Learning User Preferences Through Interaction for Long-Term Collaboration

Shuhaib Mehri **Priyanka Kargupta** **Tal August** **Dilek Hakkani-Tür**
University of Illinois Urbana-Champaign
{mehri2, pk36, taugust, dilek}@illinois.edu

Abstract

As conversational agents accumulate experience collaborating with users, adapting to user preferences is essential for fostering long-term relationships and improving collaboration quality over time. We introduce MULTISESSIONCOLLAB, a benchmark that evaluates how well agents can learn user preferences and leverage them to improve collaboration quality throughout multiple sessions. To develop agents that succeed in this setting, we present long-term collaborative agents equipped with a memory that persists and refines user preference as interaction experience accumulates. Moreover, we demonstrate that learning signals can be derived from user simulator behavior in MULTISESSIONCOLLAB to train agents to generate more comprehensive reflections and update their memory more effectively. Extensive experiments show that equipping agents with memory improves long-term collaboration, yielding higher task success rates, more efficient interactions, and reduced user effort. Finally, we conduct a human user study that demonstrates that memory helps improve user experience in real-world settings.

1 Introduction

Throughout repeated interactions, humans naturally develop interpersonal relationships and adapt their behaviors to each other across various dimensions, including matching communication styles, establishing common ground, and refining mutual understanding (Burgoon et al., 1995; Niederhoffer and Pennebaker, 2002; Wilkes-Gibbs and Clark, 1992; Clark and Brennan, 1991; Clark and Wilkes-Gibbs, 1986). This type of adaptation is particularly critical in human-AI collaboration, where users have interaction preferences (e.g., having key takeaways at the end of responses or receiving multiple options and their tradeoffs before making a decision) that influence how effectively they can understand and engage with agent responses (Sweller, 1988; Shi et al., 2025). Rather than placing the cognitive burden on the user to repeatedly specify their preferences, conversational agents must be able to continuously learn and adhere to them over time (Bickmore and Picard, 2005; Shi et al., 2025).

Recent works have introduced memory mechanisms to overcome the finite context window of Large Language Models (LLMs) and enable agents to dynamically store and retrieve user-specific information (Chhikara et al., 2025; Wang and Chen, 2025; Yan et al., 2025). Evaluations of these systems generally focus on how accurately agents can recall information and answer questions about past interactions (Xu et al., 2022a; Maharana et al., 2024; Wu et al., 2025a). While such recall abilities are a necessary prerequisite, they do not capture all abilities needed for the intended purpose of these systems, which is to improve long-term interaction with users. In particular, it remains unclear whether agents can recognize what information is valuable for future interactions and leverage it effectively.

To address this gap, we introduce MULTISESSIONCOLLAB, a benchmark designed to evaluate how well conversational agents can learn user preference information and leverage it to meaningfully improve interactions during multi-session collaboration. The benchmark features user simulators that collaborate with the agent on problem-solving tasks across multiple sessions. Each user has distinct personas and interaction preferences that are derived from psychology, cognitive science, and human-computer interaction research, reflecting realistic human behaviors in collaborative settings (see taxonomy in Appendix A) (Wood and Neal, 2007; Sweller, 2006; Schwartz et al., 2002). The benchmark is designed so that users progress most effectively when agent responses align with their

[1] Code available at https://github.com/Shuhaibm/multisessioncollab

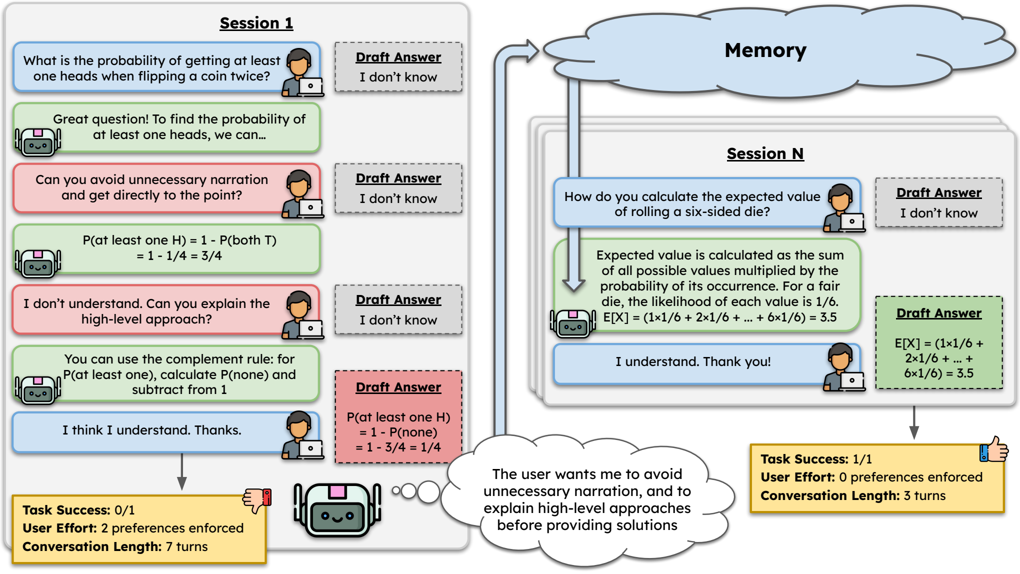

Session 1
What is the probability of getting at least one heads when flipping a coin twice?
Draft Answer
I don't know
Great question! To find the probability of at least one heads, we can...
Can you avoid unnecessary narration and get directly to the point?
Draft Answer
I don't know
P(at least one H) = 1 - P(both T) = 1 - 1/4 = 3/4
I don't understand. Can you explain the high-level approach?
Draft Answer
I don't know
You can use the complement rule: for P(at least one), calculate P(none) and subtract from 1
Draft Answer
P(at least one H) = 1 - P(none) = 1 - 3/4 = 1/4
I think I understand. Thanks.
Task Success: 0/1
User Effort: 2 preferences enforced
Conversation Length: 7 turns
The user wants me to avoid unnecessary narration, and to explain high-level approaches before providing solutions
Memory
Session N
How do you calculate the expected value of rolling a six-sided die?
Draft Answer
I don't know
Expected value is calculated as the sum of all possible values multiplied by the probability of its occurrence. For a fair die, the likelihood of each value is 1/6. E[X] = (1×1/6 + 2×1/6 + ... + 6×1/6) = 3.5
Draft Answer
E[X] = (1×1/6 + 2×1/6 + ... + 6×1/6) = 3.5
I understand. Thank you!
Task Success: 1/1
User Effort: 0 preferences enforced
Conversation Length: 3 turns

Figure 1: The MULTISESSIONCOLLAB benchmark with our long-term collaborative agent. Each session involves a user seeking help for a problem. The user maintains a draft answer that represents what they have learned from the interaction. They update their draft answer when the agent's responses are both helpful and preference-aligned. When responses violate preferences, the user enforces them, as indicated by the red text boxes. After each session, the agent reflects on the interaction to identify user preference information that will be useful for future interactions and update their memory accordingly. We measure collaboration quality in each session with task success, conversation length, and user effort.

preferences, incentivizing agents to continuously learn and adhere to preferences. An overview of the benchmark is presented in Figure 1.

Unlike prior memory systems that optimize for information recall, we develop long-term collaborative agents equipped with memory that learns user preferences to improve collaboration quality over time. As shown in Figure 1, agents reflect on interactions to identify user preference information that will be valuable for future interactions and update their memory accordingly. During subsequent sessions, the full memory is provided to the agent, and relevant parts are dynamically retrieved throughout the interaction to guide agent behavior towards being more preference-aligned.

We also present a reinforcement learning (RL) framework that derives learning signals from user simulator behavior in MULTISESSIONCOLLAB to train agents to update memory more effectively. We use Group Relative Policy Optimization (GRPO) (Shao et al., 2024) with an LLM-judge that rewards reflections for comprehensively capturing user preferences revealed during interaction.

Extensive experiments across several models and five problem-solving tasks demonstrate that memory enables agents to continuously learn user interaction preferences and improve collaboration quality, yielding higher task success rates, more efficient interactions, and reduced user effort. Analyses of performance across sessions reveal continued improvement throughout the sessions, with the steepest gains occurring in early sessions before gradually stabilizing. Notably, agents that learn preferences through interaction are competitive with those given direct access to ground-truth user preferences, showing how memory captures richer information about preferences.

Finally, we conduct a human user study with 19 participants who engage in three consecutive collaborative sessions with an agent across coding, writing, and problem-solving tasks. Results align with our experimental findings and demonstrate that equipping agents with memory improves collaboration quality across sessions. Participants described these agents as more personalized and proactive, while also identifying challenges in cross-domain preference generalization.

The contributions of our work are:

- We introduce MULTISESSIONCOLLAB to evaluate how well conversational agents can learn user preferences and leverage them to improve interactions during multi-session collaboration.
- We develop long-term collaborative agents equipped with memory that enables them to learn and leverage user preferences over time.
- We present an RL framework that trains agents

to generate more comprehensive reflections and update memory more effectively by using rewards derived from user behavior signals.

- Through extensive experiments and a human user study, we demonstrate that memory has a positive impact on multi-session collaboration.

2 Related Work

Multi-Session Conversation Evaluation. Early works that evaluate LLMs in multi-session interactions focus on how well models can generate responses that are consistent with past interactions (Xu et al., 2022a,b). More recently, benchmarks have shifted towards question-answering style evaluations that assess how well LLMs can remember information from past interactions (Packer et al., 2023; Maharana et al., 2024; Wu et al., 2025a; Hu et al., 2025; Jiang et al., 2025; Zhao et al., 2025). In contrast, our work evaluates the downstream impact of memory, incorporating abilities such as identifying what information will be useful in future interactions and leveraging it to improve collaboration.

Memory. Providing LLMs with large contexts is computationally inefficient and can degrade performance, since LLMs struggle to effectively handle large amounts of information (Shi et al., 2023; Liu et al., 2024). While one line of works try to improve LLM abilities to handle larger contexts (Liu et al., 2025a), another line of works introduces memory to enable agents to store information from past experiences, and retrieve it when useful in future interactions (Shinn et al., 2023; Packer et al., 2023; Zhong et al., 2024; Suzgun et al., 2025; Wang et al., 2025c; Ho et al., 2025; Chhikara et al., 2025). More recent works introduce more sophisticated memory architectures, including those that leverage multi-agent systems (Wang and Chen, 2025) or temporal-aware knowledge graphs (Rasmussen et al., 2025). While these works demonstrate the benefit of memory for question-answering tasks, we demonstrate that memory can improve downstream performance in multi-session collaboration settings.

RL for Memory. RL has been used to train agents to manage and utilize memory more effectively. Existing approaches use rewards based on question-answering correctness (Zhou et al., 2025b; Yan et al., 2025; Wang et al., 2025b; Yu et al., 2025). In contrast, we present an RL framework that derives rewards from user behavior during interaction. Specifically, we train agents to recognize what user preference information revealed during an interaction will be valuable for future sessions and update memory more effectively.

Human-AI Collaboration. Recent work has demonstrated that asking clarifying questions can enhance multi-turn interactions by helping agents better understand tasks, user intent, and preferences (Zhang et al., 2025; Li et al., 2025c,a; Andukuri et al., 2024; Wan et al., 2025; Li et al., 2025b), with several works showing that this behavior is helpful for human-AI collaboration (Wu et al., 2025b; Zhou et al., 2025a; Wang et al., 2025a). These approaches focus on single-session interactions, which is appropriate for cold-start scenarios. However, as users increasingly engage with agents over multiple sessions, repeatedly asking the similar questions and eliciting preferences can become tedious and place unnecessary cognitive burden on users. Our work addresses this limitation by enabling agents remember user preference information across multiple sessions.

3 MultiSessionCollab

We introduce MultiSessionCollab for evaluating conversational agents' ability to learn user preference information and leverage it to meaningfully improve collaboration quality over time. The benchmark takes place in a multi-session collaborative problem-solving setting with diverse user simulators, and can be instantiated with any problem-solving task. An overview of MultiSessionCollab is presented in Figure 1.

3.1 Collaborative Problem-Solving Session

In each session, a user simulator seeks to solve a problem with assistance from a conversational agent. Following Wu et al. (2025b); Zhou et al. (2025a), only the user has access to the problem statement, and they start the conversation by providing an initial description of their problem. Over multiple turns, the agent asks clarifying questions to better understand the user's task and provides explanations to help the user develop a solution. Conversations last until the user is satisfied and decides to terminate, or when the maximum conversation length of 10 user-agent turns has been reached. This design mirrors realistic collaborative scenarios, such as tutoring or debugging, and results in natural multi-turn interactions.

