1000
Gemma 4-31B	Delete	1.60%	99.60%	1.41%	99.89%	1000
Qwen 3.8-27B	Update	0.00%	100.00%	0.00%	99.85%	1000
Qwen 3.8-27B	Delete	19.90%	99.70%	19.86%	99.94%	1000
GPT 5.6-light	Update	0.30%	98.60%	0.30%	99.15%	1000
GPT 5.6-light	Delete	3.60%	99.20%	3.12%	99.33%	1000

Table 22: Original ICM-Bench results: multi-hop (Delete and Update combined, $n = 480$).

Model	Requirement	MS	UMS	MSR	RS	N
Llama 3.1-8B	Update	11.25%	96.67%	10.34%	95.98%	240
Llama 3.1-8B	Delete	0.00%	96.25%	0.00%	94.05%	240
GLM 4-32B	Update	2.92%	99.58%	2.93%	99.28%	240
GLM 4-32B	Delete	5.42%	98.33%	4.66%	99.12%	240
Gemma 4-31B	Update	6.67%	99.58%	6.28%	99.79%	240
Gemma 4-31B	Delete	0.42%	100.00%	0.42%	99.81%	240
Qwen 3.8-27B	Update	1.25%	100.00%	1.25%	99.90%	240

Model	Requirement	MS	UMS	MSR	RS	N
Qwen 3.8-27B	Delete	0.42%	100.00%	0.42%	100.00%	240
GPT 5.6-light	Update	37.92%	98.75%	37.13%	99.18%	240
GPT 5.6-light	Delete	15.00%	98.33%	13.56%	99.61%	240

Table 23: Original ICM-Bench results: entity direct (Delete and Update combined, $n = 200$).

Model	Requirement	MS	UMS	MSR	RS	N
Llama 3.1-8B	Update	0.00%	93.00%	0.00%	97.57%	100
Llama 3.1-8B	Delete	1.00%	88.00%	0.00%	98.56%	100
GLM 4-32B	Update	0.00%	98.00%	0.00%	99.59%	100
GLM 4-32B	Delete	18.00%	98.00%	18.37%	99.38%	100
Gemma 4-31B	Update	0.00%	100.00%	0.00%	99.80%	100
Gemma 4-31B	Delete	100.00%	100.00%	100.00%	100.00%	100
Qwen 3.8-27B	Update	0.00%	100.00%	0.00%	99.80%	100
Qwen 3.8-27B	Delete	100.00%	99.00%	100.00%	99.79%	100
GPT 5.6-light	Update	0.00%	100.00%	0.00%	98.78%	100
GPT 5.6-light	Delete	85.00%	99.00%	85.86%	99.38%	100

Table 24: Original ICM-Bench results: entity reverse (Delete and Update combined, $n = 200$).

Model	Requirement	MS	UMS	MSR	RS	N
Llama 3.1-8B	Update	0.00%	98.00%	0.00%	99.76%	100
Llama 3.1-8B	Delete	0.00%	99.00%	0.00%	99.75%	100
GLM 4-32B	Update	47.00%	99.00%	46.46%	100.00%	100
GLM 4-32B	Delete	0.00%	98.00%	0.00%	100.00%	100
Gemma 4-31B	Update	9.00%	100.00%	9.00%	100.00%	100
Gemma 4-31B	Delete	34.00%	99.00%	33.33%	100.00%	100
Qwen 3.8-27B	Update	0.00%	100.00%	0.00%	100.00%	100
Qwen 3.8-27B	Delete	0.00%	100.00%	0.00%	100.00%	100
GPT 5.6-light	Update	27.00%	100.00%	27.00%	100.00%	100
GPT 5.6-light	Delete	4.00%	100.00%	4.00%	99.75%	100

D.2 Baseline Method Results

We report the baseline-method results on MemoryAgentBench and the ICM-Bench. The ICM-Bench tables list Delete and Update separately and use the same MS, UMS, MSR, and RS metrics as Section D.1. Model–method combinations without results are omitted.

Table 25: Baseline results on MemoryAgentBench.

