EVOLVEMAS: A UNIFIED SELF-EVOLUTION FRAMEWORK FOR DECENTRALIZED MULTI-AGENT SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Decentralized multi-agent systems (MAS) enable large language model agents to interact without centralized control. Agents need to adapt to other agents at test time and reuse accumulated experience to refine their interaction strategies for future interactions across cooperative, mixed-motive, and adversarial settings. However, existing approaches often develop adaptation mechanisms for particular interaction settings. This shared requirement raises a central research question: how can a decentralized MAS framework support both test-time adaptation and reliable experience reuse across various interaction settings? To this end, we first propose EvolveMAS, a unified self-evolution framework that supports experience-driven adaptation within and across episodes. EvolveMAS pairs each main agent with a dedicated MemoryAgent for test-time memory governance, which maintains isolated Structured Memory outside the main agent's reasoning context. EvolveMAS organizes adaptation into two complementary processes: Within-Episode Online Adaptation and Cross-Episode Knowledge Evolution. During Within-Episode Online Adaptation, MemoryAgent updates episodic, semantic, and working memory from interaction evidence to inform subsequent actions. During Cross-Episode Knowledge Evolution, the main agent uses memory and episode outcomes to distill reusable interaction knowledge comprising partner profiles and interaction strategies for subsequent episodes. Across five benchmarks and six evaluation configurations spanning the three interaction settings, EvolveMAS achieves the highest normalized average score of 77.2 among the compared methods. Ablations show that test-time memory governance and cross-episode knowledge reuse both improve performance. Interaction traces further illustrate how updated memory and distilled knowledge guide subsequent interactions. Overall, the experimental results support a unified self-evolution framework for experience-driven adaptation in a decentralized MAS environment without updating model parameters. To facilitate future research, our code is available at `https://anonymous.4open.science/r/artifact_7f3a/`.

1 INTRODUCTION

Large language model (LLM) agents are increasingly deployed to assist users with personal and professional tasks (Staufer et al., 2026). Connecting agents deployed by different users creates opportunities to integrate their capabilities and support collaboration across users (Chen et al., 2025a). Decentralized multi-agent systems (MAS) support such collaboration by allowing agents to act on local information and communicate with peers without a central controller (Yang et al., 2025). From a game-theoretic perspective, multi-agent interactions are commonly categorized as cooperative, mixed-motive, or adversarial according to how agents' objectives align (Buşoniu et al., 2008; Zhang et al., 2021). Cooperative settings involve shared goals, mixed-motive settings combine shared and conflicting interests, and adversarial settings involve opposing goals, as shown in Figure 1.

When interacting with unfamiliar partners, people usually learn through interaction, using that experience to adjust immediate responses and guide future interactions. Agents similarly need to learn online from interaction experience to build profiles of other agents and refine their interaction strategies for subsequent interactions (He et al., 2016; Albrecht & Stone, 2017). This raises a central research question: *how can a unified mechanism support test-time adaptation across cooperative, mixed-motive, and adversarial settings?* Experience-driven self-evolution methods provides an in-

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

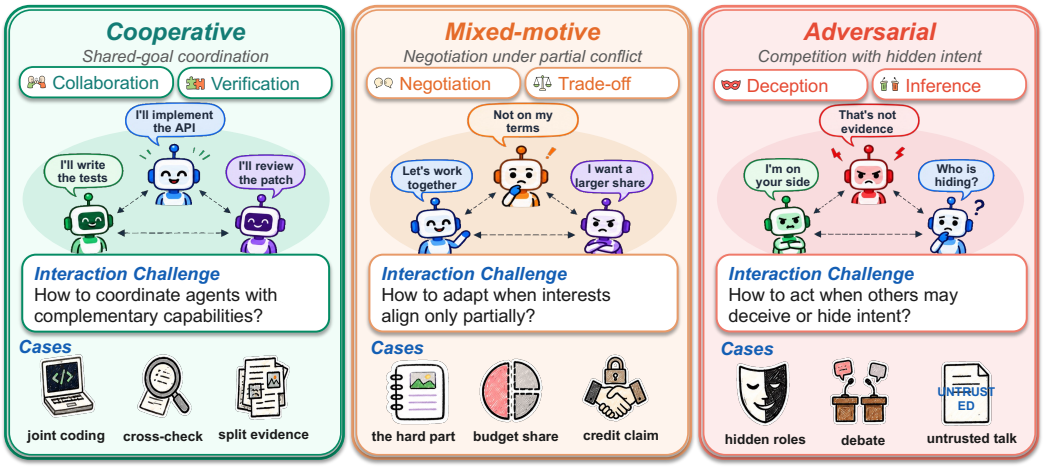


Figure 1: Three representative interaction regimes for decentralized MAS.

sight to such adaptation by using feedback and accumulated experience to improve behavior without heavy parameter updates (Wang et al., 2025b; Tang et al., 2025; Ouyang et al., 2026). However, recent research shows that accumulating long interaction histories in the reasoning context creates context bloat that can affect task performance (Du et al., 2025). Retaining inaccurate accumulated experience introduces memory pollution that can misguide subsequent interactions (Xiong et al., 2026). These two limitations further raise the question: *how can self-evolution decentralized agents achieve reliable experience reuse across three interaction settings?*

To address these questions, we propose **EvolveMAS**, a unified self-evolution framework that enables decentralized agents to learn from interaction experience across cooperative, mixed-motive, and adversarial settings. EvolveMAS pairs each main agent with a dedicated MemoryAgent for test-time memory governance to address context bloat and memory pollution. The MemoryAgent maintains Structured Memory outside the main agent's reasoning context, organizing interaction evidence into Episodic, Semantic, and Working Memory. During Within-Episode Online Adaptation phase, the MemoryAgent updates Structured Memory as interactions unfold to inform subsequent actions. During Cross-Episode Knowledge Evolution phase, the main agent uses interaction outcomes to refine reusable Knowledge comprising Partner Profiles and Interaction Strategies for future better interactions.