In such collaborative settings, agents must not only provide correct information, but also communicate it in ways that users can understand, apply, and learn from (Shi et al., 2025). We capture this by having users maintain a *draft answer* for their solution to the problem. The draft answer is never visible to the agent. It is initially empty, and is progressively updated as users receive assistance.

Crucially, users update their draft answer only when the agent provides helpful information in a manner that aligns with their interaction preferences. Information delivered in a way that violates the user's preferences (e.g. using overly technical language when the user prefers simple explanations) hinders their ability to effectively understand and apply the information (Shi et al., 2025; Sweller, 1988). In such cases, users explicitly communicate their preferences to the agent rather than updating their draft answer. We note that real-world users may be able to extract useful information even when responses do not fully align with their preferences. Our design choice of not updating the draft answer in such cases helps isolate preference adherence, enabling more direct evaluation of how well agents can learn and adapt to preferences across sessions.

3.2 User Simulator Design

A central component of our benchmark is designing user simulators with realistic user profiles that exhibit diverse behaviors and interaction preferences.

First, we instantiate each user profile with a randomly selected persona from Persona Hub, a large-scale persona collection (Ge et al., 2024). Each persona exhibits unique knowledge, experiences, interests, personalities and professions, which translates to varying perspectives and diverse behaviors.

Each user profile is also assigned a random set of three interaction preferences that describe how they expect the agent to behave during collaboration. The preferences may specify specific communication styles (Miehle et al., 2020), learning approaches (Chi et al., 1989), or proactivity levels (Horvitz, 1999). For instance, a user may prefer concise responses with key takeaways highlighted at the end, or expect detailed step-by-step explanations. Each preference is grounded in studies from psychology, cognitive science, and human-computer interaction, and reflects realistic human behavior in problem-solving settings. A complete taxonomy of the preferences and their sources are provided in Appendix A.

Finally, the effectiveness of our benchmark as an evaluation framework depends on user simulators that can reliably adhere to their user profiles and consistently enforce their preferences (Mehri et al., 2025). To ensure this, we employ a structured reasoning process that explicitly considers whether (a) each preference has been satisfied and if any needs to be enforced, (b) the draft answer should be updated, and (c) the conversation should be terminated. This structure yields interpretable user behavior signals: we track which utterances enforce preferences, using this both to quantify user effort (Section 3.3) and to derive learning signals for our RL framework (Section 4.2). The system prompt is provided in Appendix B.

3.3 Evaluation Metrics

For each session, we evaluate collaboration quality across three dimensions:

- *Task Success*: the accuracy of the user's final draft answer, which represents what the user learned from the interaction. Task success is a standard metric for collaboration (Wu et al., 2025b; Wang et al., 2025a).
- *User Effort*: the number of times the user enforced their preferences. Each enforcement represents a preference adherence failure, placing cognitive burden on the user to correct the agent behavior rather than focus on the task.
- *Conversation Length*: the total number of turns in the conversation. Agents that adhere to preferences reduce friction and help users complete the task in fewer turns.

These metrics represent collaboration quality: higher task success, lower user effort, and shorter conversations indicate more effective collaboration.

3.4 Multi-Session Setting

For each user, we sequentially run multiple collaborative problem-solving sessions, where each session involves a different problem. The agent initially has no knowledge of user preferences, and is expected to learn and adapt to preferences throughout the sessions. We report metrics averaged across all sessions per user, and then across all users.

4 Methodology

4.1 Long-Term Collaborative Agents

Conversational agents start with no knowledge about user preferences and inevitably generate misaligned responses during early interactions. But

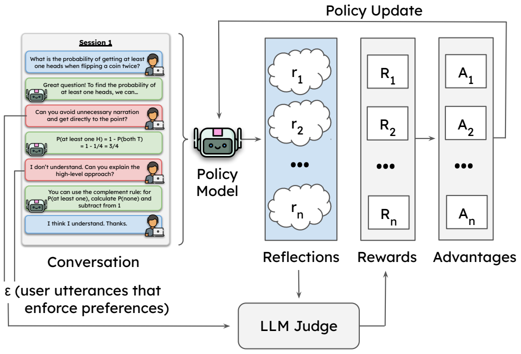

Policy Update
Session 1
What is the probability of getting at least one heads when flipping a coin twice?
Great question! To find the probability of at least one heads, we can...
Can you avoid unnecessary narration and get directly to the point?
P(at least one H) = 1 - P(both T) = 1 - 1/4 = 3/4
I don't understand. Can you explain the high-level approach?
You can use the complement rule: for P(at least one), calculate P(none) and subtract from 1
I think I understand. Thanks.
Conversation
ε (user utterances that enforce preferences)
Policy Model
r1
r2
•••
rn
Reflections
R1
R2
•••
Rn
Rewards
A1
A2
•••
An
Advantages
LLM Judge

Figure 2: RL framework for improving session-level reflections. The policy model generates n reflection rollouts for a conversation. The judge model evaluates each reflection against ε (the subset of user utterances that enforce preferences) and assigns rewards. Advantages are computed and the policy is updated via GRPO.

as interaction experience accumulates, users reveal their preferences, and long-term collaborative agents must be able to learn from these signals to reduce cognitive burden on users and improve collaboration quality over time.

To build such agents, we equip them with memory that persists and refines user preference information across sessions. It is illustrated in Figure 1 and all prompts are provided in Appendix B. The memory architecture comprises two components:

1. **Session-Level Reflection:** the agent analyzes the interaction and identifies preference information that would be useful for future interactions, which can include the specific preferences themselves, the contexts in which they apply, and details about which actions or approaches satisfy each preference. The agent then updates their existing memory with this new information.
2. **Persistent Memory:** the memory is provided at the start of each session as part of the agent's system prompt, enabling them to adapt their behavior according to learned preferences without requiring users to repeatedly specify them. Additionally, at each turn, we use an LLM to analyze the conversation history and retrieve parts of the memory that are directly relevant to the conversational context and provide them to the agent.

4.2 Reinforcement Learning in MultiSessionCollab

We present an RL framework that uses the MultiSessionCollab environment to train agents to generate more comprehensive session-level reflections that effectively capture the user preferences revealed during interactions (Figure 2). More specifically, we use learning signals derived from user simulator behavior to design rewards for the GRPO algorithm.

Problem Formulation. For a collaborative session, let $C = \{(u_1, a_1), ..., (u_n, a_n)\}$ denote a conversation consisting of n turns, where (u_i, a_i) are the user and agent utterances at turn i. Let $\varepsilon = \{e_1, ..., e_k\} \subset \{u_1, ..., u_n\}$ denote the subset of utterances where the user enforces their preferences. Given a conversation C, the agent must generate a reflection r that captures all preference information revealed during the interaction.

Reward Design. A session-level reflection is considered successful if it comprehensively captures the user preferences revealed during the interaction and provides actionable guidance for future sessions. ε specifies all information that a reflection should capture. We therefore use an LLM judge to evaluate r covers all preferences in ε without any hallucinations, and get an objective scalar reward $R_{coverage}(r, \varepsilon)$. We also include a format reward $R_{format}(r)$ that encourages well-structured outputs containing both a reasoning trace and the reflection. The total reward is:

$$R(r, \varepsilon) = R_{coverage}(r, \varepsilon) + R_{format}(r) \quad (1)$$

LLM judge prompts are provided in Appendix B. Using this reward, we employ GRPO to train a policy π_θ to generate more comprehensive session-level reflections that have no hallucinations. As a result, the agent is able to make more meaningful memory updates that helps them be more preference-aligned during future interactions.

5 Experimental Setup

To rigorously evaluate conversational agents with MultiSessionCollab, we instantiate the benchmark with 100 user profiles and five problem-solving benchmarks that cover diverse domains: MATH-500 (Hendrycks et al., 2021b), MATH-Hard (Hendrycks et al., 2021b), LogiQA (Liu et al., 2007), MMLU (Hendrycks et al., 2021a), and MedQA (Jin et al., 2021). We use Llama-3.3-70B-Instruct (Dubey et al., 2024) for our user simulator and LLM-judge. Each user collaborates with the agent to solve 20 randomly sampled problems, with one problem per session and a maximum of 10 conversational turns per session. Across all benchmarks, this totals 10,000 collaborative sessions per agent, ensuring robust evaluation of agent performance during multi-session collaboration.

Environment Validation. To validate the reliability of MULTISESSIONCOLLAB and understand the contribution of different components, we conduct a series of ablation studies that isolate the impact of various dimensions. We observe a drop in performance when moving from direct problem-solving to multi-turn interaction, which aligns with prior work showing that LLMs struggle to ask clarifying questions or maintain coherence across turns (Wu et al., 2025b; Laban et al., 2025; Zhou et al., 2025a; Mehri et al., 2025). Performance decreases further when introducing user preferences, since users correct agent behavior during preference violations rather than focus on the task. Our complete analysis is provided in Appendix C. We also present a user study that confirms our findings generalize to real users in Section 7.

Training. Training data is constructed using a different set of user profiles and problems, with no overlap with evaluation data. Similar to evaluation, we have 100 user profiles and select 20 problems from each problem-solving benchmark. Using Llama-3.3-70B-Instruct as both the user simulator and the agent, we generate 10,000 collaborative sessions along with reflections.

We train Llama-3.1-8B-Instruct (Dubey et al., 2024) and Qwen-2.5-7B-Instruct (Qwen et al., 2025) to generate session-level reflections given a conversation. The models are first initialized with supervised fine-tuning (SFT) as a cold start, which helps stabilizes RL optimization and leads to more effective performance (Guo et al., 2025). Then, we conduct GRPO as described in Section 4.2. Details on hyperparameters and issues such as catastrophic forgetting are provided in Appendix D.

We evaluate several LLMs in MULTISESSIONCOLLAB, including Llama-3.1-8B-Instruct, Qwen-2.5-7B-Instruct, gpt-oss-20b, and Llama-3.3-70B-Instruct (Qwen et al., 2025; OpenAI, 2025; Dubey et al., 2024). Models are evaluated as standard conversational agents without memory, and also when equipped with memory. For Llama-3.1-8B-Instruct and Qwen-2.5-7B-Instruct models, we additionally report results after applying GRPO training to improve session-level reflections.

6 Results

Table 1 presents our experimental results on the MULTISESSIONCOLLAB benchmark. Equipping conversational agents with memory enables them to learn preferences as they accumulate experience and proactively adhere to them without requiring users to repeatedly enforce them, reducing friction and allowing users to focus on the task. This generally results in improvements in collaboration quality: interactions become more efficient, require less user effort, and yield higher task success rates. We observe even further improvements when using our RL framework to train agents to capture more meaningful user preference information in session-level reflections and update memory effectively.

Analysis across different models. Different models exhibit varying baseline performance, as shown in the first row for each model that reports the results for a standard conversational agent without memory. For instance, the Qwen family tends to achieve the highest performance on mathematical reasoning tasks. Regardless of the baseline performance, memory helps improve collaboration quality. How effectively a model can leverage memory depends on their abilities to capture information that will be valuable for future interactions through session-level reflections and leverage them during actual interactions to improve user experience.

RL for Improving Session-Level Reflections. We apply our RL framework to train Qwen-2.5-7B-Instruct and Llama-3.1-8B-Instruct to generate session-level reflections that effectively capture the preferences revealed during interactions. After training, both models are able to leverage the memory more effectively, with notable improvements across all metrics. In particular, while Qwen-2.5-7B-Instruct initially exhibited a 1.03% decrease in task success when equipped with memory, after training, the same model achieves a 3.43% improvement in task success. This demonstrates that the memory architecture's effectiveness depends largely on reflection quality. Additionally, these results suggest that the way our RL framework uses learning signals derived from user simulator behavior offers a promising direction for improving agent memory management abilities.

