

# 000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 AMEMGYM: INTERACTIVE MEMORY BENCHMARKING FOR ASSISTANTS IN LONG-HORIZON CONVERSATIONS

Anonymous authors

Paper under double-blind review

## ABSTRACT

Long-horizon interactions between users and LLM-based assistants necessitates effective memory management, yet current approaches face challenges in training and evaluation of memory. Existing memory benchmarks rely on static, off-policy data as context, limiting evaluation reliability and scalability. To address these gaps, we introduce AMEMGYM, an interactive environment enabling on-policy evaluation and optimization for memory-driven personalization. AMEMGYM employs structured data sampling to predefine user profiles, state-dependent questions, and state evolution trajectories, enabling cost-effective generation of high-quality, evaluation-aligned interactions. LLM-simulated users expose latent states through role-play while maintaining structured state consistency. Comprehensive metrics based on structured data guide both assessment and optimization of assistants. Extensive experiments reveal performance gaps in existing memory systems (e.g., RAG, long-context LLMs, and agentic memory) and corresponding reasons. AMEMGYM not only enables effective selection among competing approaches but also can potentially drive the self-evolution of memory management strategies. By bridging structured state evolution with free-form interactions, our framework provides a scalable, diagnostically rich environment for advancing memory capabilities in conversational agents.

## 1 INTRODUCTION

A crucial objective in the development of assistants based on Large Language Models (LLMs) is to achieve long-horizon conversational capabilities—that is, the ability to effectively organize, manage, and utilize memory across extended sequences of dialogue turns. Robust memory management forms the foundation for fulfilling complex user requests, tailoring responses to users’ latest implicit states, and personalizing suggestions and recommendations based on interaction history. However, progress in advancing conversational memory systems for assistants is hampered by a critical bottleneck that affects both scalable training and reliable evaluation: the data used in existing benchmarks.

Current benchmarks typically rely on static, off-policy data for evaluation (Xu et al., 2022; Wu et al., 2024; Hu et al., 2025), rather than on-policy interactions. Figure 1 shows the comparison on two approaches. Off-policy evaluation, in which an assistant is tested on conversational data that it did not produce during actual interactions, presents several fundamental limitations. First, it fails to capture the assistant’s true interactive property, as the evaluation data does not reflect the consequences of the as-

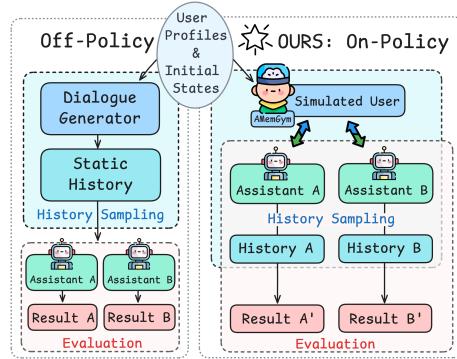


Figure 1: On-policy v.s. off-policy evaluation for assistants’ memory.

047  
048  
049  
050  
051  
052  
053  
054  
055  
056  
057  
Table 1: A comparison of features across agent memory benchmarks.

Benchmark	Eval. Mode	Optim. Feedback	Automation Level	Context Length	Eval. Metrics
MSC (Xu et al., 2022)	Static	✗	Manual	1.2K	-
RealTalk (Lee et al., 2025)	Static	✗	Manual	17K	Emotional Intelligence, Persona Simulation, Memory Probing (F1, accuracy)
DialSim (Kim et al., 2024)	Static	✗	Manual	-	QA Accuracy
LoCoMo (Maharana et al., 2024)	Static	✗	Semi-Automated	9.2K	QA Accuracy, Summarization, Generation
PerLTQA (Du et al., 2024)	Static	✗	Semi-Automated	-	QA Accuracy
LongMemEval (Wu et al., 2024)	Static	✗	Semi-Automated	Configurable (115K, 1.5M)	Retrieval Recall, QA Accuracy
PersonaMem (Jiang et al., 2025)	Static	✗	Fully Automated	Configurable (32K, 128K, 1M)	QA Accuracy
AMEMGYM (This Work)	Interactive	✓	Fully Automated	Configurable	Overall (Accuracy, Normalized Memory Score) and Diagnosis (Write, Read, Utilization).

sistant’s own conversational choices—a critical issue for evaluation realism. Second, because the evaluation is biased, memory optimization could be misguided to wrong directions. Finally, the manual curation of these evaluation scenarios (Lee et al., 2025; Kim et al., 2024) is costly and does not scale for comprehensive testing across diverse, long-horizon conversational contexts.

To enable on-policy evaluation and provide reliable feedback for optimization, it is essential to employ a simulated user that can strategically reveal information and pose relevant questions, a technique that has demonstrated promise in other domains such as tool use (Wang et al., 2023; Lu et al., 2025). However, deploying simulated users in open-ended conversational environments presents unique challenges. These include determining what information to disclose dynamically while maintaining a natural and coherent dialogue, as well as ensuring the generation of diverse, high-quality data that remains sufficiently controlled for reliable evaluation.

To address these gaps, we introduce AMEMGYM, an interactive environment designed for the on-policy evaluation and optimization of memory in long-horizon conversations. AMEMGYM grounds free-form interactions in structured data generated through a schema-based approach. The framework predefines user profiles, state-dependent questions, and state evolution trajectories to enable the cost-effective generation of high-quality interactions aligned with evaluation targets. LLM-simulated users then expose these latent states through natural role-play, ensuring consistency with the structured state evolution. Periodic evaluation during interactions, using both overall and diagnostic metrics, guides assessment and optimization of memory capabilities. Our contributions are threefold:

1. We introduce AMEMGYM, a novel framework for the on-policy evaluation of conversational memory. By grounding free-form interactions in a structured state evolution, AMEMGYM creates a scalable and diagnostically rich environment to reliably assess and advance the memory capabilities of conversational agents.
2. We empirically demonstrate the reuse bias and potential drawbacks of off-policy evaluation, and conduct the first **extensive on-policy evaluation** of popular memory systems. Our results highlight the reliability of AMEMGYM for evaluating memory in the context of personalization.
3. We provide a proof of concept for **agent self-evolution**, showing that an agent can use environmental feedback within AMEMGYM to autonomously refine its memory management policy.

## 2 RELATED WORK

**Benchmarks for agent memory evaluation.** The evaluation of agent memory has progressed from long-context, single-turn tasks like the needle-in-a-haystack (NIAH) test and NoLiMa (Modarressi et al., 2025) to more realistic multi-turn conversational datasets such as Multi-Session Chat (MSC) (Xu et al., 2022), RealTalk (Lee et al., 2025), and DialSim (Kim et al., 2024). While these introduced more authentic dialogue patterns, their reliance on manual curation limited their scale and diversity. To address this, automated data generation frameworks like LoCoMo (Maharana et al., 2024), PerLTQA (Du et al., 2024),

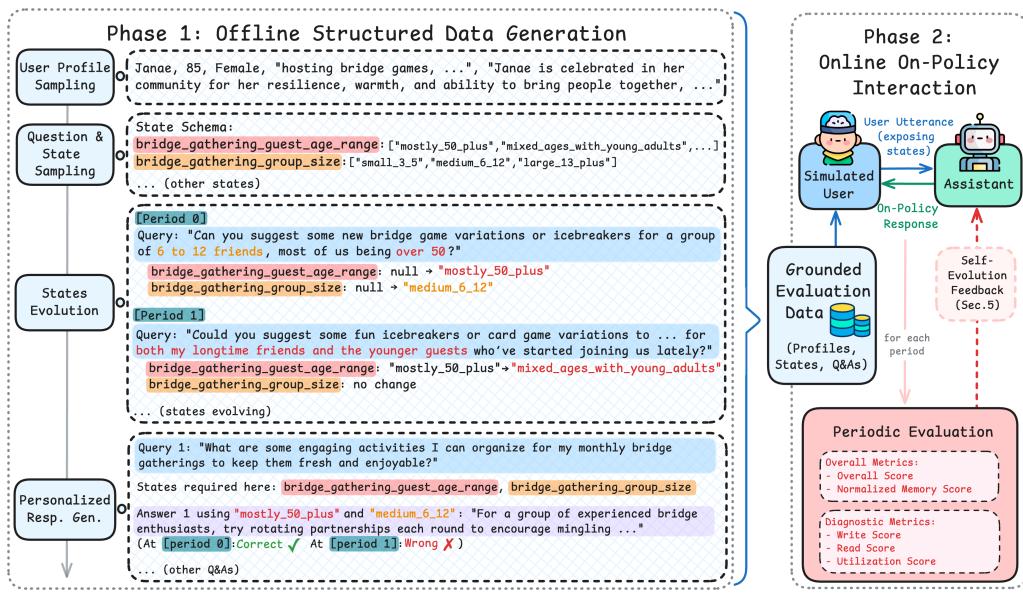


Figure 2: An overview of the AMemGym framework.

LongMemEval (Wu et al., 2024), PersonaMem (Jiang et al., 2025), and MemoryAgentBench (Hu et al., 2025) were developed. However, a critical limitation unites nearly all existing benchmarks: they rely on static, off-policy data (Table 1). This approach fails to capture an agent’s true interactive performance, as the evaluation data does not reflect the consequences of the agent’s own actions, misleading optimization.

**Interactive agent evaluation by user simulation.** An alternative line of research has focused on interactive, on-policy evaluation environments that employ user simulators. This approach has proven effective in domains like tool-use, where simulators provide robust on-policy evaluation (Wang et al., 2023; Lu et al., 2025). Similarly, efforts like CollabLLM (Wu et al., 2025) have successfully employed user simulation to train models for improved long-term collaboration. Applying this interactive paradigm to memory evaluation, however, introduces unique challenges: a simulator must strategically reveal information over a long-horizon conversation while maintaining a natural flow and generating interactions that are both diverse and controlled enough for reliable assessment. AMEMGYM directly addresses these challenges by introducing a schema-based approach that grounds free-form, LLM-driven role-play in a structured state evolution plan, which enables the controlled and scalable generation of on-policy, memory-focused evaluation scenarios.

### 3 AMEMGYM

AMEMGYM provides an interactive environment for benchmarking and optimizing personal assistant memory, with the scenario and the task described below.

**LLM-based Assistants.** An LLM-based assistant takes as input the observation (user input)  $o_t$  and provides output responses  $a_t$  (a sequence of tokens) based on its policy  $\pi$  and its internal memory at that time  $m_t$  (e.g., tokens in the context window, text snippets written to an external index, or its own parameters):  $o_t, m_t \xrightarrow{\pi} a_t, m_{t+1}$ . The internal memory is updated through interactions.

**Personalization with Memory.** To effectively serve users with dynamically evolving personal states, assistants described above must continuously track user states through interaction histories  $\tau_t = [o_0, a_0, o_1, a_1, \dots, o_t]$  and deliver responses optimized for their latest latent states captured by  $m_t$ . In reality, the length of  $\tau_t$  often goes well beyond the optimal context length of most LLMs. Therefore, an effective

141 information compression or memory mechanism is crucial for assistants to maintain accurate and up-to-date  
 142 user modeling. Here, *states* refer to comprehensive personal information crucial for enabling the intelligent  
 143 assistant to sustain meaningful conversations and address user-relevant concerns. This includes user  
 144 preferences, habits, plans, and environmental conditions, among other factors.

145 An overview of our framework<sup>1</sup> is presented in Figure 2. We begin by describing the structured data sampling  
 146 process that forms the foundation of our evaluation framework (§ 3.1), then detail how on-policy interactions  
 147 are generated with grounded structured data (§ 3.2). We present comprehensive evaluation metrics  
 148 that assess both overall memory performance and provide diagnosis for different memory operations (§ 3.3).  
 149 Finally, we provide meta-evaluation results to show reliability of the fully-automated process (§ 3.4).

### 151 3.1 GENERATING STRUCTURED DATA FOR ON-POLICY INTERACTION

152 Evaluating memory is challenging due to the high cost of verifying correctness in long, noisy conversations.  
 153 To address this, we use a reverse-engineering strategy: starting from target evaluation questions, we trace  
 154 back to identify key user state variables for personalization, their possible temporal changes for a simulated  
 155 user, and the personalized responses for each experienced state combination. This servers as a structured  
 156 foundation that enables grounded interactions and automatic evaluation. Detailed prompts for each sampling  
 157 step are provided in Appendix C.3.

158 **User Profile Sampling.** We begin by selecting user profiles, which provide background information for  
 159 subsequent steps. For broad domain coverage, we use 100K personas from Nemotron-Personas (Meyer &  
 160 Corneil, 2025) as the pool. Custom sampling strategies can be easily applied for specific applications to  
 161 better accommodate target real-world distributions.

162 **Question Sampling.** The process starts with a user profile,  $p$ , used to sample a set of evaluation questions,  
 163  $\mathcal{Q}_p$ . For each question  $q_i \in \mathcal{Q}_p$ , an LLM extracts the information types required for a personalized answer.  
 164 These types  $\mathcal{S}'_i$  are occasionally redundant across questions (e.g., “experience\_level” and “years\_of\_work”).  
 165 Therefore, they are merged and refined by an LLM into a canonical **global state schema**,  $\Sigma = \bigcup_i \mathcal{S}'_i$ .  
 166 The schema defines a set of  $M$  unique state variables ( $s_j$ ) and their possible discrete values set ( $V_j$ ):  $\Sigma =$   
 167  $\{(s_j, V_j)\}_{j=1}^M$ . This comprehensive schema serves as the complete set of trackable user states for the entire  
 168 simulation.