We evaluate EvolveMAS on five benchmarks spanning cooperative, mixed-motive, and adversarial settings. Under a shared self-evolution protocol, EvolveMAS achieves the highest normalized average score of 77.2, exceeding the strongest setting-specific baseline by 9.7 points. Ablation studies demonstrate that test-time memory governance improves interaction performance, with additional improvements from cross-episode Knowledge reuse. Additional analyses show that online memory updates improve question-answering accuracy over offline updates and that test-time memory governance reduces active-context growth.

Our contributions can be summarized as follows:

- We propose EvolveMAS, a unified self-evolution framework that supports experience-driven adaptation in decentralized MAS across cooperative, mixed-motive, and adversarial settings through Within-Episode Online Adaptation and Cross-Episode Knowledge Evolution.
- We introduce a dedicated MemoryAgent for test-time memory governance that maintains Structured Memory to support the main agent's reasoning during interaction and cross-episode Knowledge reuse through Partner Profiles and Interaction Strategies.
- We evaluate EvolveMAS on five benchmarks across the three interaction settings and demonstrate the effectiveness of test-time memory governance and cross-episode Knowledge reuse through ablation studies.

2 Related Work

Decentralized Multi-Agent Systems. Research on decentralized multi-agent systems has explored coordination and adaptation across diverse interaction regimes. Game-theoretic studies commonly classify multi-agent interactions as cooperative, mixed-motive, or adversarial (Yang & Wang, 2020; Wong et al., 2023), while recent LLM-based MAS instantiate these regimes through cooperative coding and reasoning (Qian et al., 2024; Du et al., 2024; Liang et al., 2024), mixed-motive social interaction (Zhou et al., 2024; Wang et al., 2024b), and adversarial strategic reasoning (Jin et al., 2024; Agarwal et al., 2025; Jiang et al., 2026). To support adaptation during interaction, classical opponent modeling infers other agents' behavioral patterns from interaction history to guide decision-making (Carmel & Markovitch, 1996; He et al., 2016). More broadly, recent MAS adaptation methods adjust system topology, workflows, roles, communication structures, or agent policies (Yang et al., 2025; Hu et al., 2025b;a; Zhang et al., 2025; Zhuge et al., 2024; Qian et al., 2025; Chen et al., 2024a; Wang et al., 2025a; Chen et al., 2025b; Liao et al., 2025; Liu et al., 2025; Xue et al., 2026; Park et al., 2025). These approaches improve performance on specific coordination or adaptation objectives and are typically tailored to particular interaction settings, system structures, or training procedures. In contrast, EvolveMAS maintains the same decentralized system organization and evolution framework across these interaction regimes.

Self-Evolution. Recent work explores test-time self-evolution by reusing interaction experience and feedback without updating model parameters. Reflexion (Shinn et al., 2023), ExpeL (Zhao et al., 2024), and Self-Refine (Madaan et al., 2023) demonstrate experience-driven improvements on single-agent tasks, while EvoMAC (Hu et al., 2025b) and AgentNet (Yang et al., 2025) extend iterative adaptation to multi-agent systems. Related studies also investigate self-generated data and self-play as mechanisms for agent evolution (Zelikman et al., 2022; Wang et al., 2023; Chen et al., 2024b; Huang et al., 2026; Zhao et al., 2025; Chen et al., 2025c). However, existing self-evolution methods are primarily designed for specific tasks or interaction settings. To support self-evolution across interaction settings, EvolveMAS couples within-episode adaptation with cross-episode Knowledge evolution through a shared test-time evolution mechanism for decentralized MAS.

Agent Memory. Agent memory enables experience to persist beyond a single interaction context (Park et al., 2023; Sumers et al., 2024). MemGPT (Packer et al., 2023) and MemoryBank (Zhong et al., 2024) introduce persistent memory, while Voyager (Wang et al., 2024a), Agent Workflow Memory (Wang et al., 2025b), and SkillWeaver (Zheng et al., 2025) accumulate reusable skills and workflows. More recent work explores structured memory organization and test-time memory reasoning (Xu et al., 2025; Chhikara et al., 2025; Ouyang et al., 2026; Zhou et al., 2025; Yan et al., 2025). However, in long-horizon decentralized interactions, retaining accumulated memory in the main agent's context can lead to context expansion, attention dilution, and memory pollution (Liang et al., 2026; Zhang et al., 2026; Wu et al., 2026; Wan et al., 2026). To address these challenges, EvolveMAS pairs each main agent with a dedicated external MemoryAgent. This MemoryAgent maintains an isolated Structured Memory and provides a governed interaction state to support both adaptation timescales.

3 EvolveMAS: Memory-Governed Self-Evolution

We propose EvolveMAS, a decentralized test-time self-evolution framework for heterogeneous multi-agent interactions. The framework combines per-agent external memory governance with a shared Execute–Evaluate–Evolve (EEE) process. Together, these components support within-episode adaptation and cross-episode Knowledge evolution without updating model parameters.

3.1 Framework Overview

Consider a decentralized multi-agent system (DMAS) with main agents $\mathcal{A} = \{a_i\}_{i=1}^{n}$ operating over a continuous stream of heterogeneous interactions organized into episodes. In episode e, the agents receive a task description x_e and interact autonomously over T_e turns. Each agent selects its actions based on local observations, while the task determines which agents interact and when. The resulting interaction record forms the trajectory $\tau_e = (\tau_{e,1}, \ldots, \tau_{e,T_e})$, where $\tau_{e,t}$ contains the