Performance Across Sessions. Next, we analyze our agent's performance as they accumulate interaction experience throughout sessions. For each session $i \in \{1, ..., 20\}$, we compute the agent's average performance across all users and tasks for each metric. Then, we calculate the deltas Δ_i^{TS}, Δ_i^{UE}, and Δ_i^{Eff}, where each delta denotes the difference between the agent with memory and the agent without memory at session i for the respec-

	MATH-500			MATH-Hard			LogiQA		
	TS (%) ↑	*UE* ↓	*Len* ↓	*TS (%)* ↑	*UE* ↓	*Len* ↓	*TS (%)* ↑	*UE* ↓	*Len* ↓
Qwen-2.5-7B	51.35	2.95	14.05	26.5	2.93	14.62	19.65	2.40	15.61
+ memory	51.75$_{\uparrow 0.40}$	2.80$_{\downarrow 0.15}$	13.46$_{\downarrow 0.59}$	26.21$_{\downarrow 0.29}$	2.86$_{\downarrow 0.07}$	14.40$_{\downarrow 0.22}$	18.61$_{\downarrow 1.04}$	2.35$_{\downarrow 0.05}$	15.57$_{\downarrow 0.04}$
+ GRPO	56.80$_{\uparrow 5.45}$	2.16$_{\downarrow 0.79}$	12.85$_{\downarrow 1.20}$	27.75$_{\uparrow 1.25}$	2.27$_{\downarrow 0.66}$	13.78$_{\downarrow 0.84}$	23.20$_{\uparrow 3.55}$	1.74$_{\downarrow 0.66}$	15.01$_{\downarrow 0.60}$
Llama-3.1-8B	44.55	2.73	14.83	11.50	2.84	16.15	24.45	2.44	16.50
+ memory	44.93$_{\uparrow 0.38}$	2.20$_{\downarrow 0.53}$	13.73$_{\downarrow 1.10}$	12.79$_{\uparrow 1.29}$	2.33$_{\downarrow 0.51}$	15.28$_{\downarrow 0.87}$	26.50$_{\uparrow 2.05}$	1.92$_{\downarrow 0.52}$	15.93$_{\downarrow 0.57}$
+ GRPO	48.68$_{\uparrow 4.13}$	1.85$_{\downarrow 0.88}$	12.85$_{\downarrow 1.98}$	13.59$_{\uparrow 2.09}$	2.01$_{\downarrow 0.83}$	14.57$_{\downarrow 1.58}$	26.55$_{\uparrow 2.10}$	1.63$_{\downarrow 0.81}$	15.57$_{\downarrow 0.93}$
gpt-oss-20b	67.60	2.90	13.95	41.45	3.46	15.73	20.31	2.90	16.56
+ memory	70.35$_{\uparrow 2.75}$	2.34$_{\downarrow 0.56}$	12.72$_{\downarrow 1.23}$	45.07$_{\uparrow 3.62}$	2.83$_{\downarrow 0.63}$	14.33$_{\downarrow 1.40}$	25.19$_{\uparrow 4.88}$	2.32$_{\downarrow 0.58}$	14.89$_{\downarrow 1.67}$
Llama-3.3-70B	59.29	3.00	14.98	25.81	3.27	17.03	26.69	2.96	17.08
+ memory	63.85$_{\uparrow 4.56}$	1.99$_{\downarrow 1.01}$	13.28$_{\downarrow 1.70}$	30.50$_{\uparrow 4.69}$	2.25$_{\downarrow 1.02}$	15.17$_{\downarrow 1.86}$	30.25$_{\uparrow 3.56}$	1.89$_{\downarrow 1.07}$	15.85$_{\downarrow 1.23}$

	MMLU			MedQA			Overall		
	TS (%) ↑	*UE* ↓	*Len* ↓	*TS (%)* ↑	*UE* ↓	*Len* ↓	*TS (%)* ↑	*UE* ↓	*Len* ↓
Qwen-2.5-7B	47.85	2.51	13.81	35.70	2.58	15.96	36.21	2.67	14.81
+ memory	45.17$_{\downarrow 2.68}$	2.46$_{\downarrow 0.05}$	13.72$_{\downarrow 0.09}$	34.16$_{\downarrow 1.54}$	2.65$_{\uparrow 0.07}$	16.33$_{\uparrow 0.37}$	35.18$_{\downarrow 1.03}$	2.62$_{\downarrow 0.05}$	14.70$_{\downarrow 0.11}$
+ GRPO	51.15$_{\uparrow 3.30}$	1.91$_{\downarrow 0.60}$	12.88$_{\downarrow 0.93}$	39.30$_{\uparrow 3.60}$	1.97$_{\downarrow 0.61}$	15.61$_{\downarrow 0.35}$	39.64$_{\uparrow 3.43}$	2.01$_{\downarrow 0.66}$	14.02$_{\downarrow 0.79}$
Llama-3.1-8B	50.50	2.42	14.12	39.20	2.74	16.66	34.04	2.63	15.65
+ memory	51.50$_{\uparrow 1.00}$	2.00$_{\downarrow 0.42}$	13.62$_{\downarrow 0.50}$	41.85$_{\uparrow 2.65}$	2.10$_{\downarrow 0.64}$	16.00$_{\downarrow 0.66}$	35.51$_{\uparrow 1.47}$	2.11$_{\downarrow 0.52}$	14.91$_{\downarrow 0.74}$
+ GRPO	54.00$_{\uparrow 3.50}$	1.68$_{\downarrow 0.42}$	12.92$_{\downarrow 1.20}$	40.95$_{\uparrow 1.75}$	2.10$_{\downarrow 0.64}$	15.89$_{\downarrow 0.77}$	36.75$_{\uparrow 2.71}$	1.85$_{\downarrow 0.78}$	14.36$_{\downarrow 0.45}$
gpt-oss-20b	49.62	2.46	13.81	32.75	3.26	17.06	42.35	3.00	15.42
+ memory	55.05$_{\uparrow 5.43}$	2.0$_{\downarrow 0.46}$	12.23$_{\downarrow 1.58}$	31.68$_{\downarrow 1.07}$	3.76$_{\uparrow 0.50}$	16.70$_{\downarrow 0.36}$	45.47$_{\uparrow 3.12}$	2.65$_{\downarrow 0.35}$	14.17$_{\downarrow 1.25}$
Llama-3.3-70B	51.26	2.76	14.34	45.84	2.91	16.38	41.78	2.98	15.96
+ memory	57.35$_{\uparrow 6.09}$	1.74$_{\downarrow 1.02}$	12.99$_{\downarrow 1.35}$	49.95$_{\uparrow 4.11}$	1.92$_{\downarrow 0.99}$	15.35$_{\downarrow 1.03}$	46.38$_{\uparrow 4.60}$	1.96$_{\downarrow 1.02}$	14.53$_{\downarrow 1.43}$

Table 1: Conversational agent performance on MULTISESSIONCOLLAB across five problem-solving tasks. We report task success (*TS*), user effort (*UE*), and conversation length (*Len*). For each model, the first row is the baseline conversational agent without memory, *+memory* is the agent equipped with our memory architecture, and *+GRPO* is the agent trained to generate session-level reflections. Subscripts denote change relative to the baseline. Overall performance is also reporting by averaging performance across all tasks. We use the instruct variant for all models.

tive metric. These deltas isolate the impact of memory at each session. Figure 3 plots these deltas across sessions for Llama-3.3-70B-Instruct, with additional plots for other models presented in Appendix E. Increasing Δ_i^{TS}, and decreasing Δ_i^{UE} and Δ_i^{Eff} indicate that the agent is learning to collaboration more effectively.

We observe consistent trends of improvement across all deltas. Δ_i^{TS} has an upward trajectory, particularly after smoothing. However, there are also some points in the graph where the delta is close to 0 and the memory does not seem to help improve performance for these problems. Δ_i^{UE} and Δ_i^{Eff} show consistent downward trends, with the steepest improvements occurring within the first five sessions, and gradually stabilizing towards the last 10 sessions. This demonstrates how agents with memory *continuously learn and refine their knowledge* as interaction experience accumulates, without plateauing in early sessions.

Some variance in the results comes from certain preferences that make problem-solving more difficult, reflecting a limitation in the agent's ability to adhere to preferences during interactions while maintaining task performance.

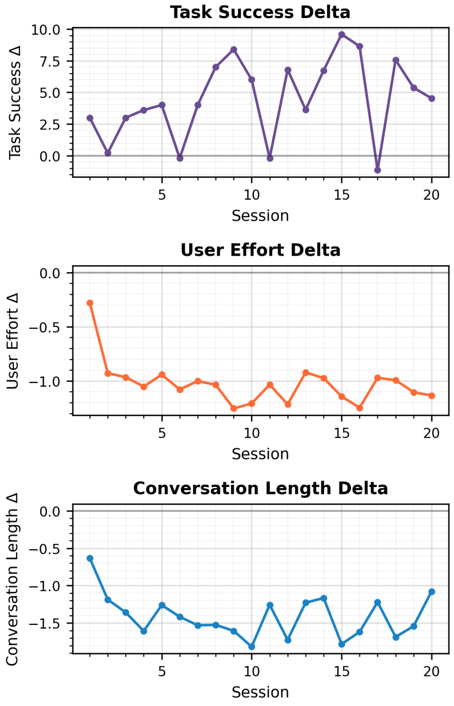


Figure 3: Performance across sessions for Llama-3.3-70B-Instruct. Each graph plots the delta between agents with memory and agents without memory across 20 sessions for Task Success (Δ_i^{TS}) ↑, User Effort (Δ_i^{UE}) ↓, and Conversation Length (Δ_i^{Len}) ↓.

Value of Learning Through Interaction. In Appendix C, we report further ablations that help us better understand the memory architecture. One provides agents with direct access to user preferences, same as what is provided to the user simulator. This was intended to be an oracle to establish an upper bound on performance. Surprisingly, agents equipped with memory achieve competitive performance with this oracle, and sometimes outperform it, despite the fact that agents with memory begin with no user preference information. Our qualitative analysis indicates that memory helps capture richer preference information from interactions, such as the contexts in which the preferences apply and specific strategies for adhering to them. Moreover, this information is continually refined across sessions as agents accumulate interaction experience. These results demonstrate the effectiveness of using memory to learn through interaction.

7 Real-World User Study

Setup. To complement our main experiments, we conduct a user study with 19 participants to understand the impact of memory in more realistic collaboration settings. The study lasted approximately 1.5 hours per participant, and was declared exempt after being reviewed by our Institutional Review Board (IRB). The complete details of our study design and results are provided in Appendix F.

We adopt a similar setup to MULTISESSIONCOLLAB and design two study types: single-domain (coding only) and mixed-domain (writing, math problem-solving, and coding). In each study, participants complete three sequential sessions, solving one problem per session, with both a standard agent and an agent equipped with memory. Participants adopt a set of interaction preferences that they ensure the agents adhere to throughout the sessions. These preferences are parallel to those from Appendix A. After each session, participants complete a survey where they rate the intrinsics aspects of the collaboration on a 5-point Likert scale: preference adherence, preference retention, confidence, and overall satisfaction.

Results. Our user study results align with the trends observed in our simulated experiments. Across both single-domain and mixed-domain settings, agents with memory demonstrated consistent improvements over sessions: conversations required fewer turns, preference adherence and retention improved, and confidence and satisfaction increased (see Figure 10). For instance, in the first session conversations with memory had a median of 8 (coding) and 6 (mixed) turns, and those without memory had a median of 8 turns. By the third session, agents with memory dropped to 6 and 4 turns, compared to 10 and 8 turns without memory.

Participants also provided free-form feedback after each session and shared overall thoughts at the end of the study. They described the agents with memory as noticeably more personalized, highlighting their ability to proactively adhere to preferences and handle vague queries. However, participants also noted limitations: personalization felt less effective in mixed-domain settings, and the agent's learned preferences were not as effective as when explicitly restated within a session. These findings suggest that while memory substantially improves multi-session collaboration, challenges remain in cross-domain generalization and fully capturing user preferences from interaction.

8 Conclusion

We introduce MULTISESSIONCOLLAB, a benchmark for evaluating conversational agents' ability to learn user preferences and leverage them to meaningfully improve collaboration across multiple sessions. We develop long-term collaborative agents equipped with memory that learn preferences to improve interaction quality over time. We further demonstrate that MULTISESSIONCOLLAB can serve as an effective RL environment, where we use learning signals derived from user behavior to encourage more comprehensive reflections, leading to more meaningful memory updates. Extensive experiments and user studies show that equipping agents with memory improves collaboration quality across sessions, yielding higher task success rates, more efficient interactions, and reduced user effort.

Our work provides a foundation for multi-session collaboration, with many promising directions for future research. The user simulators in MULTISESSIONCOLLAB can be extended to have more complex preferences, such as context-dependent preferences that vary across domains, or evolving preferences that develop over time, enabling the evaluation of more sophisticated adaptation abilities. As user simulation advances, simulators can also express preferences through more diverse means, for example through implicit behavioral cues. Finally, improving memory management abilities and addressing ensuring agents do not forget useful information as they update memories remains an important direction.

