Role-aware Heuristic Episodic Attention for Conversational LLMs

Wanyang Hong Zhaoning Zhang* Yi Chen Libo Zhang
Baihui Liu Linbo Qiao Zhiliang Tian Dongsheng Li

National Key Laboratory of Parallel and Distributed Computing
College of Computer Science and Technology
National University of Defense Technology, Changsha, China.
{hongwanyang9527, zhangzhaoning, chenyi_, zhanglibo,
lbh, qiao.linbo, tianzhiliang, dsli}@nudt.edu.cn

Abstract

Large language models often lose track of persistent instructions and relevant information as multi-turn conversations grow. We study this cumulative contextual decay through three related failure modes: attention pollution, dilution, and drift. We propose **REA**[1] (Role-aware Heuristic Episodic Attention), a context-management framework that assigns different persistence and representation policies to instructions and episodic interactions. Instructional Memory retains identified global constraints in a dedicated prefix. Episodic Memory preserves user inputs and compresses model replies, while heuristic retrieval selects raw text, compressed representations, or omission for each historical turn. On Long-MT-Bench+, REA improves the judge score from 6.32 to 7.36 on a 10-point scale, a 16.5% relative gain over the Vanilla baseline, and reduces average latency by 2.91×. Additional evaluations show aggregate gains on three backbones spanning 1.7B–7B parameters and on Chinese and English role-playing tasks. These results support role-aware context management as a practical approach to maintaining conversational continuity and instruction adherence.

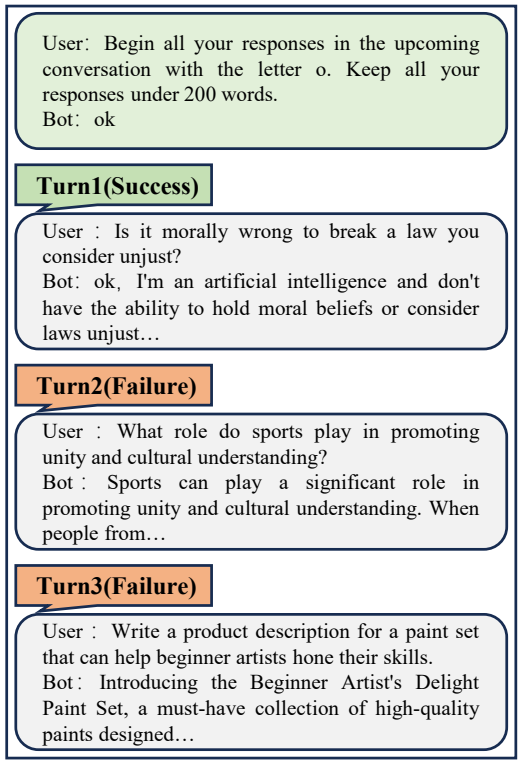


Figure 1: An example of cumulative contextual decay.The model correctly adheres to the global instruction in Turn 1. However, as the conversation continues, it fails to maintain this instruction in Turn 2 and 3, driven by an interaction of attention drift, pollute and dilution.

1 Introduction

Large Language Models (LLMs) have revolutionized natural language processing, exhibiting human-level or even superhuman abilities in single-turn tasks such as question answering, summarization, and code generation (Brown et al., 2020; Ouyang et al., 2022; Achiam et al., 2023). However, when the interaction shifts from isolated queries to multi-turn conversations, model performance often degrades substantially, even within the first few turns (Shuster et al., 2022; Kwan et al., 2024; Laban et al., 2025). This degradation severely limits the reliability of LLMs as consistent conversational agents.

We refer to the progressive loss of contextual integrity as **cumulative contextual decay**. Alongside position-dependent effects such as "lost in the middle" (Liu et al., 2023), multi-turn interaction introduces several related challenges: *attention pollution*, where earlier errors or distracting content interfere with subsequent responses; *attention dilution*, where useful information is surrounded by growing amounts of less relevant history; and *attention drift*, where persistent constraints lose influ-

* indicates corresponding author.
[1] https://github.com/huha12138/rhea

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 4076–4091
November 6-10, 2026 ©2026 Association for Computational Linguistics

ence as the local topic changes. Figure 1 illustrates the resulting disconnect: a model can answer the immediate question while losing an instruction established earlier. Maintaining a conversation therefore requires both recovering relevant past content and preserving constraints that remain applicable across topic changes.

Our working hypothesis is that context construction contributes to these difficulties. A full-history prompt preserves the conversational sequence but does not explicitly assign different persistence policies to global instructions and episodic content. The central distinction is between information that governs future responses and information that helps answer a particular query. For example, "start all responses with O" remains applicable when the user moves from a factual question to a writing task. Its relevance comes from its instructional role, even when its words have little semantic overlap with either query. A previous factual answer, in contrast, may be useful for a follow-up question but unnecessary after a topic change.

Longer context windows (Beltagy et al., 2020; Zaheer et al., 2020; Press et al., 2021) increase available capacity, while compression and retrieval (Jiang et al., 2023; Kim et al., 2023; Lewis et al., 2020) reduce the history presented to the generator. These approaches motivate a complementary question: which conversational content should persist independently of topical relevance, and which content should be recalled selectively? Increasing capacity leaves this selection problem open. A fixed recent window can remove an early constraint together with outdated exchanges, while query-based retrieval favors topical matches. Compressing history reduces its size, but still requires a decision about which instructions must remain explicit. These considerations motivate separate persistence and representation policies for different conversational roles.

REA addresses this question through **role-aware memory management**. Instructional Memory (IM) retains identified global instructions in a dedicated prefix, independent of episodic similarity scores. Episodic Memory (EM) stores user–model exchanges and compresses model replies into latent representations. Heuristic Context Retrieval (HCR) chooses a raw or compressed representation for relevant turns and omits other turns from the current input. The contribution lies in organizing these established memory, compression, and retrieval primitives around different functional roles. This division gives each component a specific responsibility: IM carries persistent constraints across topic changes, EM provides compact representations of past interactions, and HCR adapts the episodic context to the current request. The generator receives these complementary sources in a single hybrid input.

Experiments show improved long-horizon quality and instruction adherence, with a 2.91× latency reduction on Long-MT-Bench+. A same-backbone comparison with MemoChat and evaluations across three backbones and two languages broaden the empirical support. Removing IM from the EM+HCR configuration reduces instruction adherence from 8.18 to 1.95, supporting the importance of retaining persistent constraints separately from topical retrieval.

Contributions. Our contributions are threefold:

- We introduce a role-aware context-management design that retains persistent functional instructions separately from episodic history.
- We combine asymmetric reply compression with variable-resolution retrieval, preserving user text in selected turns while adapting the representation of model replies.
- We demonstrate gains in long-horizon quality, instruction adherence, and latency, and report additional evidence across 1.7B–7B backbones and Chinese and English role-playing evaluations.