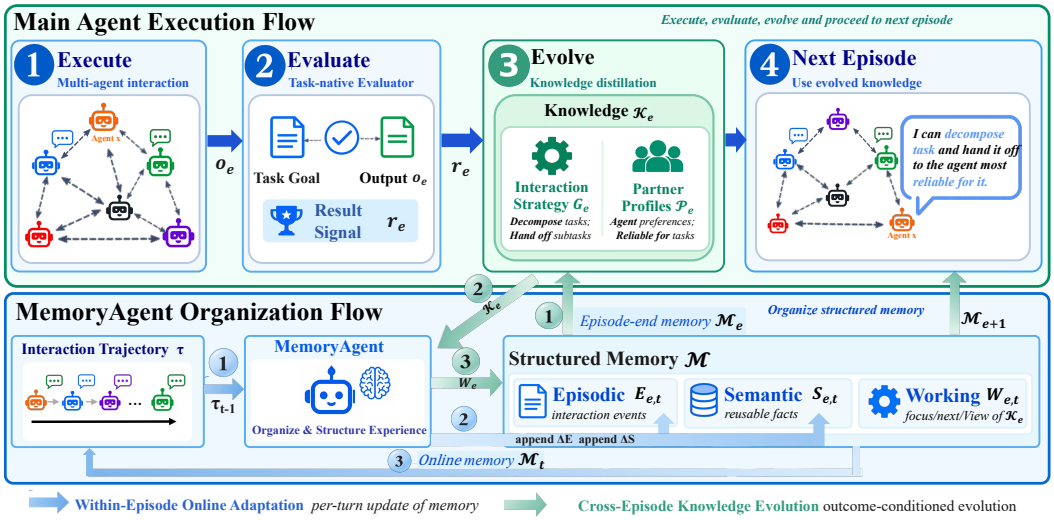


Figure 2: Overview of EvolveMAS. The upper lane is the main agent's Execute–Evaluate–Evolve cycle and the lower lane the MemoryAgent that maintains Structured Memory. Blue arrows are within-episode updates, green arrows cross-episode evolution.

messages and environment feedback produced at turn t. Each agent observes its own local view of τ_e, and the episode concludes with an interaction outcome o_e.

EvolveMAS preserves this decentralized organization (Figure 2). Each main agent a_i is paired one-to-one with a dedicated MemoryAgent m_i and an isolated memory, allowing accumulated experience to inform the main agent's subsequent behavior. The main agent handles task reasoning, communication, and environment interaction. The corresponding MemoryAgent maintains Structured Memory and Interaction Knowledge outside the main agent's reasoning context, drawing on interaction evidence observed by that main agent. All MemoryAgents share the same memory schema and governance protocol, while their contents evolve independently based on their respective main agents' local interaction histories.

Based on the interaction record, EvolveMAS represents experience at three levels of abstraction. The *Trajectory* τ_e consists of raw interaction evidence generated during task execution. Building on this evidence, EvolveMAS maintains *Structured Memory* $M_{e,t}$ as a compact state updated during interaction. At a higher level, it maintains *Interaction Knowledge* K_e to capture reusable regularities across episodes. For clarity, the subsequent formulation adopts the perspective of a focal main agent $a = a_i$, with the agent index omitted from its memory and Interaction Knowledge states.

Let $M_e \equiv M_{e,T_e}$ denote the episode-end Structured Memory. Here, o_e denotes the observed interaction outcome, and r_e denotes the corresponding task-native result signal. The EEE process is then formulated at the episode level as

$$\begin{aligned} \textsc{Execute}: &\quad (\tau_e, M_e, o_e) = \mathrm{Exec}\big(x_e, \mathcal{A}, M_{e,0}, K_{e-1}\big), \\ \textsc{Evaluate}: &\quad r_e = \mathrm{Eval}\big(x_e, \tau_e, o_e\big), \\ \textsc{Evolve}: &\quad K_e = D_a\big(M_e, r_e; K_{e-1}\big). \end{aligned} \tag{1}$$

Here, Eval denotes the task-native outcome evaluator. Accordingly, r_e may vary in form across interaction settings while providing a common interface for outcome-conditioned evolution. The operator D_a distills Interaction Knowledge from governed Structured Memory, the observed result, and previously accumulated Interaction Knowledge. The resulting K_e comprises reusable Interaction Strategies and Partner Profiles. Both EXECUTE and EVOLVE use the previous episode's Knowledge K_{e-1}. Specifically, EXECUTE uses it to guide interaction, whereas EVOLVE incorporates new experience to refine it. In contrast, EVALUATE computes r_e from x_e, τ_e, and o_e, without directly accessing accumulated memory or Knowledge. The following subsections detail the memory and Knowledge update mechanisms.

3.2 WITHIN-EPISODE ONLINE ADAPTATION

During EXECUTE in Eq. 1, main agents interact with peers and the environment. As interaction proceeds, their MemoryAgents continually update Structured Memory based on newly generated interaction evidence. At turn t, action generation and memory updating are formulated as

$$\begin{aligned}(y_{e,t}, \Delta\tau_{e,t}) &\sim \pi_a\left(\cdot \mid [x_e; M_{e,t-1}; K_{e-1}]\right), \\ M_{e,t} &= \mathrm{MA}_{\mathrm{online}}\left(M_{e,t-1}, \Delta\tau_{e,t}\right),\end{aligned} \tag{2}$$

where $y_{e,t}$ denotes the generated response or environment action, and $\Delta\tau_{e,t}$ denotes the newly observed interaction evidence at turn t.

Structured Memory comprises three complementary regions,

$$M_{e,t} = (E_{e,t}, S_{e,t}, W_{e,t}), \tag{3}$$

where $E_{e,t}$, $S_{e,t}$, and $W_{e,t}$ denote Episodic, Semantic, and Working Memory (Tulving, 1972; Baddeley & Hitch, 1974).

Episodic Memory. $E_{e,t}$ records concrete interaction experiences, including the events that occurred, the agents or sources involved, and the outcomes that followed. Together, these records provide an interaction-grounded history from which regularities can subsequently be inferred.

Semantic Memory. $S_{e,t}$ maintains reusable facts and interaction regularities inferred from experience. Depending on the interaction, these may include task-relevant facts, recurring coordination patterns, observations about interaction partners, or other stable information that supports subsequent reasoning.