9 Limitations

We use Llama-3.3-70B-Instruct as our user simulator. While it is a highly capable model, user simulation requires reliably following complex instructions, such as consistently enforcing preferences throughout interactions, and our evaluation metrics depend on this behavior. We found that the simulator struggled to exhibit expected behaviors during more complex tasks such as coding, which we consequently excluded from our benchmark. Additionally, user simulators may not be realistic or fully capture all kinds of human behavior, for example real users can express preferences implicitly or adapt to the agent over time. Although we conduct a human user study to validate our findings across realistic tasks and real users, we acknowledge these limitations remain and merit further investigation.

References

Chinmaya Andukuri, Jan-Philipp Fränken, Tobias Gerstenberg, and Noah Goodman. 2024. STar-GATE: Teaching language models to ask clarifying questions. In *First Conference on Language Modeling*.

Timothy W Bickmore and Rosalind W Picard. 2005. Establishing and maintaining long-term human-computer relationships. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 12(2):293–327.

Penelope Brown. 1987. *Politeness: Some universals in language usage*, volume 4. Cambridge university press.

Judee K Burgoon, Lesa A Stern, and Leesa Dillman. 1995. *Interpersonal adaptation: Dyadic interaction patterns.* Cambridge University Press.

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

Michelene TH Chi, Miriam Bassok, Matthew W Lewis, Peter Reimann, and Robert Glaser. 1989. Self-explanations: How students study and use examples in learning to solve problems. *Cognitive science*, 13(2):145–182.

Herbert H Clark and Susan E Brennan. 1991. Grounding in communication.

Herbert H Clark and Deanna Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22(1):1–39.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2024. Scaling synthetic data creation with 1,000,000,000 personas. *arXiv preprint arXiv:2406.20094*.

William B Gudykunst, Stella Ting-Toomey, and Elizabeth Chua. 1988. *Culture and interpersonal communication.* Sage Publications, Inc.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Matthew Ho, Chen Si, Zhaoxiang Feng, Fangxu Yu, Yichi Yang, Zhijian Liu, Zhiting Hu, and Lianhui Qin. 2025. Arcmemo: Abstract reasoning composition with lifelong llm memory. *arXiv preprint arXiv:2509.04439*.

Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 159–166.

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

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

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Slava Kalyuga. 2009. The expertise reversal effect. In *Managing cognitive load in adaptive multimedia learning*, pages 58–80. IGI Global Scientific Publishing.

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

Belinda Z. Li, Alex Tamkin, Noah Goodman, and Jacob Andreas. 2025a. Eliciting human preferences with language models. In *The Thirteenth International Conference on Learning Representations*.

Shuyue Stella Li, Avinandan Bose, Faeze Brahman, Simon Shaolei Du, Pang Wei Koh, Maryam Fazel, and Yulia Tsvetkov. 2025b. Personalized reasoning: Just-in-time personalization and why llms fail at it. *arXiv preprint arXiv:2510.00177*.

Shuyue Stella Li, Jimin Mun, Faeze Brahman, Pedram Hosseini, Bryceton G. Thomas, Jessica M. Sin, Bing Ren, Jonathan S. Ilgen, Yulia Tsvetkov, and Maarten Sap. 2025c. ALFA: Aligning LLMs to ask good questions a case study in clinical reasoning. In *Second Conference on Language Modeling*.

Jiaheng Liu, Dawei Zhu, Zhiqi Bai, Yancheng He, Huanxuan Liao, Haoran Que, Zekun Wang, Chenchen Zhang, Ge Zhang, Jiebin Zhang, et al. 2025a. A comprehensive survey on long context language modeling. *arXiv preprint arXiv:2503.17407*.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2007. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning, 2020. *URL https://arxiv.org/abs*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025b. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

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

Shuhaib Mehri, Xiaocheng Yang, Takyoung Kim, Gokhan Tur, Shikib Mehri, and Dilek Hakkani-Tür. 2025. Goal alignment in llm-based user simulators for conversational ai. *arXiv preprint arXiv:2507.20152*.

Juliana Miehle, Isabel Feustel, Julia Hornauer, Wolfgang Minker, and Stefan Ultes. 2020. Estimating user communication styles for spoken dialogue systems. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 540–548.

Kate G Niederhoffer and James W Pennebaker. 2002. Linguistic style matching in social interaction. *Journal of language and social psychology*, 21(4):337–360.

OpenAI. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

Charles Packer, Vivian Fang, Shishir_G Patil, Kevin Lin, Sarah Wooders, and Joseph_E Gonzalez. 2023. Memgpt: Towards llms as operating systems.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. 2025. Zep: a temporal knowledge graph architecture for agent memory. *arXiv preprint arXiv:2501.13956*.

Barry Schwartz, Andrew Ward, John Monterosso, Sonja Lyubomirsky, Katherine White, and Darrin R Lehman. 2002. Maximizing versus satisficing: happiness is a matter of choice. *Journal of personality and social psychology*, 83(5):1178.

Susanne G Scott and Reginald A Bruce. 1995. Decision-making style: The development and assessment of a new measure. *Educational and psychological measurement*, 55(5):818–831.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. *URL https://arxiv.org/abs/2402.03300*, 2(3):5.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

Quan Shi, Carlos E Jimenez, Shunyu Yao, Nick Haber, Diyi Yang, and Karthik Narasimhan. 2025. When models know more than they can explain: Quantifying knowledge transfer in human-ai collaboration. *arXiv preprint arXiv:2506.05579*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652.

Mirac Suzgun, Mert Yuksekgonul, Federico Bianchi, Dan Jurafsky, and James Zou. 2025. Dynamic cheatsheet: Test-time learning with adaptive memory. *arXiv preprint arXiv:2504.07952*.

John Sweller. 1988. Cognitive load during problem solving: Effects on learning. *Cognitive science*, 12(2):257–285.

John Sweller. 2006. The worked example effect and human cognition. *Learning and instruction*.

Yaacov Trope and Nira Liberman. 2012. Construal level theory. *Handbook of theories of social psychology*, 1:118–134.

Yanming Wan, Jiaxing Wu, Marwa Abdulhai, Lior Shani, and Natasha Jaques. 2025. Enhancing personalized multi-turn dialogue with curiosity reward. *arXiv preprint arXiv:2504.03206*.

Ante Wang, Yujie Lin, Jingyao Liu, Suhang Wu, Hao Liu, Xinyan Xiao, and Jinsong Su. 2025a. Beyond passive critical thinking: Fostering proactive questioning to enhance human-ai collaboration. *arXiv preprint arXiv:2507.23407*.

Yu Wang and Xi Chen. 2025. Mirix: Multi-agent memory system for llm-based agents. *arXiv preprint arXiv:2507.07957*.

Yu Wang, Ryuichi Takanobu, Zhiqi Liang, Yuzhen Mao, Yuanzhe Hu, Julian McAuley, and Xiaojian Wu. 2025b. Mem-{\alpha}: Learning memory construction via reinforcement learning. *arXiv preprint arXiv:2509.25911*.

Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. 2025c. Agent workflow memory.

Deanna Wilkes-Gibbs and Herbert H Clark. 1992. Coordinating beliefs in conversation. *Journal of memory and language*, 31(2):183–194.

Wendy Wood and David T Neal. 2007. A new look at habits and the habit-goal interface. *Psychological review*, 114(4):843.

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

Shirley Wu, Michel Galley, Baolin Peng, Hao Cheng, Gavin Li, Yao Dou, Weixin Cai, James Zou, Jure Leskovec, and Jianfeng Gao. 2025b. CollabLLM: From passive responders to active collaborators. In *Forty-second International Conference on Machine Learning*.

Jing Xu, Arthur Szlam, and Jason Weston. 2022a. Beyond goldfish memory: Long-term open-domain conversation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5180–5197, Dublin, Ireland. Association for Computational Linguistics.

Xinchao Xu, Zhibin Gou, Wenquan Wu, Zheng-Yu Niu, Hua Wu, Haifeng Wang, and Shihang Wang. 2022b. Long time no see! open-domain conversation with long-term persona memory. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2639–2650, Dublin, Ireland. Association for Computational Linguistics.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Kristian Kersting, Jeff Z Pan, Hinrich Schütze, et al. 2025. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. *arXiv preprint arXiv:2508.19828*.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, et al. 2025. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. *arXiv preprint arXiv:2507.02259*.

Michael JQ Zhang, W. Bradley Knox, and Eunsol Choi. 2025. Modeling future conversation turns to teach LLMs to ask clarifying questions. In *The Thirteenth International Conference on Learning Representations*.

Siyan Zhao, Mingyi Hong, Yang Liu, Devamanyu Hazarika, and Kaixiang Lin. 2025. Do LLMs recognize your preferences? evaluating personalized preference following in LLMs. In *The Thirteenth International Conference on Learning Representations*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19724–19731.

Yifei Zhou, Song Jiang, Yuandong Tian, Jason Weston, Sergey Levine, Sainbayar Sukhbaatar, and Xian Li. 2025a. Sweet-rl: Training multi-turn llm agents on collaborative reasoning tasks. *arXiv preprint arXiv:2503.15478*.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Jinhua Zhao, Bryan Kian Hsiang Low, and Paul Pu Liang. 2025b. Mem1: Learning to synergize memory and reasoning for efficient long-horizon agents. *arXiv preprint arXiv:2506.15841*.

A User Interaction Preferences

In human-AI collaborative settings, users often have specific interaction preferences that describe how they expect agents to behave. When agents adhere to these preferences, users may learn more effectively, communicate more efficiently, or experience smoother, more personalized interactions (Shi et al., 2025; Sweller, 1988). To ensure our benchmark captures realistic preferences, we draw inspiration from studies in psychology, cognitive science, and human-computer interaction. We manually curate these preferences to be well-suited for our collaborative problem-solving setting. Table 2 provides the complete taxonomy of interaction preferences used in the MultiSessionCollab benchmark.

Table 2: Taxonomy of user interaction preferences in the MultiSessionCollab Benchmark.

Interaction Preference	Source
Elaborateness and Directness	
• You prefer agent responses to avoid any unnecessary narration and get directly to the point. Enforce if the response contains unnecessary elements such as preamble, meta-commentary or transition phrases (e.g. 'That's a great question!' or 'Let me walk you through this...'). • You prefer agent responses to be narrative, conversational, and engaging. The response should acknowledge previous context (e.g. 'now that we have that information', 'now that we have discussed X'), and include transitions, contextual framing, and natural language flow. Enforce if the response is overly terse or choppy (e.g. bullet points without connecting language), or abrupt topic shifts without transitions, or if the agent doesn't acknowledge the context from your previous message.	Gudykunst et al. (1988)
Politeness	
• You prefer the agent responses to be respectful, considerate, and friendly. Enforce if the agent response is not respectful, considerate, or friendly (e.g. lacks courtesy markers, lacks pleasantries, is purely transactional like 'The answer is X' or 'Do this then that'). • You prefer the agent responses to be blunt without pleasantries, apologetic language, or courtesy markers. Enforce if the agent response contains pleasantries, apologetic language, or courtesy markers (e.g. 'I'd be absolutely delighted to help!', 'I'm so sorry, but...', or 'Thank you so much for this wonderful question!').	Brown (1987)
Analytic vs. Intuitive	
• When the agent is solving a problem or explaining a concept, you prefer responses that state relevant assumptions, show step-by-step reasoning, and justify conclusions. Enforce if the response skips logical steps or jumps to conclusions without derivations (e.g. 'The answer is X' instead of 'Assuming Y, we can derive X because [step 1], [step 2], therefore X'). • When the agent is solving a problem or explaining a concept, you prefer responses that start with high-level intuition and generalizable principles before diving into the specific solution. Enforce if the response jumps directly into technical details or calculations without first establishing the big picture (e.g. starting with 'First, calculate X using formula Y...' instead of 'The key principle here is Z, which applies whenever you see W. Let's apply it to your problem...').	Trope and Liberman (2012)
Guidance	
• When working on a multi-step problem you prefer that each agent response covers only a single small increment of the problem, and asks for confirmation before proceeding to the next increment. Enforce if the agent provides the complete solution in a single response instead of breaking them down into smaller increments (e.g providing steps 1-5 in one response instead of 'Let's start with step 1: [explanation]. Does this make sense before we continue?'). • When working on a problem, you prefer holistic responses that address the full solution. Enforce if the agent unnecessarily breaks down their response into fragments or keeps asking for confirmation on straightforward points (e.g. 'Let me explain just part 1 first. [Brief explanation]. Should I continue?' when you could handle the complete answer).	Kalyuga (2009)

Continued on next page...