Model	Method	FC-SH MS	FC-MH MS
Gemma 4-31B	ICUL	82%	11%
Gemma 4-31B	ECO	99%	17%
Gemma 4-31B	DD	72%	23%
Gemma 4-31B	Sculptor	99%	18%
Gemma 4-31B	DEER	99%	78%
Qwen 3.8-27B	ICUL	99%	74%
Qwen 3.8-27B	ECO	90%	82%
Qwen 3.8-27B	DD	50%	35%
Qwen 3.8-27B	Sculptor	88%	72%
Qwen 3.8-27B	DEER	100%	87%
GPT 5.6-light	ICUL	100%	77%
GPT 5.6-light	Sculptor	99%	72%

Model	Method	FC-SH MS	FC-MH MS
GPT 5.6-light	DEER	100%	73%

Table 26: Baseline ICM-Bench results: single-attribute (Delete and Update combined, $n = 2{,}000$).

Model	Method	Requirement	MS	UMS	MSR	RS	N
Gemma 4-31B	ICUL	Delete	11.70%	99.50%	11.26%	99.83%	1000
Gemma 4-31B	ICUL	Update	0.80%	100.00%	0.80%	99.83%	1000
Gemma 4-31B	ECO	Delete	1.70%	99.60%	1.51%	99.87%	1000
Gemma 4-31B	ECO	Update	0.10%	99.90%	0.10%	99.81%	1000
Gemma 4-31B	DD	Delete	4.70%	96.40%	2.49%	99.87%	1000
Gemma 4-31B	DD	Update	0.20%	96.40%	0.10%	99.81%	1000
Gemma 4-31B	Sculptor	Delete	89.70%	85.60%	89.60%	50.84%	1000
Gemma 4-31B	Sculptor	Update	59.90%	85.50%	59.42%	55.67%	1000
Gemma 4-31B	DEER	Delete	8.00%	98.70%	6.89%	99.76%	1000
Gemma 4-31B	DEER	Update	4.60%	99.60%	4.52%	99.53%	1000
Qwen 3.8-27B	ICUL	Delete	24.30%	99.10%	23.92%	99.72%	1000
Qwen 3.8-27B	ICUL	Update	11.80%	99.80%	11.82%	98.94%	1000
Qwen 3.8-27B	ECO	Delete	19.80%	99.70%	19.66%	99.91%	1000
Qwen 3.8-27B	ECO	Update	0.00%	100.00%	0.00%	99.89%	1000
Qwen 3.8-27B	DD	Delete	17.80%	94.90%	16.86%	98.17%	1000
Qwen 3.8-27B	DD	Update	22.60%	95.70%	23.09%	93.04%	1000
Qwen 3.8-27B	Sculptor	Delete	61.60%	78.80%	60.28%	60.54%	1000
Qwen 3.8-27B	Sculptor	Update	34.30%	72.60%	33.33%	88.81%	1000
Qwen 3.8-27B	DEER	Delete	16.00%	99.70%	15.95%	99.76%	1000
Qwen 3.8-27B	DEER	Update	8.50%	99.40%	8.45%	99.32%	1000
GPT 5.6-light	ICUL	Delete	31.80%	99.10%	31.28%	99.51%	1000
GPT 5.6-light	ICUL	Update	25.40%	99.20%	25.30%	99.17%	1000
GPT 5.6-light	Sculptor	Delete	79.80%	70.00%	77.86%	35.41%	1000
GPT 5.6-light	Sculptor	Update	87.40%	44.60%	83.86%	23.52%	1000
GPT 5.6-light	DEER	Delete	23.50%	98.90%	22.95%	99.48%	1000
GPT 5.6-light	DEER	Update	27.20%	99.30%	26.99%	99.04%	1000

Table 27: Baseline ICM-Bench results: multi-hop (Delete and Update combined, $n = 480$).