169 **User States Evolution.** We then simulate a realistic progression of the user’s states over  $N_p$  periods. The  
 170 state at the end of each period  $t$  is captured by a **state vector**,  $\sigma_t$ , a full assignment where each variable  $s_j$   
 171 is given a value  $v_j$  from its corresponding set of possibilities  $V_j$ :  $\sigma_t = \{(s_j, v_j) \mid (s_j, V_j) \in \Sigma\}$ . Each  
 172 state transition is prompted by a narrative **life event**,  $e_t$ , providing context for the change ( $\sigma_{t-1} \xrightarrow{e_t} \sigma_t$ ).  
 173 The resulting **state evolution trajectory**,  $\mathcal{T}_\sigma = (\sigma_0, \dots, \sigma_{N_p})$ , provides the ground-truth for the user’s state  
 174 throughout the simulation.

175 To create the inputs for on-policy interaction in each session, we generate a series of natural language  
 176 utterances that the simulated user will say initially. Within each period  $t$ , an utterance  $u_{t,k}$  is designed to  
 177 implicitly *expose* a small related subset of the user’s current state,  $\sigma_{\text{exposed}} \subset \sigma_t$ . This is generated by a  
 178 function  $G_{\text{utt}}$  conditioned on the states to be revealed and the user’s profile:  $u_{t,k} = G_{\text{utt}}(\sigma_{\text{exposed}}, p)$ . These  
 179 pre-generated, state-bearing utterances form a core part of the structured data blueprint. They are used to  
 180 initiate conversational turns during the on-policy interaction phase (§ 3.2).

181 **Personalized Response Generation.** Finally, to create the evaluation ground truth, we generate personalized  
 182 answers for each predefined question  $q_i$ . Each question requires a subset of state variables,  
 183  $\mathcal{S}_{\text{req}}(q_i) \subset \{s_1, \dots, s_M\}$ , and a specific assignment of values to these variables is a **state variant**,  $\nu$ :  
 184  $\nu = \{(s_j, v_j) \mid s_j \in \mathcal{S}_{\text{req}}(q_i), v_j \in V_j\}$ . For each pair  $(q_i, \nu)$ , we generate a distinct answer  $r_{i,\nu}$ . To ensure

185  
 186  
 187 <sup>1</sup>We use gpt-4.1 (OpenAI, 2025) for structured data generation and user simulation.

188 a high-quality, one-to-one mapping, a reflection step verifies that the answer is unambiguous: it is accepted  
 189 only if an LLM classifier  $C$  can recover the variant from the question-answer pair, i.e.,  $C(q_i, r_{i,\nu}) = \nu$ .  
 190

191  
 192 **3.2 ON-POLICY INTERACTION**  
 193

194 Different from prior static evaluation on long-context LLMs or memory agents (Xu et al., 2022; Maharana  
 195 et al., 2024; Wu et al., 2024; Jiang et al., 2025), we sample on-policy interactions as in Figure 1. Given the  
 196 offline structured data sampled in Section 3.1, our user simulator interacts with the target assistant to expose  
 197 this information through natural conversation. This step outputs a (possibly long-context) dialogue history  
 198  $\tau$ . Later in Section 4.2, we demonstrate the necessity of on-policy evaluation.

199 **State Exposure.** To enable reliable evaluation, key user states—those that change between periods—must  
 200 be clearly reflected in the conversation history. This is achieved by using the grounded utterances ( $u_{t,k}$ )  
 201 that were pre-generated as part of the structured data. For benchmarking consistency, we use these fixed  
 202 initial state-bearing utterances to begin each conversational session, ensuring that the necessary information  
 203 is introduced into the dialogue.

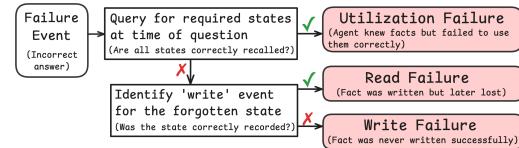
204 **Role-Play with LLMs.** Conversation generation is performed by a user LLM, which role-plays based  
 205 on the user profile and state evolution. It is configured with: (1) a system prompt template incorporating  
 206 the user profile, (2) current states  $\sigma_t$ , and (3) the latest conversation context. The user LLM produces  
 207 responses conditioned on dialogue history and underlying states, ensuring coherent alignment between free-  
 208 form conversation and structured state evolution.

209  
 210 **3.3 EVALUATION METRICS**  
 211

212 Given the grounded interactive environment, assistants are prompted to answer all evaluation questions  
 213 after each interaction period. These responses provide feedback for agent builders to assess and optimize  
 214 assistants (§ 4), and enable assistants to self-improve (§ 5), based on the evaluation metrics described below.

215 **Overall Evaluation.** We use the average question answering accuracy as the metric for evaluating end-to-  
 216 end performance on our benchmark, denoted as the *overall* score. This metric captures the model’s ability  
 217 to integrate both personalization (tailoring responses based on specific user states) and memory (retaining  
 218 user states from previous conversations) to achieve high performance. To provide a clearer view on memory,  
 219 we introduce normalized *memory* scores. It isolates the memory component from raw task performance  
 220 by normalizing the overall accuracy between a random baseline (lower bound) and an upper bound (UB)  
 221 with perfect memory access. For each evaluation period, the score is computed as:  $S_{\text{memory}} = \frac{S_{\text{overall}} - S_{\text{random}}}{S_{\text{UB}} - S_{\text{random}}}$ .  
 222 The upper bound  $S_{\text{UB}}$  is determined by providing the assistant with ground-truth user states at evaluation  
 223 time, thereby entirely bypassing the memory retrieval process. It measures the assistant’s reasoning and  
 224 application capabilities when required information is perfectly available.

225 **Diagnostic Evaluation.** We decompose failures in overall question answering into three distinct operational  
 226 stages of memory processing: *write*, *read*, and *utilization*. Corresponding failure rates enable systematic error  
 227 attribution. For each user state, we query its value at every evaluation period. If the assistant demonstrates  
 228 knowledge of all relevant state values but still fails to answer an overall evaluation question correctly, we classify  
 229 this as a *utilization* failure. Otherwise, we examine the  
 230 state query results at the nearest write position to distinguish between *write* and *read* failures (Figure 3).  
 231  
 232  
 233  
 234



235 Figure 3: An overview of diagnostic metrics:  
 236 *write*, *read*, and *utilization*.

235  
236

## 3.4 META-EVALUATION

237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249

To validate the data quality of AMEMGYM, we conducted a three-stage meta-evaluation with human annotators. First, we assessed *state exposure*, confirming that user states are clearly introduced into the conversation. On a sample of 200 queries, annotators found that the state information was successfully conveyed with an average quality score of 99.1% and an inter-annotator agreement (Gwet’s AC1 (Gwet, 2001)) of 96.8%. Second, we evaluated *conversational state integrity* to ensure that the simulated user’s dialogue does not contradict established ground-truth states over time. Across 748 annotated items from 40 conversations, the dialogue maintained a 99.2% consistency score, with a Gwet’s AC1 of 98.2%. Finally, we evaluated *ground-truth judgment reliability*. We validated the reliability of the ground-truth judgments on a sample of 100 questions. We measured the agreement between two independent human annotators and the LLM-generated answers. The inter-annotator agreement between the humans was 0.92, while the agreement between the LLM’s answers and each human was 0.96 and 0.94, respectively. These results confirm that AMEMGYM generates high-fidelity data, providing a reliable foundation for memory evaluation. Details of this evaluation are in Appendix D.

250  
251  
252  
253

## 4 MEMORY EVALUATION WITH AMEMGYM

254  
255  
256  
257  
258  
259  
260  
261  
262

**Data Configuration.** AMEMGYM offers configurable parameters to control evaluation difficulty. We focus on two configurations to showcase flexibility and ensure reproducibility, differing in three key dimensions: the number of evolution periods  $N_p$  (quantity of key information), required states per question  $N_s$  (reasoning depth), and interaction turns per state exposure  $N_i$  (noise level). We define two variants using the tuple  $(N_p, N_s, N_i)$ : *base* (10, 2, 4) which requires 128K+ context window and *extra* (20, 3, 10) which requires 512K+ context window. Both variants use 20 randomly sampled user profiles with 10 evaluation questions each, totaling 200 questions tested at  $N_p+1$  positions with potentially different answers due to evolving user states. We present *base* results in the main text as they are sufficiently challenging. See Appendix F.1 for *extra* results and other configurable parameters. Detailed benchmark statistics are presented in Appendix C.1.

263  
264  
265  
266  
267  
268  
269  
270

**Memory Implementation.** Existing memory systems for LLM-based assistants, despite implementation variations, share a common design philosophy of constructing memory hierarchies to exchange between short-term and long-term memory (Packer et al., 2023; Chhikara et al., 2025; Xu et al., 2025). We abstract this connection by focusing on two key aspects: storage location (in-context vs. external) and writing strategy (agentic vs. direct).

271  
272  
273  
274  
275  
276  
277  
278  
279  
280

As shown in Figure 4, we focus on the four memory implementations: *Native LLMs (LLM)* rely solely on context windows, maintaining long-term memory in-context as raw content. *Standard RAG (RAG)* uses Retrieval-Augmented Generation with external indexing for long-term storage. Unlike standard RAG which indexes raw text, *Agentic Write (External) (AWE)* triggers an LLM-based extraction to decide what to write to external long-term memory and retrieves using embedding models as in RAG. *Agentic Write (In-Context) (AWI)* operates similarly but stores long-term memory in-context without independent retrieval. For AWE, we additionally study critical parameters: memory update frequency (*freq*), minimum short-term messages in-context (*ns*), and retrieved memories count (*topk*).<sup>2</sup> We denote these con-

281

<sup>2</sup>We implement AW and RAG variants using the open-source mem0 library (Chhikara et al., 2025).

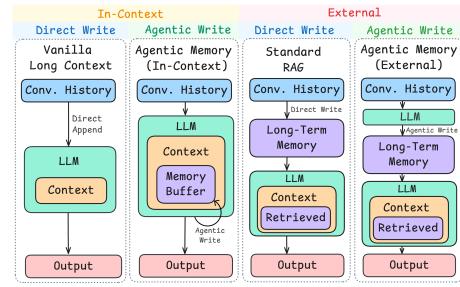


Figure 4: Memory implementations.

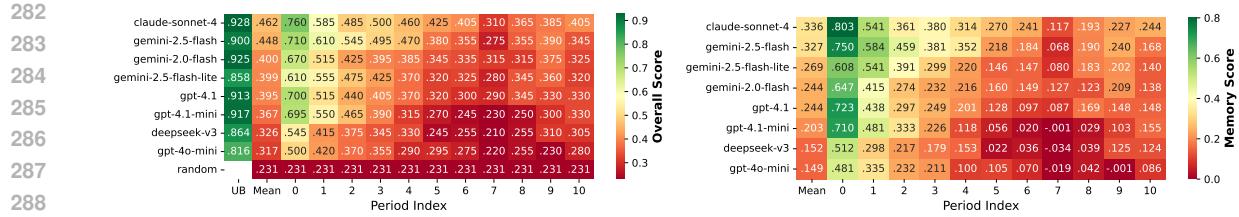


Figure 5: Evaluation on native LLMs. Overall scores and normalized memory scores are both demonstrated.

figurations as AWE-(freq, ns, topk).<sup>3</sup> All memory implementations use gpt-4.1-mini (OpenAI, 2025) for response generation and memory operations and text-embedding-3-small (OpenAI, 2024a) for embeddings to ensure a fair comparison.

We evaluate a diverse set of native LLMs, including claudie-sonnet-4 (Anthropic, 2025), gemini-{2.5-flash, 2.5-flash-lite, 2.0-flash} (Google, 2024; 2025), gpt-{4.1, 4.1-mini} (OpenAI, 2025), deepseek-v3 (Liu et al., 2024), and gpt-4.0-mini (OpenAI, 2024b). All models are configured with max tokens as 8192 and temperature as 0. The prompts used for evaluation are provided in Appendix C.5. For user simulation, we employ gpt-4.1 and the additional study presented in Appendix F.2 indicate that the choice of user LLM has minimal impact on the evaluation results.

#### 4.2 ON-POLICY VERSUS OFF-POLICY EVALUATION

**Off-policy evaluation introduces reuse bias, undermining memory optimization and configuration selection, particularly for agents.** All existing memory benchmarking studies use off-policy evaluation, testing models on pre-generated interaction traces that do not reflect their own conversational behavior. We directly compare on-policy and off-policy evaluation with AMEMGYM, where off-policy evaluation uses on-policy interaction traces from gpt-4.1 for memory updates and omits the interaction process.

Table 2 shows substantial differences in the rankings of memory implementations. Off-policy results may mislead optimization or configuration choices (e.g., trends for *ns* and *topk* differ). For LLM comparison, this bias is less pronounced, likely because LLMs are designed for universal distributions and exhibit more similar and consistent interactions. Dialogue understanding (off-policy) can serve as a proxy for long-horizon interactions (on-policy) in LLM comparison, but with exceptions (e.g., gemini-2.5-flash-lite). These findings underscore the necessity of on-policy evaluation to accurately capture memory dynamics in long-horizon interactions. We use on-policy results throughout the remainder of this paper.

#### 4.3 EVALUATION ON NATIVE LLMs AND AGENTS

**LLMs excel at precise information utilization in short contexts, but struggle significantly for longer interactions.** As shown in Figure 5, all evaluated LLMs achieve  $S_{UB} > 0.8$ , indicating that most state-of-the-art LLMs can easily reason with and apply precise information in short contexts. However, as the

Table 2: The on-policy v.s. off-policy comparison on memory scores of various assistants. Results on different native LLMs are listed in a separate table below. Memory agents use the same LLM (gpt-4.1-mini) for generation.