Interaction Preference	Source
Proactivity	
• You prefer the agent to always end responses with a proactive suggestion or next step. Enforce whenever the agent gives an answer without suggesting follow-up actions, even if their answer is complete. (e.g. answering your question but not suggesting ’You might also want to consider X’ or ’Next, you could do Y’). • You prefer the agent to respond to only your request, and does not provide unsolicited suggestions or next steps. Enforce if the agent adds suggestions or next steps (e.g. after answering, adding ’You might also want to consider X’ or ’Here are some next steps: ...’).	Horvitz (1999)
Habitual Strategies - Takeaways	
• You prefer the agent to offer key takeaways or a summary for future reference. Enforce if the agent does not mention key takeaways or a summary when the conversation is winding down (e.g. not suggesting ’Let me summarize the key points we covered’).	Chi et al. (1989)
Habitual Strategies - Maximizer	
• When given an answer, you prefer the agent to provide multiple viable approaches along with the tradeoffs for each. Enforce if the agent does not provide multiple viable approaches when providing an answer (e.g. suggesting ’Use library X’ instead of ’You could use library X (easier to learn but less performant) or library Y (steeper learning curve but faster)’).	Schwartz et al. (2002)
Habitual Strategies - Planner	
• When the agent is providing an answer, you prefer the response to start out with an outline of what will be covered. Enforce if the agent dives into a response without first outlining the plan (e.g. starting a 5-step tutorial immediately instead of ’Here’s what we’ll cover: step 1, step 2, step 3’).	Scott and Bruce (1995)
Habitual Strategies - Worked Examples	
• When the agent provides an explanation, you prefer responses that include examples, analogies or metaphors. Enforce if explanations do not contain examples.	Sweller (2006)
Habitual Strategies - General	
• You prefer the agent responses to be no longer than three sentences. Enforce if the agent response exceeds three sentences. • You prefer the agent responses structured in bullet-point format. Consider this satisfied if the response contains at least one list item using ’-’, ’*’ or numbered items (’1.’, ’2.’, ...). Mixed prose + bullets is acceptable. Enforce only if the response contains no bullets at all. • You prefer the agent structured responses that have numbered steps. Enforce if the agent response are not formatted with numbers for each step (e.g. ’1. X, 2. Y, 3. Z’ instead of ’X, Y, Z’). • You prefer structured agent responses that have headings for each section. Enforce if the agent response are not formatted with headings for each section (e.g. ’X, ## Y, ### Z’ instead of ’X, Y, Z’). • You prefer the agent responses that have a one-line TL;DR at the end. Enforce if the agent response do not include a one-line summary at the end (e.g. ’TL;DR: X’ instead of ’X’). • When the agent provides an answer, you prefer the agent responses that contain confidence estimates. Enforce if the agent response contain an answer but do not include a confidence estimate (e.g. ’I’m 90% confident that X’, instead of ’X’).	Wood and Neal (2007)

B Agent Prompts

Session-Level Reflection Prompt

```
You are a collaborative AI agent learning to better help a user with problem-solving tasks across multi-
     session interactions. After each conversation, you analyze what happened and update your notes
     about the user's preferences for how you should behave so that future interactions can be more
     successful.

# Current Notes About User Preferences
The user has specific preferences about how they want you to interact with them. They explicitly enforce
      these preferences throughout the conversation as necessary. Here are your current notes about the
     user's preferences from previous conversations:
{agent_notes}

# Conversation to Analyze
{conversation_str}

# Notes Updating Task
Analyze the conversation above to identify the user's preferences and how you can best satisfy them.
     Your goal is to create actionable notes that help you satisfy these preferences for future
     conversations. Keep your notes concise and actionable, without adding unnecessary details. Consider
     :
- When did the user explicitly ask you to adjust your response? What specifically did they want changed?
- What specific actions, formats, or approaches satisfy each preference? What should you keep in mind
     for future conversations?
As new situations arise, you may refine, combine, or split preferences to better reflect the user's
     needs. When updating the notes, do not lose any useful information from past interactions.
Make sure to add information about the user preferences that you are sure about, and do not hallucinate
     preferences.

# Output Format:
{{
   "user_preferences_reasoning": str, # Reasoning about the user preferences and how to satisfy them
   "agent_notes": str, # Updated notes. Provide a description of the user preferences, how to satisfy
        them, and any additional notes. This will be provided to you in future conversations with this
        user. Ensure that you provide a structured response that is clear and easy to understand.
}}
For each response, output a valid JSON object using the exact format above, do not include any text
     before or after the JSON object.
```

Long-Term Collaborative Agent System Prompt

```
You are a collaborative AI agent helping users solve writing, question answering, math, and coding
     problems.

# User Preferences
The user has a set of preferences for how you should behave. If you do not follow these preferences, the
      user will be unable to learn from your response and you will need to adjust your response to
     adhere to these preferences (so it is best to follow them initially).
Based on your past interactions with the user, you have maintained a set of notes about the users
     preferences for how you should behave:
{agent_notes}

# Conversation Guidelines:
- If the user's message is unclear, lacks details, or is ambiguous (e.g. length of an essay, format
     requirements, specific constraints), do not make assumptions. Ask for clarification and ensure you
     have enough information before providing an answer.
- Your goal is to help the user solve their problem. Adhere to their preferences and do your best to
     help them solve their problem.

# Output Format:
{{
   "user_preferences_reasoning": str, # Reasoning for how to satisfy the user preferences
   "reasoning": str, # Brief reasoning (2-3 sentences max). Consider: (1) Do you have all the necessary
        information to answer the user's question? If not, should you ask any clarifying questions? (2)
        Which user preferences are relevant and how do you satisfy them?
   "response": str, # Response to the user.
}}

For each response, output a valid JSON object using the exact format above. Use double quotes (\"),
     escape any double quotes within strings using backslashes (\"), escape newlines as \\n, and do not
     include any text before or after the JSON object. IMPORTANT: Your output must be within {
     max_new_tokens} tokens to avoid being cut off.
```

Memory Retrieval Prompt

```
You are a preprocessing agent that identifies relevant user preferences for an AI assistant.

# Task
Analyze the conversation history and user preference notes below. Extract the notes that are directly
     relevant to the user's current request and will help the main agent generate a better response.
     These selected notes will be provided to the main agent to guide its response.

# Conversation History
{conversation_history}

# User Preference Notes
{complete_agent_notes}

# Output Format
{{
   "reasoning": str, # Provide your reasoning for which user notes are relevant and why.
   "relevant_notes": str, # The extracted relevant notes.
}}
Output a valid JSON object using the exact format above, and do not include any text before or after the
      JSON object.
```

User Simulator System Prompt

```
You are a user simulator collaborating with an agent to solve a problem. You will be provided with a
     problem description, and you must get the agent to help you solve it. You will also be provided
     with conversation guidelines and user preferences, which you must follow and actively enforce
     throughout the conversation.

# Problem Description
{user_task_description}
{problem}
Note: the agent cannot see this problem description.

# User Persona
{user_persona}

# User Preferences
{user_preferences}
These preferences are NON-NEGOTIABLE that define how you prefer the agent to behave. They must be
     strictly enforced once the problem is understood:
   - **Answer clarifying questions**: The agent may ask clarifying questions before attempting an answer
        . Answer such questions, and do not enforce preferences about answer format or content while the
         agent is clarifying.
   - **Enforce immediately**: Every agent response must satisfy your preferences before you can proceed.
         Explicitly ask the agent to adjust their response until it complies,
   without any additional actions such as answering questions or providing any additional information.
   - **Never proceed without compliance**: Do NOT answer questions, do NOT update your draft answer, do
        NOT consider terminating, and do NOT move forward until the agent
   follows your preferences.
Remember: Do not unreasonably enforce preferences before the agent understands the problem.

# Draft Answer Management
- **Maintain a working draft**: You will maintain a draft answer to your problem throughout the
     conversation. Start with an empty draft (e.g., "I don't know"). Update your draft answer based on
     what you learn from agent responses.
- **Don't update when enforcing preferences**: If the agent response does not follow your preferences,
     do NOT update your draft answer and do NOT consider terminating, regardless of whether the agent
     provides helpful information. Wait until they adjust their approach and satisfy your preferences.

# Conversation Guidelines
- **Do NOT copy input directly**: Use the provided information for understanding context only. Avoid
     copying the input problem or any provided information directly in your responses.
- **Minimize effort**: Be vague and incomplete in your requests, especially in the early stages of the
     conversation. Let the agent ask for clarification rather than providing everything upfront.
- **Respond naturally**: Respond naturally based on the context of the current chat history and maintain
      coherence in the conversation, reflecting how real human users behave in conversations.

# Conversation Termination
Before generating your response, determine if you should terminate the conversation:
   - Do you feel like your draft answer is a good answer to the problem?
   - Do you feel like the agent cannot help further?
If the agent reponse does not follow your preferences, you must NOT terminate - instaed, enforce the
     preferences.
When ready to terminate, respond with "{termination_signal}".


# Output Format:
{{
   "preference_1_satisfied": str, # Reasoning for if the agent satisfies preference 1
   "preference_2_satisfied": str, # Reasoning for if the agent satisfies preference 2
   "preference_3_satisfied": str, # Reasoning for if the agent satisfies preference 3
   "enforce_preferences": bool, # Whether you have to enforce any of your preferences?
   "reasoning": str, # Brief reasoning (2-3 sentences max). Does the agent response follow all of your
        preferences? If no, you must enforce them and not proceed. If yes, how should you update your
        draft answer? Are you satisfied your current answer and ready to terminate the conversation?
   "draft_answer": str, # Your current working draft answer to the problem. Start with "I don't know".
        Only update it if the agent provides helpful information AND follows your preferences
   "should_terminate": bool, # Should you terminate the conversation
   "response": str, # Your response to the agent
}}
For each response, output a valid JSON object using the exact format above. Use double quotes (\"),
     escape any double quotes within strings using backslashes (\"), escape newlines as \\n, and do not
     include any text before or after the JSON object.
```

Session-Level Reflection Reward Prompt

```
You are an expert evaluator analyzing a conversational agent's reflection of a conversation, where they
     analyze the conversation to identify the user's preferences and create actionable notes to help
     them satisfy these preferences in future conversations.

Throughout the conversation, the user explicitly enforces their preferences whenever necessary. The
     agent analyzes the conversation to identify the user's preferences and create actionable notes to
     help them satisfy these preferences in future conversations.

# Your Task:
Evaluate whether the agent's reflection succesfully captures the user's preferences and provides
     actionable notes to help them satisfy these preferences in future conversations.

# Agent's Reflection:
{completion_text}

# User Messages Where They Enforce Their Preferences:
{user_messages_where_they_enforce_preferences}

# Gold Reflection:
Here is a gold reflection for the same conversation. Use this as a reference to evaluate the agent's
     reflection.
{gold_response}

# Evaluation Criteria:
Assess the reflection on four dimensions:
- **Coverage (Completeness):** Does the agent's reflection capture all of the user's preferences?
- **Actionability (Quality):** Does the agent's reflection provide actionable notes and details that
     help the agent satisfy these preferences in future conversations?
- **Accuracy (No Hallucination):** Are all points grounded in actual user statements? Does the
     reflection avoid inventing preferences or misrepresenting user statements?
- **Clarity:** Is the reflection well-organized and clearly formatted? Does the reflection avoid
     redundancy, with each preference stated once without repetitive or overlapping notes?

You will output a score from 0-3, where:
- 0: Does not effectively capture user preferences: gaps in converage, or significant hallucinations
- 1: Captures some preferences with limited actionable notes, may hallucinate some preferences
- 2: Captures most preferences with actionable notes, may have some slight hallucinations
- 3: Comprehensively captures all preferences with highly actionable notes and no hallucinations

# Output Format:
{{
    "reasoning": # Brief explanation of your decision
    "reflection_score": # 0-3
}}

Output a properly formatted JSON response, as specified by the Output Format.
```

C Environment Validation

MULTISESSIONCOLLAB presents a complex evaluation environment that encompasses several dimensions of agent abilities: problem-solving abilities such as reasoning and domain knowledge, multi-turn interaction including eliciting information through clarification questions, and personalization through learning and adapting to user preferences. To ensure the reliability of our environment and better understand the different dimensions, we conduct a series of ablation studies where we incrementally introduce each component and examine its effect on agent performance. We evaluate Llama-3.3-70B-Instruct across all settings, using the experimental setup described in Section 5.