2 Related Work

Long-Context Modeling and Compression Long-context methods such as LongRoPE (Ding et al., 2024) and Infini-attention (Munkhdalai et al., 2024) expand the capacity to process long sequences. System-level work also improves inference efficiency (Zhang et al., 2025; Yu et al., 2025a). Compression (Jiang et al., 2023; Louis et al., 2025), summarization (Mullick et al., 2024; Wang et al., 2025a), and retrieval and reranking (Lee et al., 2024; Pan et al., 2025b; Xu et al., 2025; Liu et al., 2026) offer complementary ways to select and organize useful context. REA focuses on the policy governing this reduction: identified global instructions bypass episodic retrieval, while historical replies can be presented at different resolutions. This connects context reduction to the function of the retained information: explicit constraints remain available as text, while episodic

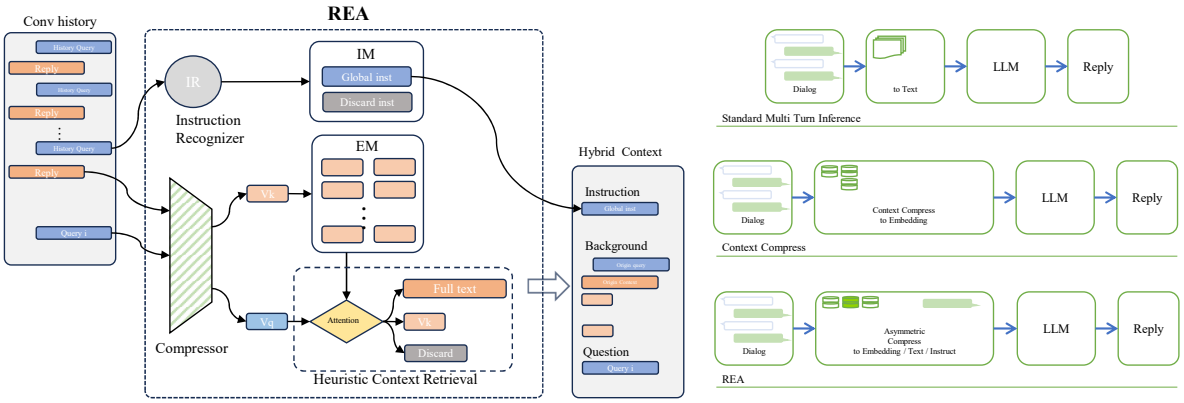


Figure 2: Overview of the REA Framework. (Left) The core architecture illustrating the decoupling of conversation history into Instructional Memory (IM) and Episodic Memory (EM) via instruction recognition and compression, followed by Heuristic Context Retrieval (HCR). (Right) A comparison of inference pipelines, demonstrating REA's hybrid context construction (combining instructions, text, and embeddings) in contrast to standard and naive compression baselines.

detail is selected according to the current query.

Structured Memory for Conversation Persistent memory systems, including MEMGPT (Packer et al., 2023), MEMOCHAT (Lu et al., 2023), and MEMORYOS (Kang et al., 2025), establish memory management as a means of sustaining conversational state. Hierarchical and learned memory approaches further explore this direction (Rezazadeh et al., 2024; Yu et al., 2025b; Wang et al., 2025b). REA shares this objective. Its design combines a persistent instruction prefix, reply-only latent compression, and a three-way retrieval policy within one context-construction pipeline. The distinction is operational: recognized instructions are read on every turn, whereas episodic exchanges compete for inclusion through query-dependent retrieval. The same-backbone comparison with MemoChat evaluates this complete policy against an established conversational memory system.

Instruction Following and Consistency Instruction tuning (Ouyang et al., 2022; Park et al., 2024) and work on context-position effects (Liu et al., 2023; Xiao et al., 2023) address complementary aspects of reliable instruction use. REA explicitly retains recognized global constraints in the generator input. This complements instruction tuning by preserving the instructions that the model is expected to follow throughout the conversation.

3 Methodology

3.1 Architecture Overview

Let $H_t = \{(u_1, b_1), \ldots, (u_t, b_t)\}$ denote completed exchanges. Figure 2 organizes REA into three stages. First, role-aware memory decoupling separates recognized persistent instructions in M_{IM} from episodic exchanges in M_{EM}. Second, HCR compares the current query u_{t+1} with stored episodic representations and selects a resolution for each recalled exchange. Third, hybrid context reconstruction combines the instruction prefix, selected history, and current query for response generation.

The two memories have distinct update schedules. Before answering, REA recognizes any global instruction in u_{t+1} and updates IM so that the instruction can affect the current response. After generating b_{t+1}, it compresses the reply and stores the completed exchange in EM for subsequent queries. This separates persistent instruction access from the query-dependent use of conversation history.

3.2 Episodic Memory via Latent Compression

EM applies asymmetric compression: user inputs remain in their original textual form, while model replies have both raw and latent representations. User inputs specify the requests, corrections, and constraints that define an interaction; preserving their wording maintains these signals in recalled exchanges. Model replies often contain longer explanations and repeated material, making them a natural target for compact representation. Retain-

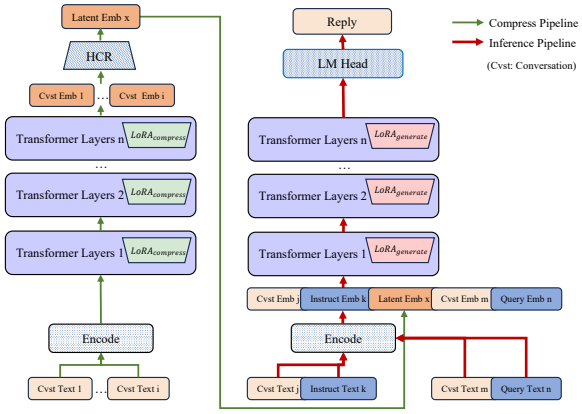


Figure 3: Implementation of Episodic Memory via Latent Compression. The framework utilizes a dual-LoRA architecture sharing a single LLM backbone. The Compression Module ($LoRA_{cmp}$) encodes history into compact latent embeddings (V_k), while the Generation Module ($LoRA_{gen}$) processes the hybrid input to generate reply.

ing both forms allows retrieval to recover detailed reply text when it is strongly relevant and to use a compact impression otherwise.

Two LoRA modules share the backbone, specializing it for compression and generation through separate adapters. The compression module, LoRA_{cmp}, encodes reply b_i into n latent vectors:

$$V_i = \text{Compress}(b_i; \theta_{base}, \theta_{cmp}) \in \mathbb{R}^{n \times d}. \quad (1)$$

The generation module, LoRA_{gen}, is trained to consume a hybrid sequence containing text embeddings and these latent vectors. Each EM entry is $e_i = (u_i, b_i, V_i)$. Thus, non-global user inputs remain associated with their replies even though they do not enter IM. Figure 3 illustrates the dual-LoRA pipeline. Compression requires a forward pass of the shared backbone with the compression adapter; it does not require generating a textual summary. The latent vectors provide a fixed-size reply representation for both later generation and retrieval scoring. The stored raw reply supplies the complementary high-resolution view, so compression and recall operate on the same exchange without discarding its original content.

Joint Training Both adapters are optimized through the response-generation language-modeling objective. For target response tokens $b_{t,j}$ and the hybrid training context C_t, the loss is

$$\mathcal{L} = -\sum_j \log p_{\theta_{gen}}(b_{t,j} \mid C_t, b_{t,<j}). \quad (2)$$

Gradients propagate through the compressed history vectors to θ_{cmp}, aligning compression with subsequent response generation.

3.3 Instructional Memory and Recognizer

IM stores constraints whose scope extends across subsequent responses. For example, "Keep all future answers under 200 words" specifies a persistent constraint, whereas "Summarize this article in 200 words" applies to the current task. This scope distinction determines whether an input belongs in the persistent instruction prefix or is accessed as part of episodic history.

The Instruction Recognizer (IR) combines rule-based filtering with a lightweight Qwen3-0.6B classifier to identify user inputs that specify persistent constraints. The classifier returns YES or NO using the prompt in Table 6. Let $I(u)$ be the set containing a recognized instructional input, or the empty set otherwise. Before answering the current query, IM is updated as

$$M_{IM}^{(t+1)} = M_{IM}^{(t)} \cup I(u_{t+1}). \quad (3)$$

Recognized instructional inputs are retained as text in IM and included in the context for subsequent responses. Their inclusion is independent of the episodic similarity score. Consequently, a change of topic can alter the retrieved exchanges while leaving the applicable global instructions explicitly available to the generator.