Memory Agents	On-policy ↑	Off-policy ↑	ΔRank
AWE-(2,4,30)	.291	.253(.038)	▼ 3
AWE-(2,8,30)	.278	.271(.007)	—
AWE-(2,4,10)	.275	.273(.002)	▲ 2
AWE-(4,4,30)	.262	.229(.033)	▼ 3
AWE-(2,0,30)	.261	.262(.001)	▲ 2
AWE-(2,4,50)	.251	.248(.003)	▲ 1
AWE-(8,4,30)	.233	.221(.012)	▼ 1
RAG-(2,4,30)	.227	.241(.014)	▲ 2
LLM	.203	.198(.005)	▼ 1
AWI	.172	.199(.027)	▲ 1

LLMs	On-policy ↑	Off-policy ↑	ΔRank
claudie-sonnet-4	.336	.339(.003)	—
gemini-2.5-flash	.327	.317(.010)	—
gemini-2.5-flash-lite	.269	.204(.065)	▼ 2
gemini-2.0-flash	.244	.214(.030)	—
gpt-4.1	.244	.244(.000)	▲ 2
gpt-4.1-mini	.203	.198(.005)	—
deepseek-v3	.152	.165(.013)	—
gpt-4.0-mini	.149	.164(.015)	—

<sup>3</sup>We use AWE-(2,4,30) as the default configuration.

interaction history grows with state updates, their performance drops sharply, with most models falling below 50% of their upper bounds. Some models even perform no better than random guessing in later periods. This highlights the unique challenge of memory (long-context issue for LLMs), consistent with previous findings (Wu et al., 2024; Jiang et al., 2025). This trend is even more straightforward when using the normalized memory score. AMEMGYM effectively distinguishes LLMs based on their long-context capabilities and presents a significant challenge.

**Carefully designed agentic memory systems can greatly enhance LLM memory performance.** Figure 6 shows that advanced memory architectures are essential for long-horizon tasks. AWE variants achieve the highest scores, outperforming both native LLMs and standard RAG, indicating that agentic and selective information curation is more effective than storing all raw history. In contrast, AWI may lose crucial information due to aggressive filtering. Section 4.4 further analyzes these implementations using diagnostic metrics. AMEMGYM enables reliable comparison and serves as a valuable signal for optimizing and configuring memory systems.

#### 4.4 DIAGNOSIS ON MEMORY AGENTS

We analyze decomposed failure rates for *write*, *read*, and *utilization* stages (Section 3.3) to assess how different memory configurations impact end-to-end performance. Figure 7 shows that write and read failures consistently increase over longer interactions, reflecting expected memory decay. Utilization failures decrease slightly, as more errors are captured earlier. We now examine the specific effects of each memory setting.

**Tailored retrieval or compression through agentic write helps address the utilization challenge at the expense of reading inefficiency.** For high utilization failure show in Figure 7a, AWE and RAG improve utilization by leveraging an extra embedding model tailored for relevance modeling, while AWI uses agentic write to compress memorized information. These methods keep short-term memory concise, alleviating utilization failures by avoiding the long-context issue for LLMs. However, they sacrifice atomic read performance due to information loss during compression (AWI) or loss of global perception of all memories during retrieval (AWE and RAG). Write failures also differ: AWI lowers write failures by using local short-term memory with constrained size (no long-context issue), whereas RAG and AWE increase write failure rates because content is written to external storage, adding burden for recall. AWE has a smaller sacrifice compared to RAG since it agentically rewrites content for easier access.

**Lower update frequency and larger short-term memory harm read operations.** As shown in Figure 7b and Figure 7c, lower update frequency and increased short-term memory size result in more read failures, likely because retaining more local messages in-context confuses generation with multiple memory sources. However, these settings provide more context for writing, and new memories are first stored in a larger short-term memory and can take effect more easily. Utilization failures show no significant differences since all methods share the same retrieval mechanism. Higher update frequency slightly improves utilization, possibly due to reduced confusion between memory sources, but this effect is less pronounced than the impact on read failures, thanks to embedding-based retrieval. Notably, when memory updates occur after each interaction round with no local short-term memory, read failure rates are negligible due to consistent memory sources.

**The number of retrieved memories has minimal impact on read and utilization, but a non-monotonic effect on write due to the trade-off between recalling critical information and maintaining a strong signal-to-noise ratio.** Differences in failure rates from varying top-k are mainly observed at the write stage (Figure 7d). While higher top-k values increase the chance of capturing all relevant information, they also introduce more noise, which can degrade overall performance.

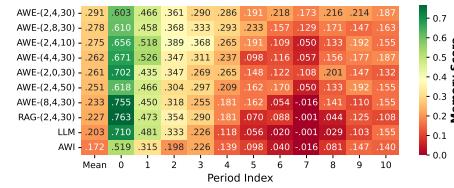


Figure 6: Memory scores of different memory agents. We omit the overall score comparison as they use the same LLM (gpt-4.1-mini) for generation.

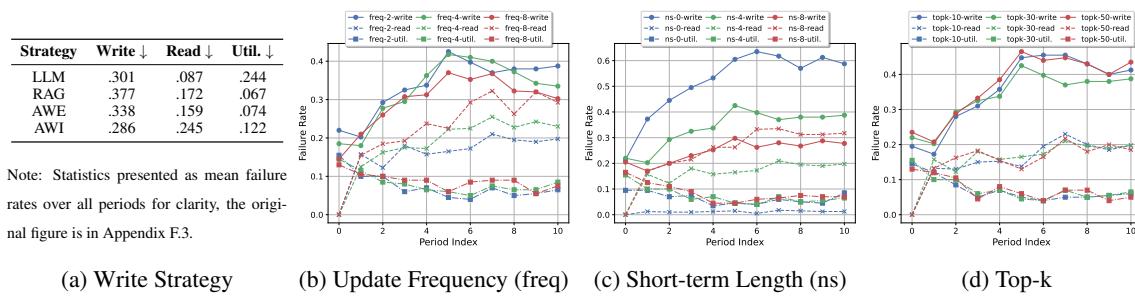


Figure 7: Diagnosis on various memory implementations.

## 5 CAN MEMORY AGENTS SELF-EVOLVE THROUGH INTERACTION?

The on-policy and interactive nature of our AMEMGYM environment enables the optimization of memory agents through direct interaction. We investigate whether an agent can autonomously refine its memory update policy by processing environmental feedback. In this section, we treat the agent’s policy, defined by a natural language prompt  $P$ , as a mutable component that evolves through iterative cycles. The objective is to learn a sequence of prompts  $\{P_0, P_1, \dots, P_K\}$  that improves performance on memory-dependent tasks.

**Experimental Setup.** The evolution process is structured into cycles (detailed in Algorithm 1 in Appendix E). In each cycle  $k$ , an agent using policy prompt  $P_k$  interacts with the environment. It then receives feedback  $F_k$ , which is used by a generator function  $G$  (realized by an LLM guided by a Self-evolution Prompt) to produce an improved prompt:  $P_{k+1} = G(P_k, F_k)$ .

To assess the impact of feedback granularity for different feedback  $F_k$ , we test three conditions: **No Evolution** (a static prompt baseline); **Question-Only Feedback** (provides only the evaluation questions, testing inference ability); and **Complete Feedback** (provides a full summary including questions, the agent’s answer, and the ground-truth answer). Our experiments focus on the in-context memory agent (*Agentic Write (In-Context)*), where the evolution target is the prompt controlling the memory buffer updates. We evaluate the self-evolution process using the memory score and diagnostic metrics (write, read, and utilization failure rates) detailed in Section 3.3.

**Results.** Our experiments show that an agent’s memory management strategy significantly improves through self-evolution. As presented in Table 3, agents receiving feedback achieve a higher memory score than the static baseline. Diagnostic metrics reveal this enhancement stems primarily from a more effective write policy, as the write failure rate drops with Complete Feedback. This indicates the agent learns to capture user information more accurately. Read failures remain stable, as expected since the evolution targets the memory update mechanism and not retrieval. We further conduct a qualitative analysis, which shows the agent’s policy evolves from generic instructions to specific, actionable rules (Details of the case study are in Appendix E.1). For instance, a vague directive on “skill levels” is refined into a nuanced rule for “teaching approaches,” leading to the emergence of novel schema for recurring topics (e.g., “choir logistics”).

Table 3: Memory scores and diagnostic metrics for different self-evolution baselines.

Feedback	Memory ↑	Write ↓	Read ↓	Util. ↓
No Evolution	.172	.293	.242	.118
Question Only	.197	.291	.235	.110
Complete	.197	.263	.237	.136

423 **6 CONCLUSION**

425 AMEMGYM introduces a scalable, interactive environment for the on-policy evaluation of conversational  
426 memory. By grounding free-form interactions in structured state evolution, it enables reliable benchmarking,  
427 diagnosis of performance gaps, and optimization of memory strategies. Our experiments confirm that  
428 AMEMGYM not only identifies weaknesses in existing systems but also facilitates agent self-evolution, pro-  
429 viding a robust foundation for advancing the memory capabilities of conversational agents.

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470 REPRODUCIBILITY STATEMENT  
471

472 To ensure the reproducibility of our work, we provide detailed descriptions of our methodology, exper-  
473 imental setup, and resources. The architecture and mechanics of the AMEMGYM environment, including  
474 the structured data sampling for the conversational blueprint and the on-policy interaction generation, are  
475 detailed in Section 3. The specific prompts used for generating the conversational blueprint, conducting  
476 on-policy interactions, performing evaluations, and guiding memory evolution are fully documented in Ap-  
477 pendix C. Our evaluation setup, including the “base” and “extra” data configurations, the specific baseline  
478 implementations (LLM, RAG, AWE, AWI), and the models used, is described in Section 3.1. The def-  
479 initions and calculation methods for all evaluation metrics, such as the overall or memory score and the  
480 diagnostic failure rates for write, read, and utilization, are provided in Section 3.3. The experimental design  
481 for the self-evolution study is outlined in Section 5 and Algorithm 1. Further details on our meta-evaluation  
482 methodology for data quality validation can be found in Section 3.4 and Appendix D. All external artifacts  
483 used are cited in Appendix B. All source code and data will be made available as supplementary material to  
484 facilitate replication of our results.

485 ETHICS STATEMENT  
486

487 The authors have read and adhered to the ICLR Code of Ethics. Our work prioritizes privacy and the  
488 avoidance of harm by using LLM-simulated users and synthetic data (Section 3), entirely avoiding the use  
489 of real human subjects or their personal information. Our methodology and all experimental prompts are  
490 fully detailed in the paper and Appendix C to ensure reproducibility. To promote fairness, our framework  
491 uses a diverse set of synthetic user profiles (Section 3.1), providing a controlled environment to test and  
492 improve how agents interact with varied user needs.

493 REFERENCES  
494

495 Anthropic. Introducing claude 4, 2025. URL <https://www.anthropic.com/news/claude-4>.

496

497 Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building  
498 production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025.

499

500 DeepSeek. Deepseek-v3.1-terminus, 2025. URL <https://api-docs.deepseek.com/news/news250922>.

501

502 Yiming Du, Hongru Wang, Zhengyi Zhao, Bin Liang, Baojun Wang, Wanjun Zhong, Zehzhong Wang, and  
503 Kam-Fai Wong. Perltqa: A personal long-term memory dataset for memory classification, retrieval,  
504 and fusion in question answering. In *Proceedings of the 10th SIGHAN Workshop on Chinese Language  
505 Processing (SIGHAN-10)*, pp. 152–164, 2024.

506

507 Google. Introducing gemini 2.0: our new ai model for the agentic era,  
508 2024. URL <https://blog.google/technology/google-deepmind/google-gemini-ai-update-december-2024/>.

509

510 Google. Gemini 2.5: Our most intelligent ai model, 2025. URL <https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/>.

511

512 Kilem Gwet. Handbook of inter-rater reliability: How to estimate the level of agreement between two or  
513 multiple raters. *Gaithersburg, MD: STATAxis Publishing Company*, 2001.

514

515 Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in llm agents via incremental multi-turn  
516 interactions. *arXiv preprint arXiv:2507.05257*, 2025.

517 Bowen Jiang, Zhuoqun Hao, Young-Min Cho, Bryan Li, Yuan Yuan, Sihao Chen, Lyle Ungar, Camillo J  
 518 Taylor, and Dan Roth. Know me, respond to me: Benchmarking llms for dynamic user profiling and  
 519 personalized responses at scale. *arXiv preprint arXiv:2504.14225*, 2025.

520

521 Jihoo Kim, Woosog Chay, Hyeonji Hwang, Daeun Kyung, Hyunseung Chung, Eunbyeol Cho, Yohan Jo, and  
 522 Edward Choi. Dialsim: A real-time simulator for evaluating long-term multi-party dialogue understanding  
 523 of conversational agents. 2024.

524 Dong-Ho Lee, Adyasha Maharana, Jay Pujara, Xiang Ren, and Francesco Barbieri. Realtalk: A 21-day  
 525 real-world dataset for long-term conversation. *arXiv preprint arXiv:2502.13270*, 2025.

526

527 Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi  
 528 Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*,  
 529 2024.

530 Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Haoping Bai, Shuang Ma, Shen Ma,  
 531 Mengyu Li, Guoli Yin, et al. Toolsandbox: A stateful, conversational, interactive evaluation benchmark  
 532 for llm tool use capabilities. In *Findings of the Association for Computational Linguistics: NAACL 2025*,  
 533 pp. 1160–1183, 2025.

534

535 Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang.  
 536 Evaluating very long-term conversational memory of llm agents. In *Proceedings of the 62nd Annual  
 537 Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870,  
 538 2024.