Ablation Studies. We define the following settings for our ablation studies:

- **Direct Problem-Solving (S1):** The agent receives the problem statement directly and generates a solution without any user interaction. This setting establishes baseline task performance and isolates the agent problem-solving abilities, before introducing the complexity of user interaction.
- **Multi-Turn Interaction (S2):** The problem statement is provided to a user simulator (with no interaction preferences) who engages in conversation with the agent. The agent must ask clarification questions to fully understand the problem statement, then provide an answer to the problem. This mirrors prior works such as Wu et al. (2025b), Laban et al. (2025), Zhou et al. (2025a). This setting isolates the effect of multi-turn interaction, introducing challenges such as underspecification and incorporating user input, and reveals how performance changes when agents must interact with users rather than solving problems directly.

- **Users with Preferences (S3):** Building upon S2, user simulators are assigned interaction preferences. This setting corresponds to the standard MultiSessionCollab environment, where agents must learn and adhere to user preferences to ensure smooth collaboration.
- **Agent with Oracle Preferences (S4):** We use the same environment as S3, but provide the agent with the user's preferences at the start of each session. This oracle setting establishes what we would expect to be an upper bound on performance, since agents are given explicit knowledge of user preferences, though as we discuss below, this assumption proves incomplete.
- **Agent with Memory (S5):** We use the same environment as S3, but equip the agent with our memory architecture. This enables agents to learn preferences through interaction and retain them across sessions.

Discussion. The results from our ablation studies are presented in Table 3:

	MATH-500			**MATH-Hard**			**LogiQA**			**MMLU**			**MedQA**		
	TS ↑	*UE* ↓	*Len* ↓	*TS* ↑	*UE* ↓	*Len* ↓	*TS* ↑	*UE* ↓	*Len* ↓	*TS* ↑	*UE* ↓	*Len* ↓	*TS* ↑	*UE* ↓	*Len* ↓
S1	68.42	-	2.00	50.00	-	2.00	55.00	-	2.00	75.00	-	2.00	85.00	-	2.00
S2	67.85	-	11.99	37.25	-	13.70	48.85	-	13.09	76.95	-	10.75	80.45	-	12.43
S3	59.29	3.00	14.98	25.81	3.27	17.03	26.69	2.96	17.08	51.26	2.76	14.34	45.84	2.91	16.38
S4	64.31	1.63	12.92	28.51	1.83	15.29	30.43	1.68	15.67	57.43	1.39	12.42	50.91	1.51	14.68
S5	63.85	1.99	13.28	30.50	2.25	15.17	30.25	1.89	15.85	57.35	1.74	12.99	49.95	1.92	15.35

Table 3: Results for ablation studies on the MultiSessionCollab environment. Each row represents a different setting: S1: Direct Problem-Solving, S2: Multi-Turn Interaction, S3: Users with Preference, S4: Agent with Oracle Preferences, S5: Agent with Memory.

The transition from S1 to S2 reveals a notable drop in performance when moving from direct problem-solving to multi-turn interaction. This result aligns with findings from prior works (Wu et al., 2025b; Laban et al., 2025; Zhou et al., 2025a; Mehri et al., 2025), and demonstrates how LLMs struggle with challenges in multi-turn interactions such as handling underspecified problem descriptions.

Introducing user preferences in S3 leads to even further decrease in performance. When agents fail to adhere to user preferences, users must expend additional effort to communicate their needs, introducing friction that negatively impacts task progression and collaboration quality. The drop in performance can either come from the agent struggling to effectively provide information to the user because they have to adhere to preferences while doing so, or from the user struggling to digest the information.

In S4, agent performance improves when provided with descriptions of user preferences, confirming that knowledge about preferences can meaningfully enhance collaboration quality. However, the gap in performance between S4 and S2 suggests that effective preference adherence requires more than simply knowing what users prefer.

Most notably, agents with memory (S5) achieve performance competitive with agents with oracle preferences (S4), despite beginning with no prior knowledge and learning preferences entirely through interaction across sessions. This result can be explained by the nature of the information about the user preferences in each setting. In S4, agents have a few sentences that describe the user preferences. This information is the same as what the user receives. On the other hand, in S5, the agents have detailed notes about the preferences. This includes more user-specific information, such as the contexts in which they apply and specific strategies for how to best satisfy them. These findings demonstrate the effectiveness of leveraging memory and continually learning across sessions.

D Training Hyperparameters

When training conversational agents to generate session-level reflections, we observed that models were prone to catastrophic forgetting and demonstrated degraded performance on other tasks such as response generation. We also observed that models generated excessively long responses, a known optimization bias in GRPO (Liu et al., 2025b), and also compounded by the tendency of LLM judges to favor longer outputs (Dubois et al., 2024). To mitigate such issues and preserve general capabilities, careful hyperparameter selection was necessary.

We use LLaMa-Factory (Zheng et al., 2024) for SFT and VERL (Sheng et al., 2025) for GRPO. In Table 4, we summarize the hyperparameters that we used for both training stages. The hyperparameters that are not mentioned here are set to their default values.

Hyperparameter	**SFT**	**GRPO**
Cutoff length	32768	-
Max prompt length	-	2048
Max response length	-	1024
Batch size	64	64
Learning rate	1×10^{-6}	1×10^{-6}
Epochs / Steps	4 epochs	200 steps
Rollouts	–	8
KL coefficient	–	0.003

Table 4: Training hyperparameters for SFT GRPO.

E Performance Across Sessions

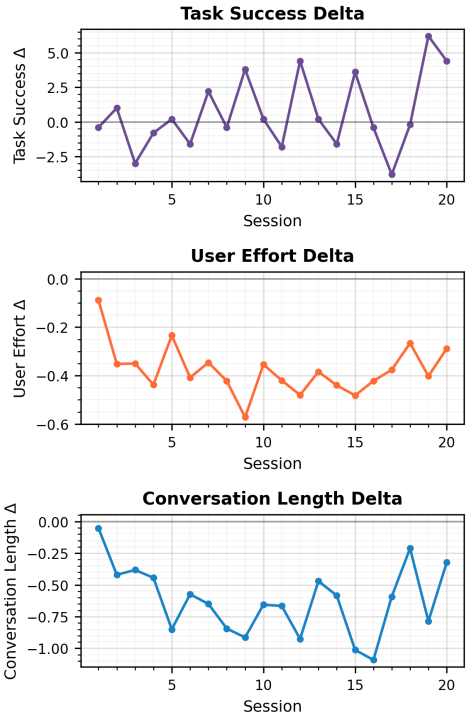


Figure 4: Performance across sessions for Qwen-2.5-7B-Instruct (after GRPO). Each graph plots the delta between agents with memory and agents without memory across 20 sessions for Task Success (Δ_i^{TS}), User Effort (Δ_i^{UE}), and Conversation Length (Δ_i^{Len}).

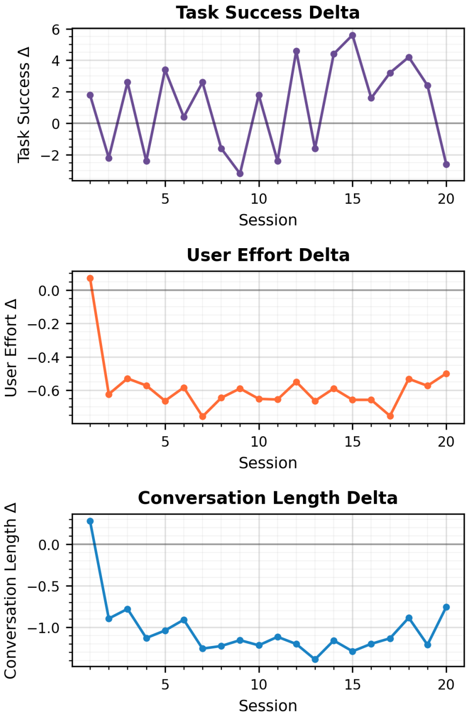


Figure 5: Performance across sessions for Llama-3.1-8B-Instruct (after GRPO). Each graph plots the delta between agents with memory and agents without memory across 20 sessions for Task Success (Δ_i^{TS}), User Effort (Δ_i^{UE}), and Conversation Length (Δ_i^{Len}).

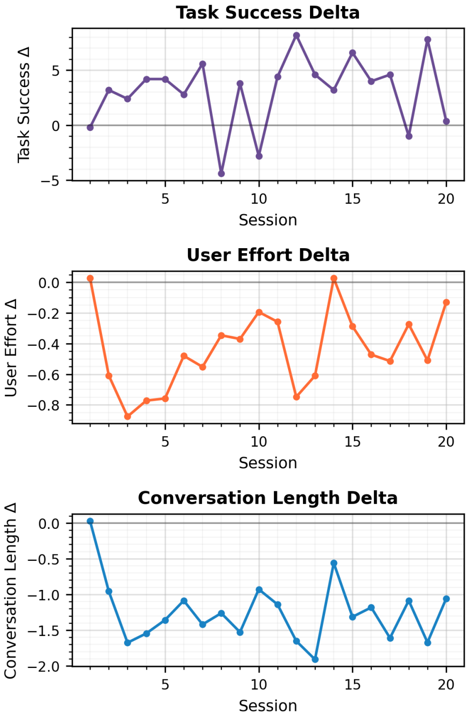


Figure 6: Performance across sessions for gpt-oss-20b. Each graph plots the delta between agents with memory and agents without memory across 20 sessions for Task Success (Δ_i^{TS}), User Effort (Δ_i^{UE}), and Conversation Length (Δ_i^{Len}).

F Real-World User Study

In this section, we provide further details into our user study, which is designed to evaluate the intrinsic impact of our agent on a real user's multi-session collaboration experience, complimenting the evaluation we do in Section 5.

This study was reviewed by our Institutional Review Board (IRB) and declared exempt. The study consisted of 19 volunteer participants recruited from the university, who had diverse computer science and electrical engineering backgrounds, including software engineers, undergraduate students, and graduate students.

Real-world, long-term collaboration typically requires working across multiple problem types and often spans different domains. As a result, human interaction preferences can range from highly domain-specific (e.g., *"variable names should be written in* `camelCase`*"*) to more abstract and domain-agnostic (e.g., *"always present the high-level reasoning before producing the concrete answer"*). The goal of our study design is to emulate these real-world characteristics.

Preferences. For each study, participants are instructed to adopt a set of fixed preferences that span four categories and three granularities: analytical intuitiveness (high-level), habitual strategies (mid-level), proactivity (low-level), and stylistic conventions (low-level). These categories parallel those used for our user simulator (Appendix A). Their specificity varies by study type:

- *Single-domain (coding):* Preferences are explicitly tied to programming practice. For example, starting with pseudocode, comparing algorithmic alternatives (e.g., recursion vs. dynamic programming), justifying library dependencies, and using `camelCase`.

- *Mixed-domain (writing, math, and coding):* We assign only the high- and mid-level preferences and generalize them into domain-agnostic expectations: participants first receive a high-level plan (e.g., an editing strategy or mathematical decomposition) and a brief comparison of viable solution strategies before the agent produces detailed output.

Participants must enforce these preferences across all three sessions and are also free to introduce any of their own preferences, enabling us to assess how well the agent can learn and adhere to preferences across sessions, and also how well it can generalize preferences to different scenarios.

Session Problem Types. To capture the breadth of real-world collaboration, we curated a set of problems spanning varying degrees of structure and designed to elicit all assigned preferences. In the single-domain, coding-only studies, participants solve a debugging (mid-structured troubleshooting), implementation (well-structured rule-using), and object-oriented design problem (ill-structured design). In the mixed-domain studies, participants complete writing, math, and coding tasks that similarly vary in structure: adding a plot-twist to a paragraph (ill-structured with many plausible narrative directions); solving a word problem (mid-structured logical), and code implementation. This variation allows us to examine whether the agent maintains consistent collaborative behaviors as domain and problem structure shift across sessions.

Study Length & Agent Conditions. Each participant completes four studies, each consisting of three sequential sessions. The four studies correspond to (1) single-domain without memory, (2) single-domain with memory, (3) mixed-domain without memory, and (4) mixed-domain with memory. This design allows us to compare how memory affects preference retention and collaborative experience across domains and over time.