Working Memory. $W_{e,t}$ maintains a compact state relevant to the ongoing interaction. This state includes the current focus, next-step information, and task-relevant views of accumulated memory and Interaction Knowledge. As the interaction context changes, Working Memory is refreshed accordingly.

At the representation level, new events are appended to Episodic Memory, reusable regularities are incorporated into Semantic Memory, and Working Memory is refreshed. These conceptual update rules do not prescribe a specific storage implementation.

The updates in Eq. 2 form a recurrent feedback loop within EXECUTE. Evidence observed at turn t updates Structured Memory, which conditions the acting policy π_a from turn $t+1$ onward. This loop enables Within-Episode Online Adaptation. Sec. 3.3 formalizes both adaptation timescales in Eq. 5.

The term *within-episode* specifies the timescale of this feedback loop. Episodic and Semantic information may persist and accumulate across episodes, whereas Working Memory is refreshed for the ongoing interaction.

3.3 CROSS-EPISODE KNOWLEDGE EVOLUTION

After episode e concludes, EVALUATE produces a task-native result signal r_e from the observed interaction outcome. This signal serves as a protocol-level interface for conditioning subsequent evolution without requiring a shared metric space across heterogeneous interactions. Conditioned on r_e, the main agent distills reusable interaction regularities from the governed Structured Memory M_e and previously accumulated Interaction Knowledge K_{e-1}. The evolved Interaction Knowledge comprises two complementary components,

$$K_e = (G_e, P_e), \tag{4}$$

which are described below.

Interaction Strategies. G_e captures reusable regularities that guide behavioral adaptation across recurring interaction situations, including coordination, verification, negotiation, and trust calibration.

Table 1: Cross-setting results. WW-V and WW-W report village win and wolf deception success in one-sided Werewolf audits. Shading identifies source settings. Bold and underlined values denote the best and second-best results in each column, respectively, including ties. Avg. is the mean of six percentage-scaled metrics: SOTOPIA is multiplied by 100 and Deliberation by $100/6$.

	Cooperative		Mixed-motive		Adversarial		Avg.
Method	BBH	HumanEval	SOTOPIA	Deliberation	WW-V	WW-W	
AgentNet	79	<u>95.7</u>	0.554	2.60	51.0	45.0	61.6
ChatDev	73	<u>95.7</u>	0.509	2.06	53.0	49.0	59.3
SOTOPIA	72	<u>95.7</u>	**0.618**	3.14	55.0	<u>57.0</u>	65.6
LLM-Deliberation	74	93.9	0.450	<u>4.92</u>	**64.0**	46.0	67.5
Werewolf/DPIns	72	92.7	0.588	3.41	55.0	44.0	63.2
Self-Memory	<u>82</u>	<u>95.7</u>	0.576	4.90	58.0	42.0	<u>69.5</u>
EvolveMAS	**83**	**97.0**	<u>0.613</u>	**5.40**	<u>62.0</u>	**70.0**	**77.2**

Partner Profiles. $P_e = \{p_{e,j}\}_{j \neq i}$ captures partner-specific regularities, including recurring behavioral tendencies, inferred capabilities, and reliability, and is updated as new interaction evidence accumulates.

In subsequent episodes, the evolved Interaction Knowledge conditions EXECUTE together with the current Structured Memory. Combined with the online memory updates in Sec. 3.2, EvolveMAS therefore operates across two coupled adaptation timescales:

$$\underbrace{\Delta\tau_{e,t} \rightarrow M_{e,t} \rightarrow \textsc{Execute}_{t+1}}_{\text{Within-Episode Online Adaptation}} \qquad \underbrace{(M_e, r_e, K_{e-1}) \rightarrow K_e \rightarrow \textsc{Execute}_{\text{future}}}_{\text{Cross-Episode Knowledge Evolution}}. \tag{5}$$

Together, within-episode evidence integration and cross-episode knowledge reuse form a unified self-evolution framework for heterogeneous interactions without updating model parameters.

4 EXPERIMENTS

4.1 EXPERIMENT SETTING

We evaluate EvolveMAS on five representative benchmarks spanning three interaction settings. The cooperative setting includes BBH (Suzgun et al., 2023) and HumanEval (Chen et al., 2021), covering multi-agent reasoning and collaborative coding under shared goals. The mixed-motive setting includes LLM-Deliberation (Abdelnabi et al., 2024) and SOTOPIA (Zhou et al., 2024), covering structured negotiation and open-ended social interaction with partially aligned objectives. The adversarial setting uses One Night Ultimate Werewolf (Jin et al., 2024), a hidden-role game involving deception, trust calibration, and strategic reasoning.

The main experiments use heterogeneous teams spanning model families and capability tiers to support Partner Profile modeling. Team size and composition follow each benchmark's native protocol and difficulty. Each main agent is paired with an isolated GPT-4o-mini MemoryAgent. We evaluate five source method families under six configurations, including benchmark-native and cross-setting comparisons. Appendices A.1 and B detail team composition, transfer construction, and additional analyses. *Self-Memory* retains the same interaction structure but manages memory within the main agent's task-reasoning context.

4.2 MAIN RESULTS ACROSS INTERACTION SETTINGS

Table 1 compares EvolveMAS with setting-specific baselines transferred across interaction settings and with Self-Memory. Under a shared self-evolution protocol, EvolveMAS ranks first in four evaluation configurations and second in the remaining two. Its normalized average score of 77.2 exceeds the strongest setting-specific baseline by 9.7 points and Self-Memory by 7.7 points.