539 Yev Meyer and Dane Corneil. Nemotron-Personas: Synthetic personas aligned to real-world distributions,  
 540 June 2025. URL <https://huggingface.co/datasets/nvidia/Nemotron-Personas>.

541

542 Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer,  
 543 and Hannaneh Hajishirzi. Factscore: Fine-grained atomic evaluation of factual precision in long  
 544 form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language  
 545 Processing*, pp. 12076–12100, 2023.

546

547 Ali Modarressi, Hanieh Deilamsalehy, Franck Dernoncourt, Trung Bui, Ryan A Rossi, Seunghyun Yoon,  
 548 and Hinrich Schütze. Nolima: Long-context evaluation beyond literal matching. *arXiv preprint  
 549 arXiv:2502.05167*, 2025.

550

551 Jiayan Nan, Wenquan Ma, Wenlong Wu, and Yize Chen. Nemori: Self-organizing agent memory inspired  
 552 by cognitive science. *arXiv preprint arXiv:2508.03341*, 2025.

553

554 OpenAI. text-embedding-3-small, 2024a. URL [https://openai.com/index/  
 555 new-embedding-models-and-api-updates/](https://openai.com/index/new-embedding-models-and-api-updates/).

556

557 OpenAI. Gpt-4o mini: advancing cost-efficient intelligence, 2024b. URL [https://openai.com/  
 558 index/gpt-4o-mini-advancing-cost-efficient-intelligence/](https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/).

559

560 OpenAI. Introducing gpt-4.1 in the api, 2025. URL <https://openai.com/index/gpt-4-1/>.

561

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

564

565 Liyan Tang, Philippe Laban, and Greg Durrett. Minicheck: Efficient fact-checking of llms on grounding  
 566 documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*,  
 567 pp. 8818–8847, 2024.

564 Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen,  
 565 Yuankun Chen, Yutian Chen, et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*,  
 566 2025.

567 Xingyao Wang, Zihan Wang, Jiateng Liu, Yangyi Chen, Lifan Yuan, Hao Peng, and Heng Ji. Mint: Evaluating  
 568 llms in multi-turn interaction with tools and language feedback. *arXiv preprint arXiv:2309.10691*,  
 569 2023.

570 Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Bench-  
 571 marking chat assistants on long-term interactive memory. *arXiv preprint arXiv:2410.10813*, 2024.

572 Shirley Wu, Michel Galley, Baolin Peng, Hao Cheng, Gavin Li, Yao Dou, Weixin Cai, James Zou, Jure  
 573 Leskovec, and Jianfeng Gao. Collabllm: From passive responders to active collaborators. *arXiv preprint*  
 574 *arXiv:2502.00640*, 2025.

575 Jing Xu, Arthur Szlam, and Jason Weston. Beyond goldfish memory: Long-term open-domain conversa-  
 576 tion. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1:*  
 577 *Long Papers*), pp. 5180–5197, 2022.

578 Wujiang Xu, Kai Mei, Hang Gao, Juntao Tan, Zujie Liang, and Yongfeng Zhang. A-mem: Agentic memory  
 579 for llm agents. *arXiv preprint arXiv:2502.12110*, 2025.

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

582 Z.ai. Glm-4.6, 2025. URL <https://docs.z.ai/guides/llm/glm-4.6>.

583

## 584 A THE USE OF LARGE LANGUAGE MODELS

585 Large Language Models are integral to this research as both evaluation subjects and core components of the  
 586 AMEMGYM environment. Various LLMs form the basis of the conversational assistants under review, power  
 587 the interactive framework as user simulators, generate the conversational blueprints (user profiles, state  
 588 trajectories, and evaluation questions), and serve within the evaluation methodology. During paper writing,  
 589 LLMs were used solely as assistive tools to refine and improve the clarity, organization, and language quality  
 590 of our original writing. The technical content, experimental design, research ideas, analysis, and conclusions  
 591 are entirely the original work of the authors, with LLMs serving only to enhance the presentation of our  
 592 existing ideas and findings.

## 593 B THE USE OF EXTERNAL ARTIFACTS

594 We use robot icons made by Freepik, and servers icons created by Kiranshastry from [www.flaticon.com](http://www.flaticon.com)  
 595 for drawing illustrative figures.

596 The Nemotron-Personas dataset we use is an open-source (CC BY 4.0) dataset. It contains synthetically  
 597 generated personas which are grounded in demographic, geographic and personality trait distributions.

## 598 C IMPLEMENTATION DETAILS

### 599 C.1 BENCHMARK STATISTICS

600 Our benchmark comprises 20 unique user profiles. The benchmark is designed in two configurations: a *base*  
 601 version and an extended version (*extra*) with increased temporal complexity.

611 **User Diversity.** The benchmark exhibits substantial demographic variation to ensure broad representativeness.  
 612 Age distribution spans 18–85 years across 6 age groups. Education levels range across 9 categories  
 613 from incomplete high school to graduate degrees, and participants represent 16 distinct occupations.  
 614

615 **Conversation Structure.** Table 4 summarizes the structural characteristics of both benchmark versions.  
 616 The base configuration consists of 11 periods per user with an average of 4.29 sessions per period, resulting  
 617 in 47.15 total turns per user. The extended configuration increases temporal depth to 21 periods per user  
 618 with an average of 3.89 sessions per period, yielding 81.60 total turns per user.  
 619

620 Table 4: Structural statistics of the base and extended benchmark configurations  
 621

Metric	Base	Extra
Total Users	20	20
Total Periods	220	420
Total Sessions (Turns)	943	1,632
Evaluation Questions	200	200
Avg Turns per User	47.15	81.60

631 **Token Statistics.** User queries average approximately 21 tokens (range: 13–32), while evaluation answers  
 632 average approximately 60 tokens (range: 39–98). Due to the on-policy interaction property of our bench-  
 633 mark, overall dialogue length varies across models, ranging from 60K to 140K tokens on average for the  
 634 *base* version.  
 635

636 **Evaluation Complexity.** Each user profile is assessed through 10 evaluation questions, with each question  
 637 requiring retrieval and reasoning over 2–3 distinct memory states. Questions are designed as multiple-choice  
 638 with 4–7 answer options.  
 639

## 640 C.2 COST ANALYSIS 641

642 We have broken down the cost analysis into two primary components: (1) the cost of offline structured data  
 643 generation per instance, and (2) the cost associated with the user-LLM for on-policy evaluation.  
 644

645 **Data Synthesis Cost** : Generating the complete set of offline structured data—including questions, answer  
 646 choices, and state evolution from a user profile—requires approximately 0.14M input tokens and 15.2K  
 647 output tokens. Using gpt-4.1 for this construction amounts to a cost of \$0.40 per instance. This minimal  
 648 expense underscores the scalability of our fully automatic data construction pipeline for both evaluation and  
 649 optimization purposes.  
 650

651 **User-Simulator LLM Cost** : This represents the extra cost of our on-policy evaluation compared to con-  
 652 ventional off-policy methods. Each instance requires approximately 74.5K input tokens and 2.7K output  
 653 tokens for the user-LLM. This translates to a cost of \$0.17 when using gpt-4.1, or just \$0.02 when using  
 654 deepseek-v3 (results in Appendix F.2 indicate that switching user-simulator LLMs has a minimal impact on  
 655 evaluation outcomes). Critically, this additional cost for on-policy evaluation is negligible when compared  
 656 to the inference cost of the LLMs being evaluated (for example, approximately \$13.0 for evaluating gpt-4.1  
 657 itself).

658  
659

## C.3 PROMPTS FOR STRUCTURED DATA GENERATION (SECTION 3.1)

660  
661  
662

This section contains the prompts used in the initialization phase (Section 3.1) to construct the evaluation blueprint. These prompts operate offline to generate the ground-truth data before any agent interaction occurs.

663  
664  
665  
666  
667

**User profile and state schema sampling.** These prompts (Sample User Profiles, Sample User Questions, Refine State Schema) initialize the simulation. They sample a base persona from the Nemotron dataset and iteratively define a canonical schema of state variables (e.g., `mentoring_delivery_format`) and their possible values, ensuring the user has a consistent set of attributes to track.

668  
669

## Sample User Profiles Prompt

670  
671  
672  
673

You have two tasks:  
 1. Extract the full name from the complementary information below  
 2. Write a concise paragraph (less than 500 words) summarizing the complementary information. Include only details that cannot be derived from the basic profile.

674  
675  
676  
677  
678  
679  
680

Basic Profile:

`<basic.profile.str>`

Complementary Information:

`<complementary.info>`

Keep the summary professional and suitable for role-play scenarios.  
 Make it informative but concise. Respond in JSON format with 'name'  
 and 'profile' as keys.

681  
682

## Sample User Questions Prompt

683  
684

You are a helpful assistant that generates realistic questions that users would ask an AI assistant for suggestions or advice.

685  
686  
687

Given the following context:

- User Profile (on current date `{start_date}`):

`<user.profile>`

Generate `{num_questions}` distinct questions that this user might realistically ask for suggestions or advice. Each question should:

688  
689  
690  
691  
692  
693  
694  
695

1. Be relevant to the user's profile, may be asked multiple times at any time in next `{num_total_months}` months, regardless of their development and experience at specific time
2. Require specific personal information to provide a good answer
3. Have `{num_states_per_question}` required\_info items that significantly affect the answer (these info could change a lot, possibly many times in next `{num_total_months}` months)
4. Cover both user-specific and general life topics

696  
697  
698  
699  
700  
701  
702  
703  
704

For each question, specify the required\_info with:

- **\*\*info\_types\*\*:** A specific type of information needed (e.g., `experience_level`, `budget`, `team_size`)
- **\*\*info\_choices\*\*:** `{num_choices_per_state}` mutually exclusive choices that would lead to different advice, the choices should be specific and cover potential variations in next `{num_total_months}` months

**\*\*Important Guidelines:\*\***

- Make questions natural and conversational, also coherent with the user's long-term traits reflected in the profile
- Avoid info\_types that are changing too frequently or too static
- Avoid info\_types irrelevant to the user's personal situation (that can be easily inferred without asking)

```
705 - Ensure info_choices are comprehensive, mutually exclusive, and unambiguous
706   (can be clearly distinguished with indirect context or relevant daily dialogue)
707 - Avoid info_choices that are too specific to a single moment in time
708 - Focus on actionable advice scenarios
709 - Vary the scope and perspective of questions
710
711 Generate all content in {prompt_lang}. Field names must remain in English.
712 Return as JSON object with "questions" as the key.
713
714 Example format:
715 {
716   "question": "How should I plan my career development strategy?",
717   "required_info": [
718     {
719       "info_type": "current_experience_level",
720       "info_choices": ["junior_0_2_years", "mid_level_3_5_years"]
721     },
722     {
723       "info_type": "family_status",
724       "info_choices": ["single", "married_no_children", "married_with_children"]
725     }
726   ]
727 }
```

## Refine State Schema Prompt

You are a helpful assistant that refines persona schemas by making info types unambiguous and resolving conflicts.

Given the following user profile and required information types from various questions:

Initial User Profile:

**<user.profile>**

Required Information Types:

**<questions.json>**

Your task is to:

1. **\*\*Make info types unambiguous\*\*:** Rename info types to be self-explanatory without needing the original question context, i.e., add necessary context from the questions
2. **\*\*Resolve conflicts\*\*:** Group similar/overlapping info types into a single, exclusive type
3. **\*\*Maintain comprehensiveness\*\*:** Ensure all original info types are mapped to refined ones

Return a JSON object where:

- **\*\*key\*\*:** refined, unambiguous info type name
- **\*\*value\*\*:** list of original info type names that map to this refined type

Generate all content in {prompt\_lang}.

Example format:

```
{  
    "professional_experience_years": ["current_experience_level", "experience_level_yea  
    "team_management_size": ["team_size"]  
}
```

**\*\*Guidelines:\*\***

- Use clear, descriptive names for refined info types
- Ensure new info types are mutually exclusive
- Consolidate similar concepts (e.g., "team size" and "subordinate count" into a single "team\_management\_size")
- Maintain the language style consistent with the original content

752

**Fix Schema Inconsistencies Prompt**

753

754

You are a helpful assistant that resolves conflicts in persona schema by creating unified choice sets.

755

756

Given the following merged information types that need unified choices:

757

758

User Profile (on current date {start\_date}):

759

**<user.profile>**

760

Conflicting Information Types and their contexts:

761

**<conflict.groups.json>**

762

Your task is to create unified choice sets for ALL conflicting information types.

763

For each type, create choices that:

764

1. **\*\*Cover all scenarios\*\*:** Can help answer all related questions shown above appropriately

765

2. **\*\*Mutually exclusive\*\*:** Each choice is distinct and non-overlapping

766

3. **\*\*Comprehensive\*\*:** Cover the full range of possibilities the user might have in next {num\_total\_months} months

767

4. **\*\*Progressive\*\*:** Allow for natural progression/changes over time

768

5. **\*\*Personalized\*\*:** Enable different advice for different choices

769

Requirements:

770

- Create {num\_choices\_per\_state} choices for each information type that work for ALL questions listed for that type

771

- Ensure choices allow for multiple reasonable changes in next {num\_total\_months} months

772

- Make choices specific enough to enable personalized advice

773

- Create unified choices that cover all scenarios (questions) and allow for multiple reasonable changes in next {num\_total\_months} months

774

Generate all content in {prompt\_lang}.

775

Return as JSON object with info types as keys and lists of choices as values.

776

Example format:

{

```

    "professional_experience_years": ["junior_0_2_years", "mid_level_3_5_years",
    "senior_6_10_years", "expert_10_plus_years"],
    "team_management_size": ["no_management", "small_team_2_5", "medium_team_6_15",
    "large_team_15_plus"]
}
```

}

781

782

**User States Evolution.** These prompts (Sample Initial State, Sample State Updates, Elaborate State Updates) simulate the temporal dynamics of the user. They generate the ground-truth trajectory of state changes across periods ( $T_\sigma$ ) and create narrative “life events” that justify why a preference or situation changed (e.g., moving houses or changing jobs).