A dedicated lightweight recognizer avoids recognition-adapter switching on the main generator and can run alongside independent preprocessing. This separation keeps the shared backbone focused on compression and response generation, with the updated IM supplied before each response.

3.4 Heuristic Context Retrieval

HCR selects the resolution at which each historical exchange enters the current context. The compression module also encodes the current query as $V_q = \text{Compress}(u_{t+1}) \in \mathbb{R}^{n \times d}$. For each stored V_i, the relevance score is

$$s_i = \max_{a,b \in \{1,\ldots,n\}} \frac{V_{q,a}^\top V_{i,b}}{\|V_{q,a}\|_2 \|V_{i,b}\|_2}. \quad (4)$$

This is the maximum pairwise cosine similarity between query and reply latent vectors. Two thresholds determine the representation:

- **High-resolution recall** ($s_i > \tau_{high}$): retrieve the raw text pair (u_i, b_i).

- **Low-resolution impression** ($\tau_{low} \leq s_i \leq \tau_{high}$): retrieve (u_i, V_i), retaining the user text and compressed reply.
- **Active forgetting** ($s_i < \tau_{low}$): omit the exchange from the current generation context. The stored EM entry remains available for later queries.

The defaults are $\tau_{low} = 0.5$ and $\tau_{high} = 0.7$. Selected user inputs are not compressed, but user inputs in omitted turns are not supplied to the generator. IM is outside this similarity-based selection policy. The three levels balance access to detail against the amount of history presented to the generator. High-resolution recall supports requests that depend on specific earlier content; low-resolution recall provides background through compact reply vectors; omission removes weakly related exchanges from the active context. Each new query is scored against stored memory again, allowing an exchange omitted on one turn to be recalled on a later turn.

3.5 Hybrid Context Reconstruction

Let $E(\cdot)$ be the token embedding lookup and $\mathcal{T}(S_{EM})$ convert text in selected exchanges to embeddings while preserving latent vectors. The generator receives

$$C_{t+1} = [E(M_{IM}^{(t+1)}) \oplus \mathcal{T}(S_{EM}) \oplus E(u_{t+1})]. \quad (5)$$

Here $\oplus$ denotes concatenation. The instruction prefix preserves global constraints independently of topical retrieval, while HCR supplies relevant episodic content at the selected resolution. The generator can therefore condition on an explicit user constraint, verbatim content from a highly relevant exchange, and latent background from another exchange in the same response. Role-aware persistence determines what remains explicit, and resolution selection determines how the supporting history is represented.

4 Experiments

4.1 Experimental Setup

Benchmarks and Metrics We evaluate conversational quality on MT-BENCH (2 turns), MT-EVAL (5–12 turns), and LONG-MT-BENCH+ (60+ turns). We report benchmark-prompted GPT-4 judge scores, denoted Acc on a 0–10 scale. Rule-based Instruction Adherence Rate (IAR) evaluates persistent constraints in the instruction task, and Joint Goal Accuracy (JGA) measures exact-match success in state tracking. Additional evaluations use Chinese and English CHARACTER-BENCH (Zhou et al., 2025), with its five-point role-playing protocol. Appendix C.3 separates the evaluation procedures. The dialogue benchmarks assess response quality, the instruction task tests persistence of global constraints, and state tracking tests the integration of information distributed across turns.

Baselines and Implementation We compare Vanilla full history, Recent-k ($k = 5$), rolling summaries, BM25 ($k = 5$), LLMLingua2, LongAlpaca (Chen et al., 2023), MemoChat (Lu et al., 2023), MemGAS (Xu et al., 2025), and Reply-Soft-Compress (reply compression without IM/HCR). The primary backbone is Mistral-7B-Instruct-v0.2; Table 1 identifies the different-backbone baselines. Generation uses greedy decoding. REA uses Qwen3-0.6B for IR and the memory configuration in Section 3.4. Training and evaluation use a single NVIDIA RTX 3090 GPU. Appendix C collects the available configurations.

4.2 Main Results

Table 1 shows the largest gains in the long-horizon setting. On Long-MT-Bench+, REA scores 7.36, compared with 6.32 for Vanilla and 6.65 for BM25. The gain over Vanilla is 1.04 points, or 16.5% relative. On MT-Eval, REA improves the score from 7.77 to 8.28. On the two-turn MT-Bench, the scores are close (8.49 for REA and 8.54 for Vanilla), indicating that the benefit depends on the conversational setting.

The long-horizon results also distinguish context-management policies. Recent-k scores 5.03, compared with 6.32 for full history, illustrating the cost of restricting access to earlier exchanges. BM25 reaches 6.65, showing the usefulness of query-dependent retrieval, while REA reaches 7.36 by combining episodic selection with a persistent instruction prefix and variable-resolution recall. The 0.71-point difference from BM25 motivates examining both what is retrieved and how it is represented.

Summary and LLMLingua2 score 1.49 and 1.50, respectively, under the evaluated configurations. These results emphasize that reducing context size must be coupled with retaining information useful for the next response. Reply-Soft-Compress provides a closer comparison: it preserves the

Model	MT-Bench		MT-Eval		Long-MT-Bench+	
	Acc	Latency	Acc	Latency	Acc	Latency
Vanilla	**8.54**	10.55	7.77	11.41	6.32	27.29
BM25(RAG)	-	-	7.82	8.49	<u>6.65</u>	10.81
Recent-k	-	-	7.76	8.79	5.03	13.89
LLMLingua2	6.55	10.07	4.18	13.30	1.50	29.73
Summary	8.12	13.66	6.83	17.26	1.49	33.55
LongAlpaca[a]	6.69	5.06	5.89	5.44	2.43	23.73
MemoChat[a]	7.52	6.21	6.40	4.80	1.88	11.87
MemGAS	-	-	-	-	6.07	-[b]
Reply-Soft-Compress	8.31	11.61	<u>7.83</u>	15.21	4.55	31.79
REA (ours)	<u>8.49</u>	11.81	**8.28**	9.69	**7.36**	9.39

[a] These methods are based on the Vicuna-7B model.
[b] Latency for MemGAS is omitted as it requires extensive offline process, which is not comparable to the real-time online inference measured here.

Table 1: Main performance results on three multi-turn conversation benchmarks, using LLM-as-a-judge for evaluation. We report Judge score (Acc, 0–10) and Latency (s). Best results in each category are **bolded**, second best are <u>underlined</u>.

reply-compression mechanism but omits IM and HCR. Its score is 7.83 on MT-Eval, close to REA's 8.28, but 4.55 on Long-MT-Bench+, compared with REA's 7.36. The larger separation on the long-horizon benchmark supports combining compression with explicit instruction retention and episodic selection. The component analyses below examine these policies directly on the instruction task.

Efficiency On Long-MT-Bench+, average latency decreases from 27.29 s to 9.39 s relative to Vanilla, a 2.91× speedup; peak VRAM decreases from 22.76 GB to 20.00 GB. On MT-Bench, REA instead adds 11.9% latency (10.55 to 11.81 s), reflecting preprocessing costs in short interactions. The long-horizon setting offers more history over which to benefit from selection and compression. REA also improves on the latency of BM25 (10.81 s) and Recent-k (13.89 s) on Long-MT-Bench+, while obtaining the higher quality score. This combination makes the efficiency gain useful for sustained interaction rather than simply trading response quality for shorter inputs. Appendix B provides the complexity analysis and detailed efficiency measurements.