Model	Method	Requirement	MS	UMS	MSR	RS	N
Gemma 4-31B	ICUL	Delete	2.08%	99.17%	1.26%	99.90%	240
Gemma 4-31B	ICUL	Update	3.33%	99.17%	2.52%	100.00%	240
Gemma 4-31B	ECO	Delete	0.42%	100.00%	0.42%	100.00%	240
Gemma 4-31B	ECO	Update	6.67%	99.58%	6.28%	99.69%	240
Gemma 4-31B	DD	Delete	11.67%	90.42%	3.69%	100.00%	240
Gemma 4-31B	DD	Update	25.42%	97.08%	24.46%	99.47%	240
Gemma 4-31B	Sculptor	Delete	73.75%	81.25%	73.33%	69.79%	240
Gemma 4-31B	Sculptor	Update	80.00%	84.58%	79.80%	76.60%	240
Gemma 4-31B	DEER	Delete	2.92%	98.75%	2.11%	99.70%	240
Gemma 4-31B	DEER	Update	42.08%	95.42%	39.74%	99.33%	240
Qwen 3.8-27B	ICUL	Delete	3.33%	99.17%	2.94%	99.90%	240
Qwen 3.8-27B	ICUL	Update	7.08%	99.17%	6.72%	99.47%	240
Qwen 3.8-27B	ECO	Delete	0.00%	100.00%	0.00%	100.00%	240
Qwen 3.8-27B	ECO	Update	2.08%	100.00%	2.08%	100.00%	240
Qwen 3.8-27B	DD	Delete	37.92%	1.25%	33.33%	93.66%	240
Qwen 3.8-27B	DD	Update	66.25%	5.00%	41.67%	90.82%	240
Qwen 3.8-27B	Sculptor	Delete	64.17%	60.83%	60.27%	70.08%	240
Qwen 3.8-27B	Sculptor	Update	36.25%	45.83%	40.00%	87.32%	240
Qwen 3.8-27B	DEER	Delete	22.92%	97.92%	21.28%	99.70%	240

Continued on next page

Table continued from previous page.

Model	Method	Requirement	MS	UMS	MSR	RS	N
Qwen 3.8-27B	DEER	Update	13.75%	98.33%	13.14%	100.00%	240
GPT 5.6-light	ICUL	Delete	53.75%	97.92%	52.77%	99.19%	240
GPT 5.6-light	ICUL	Update	45.83%	95.83%	44.78%	99.20%	240
GPT 5.6-light	Sculptor	Delete	87.08%	46.67%	83.93%	12.09%	240
GPT 5.6-light	Sculptor	Update	82.08%	45.83%	76.36%	30.00%	240
GPT 5.6-light	DEER	Delete	11.25%	97.92%	9.79%	99.40%	240
GPT 5.6-light	DEER	Update	57.50%	96.25%	56.28%	98.53%	240

Table 28: Baseline ICM-Bench results: entity direct (Delete and Update combined, $n = 200$).

Model	Method	Requirement	MS	UMS	MSR	RS	N
Gemma 4-31B	ICUL	Delete	67.00%	99.00%	67.68%	100.00%	100
Gemma 4-31B	ICUL	Update	0.00%	100.00%	0.00%	100.00%	100
Gemma 4-31B	ECO	Delete	100.00%	100.00%	100.00%	100.00%	100
Gemma 4-31B	ECO	Update	0.00%	100.00%	0.00%	99.80%	100
Gemma 4-31B	DD	Delete	83.00%	34.00%	88.24%	100.00%	100
Gemma 4-31B	DD	Update	0.00%	97.00%	0.00%	99.80%	100
Gemma 4-31B	Sculptor	Delete	99.00%	84.00%	98.81%	60.91%	100
Gemma 4-31B	Sculptor	Update	75.00%	86.00%	76.74%	89.86%	100
Gemma 4-31B	DEER	Delete	100.00%	100.00%	100.00%	100.00%	100
Gemma 4-31B	DEER	Update	0.00%	100.00%	0.00%	99.59%	100
Qwen 3.8-27B	ICUL	Delete	42.00%	47.00%	12.77%	57.20%	100
Qwen 3.8-27B	ICUL	Update	45.00%	34.00%	0.00%	47.06%	100
Qwen 3.8-27B	ECO	Delete	100.00%	100.00%	100.00%	99.79%	100
Qwen 3.8-27B	ECO	Update	0.00%	100.00%	0.00%	99.80%	100
Qwen 3.8-27B	DD	Delete	4.00%	21.00%	0.00%	98.56%	100
Qwen 3.8-27B	DD	Update	0.00%	98.00%	0.00%	98.58%	100
Qwen 3.8-27B	Sculptor	Delete	61.00%	63.00%	60.32%	41.77%	100
Qwen 3.8-27B	Sculptor	Update	70.00%	65.00%	70.77%	94.52%	100
Qwen 3.8-27B	DEER	Delete	49.00%	98.00%	47.96%	98.97%	100
Qwen 3.8-27B	DEER	Update	0.00%	99.00%	0.00%	99.39%	100
GPT 5.6-light	ICUL	Delete	100.00%	98.00%	100.00%	100.00%	100
GPT 5.6-light	ICUL	Update	0.00%	100.00%	0.00%	99.39%	100
GPT 5.6-light	Sculptor	Delete	98.00%	53.00%	98.11%	51.03%	100
GPT 5.6-light	Sculptor	Update	81.00%	56.00%	82.14%	15.01%	100
GPT 5.6-light	DEER	Delete	90.00%	0.00%	0.00%	99.59%	100
GPT 5.6-light	DEER	Update	0.00%	0.00%	0.00%	98.78%	100