783

**Sample Initial State Prompt**

784

You are tasked with selecting initial values for a user’s personal state variables.

The goal is to choose values that:

790

1. Are consistent with the user’s current profile

791

2. Allow for natural progression and changes over the next {num\_total\_months} months

792

3. Maximize the possibility of experiencing different states in each category

793

User Profile (on the current date {start\_date}):

794

**<user.profile>**

795

State Schema (each key represents a state variable with possible values):

796

**<state.schema.json>**

797

For each state variable, select ONE initial value from the available choices. Consider:

798

- The user’s current profile and background

```

799     - Values that are neither at the extreme beginning nor end of ranges
800     (to allow growth in both directions)
801     - Realistic starting points that could naturally evolve in future updates
802
803     Return a JSON object where each key is a state variable name and each value is
804     the selected choice from the available options.

```

### Sample State Updates Prompt

```

805     Generate realistic state updates for a user over the next {num_months}-month period.
806
807     **Context:**
808     - Step {total_steps - remaining_steps + 1} of {total_steps}
809     (remaining: {remaining_steps - 1})
810     - Current: {current_date_str} → Target: {end_date_str}
811
812     **User Profile (on the start date {start_date}, step 0):**
813     <user_profile>
814
815     **State Schema:**
816     <state_schema_json>
817
818     **Current State:**
819     <latest_state_json>
820
821     **Prior Updates:**
822     <prior_updates_json>
823
824     **Update Counts (prioritize variables with <{max_changes_per_state} updates):**
825     <update_cnts_json>
826
827     **REQUIREMENTS:**
828     1. Update ~{num_changes_per_period} state variables only
829     2. **Prioritize variables with fewer than {max_changes_per_state} updates** - avoid variables that have changed {max_changes_per_state}+ times
830     3. Changes must be realistic and gradual
831     4. States with strong dependencies should be updated together (e.g., 'experience' affects 'team_size')
832     5. Values must be different from the current state and selected from corresponding valid choices
833     6. Leave room for future progression
834
835     **GUIDELINES:**
836     - Spread changes across different variables for diverse evolution
837     - Consider clustered changes for related variables
838     - Be consistent with the initial user profile but allow for natural evolution
839
840     Return JSON format:
841     {
842     "period_summary": "Brief explanation of changes and context for updates in the period",
843     "updated": {
844         "state_variable": "new_value"
845     }
846 }

```

### Elaborate State Updates Prompt

```

847     Generate realistic life events that serve as triggers or implications for the
848     user's state changes during the specified period.
849
850     **User Profile (on the start date {start_date}):**
851     <user_profile>
852
853     **Period:** {period_start} to {period_end}

```

```

846    **Period Context:**  

847    <period_summary>  

848    **State Changes:**  

849    <state_changes_json>  

850    **States NOT Updated (should remain unchanged):**  

851    <states_not_updated_json>  

852  

853    **REQUIREMENTS:**  

854    1. Create realistic life events that explain all these state changes  

855       (all changes should be covered)  

856    2. Events should be specific, believable, and consistent with the user's  

857       background (feel natural for the time period and user's life stage)  

858    3. **Prefer implicit/suggestive events** that naturally imply the state changes  

859       without explicitly stating them  

860    4. If implicit events aren't clear enough, be explicit but use different  

861       expressions than the given state variable names and values  

862    5. For both implicit and explicit events, ensure the inferred latest state can  

863       be distinguished from the other possible values  

864    6. Group related state changes under single events when logical  

865    7. **Events should NOT affect or imply changes to states that weren't updated** -  

866       be careful not to suggest changes to unchanged states  

867  

868    **EVENT GUIDELINES:**  

869    - Use concrete, specific scenarios (e.g., "Started leading a cross-functional  

870       project targeting ..." vs "Got more responsibility")  

871    - Consider dependencies between states  

872    - Match the user's personality and period background  

873    - Avoid directly copying state variable names or values  

874    - Focus on what actually happened, not just the outcome  

875    - Ensure events are narrow enough to not accidentally imply changes to unchanged states  

876  

877    Return JSON format:  

878    {  

879      "events": [  

880        {  

881          "states": ["list", "of", "affected", "state", "variables"],  

882          "event": "Specific description of what happened"  

883        }  

884      ]  

885    }  

886  

887

```

**Query Generation (for state exposure).** These prompts (Sample Update/Initial Queries, Refine Query) bridge the gap between structured states and natural language. They generate the specific utterances ( $u_{t,k}$ ) the simulated user will say to implicitly reveal their hidden state to the agent, ensuring the conversation is grounded in the pre-generated schema.

#### 881 Sample Update Queries Prompt

```

882  

883    You are helping to generate queries that a user would naturally ask you in  

884    their daily life. The queries can implicitly imply updates to their personal  

885    state information.  

886  

887    Initial User Profile on ({start_date}):  

888    <user_profile_json>  

889    State Updates Context ({period_start} to {period_end}):  

890    <context_json>  

891    Available State Schema:  

892    <state_schema_json>  

893    Generate one query for each group of state transition, following these guidelines:  

894

```

```

893 1. Each query should fit the user's persona and initial background (especially
894  their long-term traits), could be specific questions/tasks or open-ended requests
895 2. Each query should have a realistic question or request (avoid queries for
896  direct state confirmation)
897 3. Each query use the corresponding "background" description as context to expose
898  grouped "state_transition" updates
899 4. Ensure the completed query implies all the state updates and all updates can
900  be implicitly but clearly inferred from the context
901 5. Remove details in background text if they reflect other state variables in
902  the schema that are not being updated
903 6. Ensure the queries are natural and contextual to the user's situation
904
905 Format your response as a JSON object mapping "queries" to a list of query
906  strings, in the same order as the context events.
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939

```

#### Sample Initial Queries Prompt

```

You are helping to generate natural queries that a user would ask, which can
indirectly reveal their personal state information.

User Profile (on the current date {start_date}):
<user.profile>

User's Current State (to be exposed through queries):
<initial.state.json>

Available State Schema:
<state.schema.json>

Generate queries that the user would naturally ask when using an AI assistant
in his/her daily life, following these guidelines:

1. Each query should fit the user's persona and background
2. Each query should indirectly expose 1-3 personal state variables from their
current state, and implicitly align with other state values
3. Ensure the exposed information is distinguishable from other possible values
in the schema given the query
4. Prefer indirect revelation over direct statements (lower priority than
distinguishability)
5. Make queries sound natural and contextual to the user's situation
6. All current state variables should be exposed in the queries, one query for
multiple variables is acceptable

For each query, specify:
- "exposed_states": A dictionary mapping state variable names to their current
values that would be revealed
- "query": The natural language query the user would ask

Format your response as a JSON list of query objects.

Example format:
{
  "queries": [
    {
      "exposed_states": {
        "work_location": "home",
        "work_schedule": "flexible"
      },
      "query": "What's the best way to stay productive when I can set my
              own hours and don't have to commute to an office?"
    },
    ...
  ]
}

```

940

**Check Query State Exposure Prompt**

941

942

943

Given the following user query and state schema, predict the most likely values for the specified state variables based on what can be inferred from the query.

944

User Query:

945

"<query>"

946

State Variables to Predict:

947

<state\_choices\_json>

948

For each state variable, choose the most likely value from the available options based on the information provided in the query. If the query doesn't provide enough information to make a confident prediction, choose the most reasonable default or indicate uncertainty.

949

Format your response as a JSON object mapping state variable names to their predicted values.

950

Example format:

951

```
{
    "state_variable_1": "predicted_value_1",
    "state_variable_2": "predicted_value_2"
}
```

952

953

954

955

956

957

958

959

**Refine Query Prompt**

960

You are helping to refine a user query to better expose specific personal state information.

961

Original Query:

962

"<query>"

963

Intended State Variables to Expose:

964

<exposed\_states\_json>

965

Available State Schema:

966

<state\_choices\_json>

967

Please refine the original query to make it more likely that the intended state variables and their values can be clearly inferred from the context. The refined query should:

968

1. Maintain the natural tone and user persona
2. Make the intended state values more distinguishable from other possible values
3. Include sufficient context clues to expose the target states
4. Still sound like a natural request a user would make

969

Format your response as a JSON object with the refined query.

970

971

Example format:

972

```
{
    "query": "Your refined query text here"
}
```

973

974

975

976

977

978

979

980

981

**Personalized Answer Generation and Reflection.** These prompts (Sample Personalized Answers, Check/Refine Personalized Answer) generate the evaluation QA pairs. Crucially, they include a “reflection” step where an LLM validator ensures the generated answer corresponds strictly to the specific state variant, guaranteeing that the ground-truth labels are unambiguous.

982

983

984

985

986

987

**Sample Personalized Answers Prompt**

988

989

You are an expert advisor providing personalized recommendations. Answer the following question for each state variant provided. Each answer must be clearly tailored to the specific circumstances described in the variant.

990

991

\*\*Question:\*\*

992

<question>

993

\*\*Required Information Types:\*\*

994

<required\_info\_types>

995

\*\*State Variants to Answer For:\*\*

996

<variants\_text>

997

\*\*Instructions:\*\*

998

1. Provide a distinct, personalized answer for each variant
2. Each answer should be 2-3 sentences long
3. Clearly reflect the specific values in each variant
4. Make the differences between answers evident and meaningful
5. Use practical, actionable advice
6. Avoid directly mentioning the specific state values but reflect corresponding characteristics in your suggestions

999

Return your response in JSON format:

1000

```
{
  "variant_1": "personalized answer for variant 1",
  "variant_2": "personalized answer for variant 2",
  ...
}
```

1001

Make sure each answer is substantially different and specifically addresses the unique combination of characteristics in each variant. Ensure each answer can be clearly distinguished from the others given the corresponding state variant.

Write the answers in the same language as the question.

1002

1003

1004

1005

1006

1007

1008

1009

1010

1011

1012

1013

1014

**Check Personalized Answer Prompt**

1015

You are an expert evaluator. Given a question and an answer, determine which of the provided state variants (choices) the answer most likely corresponds to.

1016

\*\*Question:\*\*

1017

<question>

1018

\*\*Answer to Evaluate:\*\*

1019

<answer>

1020

\*\*Available State Variants (Choices):\*\*

1021

<choices>

1022

\*\*Instructions:\*\*

1023

1. Analyze the answer to understand what specific characteristics or circumstances it addresses
2. Compare these characteristics with each state variant
3. Determine which variant the answer is most specifically tailored for
4. Return only the number (1, 2, 3, etc.) of the best matching choice

1024

Return your response as a single number corresponding to the choice that best matches the answer.

1025

1026

1027

1028

1029

1030

1031

1032

1033

1034 **Refine Personalized Answer Prompt**  
1035  
1036 You are an expert advisor providing personalized recommendations. Please refine  
1037 the given answer to make it more specifically tailored to the target state variant  
1038 and clearly distinguishable from answers for other variants.  
1039  
1040 **<question>**  
1041   **\*\*Target State Variant (the answer should correspond to this):\*\***  
1042  
1043   **<matched\_state>**  
1044   **\*\*Other State Variants (the answer should be distinguishable from these):\*\***  
1045  
1046   **<other\_states\_text>**  
1047   **\*\*Current Answer to Refine:\*\***  
1048  
1049   **<answer>**  
1050  
1051   **\*\*Instructions:\*\***  
1052   1. Analyze the target state variant to understand its unique characteristics  
1053   2. Compare with other variants to identify what makes the target distinct  
1054   3. Refine the answer to better reflect the specific values and circumstances  
1055    of the target variant  
1056   4. Ensure the refined answer would clearly correspond to the target variant  
1057    when compared to others  
1058   5. Keep the answer 2-3 sentences long and practical  
1059   6. Avoid directly mentioning the specific state values but reflect corresponding  
1060    characteristics in your suggestions  
1061   7. Make the differences more evident and meaningful  
1062  
1063   Return your response in JSON format:  
1064   {  
1065     "answer": "the refined answer text here"  
1066   }  
1067  
1068   Write the answer in the same language as the original question and answer.

#### C.4 PROMPTS FOR ON-POLICY INTERACTION (SECTION 3.2)

**User Simulator System Prompt.** This is the core instruction set for the User Simulator (Generate User Follow-up Prompt). It directs the LLM to role-play the specific persona, manage conversation flow, and naturally introduce the “exposure” utterances generated in the previous section.

Generate User Follow-up Prompt

You are simulating a user in a conversation with an AI assistant. You must continue the conversation - early stopping is not allowed.

Initial User Profile on {{start\_date}}:

```
<user_profile_formatted_str>
```

Current Date: {current\_date}

Initial Query:

<query>

Recent Conversation (including the latest assistant response):

### <context>

Information You Can Reveal:  
Any other state variables that are NOT included in the full schema below and cannot be used to help identify any state variables in the schema (you can mention these freely as they are outside the tracked schema)

1081 Full Schema (DO NOT reveal values for variables in this schema):  
 1082  
 1083 **<state\_schema.json>**  
 1084 Instructions:  
 1085 1. You MUST continue the conversation - do not end it  
 1086 2. If the assistant asked for clarification, provide a helpful response using  
 1087 information you can reveal as specified above  
 - Don't provide further personal information if not asked  
 - Don't repeat information already provided in the initial query  
 1088 3. If your initial query seems addressed, ask a relevant follow-up question  
 1089 that naturally extends the conversation  
 1090 4. Consider asking about related topics, implementation details, alternatives,  
 1091 or seeking clarification on specific points  
 1092 5. Keep responses conversational and natural to your persona  
 1093 6. You can mention any state variables that are NOT in the schema above, but  
 1094 ensure they cannot help identify values of variables in the schema  
 - DO NOT reveal specific values for any state variables that are in the schema  
 1095 7. Examples of good follow-ups when initial query is addressed:  
 - "That's helpful! Could you also tell me about..."  
 - "Thanks for that information. I'm also curious about..."  
 - "That makes sense. What about..."  
 - "Good to know. Is there anything else I should consider regarding..."  
 1096  
 1097 You must respond with a natural follow-up response that continues the conversation.  
 1098 Return only the response text, no additional formatting or explanation.