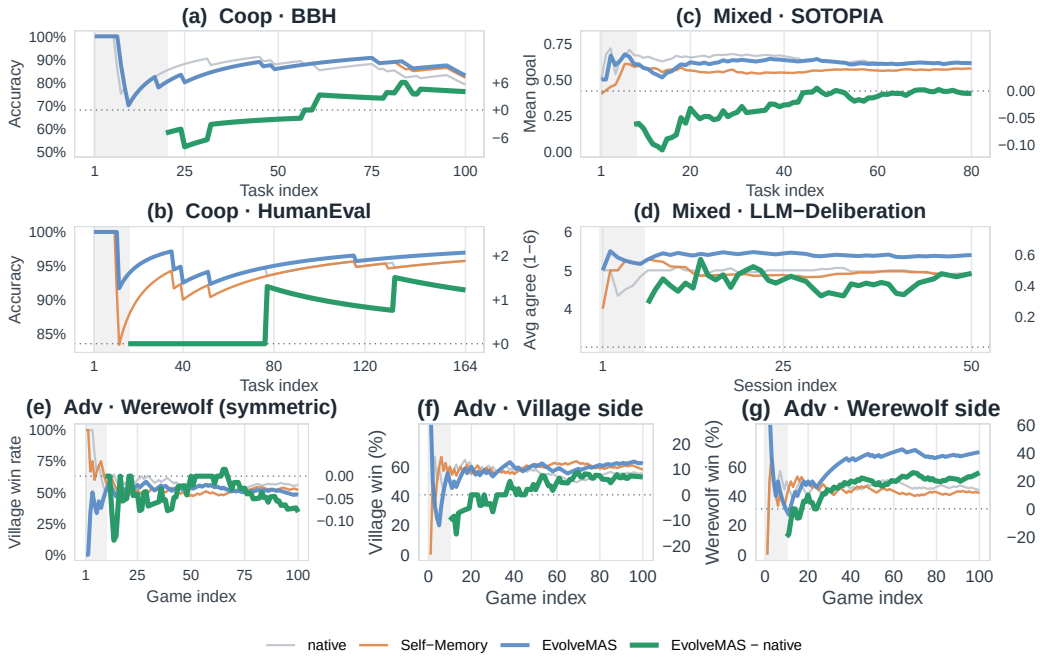


Figure 3: Within-session cumulative evolution. The cooperative (a–b) and mixed-motive (c–d) blocks are stacked side by side, and the adversarial row reports symmetric Werewolf (e) with the one-sided ablations, scored as village win (f) and as wolf deception success (g). Lines are cumulative means; the thicker green line is the paired difference (EvolveMAS − native), on the right-hand axis, with its zero dotted and its scale set independently in each panel. Shading marks each panel's early, unstable tasks or games.

Cooperative setting. EvolveMAS achieves 83.0 on BBH, exceeding the native AgentNet baseline of 79.0 by four percentage points. On HumanEval, it achieves 97.0, compared with 95.7 for both ChatDev and Self-Memory. Appendix Table 5 further examines performance under an alternative BBH model configuration.

Mixed-motive setting. On LLM-Deliberation, EvolveMAS achieves the highest average agreement of 5.40 parties, compared with 4.92 for the native baseline. On SOTOPIA, it ranks second at 0.613, close to the native baseline of 0.618 and 0.037 above Self-Memory. Both settings provide partner-specific and negotiation regularities that can be reused through cross-episode evolution.

Adversarial setting. Because adversarial roles have conflicting objectives, the one-sided audits apply EvolveMAS to one side at a time. On WW-V, EvolveMAS scores 62.0, compared with the highest score of 64.0 achieved by LLM-Deliberation. On WW-W, it achieves the highest score of 70.0, exceeding the best competing method by 13 points. Appendix E.4 reports both one-sided conditions and the two symmetric references on a common village-win scale.

Detailed metrics and per-benchmark breakdowns for Table 1 are provided in Appendices A and E.

4.3 Cross-Setting Evolution

We examine how performance evolves as interaction experience accumulates. Figure 3 presents cumulative performance over tasks, sessions, or games, grouped by interaction setting. The trajectories compare EvolveMAS with the native baseline and Self-Memory to assess whether accumulated experience continues to influence subsequent behavior under the same evolution protocol.

In the cooperative setting, the relative performance of EvolveMAS on BBH improves from an initial deficit to a sustained advantage over the later tasks, ending at $+4$ percentage points. On HumanEval, performance remains comparable to the native baseline for roughly the first half of the task stream,

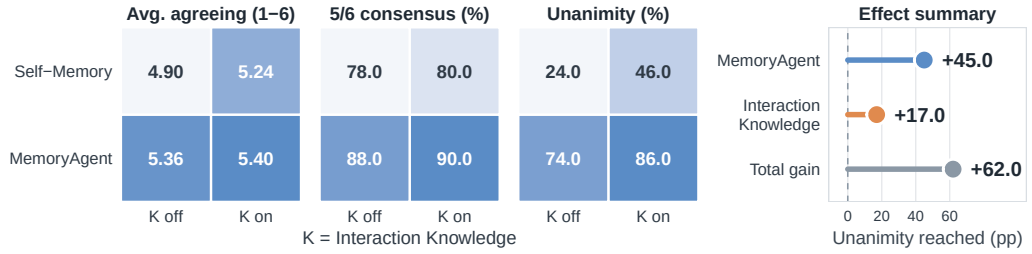


Figure 4: LLM-Deliberation ablation. Rows vary memory governance; columns vary the availability of cross-episode Knowledge $K_e = (G_e, P_e)$. Each cell reports one run of 50 sessions; color scales are panel-specific. The effect summary reports average marginal effects on unanimity and the total gain.

followed by a modest advantage that ends at approximately $+1.2$ percentage points. In the mixed-motive setting, SOTOPIA shows an overall upward trend in relative performance, with the early deficit narrowing toward parity despite intermediate fluctuations. On LLM-Deliberation, the cumulative advantage is positive throughout the displayed interval and persists through the final session. This pattern is consistent with adaptation taking effect early and providing sustained benefits in subsequent interactions. In the adversarial setting, symmetric Werewolf tracks village-side performance as both opposing sides evolve. The one-sided evaluations show positive advantages that persist over the later games, with larger gains on the wolf side.

The trajectories in Figure 3 indicate that accumulated interaction experience supports continual evolution across cooperative, mixed-motive, and adversarial settings under a shared EvolveMAS protocol. Although trajectory shapes vary with the interaction structure, the self-evolution process, memory representation, and update schedule remain unchanged.