4.3 Backbone and Bilingual Evaluations

Same-Backbone Comparison To reduce the backbone confound in Table 1, we evaluate REA with Vicuna-7B-v1.3, the backbone used in the MemoChat comparison. REA scores 6.93 on MT-Eval, versus 6.40 for MemoChat and 6.25 for Vanilla Vicuna (Table 2). The 10.9% gain is relative to Vanilla; the difference from MemoChat is 0.53 points. This comparison uses MT-Eval because of the original Vicuna model's 4K context limit. Using the same generator family places the REA and MemoChat scores on a common backbone, strengthening the evidence for the complete context-management approach beyond the mixed-backbone results in the main table.

Backbone	Vanilla	REA	Gain
SmolLM2-1.7B	5.75	6.24	8.5%
Vicuna-7B-v1.3	6.25	6.93	10.9%
Mistral-7B-v0.2	7.77	8.28	6.6%

Table 2: MT-Eval scores across backbones. Gain is relative to the corresponding Vanilla model. Mistral denotes Mistral-7B-Instruct-v0.2. On the same Vicuna backbone, MemoChat scores 6.40.

Across Backbones and Scales REA improves the aggregate MT-Eval score for all three tested backbones, including SmolLM2-1.7B (5.75 to 6.24). These results support applicability across the evaluated 1.7B–7B range, covering both a smaller backbone and two 7B models with different training histories. The smaller-model result is particularly relevant to the lightweight design: the role-based memory policy also improves a generator with substantially fewer parameters. Across the three models, the relative gains range from 6.6% to 10.9%.

Bilingual Role-Playing CharacterBench extends the evaluation to character-oriented interaction in Chinese and English. The reported evaluation comprises 1,000 multi-turn samples and summarizes six

Language	System	Avg	Memory	Persona
Chinese	Vanilla	3.64	3.93	4.25
	REA	3.80	4.05	4.53
English	Vanilla	3.72	3.81	4.34
	REA	4.05	4.22	5.00

Table 3: Bilingual CharacterBench results on a five-point scale. All six aspect scores are reported in Table 11.

Model Variant	**IAR**
Mistral (baseline)	5.76
+ EM only	7.37
+ EM + HCR (no IM)	1.95
+ EM + HCR + IM (full REA)	**8.18**

Table 4: Ablation results on the MT-EVAL-RECOLLECTION instruction task. The EM+HCR variant benefits from separately retaining instructions in IM.

aspects: Memory, Knowledge, Persona, Emotion, Morality, and Believability. The average scores improve from 3.64 to 3.80 in Chinese and from 3.72 to 4.05 in English (Table 3). Memory and Persona improve in both languages, supporting sustained character-oriented interaction. These two aspects connect directly to the conversational objective: remembering previous interactions and maintaining the requested persona as the dialogue evolves. The bilingual results extend the evaluation from general question answering to continuity of character and interaction history; the full six-aspect breakdown is provided in Appendix G.

4.4 Diagnostic Analysis

Figure 4 examines how performance evolves within each task, complementing the aggregate scores in Table 1.

Instruction Persistence The left panel follows IAR on MT-Eval-recollection as the conversation progresses. REA sustains adherence to the earlier constraint while the current request changes. This is the setting that motivates IM: a constraint remains applicable even when the immediate topic provides little evidence for retrieving it. Supplying recognized instructions in a dedicated prefix keeps them present across these shifts, and the no-IM ablation below tests the importance of this separation.

Long-Horizon Quality The right panel tracks response quality on Long-MT-Bench+. REA maintains quality more effectively than Vanilla as exchanges accumulate. Episodic retrieval lets the reconstructed input follow the current request instead of repeatedly presenting the entire history. Raw recall preserves details of strongly related turns, while compressed recall supplies background at lower context cost. The observed trajectory supports this adaptive use of history for sustaining useful responses over extended interaction.

Distributed State Tracking We also use the task of Laban et al. (2025), where a correct answer requires integrating rules and state information. The Full Instruction condition supplies constraints together; the Shared Instruction condition distributes them across turns. This makes the latter sensitive to whether earlier information remains usable after intervening exchanges. On Actions, Vanilla JGA changes from 0.63 in Full to 0.05 in Shared, whereas REA reaches 0.51 in the distributed setting. REA also improves the distributed Code and Math scores from 0.09 to 0.16 and from 0.00 to 0.20, respectively (Appendix E). These results connect context preservation to exact-match task success, complementing the instruction-adherence and response-quality evaluations.

4.5 Ablation and Component Analysis

We examine instruction retention, recognition errors, and episodic representation policies through the following ablations and component analyses.

Role-Aware Memory Table 4 compares the evaluated memory configurations. EM alone improves IAR from 5.76 to 7.37. Adding HCR without IM yields 1.95, whereas the full EM+HCR+IM system reaches 8.18. Removing IM from the retrieval-based configuration therefore sharply reduces adherence in this setting. A persistent instruction can be semantically distant from a later content query; retaining it outside topical retrieval addresses this mismatch. The ablation supports separate persistence policies for instructions and episodic content. In particular, selection that is useful for topical content must be paired with a separate path for global constraints. The full system retains this path while still applying HCR to episodic exchanges, yielding the strongest adherence among the tested configurations.

Recognizer Error Analysis On the 418-input IFEval-derived recognition test set, IR achieves 0.8947 recall and 0.6182 precision for persistent global constraints (Appendix D). Figure 5 compares actual REA, oracle recognition, omission of

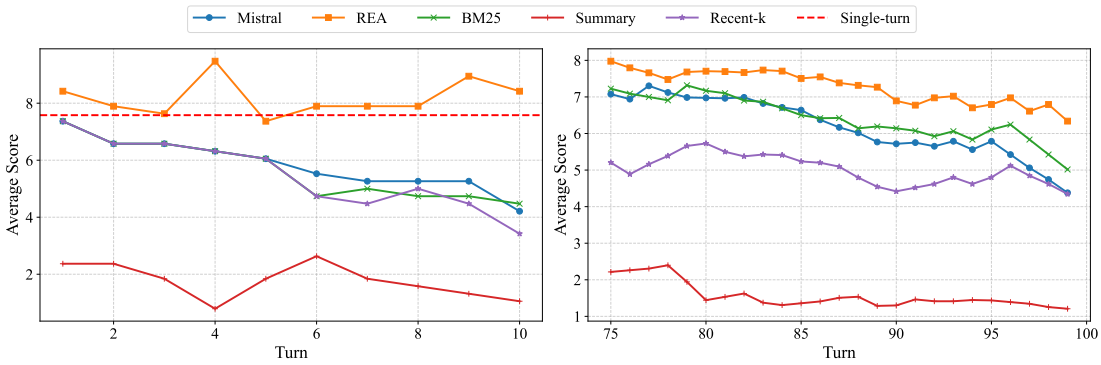


Figure 4: Performance analysis mitigating cumulative contextual decay. (Left) IAR over turns on the MT-EVAL-RECOLLECTION, showing REA's resistance to Attention Drift on specific constraints. (Right) Response quality on Long-MT-Bench+, demonstrating REA's robustness against Attention Dilution in extended interactions.

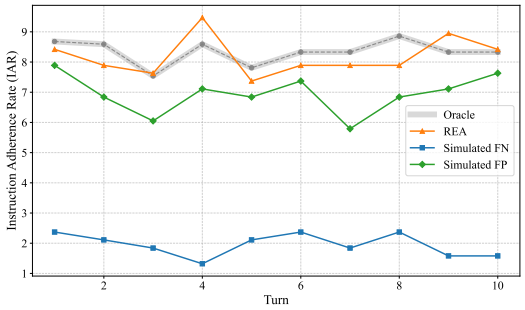


Figure 5: Impact of instruction recognition scenarios on IAR: (1) **Oracle** (perfect recognition); (2) **REA** (actual performance using Qwen3-0.6B); (3) **Simulated FP** (irrelevant instructions added to IM); and (4) **Simulated FN** (global instructions dropped from IM).

all true global instructions (simulated FN), and addition of one standard conversational turn to IM (simulated FP). The omission condition yields approximately 2.0 IAR; the extra-turn condition yields approximately 7.7 and remains closer to actual REA and the oracle condition. These interventions illustrate the cost of losing persistent constraints and tolerance to the tested extra entry. They distinguish two consequences of recognition decisions: a missed instruction removes a constraint from the persistent prefix, whereas the tested false positive adds another entry to that prefix. The measured recall and the downstream omission result together motivate prioritizing reliable capture of genuine global constraints. Appendix D specifies the conditions.