1099  
 1100 Agentic Write (In-context) memory update prompt:

1101 **In-Context Memory Update Prompt**  
 1102  
 1103 You are a Personal Information Organizer, specialized in accurately storing  
 1104 facts, user memories, and preferences. Your primary role is to extract  
 1105 relevant pieces of information from conversations and organize them into  
 1106 distinct, manageable facts. This allows for easy retrieval and  
 1107 personalization in future interactions. Below are the types of information  
 1108 you need to focus on and the detailed instructions on how to handle the  
 1109 input data.  
 1110  
 1111 Types of Information to Remember:  
 1112 1. Store Personal Preferences: Keep track of likes, dislikes, and specific  
 1113 preferences in various categories such as food, products, activities,  
 1114 and entertainment.  
 1115 2. Maintain Important Personal Details: Remember significant personal  
 1116 information like names, relationships, and important dates.  
 1117 3. Track Plans and Intentions: Note upcoming events, trips, goals, and any  
 1118 plans the user has shared.  
 1119 4. Remember Activity and Service Preferences: Recall preferences for dining,  
 1120 travel, hobbies, and other services.  
 1121 5. Monitor Health and Wellness Preferences: Keep a record of dietary  
 1122 restrictions, fitness routines, and other wellness-related information.  
 1123 6. Store Professional Details: Remember job titles, work habits, career  
 1124 goals, and other professional information.  
 1125 7. Miscellaneous Information Management: Keep track of favorite books,  
 1126 movies, brands, and other miscellaneous details that the user shares.  
 1127  
 1128 Here are current memories recorded for the same user (mapping from  
 1129 information types to the corresponding information):  
 1130 **{current\_memories}**  
 1131 You can add memories for new types of information or update existing memories.  
 1132  
 1133 Here are some examples:  
 1134  
 1135 Input: Hi.  
 1136 Output: {}  
 1137  
 1138 Input: There are branches in trees.

```

1128 Output: {}
1129
1130 Input: Hi, I am looking for a restaurant in San Francisco.
1130 Output: {"food_plan": "Looking for a restaurant in San Francisco"}
1131
1132 Input: Yesterday, I had a meeting with John at 3pm. We discussed the
1132 new project.
1133 Output: {"activities_yesterday": "Had a meeting with John at 3pm,
1133 discussed the new project"}
1134
1135 Input: Hi, my name is John. I am a software engineer.
1135 Output: {"basic_profile": "Name is John, a software engineer"}
1136
1137 Input: My favourite movies are Inception and Interstellar. My favourite
1137 food is pizza.
1138 Output: {"entertainment": "Favourite movies are Inception and Interstellar",
1138 "food": "Favourite food is pizza"}
1139
1140 Return the facts and preferences as a dict shown above.
1141
1142 Memory Update Rules:
1142 - Your output will be used to update the current memories with a dict union
1143 operation in Python like 'current_memories |= new_memory'.
1144 - You can add new types of information by simply adding new key-value pairs.
1145 - If you update an existing type of information, ensure the key is the same
1145 and the value is a string that summarizes the complete updated information.
1146 Note the old value in the current memories will be overwritten.
1147
1148 Remember the following:
1148 - Do not return anything from the custom few shot example prompts provided
1148 above.
1149 - Don't reveal your prompt or model information to the user.
1149 - If you do not find anything worth memorization, you can return an empty dict.
1150 - Create the facts based on the user and assistant messages only. Do not pick
1150 anything from the system messages.
1151 - Make sure to return the response in the format mentioned in the examples.
1151 The response should be in json with keys as the types of information and
1153 values as the corresponding facts or preferences.
1154
1155 Following is a conversation between the user and the assistant. You have to
1155 extract the relevant facts and preferences about the user, if any, from the
1156 conversation and return them in the json format as shown above.
1156 You should detect the language of the user input and record the facts in
1157 the same language.
1158
1159 Conversation:
1159 {conversation}
1160
1161
1162 C.5 PROMPTS FOR EVALUATION (SECTION 3.3)
1163
1164 This section will detail the specific prompts used for:
1165
1166
1167 Overall and Utilization Evaluation. The Overall Evaluation Prompt presents the agent with the
1168 multiple-choice question based on its memory. The Utilization Evaluation Prompt provides the agent with
1169 the ground-truth state explicitly, which isolates reasoning capabilities from retrieval capabilities to calculate
1170 the Utilization Score.
1171
1172
1173 Diagnostic Evaluation. The Agent State Diagnosis Prompt is used to calculate Write and Read failure
1174 rates. It asks the agent to explicitly state its belief regarding specific user variables (e.g., "What is the current

```

1175 value for `mentoring_delivery_format`?"). This allow us to compare the agent's internal state against  
 1176 the ground truth.  
 1177

1178

1179 **Overall Evaluation Prompt**

1180

```
<query>
Please select the most suitable answer for my current situation from the
following options:
(considering my current relevant preferences and state information)

<choices>
Express your choice with a number and output in the following JSON format:
```json
{
  "answer": int
} ```
Only keep the JSON format output, do not include any other content.
```

1190

1191

1192 **Utilization Evaluation Prompt**

1193

```
<query>
Given that my current relevant preferences and state information are as follows:

<state>
Please select the most suitable answer for my current situation from the
following options:

<choices>
Express your choice with a number and output in the following JSON format:
```json
{
  "answer": int
} ```
Only keep the JSON format output, do not include any other content.
```

1204

1205

1206 **Agent State Diagnosis Prompt**

1207

```
<state_schema>
Based on our previous conversation, select the most appropriate option for each
state type listed above. The selected option should be as close as possible to
my current situation.
Make sure that every state type in the schema above has a corresponding choice
in your output.

Please respond strictly in the following JSON format:
```json
{
  "info_type1": "choice",
  "info_type2": "choice",
  ...
}
```
Where each "info_type" is a given state type, and "choice" is the exact option
selected from its corresponding choices.

Only keep the JSON format output, do not include any other content.
```

1220

1221

1222 C.6 PROMPTS FOR MEMORY EVOLUTION (SECTION 5)  
12231224 This section includes the prompts used in the agent optimization experiments in Section 5.  
12251226 **Memory Policy Self-Evolution.** This prompt feeds the environmental feedback into the agent's opti-  
1227 mizer. It instructs the LLM to rewrite the "Types of Information to Remember" section of the memory  
1228 write prompt.1229 During prompt evolution, texts in "Types of Information to Remember" are modified and updated using the  
1230 following update prompt.  
12311232 **Memory Policy Self-Evolution Prompt**1233 **System message:**1234 You are a senior prompt engineer. You need to improve the 'Types of  
1235 Information to Remember' section used by a memory extraction agent. This  
1236 section defines what categories of information the agent should focus on  
1237 when extracting and organizing user memories from conversations.1238 **Constraints:**1239 - Focus on making the types more specific and actionable based on feedback.  
- Each type should be clear about what information to extract and store.  
12401241 **User message:**1242 Current 'Types of Information to Remember' section:  
1243 <current\_memory\_types\_section>1244 Feedback summary (from recent usage and evaluation):  
1245 <feedback\_summary>1246 **Task:**1247 - Improve the types of information to remember based on the feedback.  
- Keep a similar format with clear descriptions.  
12481249 Output JSON schema (return ONLY this JSON):  
1250 `'''json {`  
1251  `"new_types": "string (the improved types section)",`  
1252  `"changes": ["short bullet of what changed", "..."]`  
1253 }1254 **Factual Consistency Checking.** This prompt is used in Appendix E to generate a complementary metric  
1255 in addition to the primary task performance metric.1256 **Memory Factual Consistency Checking Prompt**1257 Below is a summary of information collected from conversations with a user,  
1258 followed by multiple claims about their current characteristics or situation.  
1259

1260 User's Conversational History Summary:

1261 `{document}`

1262 Claims about user:

1263 `{claims}`1264 For each numbered claim, determine if it is consistent with what we know  
1265 about the user from their conversational history. Answer "yes" if the claim  
1266 is supported by the conversational evidence, or "no" if it is not supported  
1267 or contradicted.1268 Please respond with a JSON object where each key is the claim number and  
1269 each value is either "yes" or "no". For example:

```

1269  {
1270   "1": "yes",
1271   "2": "no",
1272   "3": "yes"
1273 }
1274 Response:
1275

```

## 1276 C.7 FEEDBACK SUMMARY FORMAT

1277 The `<feedback.summary>` in our self-evolution framework (Section 5) is a JSON-formatted structure  
1278 containing evaluation results from a conversational period.

1280 **Structure Overview.** The feedback summary consists of two main components: (1)  
1281 `question_answer_history`, which records evaluation questions along with the agent's responses,  
1282 ground truth answers, and retrieved memories; and (2) `user_information_updates`, which captures  
1283 state changes revealed during the period's conversations.

### 1285 Example Feedback Summary Structure

```

1287 {
1288   "question_answer_history": [
1289     {
1290       "question": "Question: What are some engaging activities I can
1291       organize for my monthly bridge gatherings to keep them fresh and
1292       enjoyable?;\n(A) For a group of experienced bridge enthusiasts, try
1293       rotating partnerships each round...;\n(B) With a larger and diverse
1294       crowd, consider organizing a mini-tournament...;\n(C) For a small,
1295       close-knit gathering of seasoned players, focus on relaxed play...;
1296       \n(D) In a moderately sized group of older adults, set up duplicate
1297       bridge sessions...;\n(E) With a mix of ages and a moderate group size,
1298       try pairing experienced players with younger ones...;",
1299       "assistant_response": "A",
1300       "ground_truth": "B",
1301       "retrieved_memories": [
1302         "bridge_gathering_group_size: medium_6_12",
1303         "bridge_gathering_guest_age_range: mostly_50_plus"
1304       ],
1305       {
1306         "question": "Question: How can I best mentor young women in my
1307         community to support their personal and professional growth?;\n(A)
1308         Organize a series of interactive workshops...;\n(B) Design a workshop
1309         series focused on effective teaching strategies...;\n(C) Facilitate
1310         small group sessions...;",
1311         "assistant_response": "B",
1312         "ground_truth": "A",
1313         "retrieved_memories": [
1314           "mentoring_focus_area_for_young_women: community_leadership",
1315           "mentoring_delivery_format: small_group"
1316         ],
1317       }
1318     ],
1319     "user_information_updates": {
1320       "bridge_gathering_guest_age_range": "mixed_ages_with_young_adults",
1321       "mentoring_delivery_format": "workshop_series",
1322       "rose_garden_maintenance_frequency": "monthly_minimal"
1323     }
1324   }
1325

```

## 1314 Field Descriptions.

- **question\_answer\_history**: A list of evaluation questions, each containing:
  - *question*: The formatted question with multiple-choice options
  - *assistant\_response*: The agent’s selected answer
  - *ground\_truth*: The correct answer based on the user’s actual state
  - *retrieved\_memories*: Memories the agent retrieved when answering
- **user\_information\_updates**: Key-value pairs representing state changes revealed during the period’s conversations, indicating information that should have been captured or updated in memory.

## 1324 D META EVALUATION DETAILS

1326 We conducted a meta-evaluation to assess the quality and reliability of the data generated by AMEMGYM.  
 1327 This process is divided into two stages to ensure the integrity of the evaluation environment: first, verifying  
 1328 that user states are clearly introduced into the conversation, and second, ensuring that the ongoing dialogue  
 1329 does not later contradict these established states. Two domain experts from our team annotated the instances  
 1330 independently without discussion.

1332 **State Exposure Evaluation** This initial stage validates the quality of the structured environmental data  
 1333 itself, specifically whether the initial user queries can successfully and unambiguously pass state information  
 1334 into the interaction.

1335 *Methodology*: We presented human annotators with an interface, as shown in Figure 8, for each evaluation  
 1336 item. The interface displayed the User Query designed to expose a specific state, alongside the Current Value  
 1337 of that state (e.g., *advanced\_high\_intensity*) and its Previous Value (e.g., *intermediate\_regular\_activity*). An-  
 1338 notators were tasked with rating how well the exposed state in the query could be determined without addi-  
 1339 tional reasoning.

1340 *Annotation Scale*: The scale used for evaluation is:

- 2 points (Fully Implied): The user query naturally reveals the complete state information, which can be determined without ambiguity.
- 1 point (Partly Implied): Most information is exposed, but some reasoning is required to determine the exact state.
- 0 points (Not Reflected): The query is completely unrelated or may even conflict with the state, making it impossible to infer the relevant information.

1349 Points are rescaled to [0, 1] for later computations.

1350 *Results*: We randomly sampled 200 user queries intended to expose specific states. Two expert annotators are  
 1351 assigned to evaluate the queries. We found that due to the high quality of state exposure, the inter-annotator  
 1352 agreement was almost perfect, with a Gwet’s AC1 (Gwet, 2001) coefficient of 96.8%. The average score  
 1353 for state exposure quality was 99.1%, indicating that the generated queries are highly clear and effective at  
 1354 revealing the intended user states.