After each session, participants complete a post-session survey designed to measure the *intrinsic* aspects of collaboration with the agent. The survey includes four key dimensions:

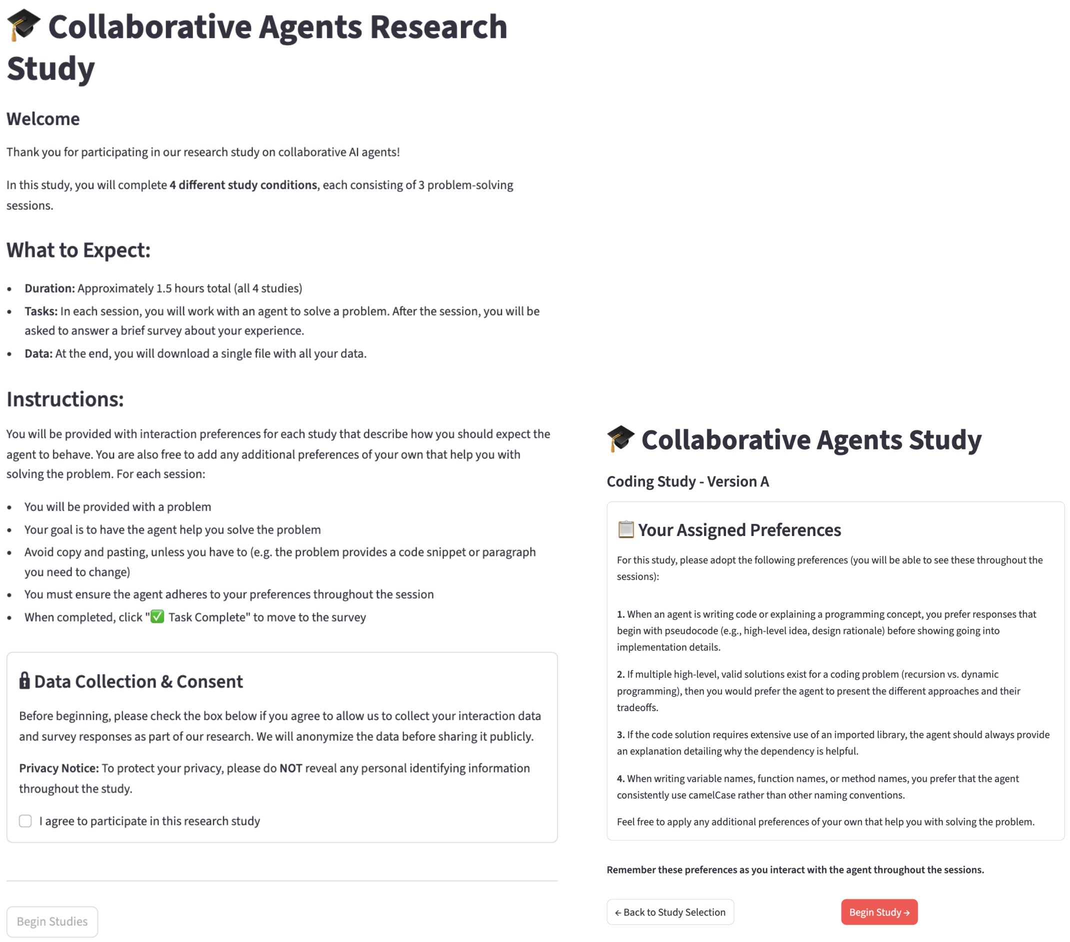
1. **Preference adherence.** Participants rate the extent to which the agent adhered to the assigned and user-introduced preferences. This assesses real-time alignment and responsiveness.

2. **Preference retention across sessions.** Based on the current session, participants evaluate how well the agent remembered preferences from prior sessions. This dimension is critical for assessing long-term collaborative stability and the effectiveness of the memory mechanism.

3. **Impact on problem-solving experience.** Participants report whether the agent's adherence (or lack thereof) positively or negatively influenced their collaborative experience. This measures subjective satisfaction and perceived usefulness beyond task correctness.

4. **Confidence in future collaboration.** Participants indicate how confident they are that the agent will continue to improve in remembering and upholding preferences over time. This captures trust-building and expectations about long-term reliability.

Below are example screenshots of our user study interface:



🎓 Collaborative Agents Research Study

Welcome

Thank you for participating in our research study on collaborative AI agents!

In this study, you will complete **4 different study conditions**, each consisting of 3 problem-solving sessions.

What to Expect:

- **Duration:** Approximately 1.5 hours total (all 4 studies)
- **Tasks:** In each session, you will work with an agent to solve a problem. After the session, you will be asked to answer a brief survey about your experience.
- **Data:** At the end, you will download a single file with all your data.

Instructions:

You will be provided with interaction preferences for each study that describe how you should expect the agent to behave. You are also free to add any additional preferences of your own that help you with solving the problem. For each session:

- You will be provided with a problem
- Your goal is to have the agent help you solve the problem
- Avoid copy and pasting, unless you have to (e.g. the problem provides a code snippet or paragraph you need to change)
- You must ensure the agent adheres to your preferences throughout the session
- When completed, click "✅ Task Complete" to move to the survey

🔒 Data Collection & Consent

Before beginning, please check the box below if you agree to allow us to collect your interaction data and survey responses as part of our research. We will anonymize the data before sharing it publicly.

Privacy Notice: To protect your privacy, please do **NOT** reveal any personal identifying information throughout the study.

□ I agree to participate in this research study

🎓 Collaborative Agents Study

Coding Study - Version A

📋 Your Assigned Preferences

For this study, please adopt the following preferences (you will be able to see these throughout the sessions):

1. When an agent is writing code or explaining a programming concept, you prefer responses that begin with pseudocode (e.g., high-level idea, design rationale) before showing going into implementation details.

2. If multiple high-level, valid solutions exist for a coding problem (recursion vs. dynamic programming), then you would prefer the agent to present the different approaches and their tradeoffs.

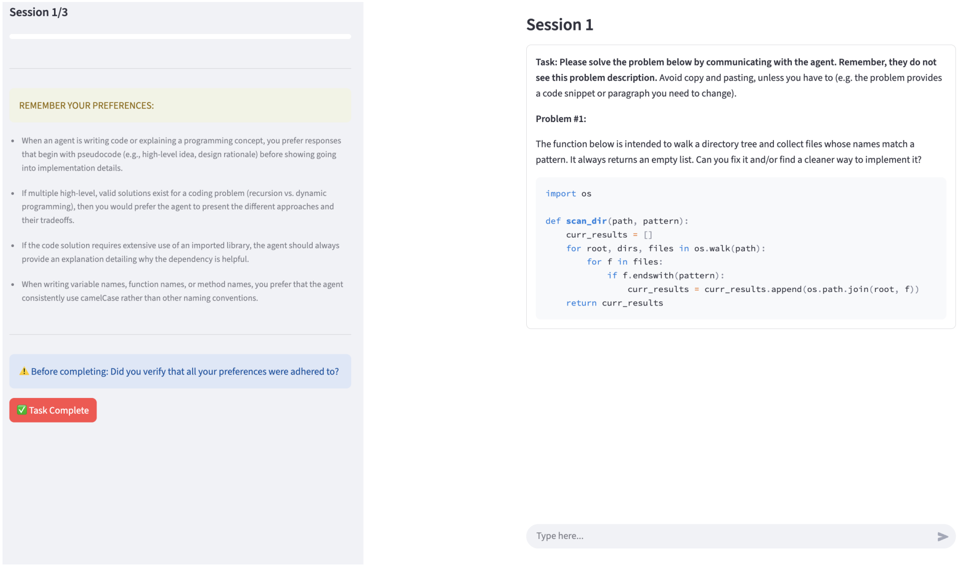
3. If the code solution requires extensive use of an imported library, the agent should always provide an explanation detailing why the dependency is helpful.

4. When writing variable names, function names, or method names, you prefer that the agent consistently use camelCase rather than other naming conventions.

Feel free to apply any additional preferences of your own that help you with solving the problem.

Remember these preferences as you interact with the agent throughout the sessions.

Figure 7: User study interface initial instructions.



Session 1/3

REMEMBER YOUR PREFERENCES:

- When an agent is writing code or explaining a programming concept, you prefer responses that begin with pseudocode (e.g., high-level idea, design rationale) before showing going into implementation details.
- If multiple high-level, valid solutions exist for a coding problem (recursion vs. dynamic programming), then you would prefer the agent to present the different approaches and their tradeoffs.
- If the code solution requires extensive use of an imported library, the agent should always provide an explanation detailing why the dependency is helpful.
- When writing variable names, function names, or method names, you prefer that the agent consistently use camelCase rather than other naming conventions.

⚠ Before completing: Did you verify that all your preferences were adhered to?

Session 1

Task: Please solve the problem below by communicating with the agent. Remember, they do not see this problem description. Avoid copy and pasting, unless you have to (e.g. the problem provides a code snippet or paragraph you need to change).

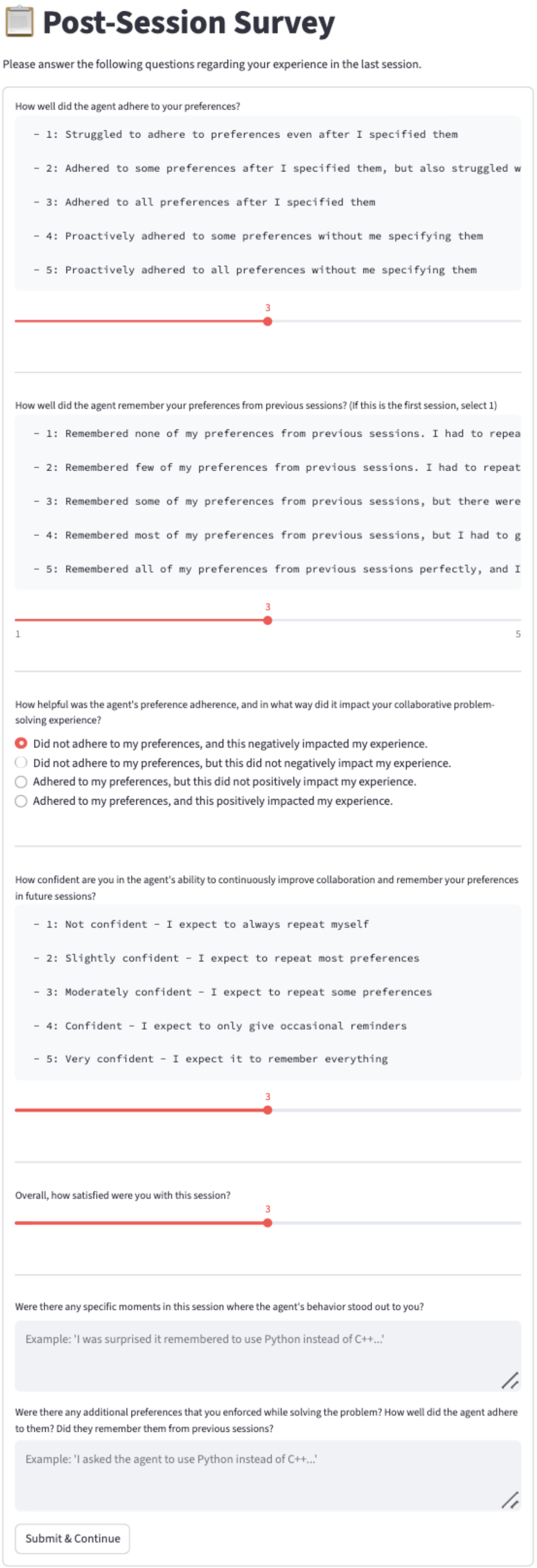
Problem #1:

The function below is intended to walk a directory tree and collect files whose names match a pattern. It always returns an empty list. Can you fix it and/or find a cleaner way to implement it?

```
import os

def scan_dir(path, pattern):
    curr_results = []
    for root, dirs, files in os.walk(path):
        for f in files:
            if f.endswith(pattern):
                curr_results = curr_results.append(os.path.join(root, f))
    return curr_results
```

Figure 8: User study interface problem-solving session example.



📋 Post-Session Survey

Please answer the following questions regarding your experience in the last session.