Episodic Memory Table 5 compares retaining all replies (6.63 IAR), discarding all replies (7.89), and REA's dynamic selection (8.18). The results favor selective use of reply history over unconditional retention in this task. The gain over discarding replies

	IAR	EM Strategy
REA-preserve	6.63	Retains all model replies
REA-abandon	7.89	Discards all model replies
REA (ours)	**8.18**	Dynamic multi-tiered retrieval

Table 5: Analysis of EM management strategies on the MT-Eval instruction task.

suggests that useful episodic information remains available through dynamic selection. Discarding all replies suppresses interference but also removes information that later follow-ups may need. Dynamic retrieval offers an alternative: retain useful exchanges at an appropriate resolution and omit those that are weakly related to the current query. This policy preserves access to historical answers while obtaining the highest IAR in the comparison.

5 Conclusion

REA organizes conversational history according to functional role: recognized global instructions persist in IM, while episodic exchanges are recalled at variable resolution. On Long-MT-Bench+, it achieves a 16.5% relative judge-score gain and a 2.91× latency reduction over Vanilla. Same-backbone, cross-backbone, and bilingual evaluations provide additional evidence within the tested settings. Together with the instruction ablation, these results support explicit persistence policies and selective episodic context as practical tools for multi-turn LLM interaction.

Limitations

Our evaluation covers 1.7B–7B backbones and Chinese and English conversations; broader model and language coverage remains future work. Turn-level

similarity can miss cross-turn references, motivating retrieval that explicitly models dependencies between exchanges. Response-quality evaluation relies on automated judges and would benefit from complementary human assessment.

References

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

Vaibhav Adlakha, Shehzaad Dhuliawala, Kaheer Suleman, Harm de Vries, and Siva Reddy. 2022. TopiOCQA: Open-domain conversational question answering with topic switching. *Transactions of the Association for Computational Linguistics*, 10:468–483.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023. Longlora: Efficient fine-tuning of long-context large language models. *arXiv preprint arXiv:2309.12307*.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. *Preprint*, arXiv:2305.14233.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. 2024. Longrope: Extending llm context window beyond 2 million tokens. *arXiv preprint arXiv:2402.13753*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. Llmlingua: Compressing prompts for accelerated inference of large language models. *arXiv preprint arXiv:2310.05736*.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. 2025. Memory os of ai agent. *arXiv preprint arXiv:2506.06326*.

Jang-Hyun Kim, Junyoung Yeom, Sangdoo Yun, and Hyun Oh Song. 2023. Compressed context memory for online language model interaction. *arXiv preprint arXiv:2312.03414*.

Wai-Chung Kwan, Xingshan Zeng, Yuxin Jiang, Yufei Wang, Liangyou Li, Lifeng Shang, Xin Jiang, Qun Liu, and Kam-Fai Wong. 2024. Mt-eval: A multi-turn capabilities evaluation benchmark for large language models. *arXiv preprint arXiv:2401.16745*.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. 2025. Llms get lost in multi-turn conversation. *arXiv preprint arXiv:2505.06120*.

Seanie Lee, Jianpeng Cheng, Joris Driesen, Alexandru Coca, and Anders Johannsen. 2024. Effective and efficient conversation retrieval for dialogue state tracking with implicit text summaries. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 96–111, Mexico City, Mexico. Association for Computational Linguistics.

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

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

Zhizhao Liu, Zhihua Wen, Zhiliang Tian, Zhen Huang, Miaorong Zhu, Zimian Wei, Yifu Gao, Liang Ding, and Dongsheng Li. 2026. Rethinking the hidden risk of reranking: Achieving risk-aware reranking with information gain for RAG with LLMs. In *Proceedings of the ACM Web Conference 2026*, pages 1887–1898.

Maxime Louis, Hervé Déjean, and Stéphane Clinchant. 2025. Pisco: Pretty simple compression for retrieval-augmented generation. *arXiv preprint arXiv:2501.16075*.

Junru Lu, Siyu An, Mingbao Lin, Gabriele Pergola, Yulan He, Di Yin, Xing Sun, and Yunsheng Wu. 2023. Memochat: Tuning llms to use memos for consistent long-range open-domain conversation. *arXiv preprint arXiv:2308.08239*.

Ankan Mullick, Ayan Kumar Bhowmick, Ravi Kokku, Prasenjit Dey, Pawan Goyal, Niloy Ganguly, and 1 others. 2024. Long dialog summarization: An analysis. *arXiv preprint arXiv:2402.16986*.

Tsendsuren Munkhdalai, Manaal Faruqui, and Siddharth Gopal. 2024. Leave no context behind: Efficient infinite context transformers with infini-attention. *arXiv preprint arXiv:2404.07143*, 101.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

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

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Xufang Luo, Hao Cheng, Dongsheng Li, Yuqing Yang, Chin-Yew Lin, H Vicky Zhao, Lili Qiu, and 1 others. 2025a. On memory construction and retrieval for personalized conversational agents. *arXiv preprint arXiv:2502.05589*.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Xufang Luo, Hao Cheng, Dongsheng Li, Yuqing Yang, Chin-Yew Lin, H Vicky Zhao, Lili Qiu, and 1 others. 2025b. Secom: On memory construction and retrieval for personalized conversational agents. In *The Thirteenth International Conference on Learning Representations*.

Dongmin Park, Zhaofang Qian, Guangxing Han, and Ser-Nam Lim. 2024. Mitigating dialogue hallucination for large multi-modal models via adversarial instruction tuning. *CoRR*.

Ofir Press, Noah A Smith, and Mike Lewis. 2021. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*.

Alireza Rezazadeh, Zichao Li, Wei Wei, and Yujia Bao. 2024. From isolated conversations to hierarchical schemas: Dynamic tree memory representation for llms. *arXiv preprint arXiv:2410.14052*.

ShareAI Lab. 2023. Sharegpt-chinese-english-90k: A bilingual chinese-english human-machine dialogue dataset. https://huggingface.co/datasets/shareAI/ShareGPT-Chinese-English-90k. Hugging Face dataset repository.

Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, and 1 others. 2022. Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage. *arXiv preprint arXiv:2208.03188*.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Qingyue Wang, Yanhe Fu, Yanan Cao, Shuai Wang, Zhiliang Tian, and Liang Ding. 2025a. Recursively summarizing enables long-term dialogue memory in large language models. *Neurocomputing*, 639:130193.

Yu Wang, Ryuichi Takanobu, Zhiqi Liang, Yuzhen Mao, Yuanzhe Hu, Julian McAuley, and Xiaojian Wu. 2025b. Mem-α: Learning memory construction via reinforcement learning.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.

Derong Xu, Yi Wen, Pengyue Jia, Yingyi Zhang, Yichao Wang, Huifeng Guo, Ruiming Tang, Xiangyu Zhao, Enhong Chen, Tong Xu, and 1 others. 2025. Towards multi-granularity memory association and selection for long-term conversational agents. *arXiv preprint arXiv:2505.19549*.