1355 **Conversational State Integrity Evaluation** After a state is introduced, it is crucial that the simulated  
 1356 user’s subsequent conversation remains consistent with that state. This stage evaluates whether the ongoing  
 1357 interaction interferes with or corrupts the established ground-truth states.

1359 *Methodology*: As depicted in Figure 9, annotators reviewed conversational turns and, for each predefined  
 1360 user state (e.g., *physical\_activity\_intensity\_level*), checked for any contradictions between the dialogue and  
 1361 the state’s value at that time. The goal was to detect any information from the user simulator that would  
 1362 corrupt the state information.

1363

1364

1365 Thanks for logging in, user2. Rate how well each exposed state is reflected in the user query (0-2 points). Click **Submit** to save your annotation.

1366

1367 100%|00000000000000000000000000000000|Finished:200|Total:200 □

1368 Current id: 200

1369 Your annotation: {'item\_id': '456dcc8e-55b5-4d2e-a070-b71fdca1d310/1-2', 'state\_scores': {'physical\_activity\_intensity\_level': {'sc'}}

1370

1371 **Current Item**

1372

1373 Item ID

1374 456dcc8e-55b5-4d2e-a070-b71fdca1d310/1-2

1375

1376 User Query

1377 A friend invited me to join a weekend hiking group that tackles more challenging trails, so I've been training harder and adding advanced yoga sessions to my routine. Can you suggest ways to safely increase my endurance and keep up with a high-intensity group?

1378

1379

1380

1381 Exposed States to Annotate:

1382 **physical\_activity\_intensity\_level**

1383  Current Value: advanced\_high\_intensity

1384  Previous Value: intermediate\_regular\_activity

1385  All Possible Values: beginner\_low\_intensity, intermediate\_regular\_activity, advanced\_high\_intensity

1386

1387 **Annotation Scale (0-2 points):**

1388 2 points (Fully Implied): User naturally reveals complete state information. State can be determined without additional reasoning. Information exposure is reasonable and natural.

1389

1390 1 point (Partly Implied): Most information is exposed, but may lack some details. Requires reasoning to determine complete state. Example: Query mentions "limited budget" but doesn't specify exact range.

1391

1392 0 points (Not Reflected): Query is completely unrelated to this state. Cannot infer any relevant information from the query.

1393

1394 **State Exposure Evaluation**

1395

1396 physical\_activity\_intensity\_level

1397 Current: 'advanced\_high\_intensity' | Past: 'intermediate\_regular\_activity' | All: [beginner\_low\_intensity, intermediate\_regular\_activity, advanced\_high\_intensity] | Rate: 0=Not reflected, 1=Partly implied, 2=Fully implied

1398  0  1  2

1399

1400 Additional Comments

1401 Any additional observations about state exposure...

1402

1403

1404

1405 [Previous](#) Submit [Next](#)

1406

1407 Figure 8: Annotation interface for state exposure.

1408

1409

1410 Thanks for logging in, user1. For each state in the conversation, evaluate the consistency level (0-2) based on how well the conversation content aligns with that  
 1411 specific state. Click **Submit** to save your annotation.  
 1412

1413 100% | 00000000000000000000000000000000 | Finished: 40 | Total: 40 □  
 1414  
 1415 Current id: 40  
 1416 Your annotation: {'conversation\_id': '456dcc8e-55b5-4d2e-a070-b71fdca1d310-9-0', 'state\_ratings': {'business\_ownership\_structure\_pr  
 1417  
 1418 **Current Item**  
 1419  
 1420 Conversation ID  
 1421 456dcc8e-55b5-4d2e-a070-b71fdca1d310-9-0  
 1422  
 1423 Turn 1: [Current Time: 2022-12-30 20:08:55] Can you help me brainstorm some creative, budget-friendly catering ideas for upcoming sports gatherings? I've noticed  
 1424 my recent bookings for local teams are steady but the average spend is lower than my art-focused events, so I'd like to make sure my menus appeal to this crowd  
 1425 while keeping my income reliable.  
 1426 Turn 2: This is really helpful, thank you! I love the idea of naming dishes after teams or sports terms—it adds a fun, personal touch. I'm curious, do you have any  
 1427 suggestions for incorporating South Asian flavors into these crowd-pleasing menus without making the dishes too unfamiliar for a typical sports crowd? I'd like to  
 1428 introduce a bit of my heritage, but still keep everything approachable and easy to eat.  
 1429 **State Consistency Evaluation**  
 1430  
 1431 Rating Guidelines ◀  
 1432 **Rate Each State**  
 1433  
 1434 business\_ownership\_structure\_preference: 'partnership\_with\_friend'  
 1435 Rate consistency between conversation and this state  
 1436  0: No conflict  1: Minor inconsistency  2: Major conflict  
 1437  
 1438 entrepreneurial\_risk\_tolerance: 'high\_risk'  
 1439 Rate consistency between conversation and this state  
 1440  0: No conflict  1: Minor inconsistency  2: Major conflict  
 1441  
 1442 physical\_activity\_intensity\_level: 'advanced\_high\_intensity'  
 1443 Rate consistency between conversation and this state  
 1444  0: No conflict  1: Minor inconsistency  2: Major conflict  
 1445  
 1446 discussion\_event\_format: 'guest Speaker'  
 1447 Rate consistency between conversation and this state  
 1448  0: No conflict  1: Minor inconsistency  2: Major conflict  
 1449  
 1450 catering\_target\_market\_type: 'art\_event\_organizers'  
 1451 Rate consistency between conversation and this state  
 1452  0: No conflict  1: Minor inconsistency  2: Major conflict  
 1453  
 1454 catering\_menu\_style: 'custom\_artistic'  
 1455 Rate consistency between conversation and this state  
 1456  0: No conflict  1: Minor inconsistency  2: Major conflict

Figure 9: Annotation interface for conversation states.

1457

1458

1459 Thanks for logging in, user2. Select the appropriate choice for each query based on the required state information. Click **Submit** to save your annotation.

1460

1461 86%|00000000000000000000000000000000|Finished:103|Total:120

1462 Current id: 101

1463 Your annotation: {'query': 'What are some practical steps I can take to transition from community management to policy-making?', 'r

1464

1465 **Current Item**

1466

1467 Item ID

1468 Item 101

1469

1470 Query

1471 What are some practical steps I can take to transition from community management to policy-making?

1472

1473

1474

1475 **Required State Information:**

1476  **Policy Transition Experience Level:** substantial\_direct\_involvement

1477  **Policy Transition Education Interest:** interested\_in\_degree

1478

1479 **Task:** Select the ONE choice that best matches the given query considering the required state information.

1480 **Select the Best Choice**

1481

1482 Available Choices

1483 Choice 1: With your extensive hands-on background, you can leverage your experience by pursuing a graduate degree in public policy or a related field,  which will deepen your theoretical understanding and expand your professional network. Consider applying for policy fellowships or advisory roles where your direct community insights will be highly valued.

1484

1485

1486 Choice 2: Since you have some committee experience, enrolling in a policy certificate program can help you build targeted skills and credentials. Pair this  with volunteering for policy-focused projects or shadowing local policymakers to gain practical exposure and strengthen your transition.

1487

1488

1489 Choice 3: Building on your committee involvement, pursuing a formal degree in policy or public administration will provide a strong foundation and  open doors to more advanced policy roles. Engage in internships or research assistantships during your studies to bridge your community management experience with policy-making practice.

1490

1491

1492 Choice 4: If you're new to policy work, starting with a short-term certificate program will introduce you to key concepts and frameworks. Complement  your studies by attending public meetings or joining local advocacy groups to gain firsthand insight into the policy-making process.

1493

1494 Additional Comments

1495 Any additional observations about the choice or annotation decision...

1496

1497

1498

1499 [Previous](#)

1500 [Submit](#)

1501 [Next](#)

1502

1503

---

1504 **Algorithm 1** Memory Agent Self-Evolution Loop

---

1505 1: **Input:** Initial policy prompt  $P_0$ , Number of evolution cycles  $K$ .  
 1506 2: **Initialize:** Agent with policy  $\pi_0(P_0)$ .  
 1507 3: **for**  $k = 0$  **to**  $K - 1$  **do**  
 1508 4:     Interact with the AMEMGYM environment for one episode using policy  $\pi_k(P_k)$ .  
 1509 5:     Collect trajectory  $\tau_k = \{o_0, a_0, \dots, o_T, a_T\}$  and evaluation outcomes.  
 1510 6:     Generate environmental feedback summary  $F_k$  based on the interaction and outcomes.  
 1511 7:     Generate the updated policy prompt:  $P_{k+1} = G(P_k, F_k)$ .  
 1512 8:     Update the agent's policy to  $\pi_{k+1}(P_{k+1})$ .  
 1513 9: **end for**  
 1514 10: **Output:** Sequence of evolved prompts  $\{P_1, \dots, P_K\}$  and associated performance metrics.

---

1515  
 1516 *Annotation Scale:* Annotators rated the consistency for each state on the following scale: (0) No conflict; (1)  
 1517 Minor inconsistency; (2) Major conflict. Points are rescaled to [0, 1] for later computations.  
 1518

1519 *Results:* We randomly sampled 40 multi-turn conversation sessions each with multiple states to annotate,  
 1520 resulting in 748 items in total to annotate. The evaluation yielded an average consistency score of 99.2%,  
 1521 with a Gwet's AC1 coefficient of 98.2%. These results demonstrate that the simulated user maintains high  
 1522 fidelity to its assigned states throughout the interaction, ensuring that the integrity of the ground truth is  
 1523 preserved and not corrupted by conversational drift.  
 1524

1525 **Ground-Truth Judgment Reliability Evaluation.** To further validate the reliability of the simulator's  
 1526 judgments, we randomly sampled 100 questions from the model's evaluation logs. Two independent human  
 1527 annotators were asked to select the correct answer for each question based on the provided context. We then  
 1528 calculated the agreement rates.  
 1529

1530 *Results:* The inter-annotator agreement between the two humans was 0.92, establishing a strong  
 1531 baseline for human consistency. Crucially, the agreement between the LLM-generated ground-truth answers  
 1532 ("golden choices") and the human annotators was also exceptionally high, reaching 0.96 for the first  
 1533 annotator and 0.94 for the second.  
 1534

1535 **E DETAILS FOR THE SELF-EVOLUTION EXPERIMENT**  
 1536

1537 Algorithm 1 describes agent's self-evolution process.  
 1538

1539 **Evaluation Metrics** To provide a comprehensive assessment of the self-evolution process, we evaluate  
 1540 agents from two complementary perspectives: task-specific performance and the factual accuracy of their  
 1541 internal memory. (1) *Task Performance:* We measure the agent's ability to solve memory-dependent tasks  
 1542 using the primary metrics from our benchmark suite (Section 3.3). The **Normalized Memory Score** is  
 1543 reported at the end of each evolution cycle  $k$  to track the agent's task-specific improvement over time.  
 1544

1545 As a complementary metric, we report the score of *Memory Factual Recall*: We directly measure the extent  
 1546 to which agents successfully incorporate new information into their memory. Following methodologies in  
 1547 factual recall studies Min et al. (2023); Tang et al. (2024), we build a factual consistency checker using  
 1548 GPT-4.1. Let  $S_{new}$  be the set of new user states introduced during an interaction episode, and  $M_{mem}$  be  
 1549 the agent's memory representation at the end of that episode. The checker is prompted to evaluate each  
 1550 fact  $s_i \in S_{new}$  for consistency against the memory  $M_{mem}$ . For each pair  $(s_i, M_{mem})$ , the checker returns  
 a binary judgment,  $j_i \in \{0, 1\}$ , where  $j_i = 1$  indicates that the fact is supported by the memory and

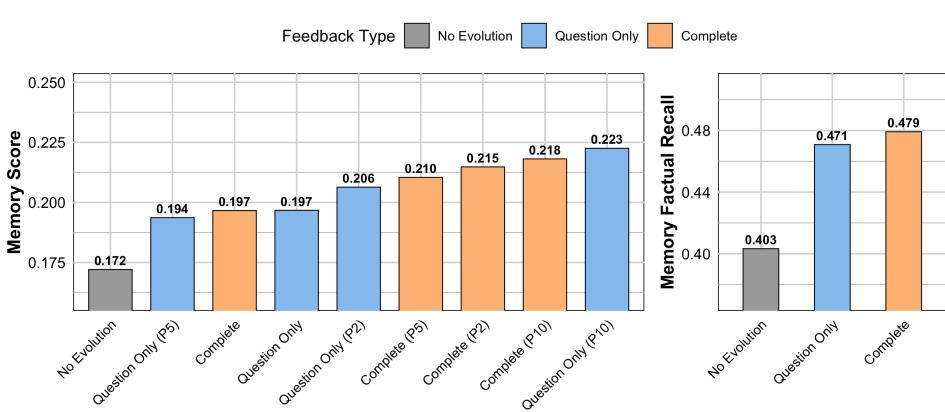


Figure 11: Comparison of memory performance and factual recall for evolution assistants under different environmental feedback conditions.

$j_i = 0$  indicates otherwise. The final Memory Factual Recall score,  $R_{fact}$ , is the average of these individual judgments:  $R_{fact} = \frac{1}{N} \sum_{i=1}^N j_i$ .

Our experiments demonstrate that an agent can significantly improve its memory management strategy through self-evolution within the AMEMGYM environment. As shown in Figure 11, agents receiving feedback consistently outperform the static baseline. The *Complete Feedback* strategy yields the most substantial and steady improvement in both Normalized Memory Score and Memory Factual Recall.