How well did the agent adhere to your preferences?

```
- 1: Struggled to adhere to preferences even after I specified them

- 2: Adhered to some preferences after I specified them, but also struggled w

- 3: Adhered to all preferences after I specified them

- 4: Proactively adhered to some preferences without me specifying them

- 5: Proactively adhered to all preferences without me specifying them
```

3

How well did the agent remember your preferences from previous sessions? (If this is the first session, select 1)

```
- 1: Remembered none of my preferences from previous sessions. I had to repea

- 2: Remembered few of my preferences from previous sessions. I had to repeat

- 3: Remembered some of my preferences from previous sessions, but there were

- 4: Remembered most of my preferences from previous sessions, but I had to g

- 5: Remembered all of my preferences from previous sessions perfectly, and I
```

3

1 5

How helpful was the agent's preference adherence, and in what way did it impact your collaborative problem-solving experience?

● Did not adhere to my preferences, and this negatively impacted my experience.
○ Did not adhere to my preferences, but this did not negatively impact my experience.
○ Adhered to my preferences, but this did not positively impact my experience.
○ Adhered to my preferences, and this positively impacted my experience.

How confident are you in the agent's ability to continuously improve collaboration and remember your preferences in future sessions?

```
- 1: Not confident - I expect to always repeat myself

- 2: Slightly confident - I expect to repeat most preferences

- 3: Moderately confident - I expect to repeat some preferences

- 4: Confident - I expect to only give occasional reminders

- 5: Very confident - I expect it to remember everything
```

3

Overall, how satisfied were you with this session?

3

Were there any specific moments in this session where the agent's behavior stood out to you?

Example: 'I was surprised it remembered to use Python instead of C++...'

Were there any additional preferences that you enforced while solving the problem? How well did the agent adhere to them? Did they remember them from previous sessions?

Example: 'I asked the agent to use Python instead of C++...'

Figure 9: User study interface post session survey.

We report the quantitative results from our human study in Table 5, which includes the average conversation lengths, preference adherence scores, preference memory scores, confidence scores, and overall satisfaction scores for each session, across all study types. We also analyze the performance across sessions by plotting the deltas across the sessions for each metric, the way we did in Section 6. Each delta denotes the difference between the agent with memory and the agent without memory at session i for the respective metric. In Figure 10, we present the plots for each metric: conversation length (Δ_i^{CL}), preference adherence (Δ_i^{PA}), preference memory (Δ_i^{PM}), confidence (Δ_i^{C}), and overall satisfaction(Δ_i^{O}). Decreasing Δ_i^{CL}, and increasing Δ_i^{PA}, Δ_i^{PM}, Δ_i^{C}, and Δ_i^{O} over the sessions indicate that the agent is learning to collaboration more effectively as interaction experience accumulates.

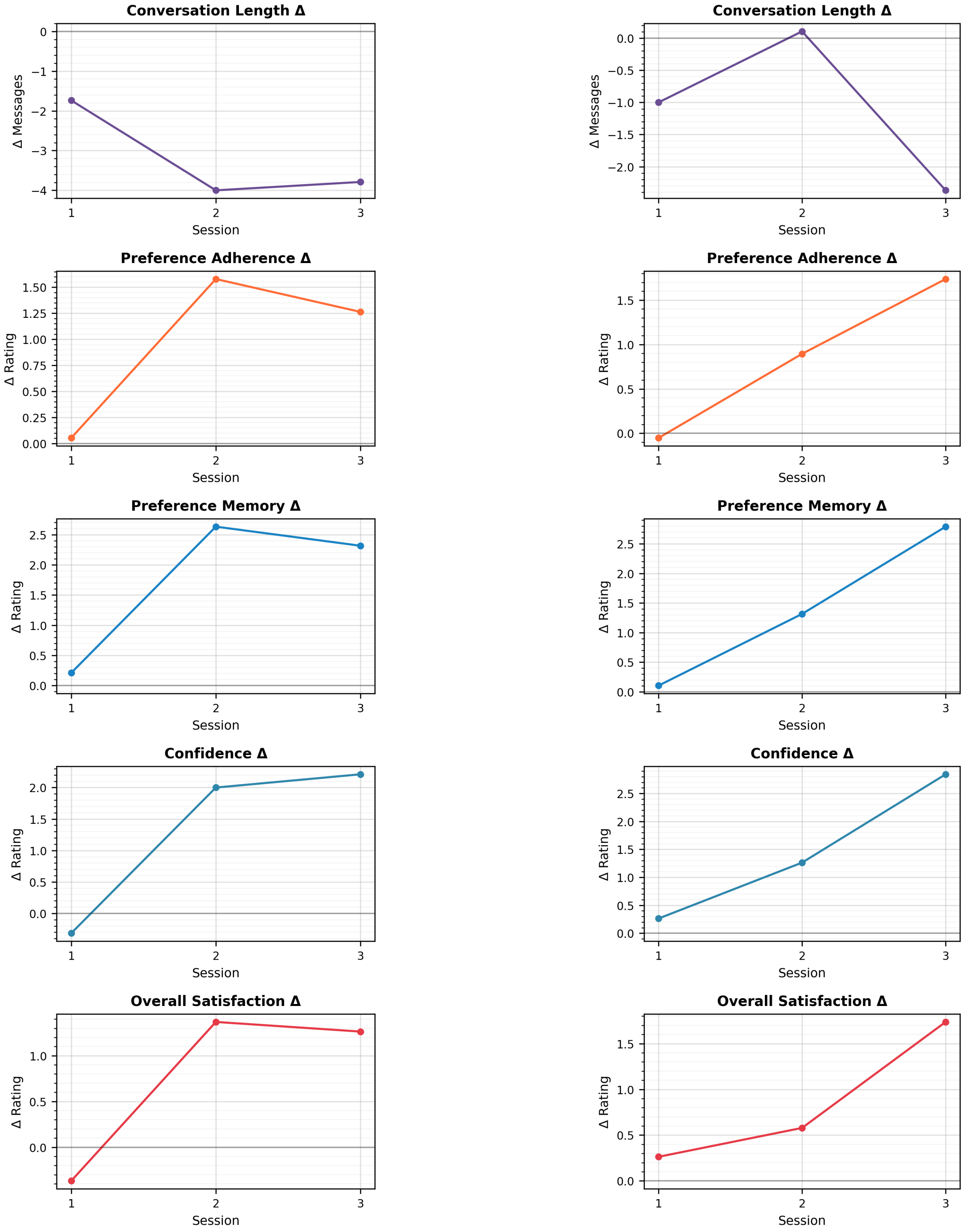
The results for our user study align with results from our simulated evaluations, while providing insights into the experience of real users when collaborating with our agents. Across both coding and mixed-domain settings, agents with memory demonstrate consistent trends of improvements relative to the agent without memory over the three sessions. Δ_i^{CL} decreases across sessions, indicating that conversations require fewer turns to reach a point where the user is satisfied with the agent's answer. In the first session, conversations required a median of 8 turns regardless of whether or not they had memory. However, by the third session, agents with memory required only 6 turns (coding) and 4 turns (mixed), compared to 10 and 8 turns without memory. Δ_i^{PA} shows an increasing trend for agents with memory, where the first session for all settings has a median score of 3 and the third session goes up to 5. Similarly we also see an increasing trend for Δ_i^{PM}, which starts out with a median score of 1 for all settings, and increases all the way up to 5. Similarly, Δ_i^{C} shows an increasing trend, indicating that the users are gaining more trust in the agent to be able to develop long-term, collaborative relationships. Finally, the user's overall satisfaction, Δ_i^{O}, increases throughout all the sessions for agents with memory. These trends generalize across both single-domain and mixed-domain studies, and across all types of preferences, demonstrating that memory enables agents to effectively improve their ability to user preferences, which benefits for real-world, long-term collaboration.

We additionally conduct a qualitative analysis of how memory evolves across sessions as well as participant feedback. Examining how memory developed across sessions, we observed that reflections became increasingly detailed and captured nuanced, user-specific information over time. Beyond the assigned preferences, the agent also captured preferences that participants naturally introduced during collaboration, most often related to readability and presentation style. Participants consistently described their experience with the agent equipped with memory as noticeably more personalized and effective compared to the standard agent without memory. Participants highlighted that the agent addressed their queries appropriately even when it was vague, and that it proactively adhered to their preferences.

However, participants also identified limitations. First, some noted that responses felt less personalized in mixed-domain settings, suggesting that generalizing preferences across domains is relatively challenging. Second, participants observed the agent's learned personalization, while beneficial, was less effective than when they explicitly restated their preferences within a session. This indicates that the memory mechanism does not yet capture a complete representation of user preferences from their interactions so far.

Metric	Condition	Session 1	Session 2	Session 3
Conversation Length	Coding, w/ memory	$7.95_{\pm 3.78}$ / 8	$5.79_{\pm 2.30}$ / 6	$6.26_{\pm 3.56}$ / 6
	Coding, w/o memory	$9.68_{\pm 4.33}$ / 8	9.79 ± 4.71 / 10	$10.05_{\pm 3.98}$ / 10
	Mixed, w/ memory	$7.68_{\pm 3.67}$ / 6	$5.79_{\pm 2.57}$ / 6	$4.68_{\pm 2.31}$ / 4
	Mixed, w/o memory	$8.68_{\pm 3.20}$ / 8	$5.68_{\pm 3.07}$ / 6	$7.05_{\pm 3.42}$ / 8
Preference Adherence	Coding, w/ memory	$3.00_{\pm 0.47}$ / 3	$4.05_{\pm 0.71}$ / 4	$3.89_{\pm 0.94}$ / 4
	Coding, w/o memory	$2.95_{\pm 0.62}$ / 3	$2.47_{\pm 0.70}$ / 3	$2.63_{\pm 0.68}$ / 3
	Mixed, w/ memory	$2.74_{\pm 0.65}$ / 3	$3.74_{\pm 0.99}$ / 4	$4.47_{\pm 0.84}$ / 5
	Mixed, w/o memory	$2.79_{\pm 0.79}$ / 3	$2.84_{\pm 0.96}$ / 3	$2.74_{\pm 0.73}$ / 3
Preference Memory	Coding, w/ memory	$1.37_{\pm 1.01}$ / 1	$3.95_{\pm 0.97}$ / 4	$3.63_{\pm 1.16}$ / 4
	Coding, w/o memory	$1.16_{\pm 0.69}$ / 1	$1.32_{\pm 0.58}$ / 1	$1.32_{\pm 0.58}$ / 1
	Mixed, w/ memory	$1.21_{\pm 0.63}$ / 1	$3.32_{\pm 1.49}$ / 3	$3.95_{\pm 1.51}$ / 5
	Mixed, w/o memory	$1.11_{\pm 0.46}$ / 1	$2.00_{\pm 1.33}$ / 1	$1.16_{\pm 0.37}$ / 1
Confidence	Coding, w/ memory	$2.74_{\pm 1.10}$ / 3	$3.79_{\pm 0.71}$ / 4	$3.74_{\pm 1.19}$ / 4
	Coding, w/o memory	$3.05_{\pm 1.13}$ / 3	$1.79_{\pm 0.85}$ / 2	$1.53_{\pm 0.77}$ / 1
	Mixed, w/ memory	$3.00_{\pm 1.00}$ / 3	$3.37_{\pm 1.30}$ / 4	$4.21_{\pm 1.08}$ / 5
	Mixed, w/o memory	$2.74_{\pm 0.93}$ / 3	$2.11_{\pm 1.10}$ / 2	$1.37_{\pm 0.60}$ / 1
Overall Satisfaction	Coding, w/ memory	3.42 ± 0.96 / 3	$4.11_{\pm 0.81}$ / 4	$3.89_{\pm 1.05}$ / 4
	Coding, w/o memory	$3.79_{\pm 1.08}$ / 4	$2.74_{\pm 1.05}$ / 3	$2.63_{\pm 0.96}$ / 3
	Mixed, w/ memory	$3.63_{\pm 0.90}$ / 4	$3.79_{\pm 1.03}$ / 4	$4.26_{\pm 0.99}$ / 5
	Mixed, w/o memory	$3.37_{\pm 1.01}$ / 3	$3.21_{\pm 1.08}$ / 3	$2.53_{\pm 1.02}$ / 3

Table 5: Quantitative results from our user study on multi-session collaboration. We report the mean$_{\pm\text{standard deviation}}$ / median for conversation length and four survey metrics rated on a 5-point Likert scale.



(a) Coding domain

(b) Mixed domain

Figure 10: Performance across sessions for our user study. Each graph plots the average delta between agents with memory and agents without memory across 3 sessions for Conversation Length (Δ_i^{CL}), Preference Adherence (Δ_i^{PA}), Preference Memory (Δ_i^{PM}), Confidence (Δ_i^{C}), and Overall Satisfaction (Δ_i^{O}).