4.4 Ablating the Two Evolution Mechanisms

To distinguish the contributions of the two mechanisms, the evaluation requires both emerging information within sessions and recurring partners across sessions. These conditions support within-episode adaptation and cross-episode Knowledge reuse, respectively. LLM-Deliberation meets both conditions: participants reveal their positions over successive rounds, and the same six parties participate in every session. Its graded outcomes also allow the contribution of each mechanism to be measured. We report average agreement, measured as the mean number of parties accepting the final proposal, and 5/6 consensus on that proposal. The unanimity rate measures the fraction of sessions in which at least one lead-party proposal is accepted by all six parties.

We independently vary memory governance (Self-Memory vs. external MemoryAgent) and the availability of cross-episode Interaction Knowledge ($K = \varnothing$ vs. $K_e = (G_e, P_e)$). This yields the 2×2 factorial design in Figure 4. The lower-right condition in each metric panel combines external memory governance with cross-episode Interaction Knowledge and corresponds to full EvolveMAS.

Unanimity rates differ across all four conditions. Replacing Self-Memory with an external MemoryAgent increases unanimity from 24.0% to 74.0% without cross-episode Knowledge and from 46.0% to 86.0% with it. These gains are consistent with updated interaction states helping agents respond to positions revealed during deliberation. Adding K_e further increases unanimity from 74.0% to 86.0% with an external MemoryAgent and from 24.0% to 46.0% with Self-Memory. These improvements support the value of reusing negotiation regularities across episodes. Average agreement follows the same ordering (4.90, 5.24, 5.36, 5.40). Appendix A.2 reports additional metrics, which follow this ordering except for final-proposal unanimity. Appendix E.2 extends the ablation to three additional benchmarks.

4.5 Case Study

Figure 5 illustrates the two adaptation processes in Eq. 5 through an instrumented EvolveMAS trajectory in SOTOPIA negotiation.

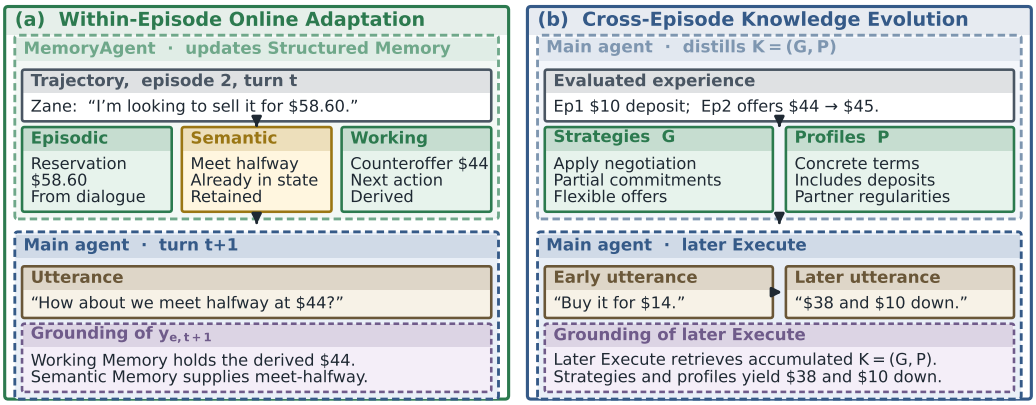


Figure 5: Instrumented SOTOPIA example illustrating Within-Episode Online Adaptation (a) and Cross-Episode Knowledge Evolution (b). In (a), entries written at the current turn are green and carried-over entries are yellow.

Within-Episode Online Adaptation. In episode 2, the seller states an asking price of $58.60 (Figure 5a). The MemoryAgent incorporates this evidence into Structured Memory while retaining the existing meet-halfway regularity. The resulting Working Memory contains a $44 counteroffer, which the buyer uses on the next turn. This sequence illustrates how newly observed evidence informs subsequent decisions within the same episode.

Cross-Episode Knowledge Evolution. Outcome-conditioned distillation produces reusable Interaction Strategies and Partner Profiles across episodes (Figure 5b). Experiences involving a $10 deposit and a counteroffer revised from $44 to $45 yield negotiation regularities. The Partner Profile captures the counterpart's responsiveness to concrete terms. The resulting Interaction Knowledge guides subsequent offer formulation during EXECUTE.

Appendix G provides further examples across all three interaction settings.

4.6 ADDITIONAL ANALYSES

System-level analyses show stable performance across the evaluated team sizes and favorable performance relative to static aggregation. Further diagnostics show stable short-horizon performance and lower per-episode cost and acting-context occupancy than retaining raw interaction histories as experience accumulates (Appendix C). Analyses of Knowledge accumulation indicate that stronger acting models distill more transferable Interaction Knowledge (Appendix D). Finally, a long-horizon memory ablation shows higher overall question-answering accuracy with online MemoryAgent updates than with offline updates or self-managed memory (Appendix F).

5 CONCLUSION

We presented EvolveMAS, a unified self-evolution framework for decentralized multi-agent systems (DMAS). Evaluations on five benchmarks demonstrate its applicability to cooperative, mixed-motive, and adversarial settings, while ablations highlight the value of test-time memory governance and cross-episode Knowledge reuse. Together, these results support experience-driven adaptation across the three interaction settings under a shared self-evolution framework without model parameter updates. Designed as a plug-in extension to existing agents, EvolveMAS supports decentralized interaction while preserving each agent's autonomy. This provides a basis for integrating the capabilities of agents deployed by different users without centralized control. Building on this paradigm, future work can investigate empirical scaling laws by varying agent capability, population size, and accumulated experience. Such studies would clarify the design trade-offs among DMAS reasoning performance, computational cost, and safety.