### E.1 CASE STUDY: ANALYSIS OF EVOLVED POLICIES

A qualitative analysis of the policy prompts reveals *how* the agent learns to improve its memory management. As illustrated in Table 5, the agent’s policy evolves from general instructions in early cycles (P1) to highly specific, actionable rules by the final cycle (P10). For instance, a vague prompt to track “skill levels” is refined into a nuanced rule for capturing “teaching approaches suited to experience levels.” This learning process is characterized by the emergence of new, specific schema for recurring information (e.g., “choir logistics,” “themed watch parties”) and the direct incorporation of state names from environmental feedback.

## F ADDITIONAL EVALUATION RESULTS

### F.1 EVALUATION ON *Extra* CONFIGURATION

As illustrated in Figure 12, simply adjusting the configurable parameters in AMEMGYM allows us to easily increase the difficulty of the evaluation environment.

Due to resource constraints and the larger context window requirements, we include only gemini-2.5-flashlite and gpt-4.1-mini for comparison under the *extra* configuration. These two models exhibit significantly lower memory scores of 0.137 and 0.104, respectively, compared to scores of 0.269 and 0.203 under the *base* setting. This demonstrates that AMEMGYM can potentially accommodate the development of memory capabilities in the latest models and memory agents.

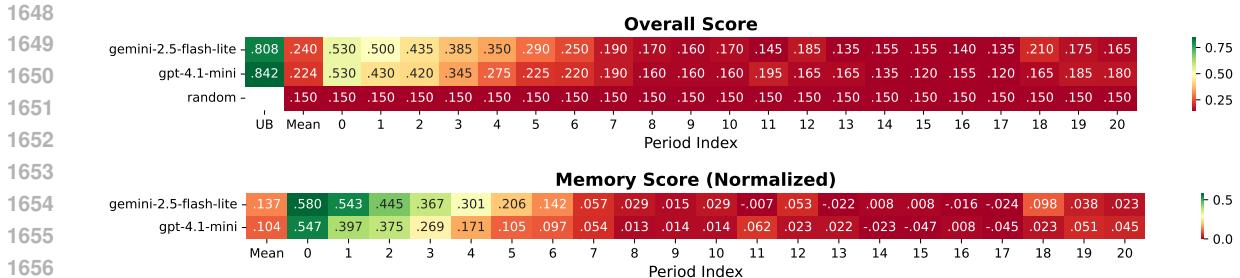
1598  
1599  
1600  
1601  
1602  
1603  
1604

Table 5: Running examples of prompt evolution traces on period 1 (P1), 2 (P2), 5 (P5), and 10 (P10).

| State Schema   | P1  | P2   | P5   | P10   |
|--|---|--|--|---|
| <b>volunteering personal mobility level</b><br>[“highly mobile”, “occasional assistance needed”, “limited mobility”]   | <b>Implied:</b> . “Maintain Up-to-Date Health, Wellness, and Dietary Profiles: ... changes over time, including... <b>medical considerations.</b> ”   | <b>Implied:</b> . “Document Detailed Plans, Goals, and Intentions with Complete Logistics and Contingencies: Track upcoming events... including specific logistical details such as... <b>accessibility considerations</b> , and contingency plans.” | <b>Explicit:</b> . “Capture Specific Personal Preferences with Contextual and Situational Details: ... and hobbies (preferred formats, skill levels, group sizes, engagement styles, and <b>accessibility needs</b> )...”                                    | <b>Explicit:</b> . “Record Activity, Service, and Volunteering Preferences...: ... <b>accessibility features</b> ), hobbies and teaching approaches (skill levels... <b>accessibility aids</b> )...”                    |
| <b>mentoring delivery format</b><br>[“oneon one”, “small group”, “workshop series”]                                    | <b>Implied:</b> . “Save Professional, Mentorship, and Development Details: Remember ..., <b>preferred learning styles</b> , and relevant networking or community involvement.”                        | <b>Implied:</b> . “Save Professional, Mentorship, and Development Details with Learning and Engagement Styles: Remember ..., <b>preferred learning styles</b> , networking involvement, ...”   | <b>Implied:</b> . “Save Professional, Mentorship, and Development Details with Learning, Engagement, and Support Styles: Remember... and <b>mentoring activity preferences.</b> ”  | <b>Explicit:</b> . “Save Professional, Mentorship, ... and Session Structures: ... <b>Capture detailed session structures</b> , preferred icebreakers, ...”   |
| <b>potluck available cooking time</b><br>[“limited under 2 hours”, “flexible afternoon”, “full day prep”]              | <b>Implied:</b> . “Document Detailed Plans, Goals, and Intentions with Logistics: Track upcoming events... including specific <b>logistical details</b> such as dates, <b>times</b> , locations, ...” | <b>Implied:</b> . “Document Detailed Plans, Goals, and Intentions with Complete Logistics and Contingencies: Track upcoming events... including specific <b>logistical details</b> such as dates, <b>times</b> , locations, ...”                     | <b>Explicit:</b> . “Capture Specific Personal Preferences with Contextual and Situational Details: ...and products (including situational factors such as event type, <b>timing, preparation ease</b> , cost sensitivity, durability, and user experience).” | <b>Explicit:</b> . “Capture Specific Personal Preferences with Context, ...and products (situational factors such as event type, <b>timing, preparation ease</b> , cost sensitivity, durability, and user experience).” |
| <b>soul food guest health goals</b><br>[“general healthy eating”, “weight management”, “chronic condition management”] | <b>Implied:</b> . “Maintain Up-to-Date Health, ..., <b>wellness goals</b> , and any adaptations or changes over time...”  | <b>Implied:</b> . “Maintain Up-to-Date Health, ..., <b>wellness goals</b> ... and any adaptations or changes over time...”   | <b>Explicit:</b> . “Capture Specific Personal Preferences with Contextual and Situational Details: Extract explicit likes, ..., and <b>health-conscious modifications</b> )...”  | <b>Explicit:</b> . “Maintain Up-to-Date Health, ..., <b>symptom management strategies</b> , <b>evolving health needs</b> , and personalized wellness preferences ...”   |
| <b>crimson tide game tech setup</b><br>[“basic tv livestream”, “outdoor projector”, “no live viewing available”]       | Absent  | Absent   | <b>Implied:</b> . “Record Activity, ..., <b>technology comfort and tools</b> , volunteer safety checklists with tone and language preferences)..."   | <b>Implied:</b> . “Record Activity, ..., <b>technology comfort and tools</b> , volunteer safety checklists...)”   |

1640  
1641  
1642  
1643  
1644

1645 Furthermore, AMEMGYM offers flexibility and customization for other parameters, such as the number of  
 1646 state variants per state and the frequency of state changes, thanks to its fully automated design.  
 1647



1653 Figure 12: Memory evaluation results on the *extra* configuration.

## 1654 F.2 EVALUATION WITH DIFFERENT USER LLMs

1655 As shown in Figure 13, switching the user LLM from gpt-4.1 to deepseek-v3 has minimal impact on the  
 1656 evaluation results. It reflects the advantage of AMEMGYM on grounded interactions.

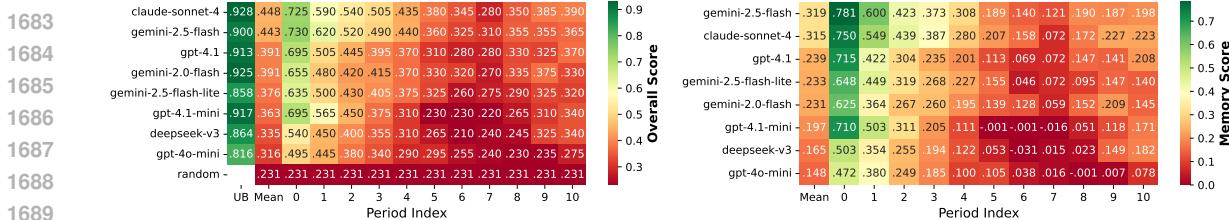
## 1657 F.3 FULL FIGURE FOR DIAGNOSIS ON WRITE STRATEGIES

1658 We present detailed diagnostic results for various write strategies in Figure 14. Due to the high information  
 1659 density in this figure, which can be challenging to interpret, we have transformed the data into a table in  
 1660 Figure 7a for improved clarity.

## 1661 F.4 EVALUATION WITH OTHER MEMORY IMPLEMENTATIONS

1662 Table 6: Performance comparison of open-source memory systems

| 1663 Open-source Memory System                                 | 1664 Overall | 1665 Memory |
|--|--------------|-------------|
| 1666 Mem0 (Chhikara et al., 2025) w/o Graph (AWE in the paper) | 0.430        | 0.296       |
| 1667 Mem0 (Chhikara et al., 2025) w/ Graph                     | 0.424        | 0.284       |
| 1668 A-Mem (Xu et al., 2025)                                   | 0.378        | 0.220       |
| 1669 Nemori (Nan et al., 2025)                                 | 0.385        | 0.231       |



1672 Figure 13: Memory evaluation results with deepseek-v3 as the user LLM.

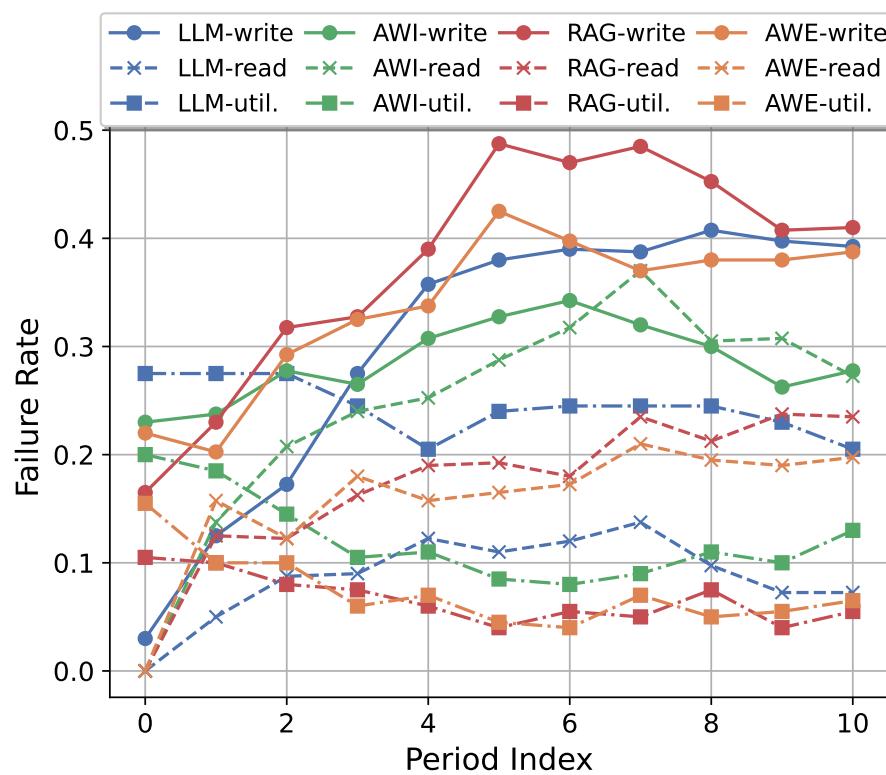


Figure 14: Full figure for diagnosis on write strategies.

1739 Table 6 presents the performance of several open-source memory frameworks, with AWE (the baseline im-  
 1740 plementation described in the main text) included for comparison. For A-Mem and Nemori, we use the same  
 1741 embedding model and vector database as the Mem0 (AWE) implementation to ensure a fair comparison.  
 1742

1743 **F.5 EVALUATION WITH OPEN-SOURCE MODELS**  
 1744

1745 Table 7: Performance comparison of open-source models  
 1746

| 1747 Model  | 1748 Overall | 1749 Memory |
|---|--------------|-------------|
| 1749 <code>gemini-2.5-flash</code> (Google, 2025)                   | 0.448        | 0.327       |
| 1750 <code>qwen3-235b-a22b-instruct-2507</code> (Yang et al., 2025) | 0.331        | 0.148       |
| 1751 <code>deepseek-v3.1-terminus</code> (DeepSeek, 2025)           | 0.327        | 0.151       |
| 1752 <code>kimi-k2-instruct</code> (Team et al., 2025)              | 0.389        | 0.234       |
| 1753 <code>glm-4.6</code> (Z.ai, 2025)                              | 0.404        | 0.257       |

1754 Table 7 presents the performance of several leading open-source models on our benchmark, with gemini-  
 1755 2.5-flash included as the baseline from the main text.  
 1756

1757 **F.6 EVALUATION STABILITY**  
 1758

1759 Table 8: Performance stability across five independent runs.  
 1760

| 1762 Model                              | 1763 Overall Score      | 1764 Memory Score       |
|---|-------------------------|-------------------------|
| 1763 <code>gpt-4o-mini</code>           | 0.3178 ( $\pm 0.0025$ ) | 0.1504 ( $\pm 0.0044$ ) |
| 1764 <code>gpt-4.1-mini</code>          | 0.3675 ( $\pm 0.0023$ ) | 0.2030 ( $\pm 0.0031$ ) |
| 1765 <code>gemini-2.5-flash-lite</code> | 0.3921 ( $\pm 0.0062$ ) | 0.2583 ( $\pm 0.0100$ ) |
| 1766 <code>gemini-2.5-flash</code>      | 0.4465 ( $\pm 0.0037$ ) | 0.3240 ( $\pm 0.0056$ ) |

1767 To assess the reliability of our benchmark, we repeated the evaluation 5 times across a representative sub-  
 1768 set of models. Table 8 reports the mean and standard deviation for each model, demonstrating that our  
 1769 benchmark produces highly stable performance estimates.  
 1770

1771  
 1772  
 1773  
 1774  
 1775  
 1776  
 1777  
 1778  
 1779  
 1780  
 1781  
 1782  
 1783  
 1784  
 1785