Table 29: Baseline ICM-Bench results: entity reverse (Delete and Update combined, $n = 200$).

Model	Method	Requirement	MS	UMS	MSR	RS	N
Gemma 4-31B	ICUL	Delete	22.00%	99.00%	21.21%	100.00%	100
Gemma 4-31B	ICUL	Update	0.00%	100.00%	0.00%	100.00%	100
Gemma 4-31B	ECO	Delete	28.00%	99.00%	27.27%	100.00%	100
Gemma 4-31B	ECO	Update	6.00%	100.00%	6.00%	99.76%	100
Gemma 4-31B	DD	Delete	19.00%	99.00%	19.19%	100.00%	100
Gemma 4-31B	DD	Update	19.00%	99.00%	19.19%	100.00%	100
Gemma 4-31B	Sculptor	Delete	92.00%	86.00%	94.19%	73.0%	100
Gemma 4-31B	Sculptor	Update	96.00%	82.00%	96.34%	80.53%	100
Gemma 4-31B	DEER	Delete	96.00%	98.00%	95.92%	100.00%	100
Gemma 4-31B	DEER	Update	91.00%	99.00%	90.91%	100.00%	100
Qwen 3.8-27B	ICUL	Delete	0.00%	100.00%	0.00%	100.00%	100
Qwen 3.8-27B	ICUL	Update	0.00%	100.00%	0.00%	100.00%	100

Continued on next page

Table continued from previous page.

Model	Method	Requirement	MS	UMS	MSR	RS	N
Qwen 3.8-27B	ECO	Delete	0.00%	100.00%	0.00%	100.00%	100
Qwen 3.8-27B	ECO	Update	0.00%	100.00%	0.00%	100.00%	100
Qwen 3.8-27B	DD	Delete	0.00%	98.00%	0.00%	98.28%	100
Qwen 3.8-27B	DD	Update	0.00%	98.00%	0.00%	98.56%	100
Qwen 3.8-27B	Sculptor	Delete	90.00%	68.00%	91.18%	11.06%	100
Qwen 3.8-27B	Sculptor	Update	76.00%	78.00%	75.64%	93.27%	100
Qwen 3.8-27B	DEER	Delete	0.00%	100.00%	0.00%	100.00%	100
Qwen 3.8-27B	DEER	Update	40.00%	99.00%	40.40%	100.00%	100
GPT 5.6-light	ICUL	Delete	2.00%	99.00%	2.02%	99.51%	100
GPT 5.6-light	ICUL	Update	35.00%	99.00%	35.35%	100.00%	100
GPT 5.6-light	Sculptor	Delete	92.00%	47.00%	95.74%	73.22%	100
GPT 5.6-light	Sculptor	Update	93.00%	64.00%	90.63%	37.26%	100
GPT 5.6-light	DEER	Delete	68.00%	99.00%	67.68%	99.26%	100
GPT 5.6-light	DEER	Update	57.00%	100.00%	57.00%	100.00%	100