AI USE STATEMENT

We used generative AI tools to assist with code execution under experimental protocols specified by the authors. The research questions, framework design, experimental design, analysis methodology, and interpretation of results were determined by the authors. Generative AI tools were not used to propose or refine hypotheses, develop theoretical models or conceptual frameworks, implement the EvolveMAS framework itself, generate synthetic datasets, or clean and reformat datasets. Tasks involving the formulation or proof of mathematical claims are not applicable to this work.

Additionally, we used generative AI tools to edit author-written text for clarity and readability, search for and identify relevant literature, and format references.

The authors reviewed all AI-assisted work, including code execution and the resulting outputs. Manuscript edits and translations were checked against the authors' intended meaning, and cited references were verified against authoritative records. The authors take responsibility for the final manuscript and all reported claims, results, and artifacts, including those produced with assistance from generative AI.

ETHICS STATEMENT

This work studies test-time self-evolution among large language model agents on public benchmarks. It does not involve human subjects, personal data, or the release of new datasets. The adversarial Werewolf evaluation uses simulated hidden-role interaction among model agents. We report both village-side and wolf-side outcomes so that deception capability is not presented in isolation. All authors have read and adhere to the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

We report the full cross-setting comparison in Table 1 and the transfer instantiation of each source method family in Appendix A.1. Benchmarks and team composition are specified in Section 4.1. MAS-level properties, model-wise analyses, within-setting results, and the long-horizon memory ablation appear in Appendices C, D, E, and F. The anonymous repository at `https://anonymous.4open.science/r/artifact_7f3a/` holds the framework and the per-benchmark code, together with a README and a dependency list. The scripts that generate the reported figures ship with the paper source rather than with that repository.

REFERENCES

Sahar Abdelnabi, Amr Gomaa, Sarath Sivaprasad, Lea Schönherr, and Mario Fritz. Cooperation, competition, and maliciousness: LLM-stakeholders interactive negotiation. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024), Datasets and Benchmarks Track*, 2024. Released as the LLM-Deliberation benchmark; arXiv:2309.17234.

Mrinal Agarwal, Saad Rana, Theo Sundoro, Hermela Berhe, Spencer Kim, Vasu Sharma, Sean O'Brien, and Kevin Zhu. WOLF: Werewolf-based observations for LLM deception and falsehoods. In *Multi-Turn Interactions in Large Language Models Workshop at NeurIPS*, 2025. arXiv:2512.09187.

Stefano V. Albrecht and Peter Stone. Reasoning about hypothetical agent behaviours and their parameters. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems*, pp. 547–555. International Foundation for Autonomous Agents and Multiagent Systems, 2017. URL https://www.ifaamas.org/AAMAS/aamas2017/proceedings/pdfs/p547.pdf.

Alan D. Baddeley and Graham Hitch. Working memory. *Psychology of Learning and Motivation*, 8:47–89, 1974.

Lucian Buşoniu, Robert Babuška, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008. doi: 10.1109/TSMCC.2007.913919. URL https://doi.org/10.1109/TSMCC.2007.913919.

David Carmel and Shaul Markovitch. Opponent modeling in multi-agent systems. In Gerhard Weiß and Sandip Sen (eds.), *Adaption and Learning in Multi-Agent Systems*, volume 1042, pp. 40–52. Springer, 1996.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. AgentVerse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *International Conference on Learning Representations*, 2024a.

Weize Chen, Ziming You, Ran Li, Yitong Guan, Chen Qian, Chenyang Zhao, Cheng Yang, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. Internet of agents: Weaving a web of heterogeneous agents for collaborative intelligence. In *International Conference on Learning Representations*, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/59c27bf8d56d3d50c7aeaf7535dee975-Abstract-Conference.html.

Weize Chen, Jiarui Yuan, Chen Qian, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Optima: Optimizing effectiveness and efficiency for LLM-based multi-agent system. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 11534–11557, 2025b.

Yixing Chen, Yiding Wang, Siqi Zhu, Haofei Yu, Tao Feng, Muhan Zhang, Mostofa Patwary, and Jiaxuan You. Multi-agent evolve: LLM self-improve through co-evolution. *arXiv preprint arXiv:2510.23595*, 2025c.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. In *Proceedings of ICML*, 2024b. arXiv:2401.01335.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready AI agents with scalable long-term memory. In *Proceedings of the 28th European Conference on Artificial Intelligence*, pp. 2993–3000, 2025. doi: 10.3233/FAIA251160.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, PMLR, 2024.

Yufeng Du, Minyang Tian, Srikanth Ronanki, Subendhu Rongali, Sravan Babu Bodapati, Aram Galstyan, Azton Wells, Roy Schwartz, Eliu A Huerta, and Hao Peng. Context length alone hurts LLM performance despite perfect retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 23281–23298. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.1264. URL https://aclanthology.org/2025.findings-emnlp.1264/.

He He, Jordan Boyd-Graber, Kevin Kwok, and Hal Daumé III. Opponent modeling in deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1804–1813. PMLR, 2016. URL https://proceedings.mlr.press/v48/he16.html.

Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. In *International Conference on Learning Representations*, 2025a.

Yue Hu, Yuzhu Cai, Yaxin Du, Xinyu Zhu, Xiangrui Liu, Zijie Yu, Yuchen Hou, Shuo Tang, and Siheng Chen. Self-evolving multi-agent collaboration networks for software development. In *International Conference on Learning Representations*, 2025b.

Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. R-Zero: Self-evolving reasoning LLM from zero data. In *International Conference on Learning Representations*, 2026. arXiv:2508.05004.

Bowen Jiang, Taiwei Shi, Ryo Kamoi, Yuan Yuan, Camillo J. Taylor, Longqi Yang, Pei Zhou, and Sihao Chen. One model, all roles: Multi-turn, multi-agent self-play reinforcement learning for conversational social intelligence. *arXiv preprint arXiv:2602.03109*, 2026.

Xuanfa Jin, Ziyan Wang, Yali Du, Meng Fang, Haifeng Zhang, and Jun Wang. Learning to discuss strategically: A case study on one night ultimate werewolf. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, 2024. arXiv:2405.19946.