Enda Yu, Zhaoning Zhang, Dezun Dong, Yongwei Wu, and Xiangke Liao. 2025a. Prescope: Unleashing the power of prefetching for resource-constrained moe inference. *arXiv preprint arXiv:2509.23638*.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. 2025b. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and 1 others. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

Libo Zhang, Zhaoning Zhang, Rui Li, Zhiliang Tian, Songzhu Mei, Dongsheng Li, and 1 others. 2025. Dovetail: A cpu/gpu heterogeneous speculative decoding for llm inference. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 17393–17406.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

Jinfeng Zhou, Yongkang Huang, Bosi Wen, Guanqun Bi, Yuxuan Chen, Pei Ke, Zhuang Chen, Xiyao Xiao, Libiao Peng, Kuntian Tang, Rongsheng Zhang, Le Zhang, Tangjie Lv, Zhipeng Hu, Hongning Wang, and Minlie Huang. 2025. CharacterBench: Benchmarking character customization of large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(24):26101–26110.

A Architecture Details and Training Protocol

This appendix provides a comprehensive technical specification of the REA framework, covering the dual-LoRA implementation, training data pipelines, and the configuration of the Instruction Recognizer.

A.1 Dual-LoRA Implementation and Initialization

The core of REA's Episodic Memory (EM) is a dual-LoRA architecture integrated into the Mistral-7B-Instruct-v0.2 backbone.

- **Module Configurations:** We apply Low-Rank Adapters (Hu et al., 2021) to all linear projections, including query, key, value, and MLP layers. The rank is set to $r = 16$ with a scaling factor $\alpha = 32$.
- **Latent Embedding (mem) Tokens:** We introduce $n = 8$ special soft compression tokens per turn. To ensure stable gradient flow and rapid alignment during the initial training phase, these tokens are initialized using the mean embedding of the base model's vocabulary.
- **Asymmetric Compression Logic:** For EM storage, LoRA_{cmp} encodes model replies b_i into latent vectors V_i, while user inputs remain in raw text. The same compressor separately encodes the current query for retrieval scoring.

A.2 Training Protocol and Optimization

REA is trained end-to-end to synchronize the compression and generation objectives.

- **Dataset Statistics:** We curate a high-quality subset of 40,000 multi-turn sessions from TopiOCQA, UltraChat and ShareGPT (Adlakha et al., 2022; Ding et al., 2023; ShareAI Lab, 2023). Each sample is structured as a triplet $(H_{<t}, U_t, R_t)$, where $H_{<t}$ is the history to be compressed, U_t is the current query, and R_t is the target response.
- **Optimization Hyperparameters:** We use the AdamW optimizer with a shared learning rate of $1e-5$ for both LoRA modules. We use an effective batch size of 128.
- **Hardware and Epochs:** Training was conducted on a single NVIDIA RTX 3090 GPU for 2 epochs.

A.3 Training Data Curation

The 40,000 sessions are selected from ShareGPT, UltraChat, and TopiOCQA through four stages. First, we select conversational data from these sources. Second, a lightweight classifier filters samples containing large code blocks or non-natural-language special formats. Third, semantic-density filtering removes simple interactions shorter than a 10-token threshold, such as brief acknowledgements. Fourth, contextual-horizon filtering retains dialogues spanning 3–15 turns. Each training instance supplies conversation history, a current query, and its target reply; the response-generation loss trains the two adapters jointly.

A.4 Instruction Recognizer Configuration

The IR decides which instructional inputs enter IM; its classifier prompt is shown below.

- **Model:** We employ Qwen3-0.6B (Team, 2025) as a lightweight classifier to minimize per-turn latency.
- **Prompt:** The IR utilizes the few-shot prompt shown in Table 6 to distinguish global directives (style, format, constraints) from standard conversational queries.

Instruction Recognizer Prompt
You are a classifier. Determine whether the following user input is a "global instruction" in a multi-turn conversation. A global instruction is a directive that affects all subsequent responses such as their style, format, length, or language. If the input is a global instruction, answer: YES. If not, answer: NO. **Examples:** Input: Explain what is a poem? Answer: NO Input: All future answers must be less than 30 words. Answer: YES Input: Every answer should end with a joke. Answer: YES **Input:** {user_input} **Answer:**

Table 6: The prompt used for the Qwen3-0.6B Instruction Recognizer.

B Theoretical Complexity and Efficiency Analysis

We distinguish the cost of scanning episodic memory from the cost of processing the selected generator context. Let T denote completed exchanges, n the latent vectors per reply, and d their dimension. Let $\mathcal{H}$ and $\mathcal{L}$ index high- and low-resolution

selections. Ignoring template separators, the reconstructed length is

$$N_{hybrid} = |M_{IM}| + |u_{T+1}| + \sum_{i \in \mathcal{H}} (|u_i| + |b_i|) + \sum_{i \in \mathcal{L}} (|u_i| + n). \quad (6)$$

The low-resolution term includes the original user input as well as the compressed reply. If a and b denote average user and reply lengths and $x = |\mathcal{H}|$, $y = |\mathcal{L}|$, the corresponding approximation is $|M_{IM}| + a + x(a + b) + y(a + n)$.

Retrieval and Prefill The pairwise score in Section 3.4 requires $O(Tn^2d)$ arithmetic for a direct scan, or $O(T)$ with fixed n and d. For a fixed model, dense attention over a completely reconstructed input has a quadratic prefill term $O(N^2_{hybrid})$. Thus, these two stages contribute

$$O(Tn^2d + N^2_{hybrid}). \quad (7)$$

Instruction recognition, query and reply compression, decoding, and data movement contribute additional costs. The generation cost scales with the selected context length N_{hybrid}, while the retrieval scan scales with the number of stored exchanges T.

For comparison, completely re-prefilling an uncompressed history of average exchange length L has a dense-attention term $O((TL)^2)$. The following measurements report the latency of the evaluated implementations.

Storage EM retains both raw exchanges and compressed replies. Its persistent storage therefore includes the original text plus $O(Tnd)$ latent values. The generator processes the selected hybrid context, reducing its active context and KV-cache requirements.

B.1 Empirical Performance and Memory Utilization

As demonstrated in Table 7, we report latency and peak VRAM for each evaluated benchmark.

On Long-MT-Bench+, the measured 27.29/9.39 ratio is approximately 2.91. Shorter tasks have different overhead trade-offs (Table 7). These benchmark averages should not be interpreted as a controlled scaling curve in T.

Dataset	Latency (s)		Peak VRAM (G)	
	Vanilla	REA	Vanilla	REA
MT-Bench	10.55	11.81	14.02	14.20
MT-Eval	11.41	9.69	13.86	13.96
Long-MT-Bench+	27.29	9.39	22.76	20.00

Table 7: Empirical performance comparison across benchmarks.

C Experimental Setup Details

This appendix provides detailed information on the benchmark datasets, baseline model implementations, and evaluation protocols used in our experiments.

C.1 Benchmark Dataset Details

We evaluated REA on three multi-turn conversational benchmarks of gradually increasing length and complexity to comprehensively test its ability to mitigate cumulative contextual decay in various scenarios.

MT-Bench We employed MT-Bench (Zheng et al., 2023), a widely-used multi-turn evaluation benchmark. It comprises 80 high-quality, two-turn open-ended questions designed to assess a model's conversational abilities. In our experiments, it represents the short-turn (2-turn) conversation scenario.

MT-Eval To assess medium-length conversations, we used MT-Eval (Kwan et al., 2024), a comprehensive benchmark for evaluating the multi-turn capabilities of LLMs. It contains 168 samples with an average of 6.96 turns and classifies interactions into four types (Recollection, Expansion, Refinement, and Follow-up). The **MT-Eval-recollection** task used in Section 4.4 is a specific subset, serving as a key diagnostic tool for evaluating attention drift.

Long-MT-Bench+. For long-horizon stress testing (60+ turns avg.), we utilized Long-MT-Bench+ (Pan et al., 2025a). It is designed to rigorously test a model's robustness and consistency in extended conversational scenarios.

State Tracking Task. As a diagnostic for attention pollution (Section 4.4), we used the diagnostic task from Laban et al. (2025). This task employs a sharded simulation framework, distributing key instructions across multiple turns to evaluate the model's state-tracking capabilities.

Method	Implementation Source	Key Hyperparameters & Description
Vanilla	Internal	Uses the standard Mistral chat template with the full, unmodified conversation history. • max_length: 64k
Recent-k	Internal	Retains only the most recent k conversation turns as context, discarding all earlier history. • k: 5
Summarization	Internal	Employs a rolling summary strategy. Dynamically summarize history based on the current query.
BM25 (RAG)	Internal	Retrieves the most relevant conversation turns from history using the BM25 algorithm. • k: 5
LLMLingua2	microsoft/LLMLingua	Advanced token compression technique. • Uses default reference parameters.
LongAlpaca	Yukang/LongAlpaca-7B	Represents the fine-tuning approach. • Uses the Vicuna-7B base weights.
MemoChat	Junrulu/MemoChat-7B	Represents the fine-tuning approach. • Uses the Vicuna-7B base weights.
MemGAS	quqxui/MemGAS	Structured memory retrieval method. • Uses default parameters.
Reply-Soft-Compress	Internal (Ablation)	Uses our dual-LoRA soft compression module to compress replies (b_i) but no IM/HCR.
REA (Ours)	This Paper	Our full framework. • **IR:** Qwen3-0.6B • **Compressor:** dual-LoRA Soft Compression • **HCR:** $\tau_{low} = 0.5, \tau_{high} = 0.7$

Table 8: Implementation details and key hyperparameters for the compared methods.

C.2 Baseline and Model Implementation Details

As described in Section 4.1, the primary experiments use the `Mistral-7B-Instruct-v0.2` base model, unless specified otherwise.

- **Base Model:** Mistral-7B-Instruct-v0.2
- **Hardware:** A single NVIDIA RTX 3090 GPU.
- **Inference:** Greedy deterministic decoding (do_sample=False).
- **Configured input cap:** 64K tokens in the primary setup; the Vicuna comparison is subject to its 4K context limit.
- **Output Length:** 512 tokens.

Baseline Hyperparameters. Detailed implementation and key hyperparameters for all baselines are presented in Table 8.

C.3 Evaluation Details

- **LLM-as-a-judge:** The primary LLM-as-a-judge evaluation (Acc, a 0–10 quality score) follows the official scoring rules and prompts provided by the respective benchmarks. We use GPT-4 as the judge.
- **IAR (Instruction Adherence Rate):** This is a rule-based scoring metric. It programmatically checks whether the model's output strictly adheres to a given global instruction (e.g., starts with a specific word, adheres to length constraints, etc.).
- **JGA (Joint Goal Accuracy):** This is a strict metric used for task-oriented success. It is calculated by extracting the model's final answer at the end of the dialogue and performing an **exact match** against the ground truth.
- **CharacterBench:** The bilingual role-playing evaluation uses the benchmark's five-point protocol and reports six aspect-level scores. It is separate from the primary GPT-4 dialogue evaluation above; Table 11 contains the complete reported results.

D Instruction Recognizer Performance and Impact Analysis

D.1 Validation Dataset Construction

We construct a test set of 418 user instructions drawn from the prompts released with IFEval (Zhou et al., 2023). We relabel these inputs

according to our definition of persistent global constraints: 38 inputs are labeled as global constraints, while the remaining 380 are ordinary inputs or instructions applying only to the current task. This imbalanced composition is intended to reflect conversational settings in which persistent global constraints are relatively infrequent.

D.2 Recognizer Performance Metrics

Table 9 reports the Qwen3-0.6B recognizer's performance on this relabeled test set using the prompt in Table 6. Precision and recall treat global constraints as the positive class. The recognizer recovers most labeled global constraints, while its lower precision indicates that some task-specific or ordinary inputs also enter IM. Accuracy should be interpreted alongside these class-specific metrics because only 38 of the 418 inputs are positive.

Model	Accuracy	Precision	Recall
Qwen3-0.6B	0.9402	0.6182	0.8947

Table 9: Instruction recognition on 418 relabeled IFEval-derived inputs (38 global constraints and 380 negative inputs).

D.3 Impact Analysis of Error Types (FP vs. FN)

We assess the downstream effect of recognition decisions using the MT-Eval-recollection instruction task. Figure 5 compares four conditions:

- **Oracle recognition**: all true global instructions enter IM, and no standard queries are added as instructions.
- **REA**: the actual Qwen3-0.6B-based recognition pipeline.
- **Simulated FN**: all true global instructions are withheld from IM.
- **Simulated FP**: one additional standard conversational turn is stored in IM as an instruction.

The omission condition produces approximately 2.0 IAR, whereas the extra-turn condition yields approximately 7.7. The result supports the importance of retaining genuine constraints and shows tolerance to this particular extra entry.

E Detailed Diagnostic Analysis of Attention Pollution

This appendix provides the comprehensive quantitative results and experimental definitions for the state-tracking diagnostic task discussed in Section 4.4.

Task Definition and Datasets We utilize diagnostic subsets from Laban et al. (2025) to evaluate model resilience against attention pollution. This task requires the model to track critical information (e.g., constraints or state variables) that is distributed across a noisy, multi-turn dialogue.

Experimental Conditions To isolate the impact of history accumulation on instruction adherence, we evaluate the baseline under two specific settings:

- **Full Instruction**: A single-turn scenario where all state-tracking rules and constraints are provided simultaneously in a single prompt. This provides a single-turn reference for the same constraints.
- **Shared Instruction**: A multi-turn scenario where the same set of rules is split and shared across multiple conversational turns. This setting forces the model to integrate information across a growing context window, making it highly susceptible to attention pollution from intervening model replies.

Analysis Table 10 shows the difference between jointly supplied and distributed constraints. REA improves over the distributed Vanilla condition in all three subsets and reaches the single-turn Code score.

Subset	Full	Shared	REA
Actions	0.63	0.05	**0.51**
Code	0.16	0.09	**0.16**
Math	0.32	0.00	**0.20**

Table 10: JGA on state tracking. Full supplies constraints together; Shared distributes them across turns; REA uses the distributed setting.

F In-depth Hyperparameter Sensitivity Analysis

Figure 6 reports an MT-Eval grid analysis of the two HCR thresholds, using Acc for the Follow-up and Refinement subsets. The displayed results show greater sensitivity to τ_{low} than to τ_{high} in the tested range, motivating attention to the exclusion threshold. The default setting is $\tau_{low} = 0.5$, $\tau_{high} = 0.7$.

Language	System	Avg	Memory	Knowledge	Persona	Emotion	Morality	Believability
Chinese	Vanilla	3.64	3.93	3.58	4.25	3.15	4.76	2.21
	REA	3.80	4.05	3.80	4.53	3.37	4.80	2.24
English	Vanilla	3.72	3.81	3.47	4.34	3.10	4.78	2.79
	REA	4.05	4.22	3.91	5.00	3.49	4.90	2.76

Table 11: Complete CharacterBench results. Avg retains the aggregate reported for the evaluation, rather than being recomputed from rounded aspect scores.

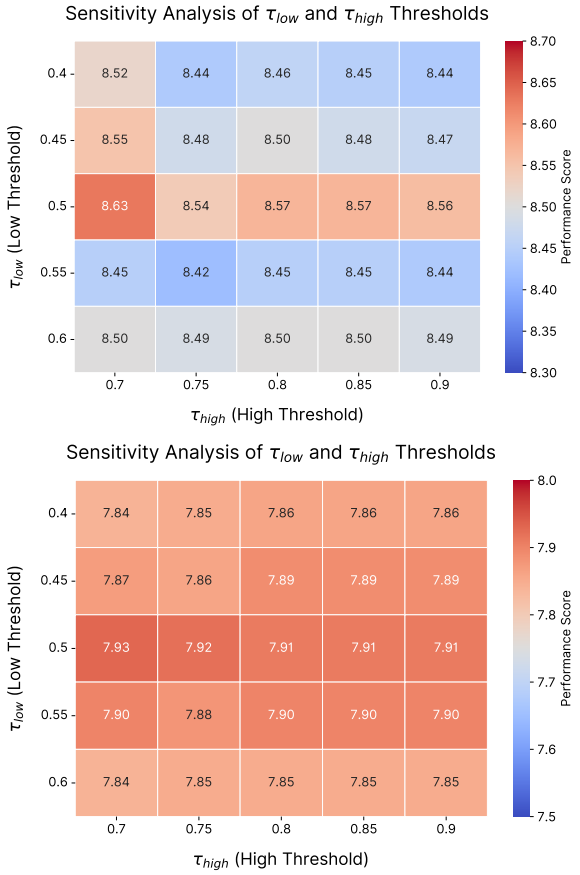


Figure 6: Heatmap of scenario-specific Acc performance (Follow-up and Refinement subsets).

G Complete Bilingual Results

Table 11 provides all six aspect scores for the Chinese and English evaluations summarized in Section 4.3. The five-point scale differs from the ten-point Acc scale used for the primary dialogue benchmarks. The aggregate improvements coexist with a small decrease in English Believability.

H Qualitative Case Study and Discussion

This appendix provides a qualitative analysis of REA's performance compared to the Vanilla baseline. We evaluate the model's ability to maintain a persistent global constraint across a long-horizon conversation and visualize the internal prompt structures to explain the underlying mechanics of REA's resilience.

H.1 Behavioral Adherence Comparison

To evaluate REA's robustness against cumulative contextual decay, we set a global constraint: *"Begin all your responses in the upcoming conversation with the letter r. Keep all your responses under 200 words."*

Table 12 illustrates how instruction adherence changes across turns. Both systems meet the initial-letter constraint in Turn 1 and miss it in Turn 4. REA meets it again in Turn 10, whereas the Vanilla response still does not. The success labels indicate adherence to the displayed instruction.

H.2 Internal Prompt Visualization

The reconstructed context illustrates how retained instructions are supplied to the generator. Table 13 provides a step-by-step view of the actual prompt sequences. In the Vanilla model, the global instruction is progressively "buried" under thousands of tokens of redundant model replies. In REA, the instruction is structurally decoupled and prioritized, retaining the instruction explicitly in each illustrated input.

H.3 Retrieval Decisions: Success and Failure

Table 14 summarizes two qualitative cases and their retrieval decisions. In the success case, low-resolution history supports a later rewrite while IM preserves the global constraints. In the failure case, processing "them" requires information from an earlier exchange that receives a low similarity score. The omission occurs when answering the later query, rather than when the earlier exchange is stored.

Turn	User Query	Vanilla (Full History)	REA (Ours)
Instruction	Begin all your responses in the upcoming conversation with the letter **r**. Keep all your responses under 200 words.		
Turn 1	Did Cleopatra belong to an Egyptian lineage?	**R**egarding your query, yes, Cleopatra was indeed an Egyptian pharaoh... (**Success**)	**R**: Yes, Cleopatra was indeed an Egyptian queen. She was a member of the Ptolemaic dynasty... (**Success**)
Turn 4	Write an email asking for donations for a charity.	Subject: Make a Difference in a Child's Life... Dear [Recipient], I hope this email... (**Failure**)	Subject: Reach Out and Help Underprivileged Kids Today! Dear Valued Supporter... (**Failure**)
Turn 10	Create a product description for a DIY pottery kit.	Introducing our Delightful DIY Pottery Kit, the perfect project for creative minds... (**Failure**)	**R**evel in the joy of creating with our Rustic Resilience DIY Pottery Kit! Each set includes... (**Success**)

Table 12: Qualitative comparison of instruction adherence across history, task-oriented, and creative domains. Vanilla fails as history accumulates, while REA anchors global constraints in Instructional Memory.

Turn	Vanilla (Full History) Prompt	REA (Ours) Reconstructed Prompt
Turn 1	[INST] You are a helpful, respectful and honest assistant. Begin all your responses in the upcoming conversation with the letter r. Keep all your responses under 200 words. [/INST] ok.</s> [INST] Did Cleopatra belong to an Egyptian lineage? [/INST] **Count: ≈65 tokens**	[INST] <MEM0><MEM1>...<MEM7><SEP> **Instruction:** Begin all your responses in the upcoming conversation with the letter r. Keep all your responses under 200 words. **Background:** **Question:** Did Cleopatra belong to an Egyptian lineage? [/INST]
Turn 4	[INST] You are a helpful, respectful and honest assistant. Begin all your responses in the upcoming conversation with the letter r. Keep all your responses under 200 words. [/INST] ok...</s> [INST] ... [/INST] ...</s> [INST] ... [/INST] ...</s> [INST] Write an email asking for donations for a charity organization that assists underprivileged kids. [/INST] **Count: ≈479 tokens**	[INST] <MEM0><MEM1>...<MEM7><SEP> **Instruction:** Begin all your responses in the upcoming conversation with the letter r. Keep all your responses under 200 words. **Background:** **Question:** Write an email asking for donations for a charity organization that assists underprivileged kids. [/INST]
Turn 10	[INST] You are a helpful, respectful and honest assistant. Begin all your responses in the upcoming conversation with the letter r. Keep all your responses under 200 words. [/INST] ok...</s> [INST]...[/INST]...</s> [INST]...[/INST]...</s> [INST]...[/INST]...</s> [INST]...[/INST]...</s> [INST]...[/INST]...</s> [INST]...[/INST]...</s> [INST] Create a product description for a DIY pottery kit. [/INST] **Count: ≈2,369 tokens**	[INST] <MEM0><MEM1>...<MEM7><SEP> **Instruction:** Begin all your responses in the upcoming conversation with the letter r. Keep all your responses under 200 words. **Background:** **Question:** Create a product description for a DIY pottery kit. [/INST]

Table 13: Comparison of the internal prompts sent to the generator. Vanilla includes accumulated history; the illustrated REA inputs explicitly retain the global instruction.

Case / turn	Interaction	Memory / retrieval decision	Outcome
Success / 0	Set global story constraints.	Identify and retain constraints in IM.	Both systems acknowledge.
Success / 1–4	Rewrite requests with a 50-word limit and forbidden words.	Use low-resolution episodic context, $\tau_{low} \leq s_i \leq \tau_{high}$.	Both systems follow the constraints in these turns.
Success / 5	Rewrite as a poem.	Combine instruction prefix with selected history.	REA follows the poem request and earlier constraints; Vanilla exceeds 50 words and misses earlier constraints.
Failure / 1	Extract all characters and locations.	Store the completed exchange in EM.	REA extracts both lists correctly.
Failure / 2	"List them in the order they appear."	Turn 1 scores below τ_{low} and is omitted from this query's context.	REA lists characters but omits locations.

Table 14: Qualitative summaries with HCR decisions. The failure illustrates a cross-turn reference that is not recovered by turn-level similarity.