Sirui Liang, Pengfei Cao, Jian Zhao, Wenhao Teng, Xiangwen Liao, Jun Zhao, and Kang Liu. Learning how to remember: A meta-cognitive management method for structured and transferable agent memory. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 30733–30753, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.1535.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 17889–17904, 2024.

Junwei Liao, Muning Wen, Jun Wang, and Weinan Zhang. MARFT: Multi-agent reinforcement fine-tuning. *arXiv preprint arXiv:2504.16129*, 2025.

Shuo Liu, Tianle Chen, Zeyu Liang, Xueguang Lyu, and Christopher Amato. LLM collaboration with multi-agent reinforcement learning. *arXiv preprint arXiv:2508.04652*, 2025.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, 2023.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, 2024. doi: 10.18653/v1/2024.acl-long.747.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long Le, Samira Daruki, Xiangru Tang, Vishy Tirumalashetty, George Lee, Mahsan Rofouei, Hangfei Lin, Jiawei Han, Chen-Yu Lee, and Tomas Pfister. ReasoningBank: Scaling agent self-evolving with reasoning memory. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=jL7fwchScm.

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

Chanwoo Park, Seungju Han, Xingzhi Guo, Asuman Ozdaglar, Kaiqing Zhang, and Joo-Kyung Kim. MAPoRL: Multi-agent post-co-training for collaborative large language models with reinforcement learning. In *Proceedings of ACL*, 2025. arXiv:2502.18439.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of UIST*, 2023.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. ChatDev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15174–15186. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.810.

Chen Qian, Zihao Xie, YiFei Wang, Wei Liu, Kunlun Zhu, Hanchen Xia, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Scaling large language model-based multi-agent collaboration. In *International Conference on Learning Representations*, 2025. arXiv:2406.07155.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-0377. URL https://papers.nips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Leon Staufer, Kevin Feng, Kevin Wei, Luke Bailey, Yawen Duan, Mick Yang, A. Pinar Ozisik, Stephen Casper, and Noam Kolt. The 2025 AI agent index: Documenting technical and safety features of deployed agentic AI systems. In *Proceedings of the 2026 ACM Conference on Fairness, Accountability, and Transparency*, pp. 1536–1576. ACM, 2026. doi: 10.1145/3805689.3806728. URL https://doi.org/10.1145/3805689.3806728.

Theodore R. Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L. Griffiths. Cognitive architectures for language agents. In *Transactions on Machine Learning Research*, 2024.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging BIG-Bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, 2023.

Xiangru Tang, Tianyu Hu, Muyang Ye, Daniel Shao, Xunjian Yin, Siru Ouyang, Wangchunshu Zhou, Pan Lu, Zhuosheng Zhang, Yilun Zhao, Arman Cohan, and Mark Gerstein. ChemAgent: Self-updating memories in large language models improves chemical reasoning. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/fa7f64b45970e6a7f8824781e7e01501-Abstract-Conference.html.

Endel Tulving. Episodic and semantic memory. *Organization of Memory*, 1972.

Guangya Wan, Mingyang Ling, Xiaoqi Ren, Rujun Han, Sheng Li, and Zizhao Zhang. COMPASS: Enhancing agent long-horizon reasoning with evolving context. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3360–3380, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.152.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024a. URL https://openreview.net/forum?id=ehfRiF0R3a.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. Mixture-of-agents enhances large language model capabilities. In *Proceedings of ICLR*, 2025a.

Ruiyi Wang, Haofei Yu, Wenxin Zhang, Zhengyang Qi, Maarten Sap, Graham Neubig, Yonatan Bisk, and Hao Zhu. SOTOPIA-π: Interactive learning of socially intelligent language agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12912–12940. Association for Computational Linguistics, 2024b.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of ACL*, 2023. arXiv:2212.10560.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 63897–63911. PMLR, 2025b. URL https://proceedings.mlr.press/v267/wang25bx.html.

Annie Wong, Thomas Bäck, Anna V. Kononova, and Aske Plaat. Deep multiagent reinforcement learning: Challenges and directions. *Artificial Intelligence Review*, 56(6):5023–5056, 2023. doi: 10.1007/s10462-022-10299-x.

Yifan Wu, Lizhu Zhang, Yuhang Zhou, Mingyi Wang, Bo Peng, Serena Li, Xiangjun Fan, and Zhuokai Zhao. Remember when it matters: Proactive memory agent for long-horizon agents. *arXiv preprint arXiv:2607.08716*, 2026. doi: 10.48550/arXiv.2607.08716.

Zidi Xiong, Yuping Lin, Wenya Xie, Pengfei He, Zirui Liu, Jiliang Tang, Himabindu Lakkaraju, and Zhen Xiang. How memory management impacts LLM agents: An empirical study of experience-following behavior. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 623–645. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.27. URL https://aclanthology.org/2026.acl-long.27/.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-Mem: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems*, 2025.

Xiangyuan Xue, Yifan Zhou, Guibin Zhang, Zaibin Zhang, Yijiang Li, Chen Zhang, Zhenfei Yin, Philip Torr, Wanli Ouyang, and Lei Bai. CoMAS: Co-evolving multi-agent systems via interaction rewards. In *International Conference on Learning Representations*, 2026. arXiv:2510.08529.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z. Pan, Hinrich Schütze, Volker Tresp, and Yunpu Ma. Memory-R1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. *arXiv preprint arXiv:2508.19828*, 2025.

Yaodong Yang and Jun Wang. An overview of multi-agent reinforcement learning from game theoretical perspective. *arXiv preprint arXiv:2011.00583v1*, 2020.

Yingxuan Yang, Huacan Chai, Shuai Shao, Yuanyi Song, Siyuan Qi, Renting Rui, and Weinan Zhang. AgentNet: Decentralized evolutionary coordination for LLM-based multi-agent systems. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-3578. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/9a379c1b05793d1c42dc832269834515-Abstract-Conference.html.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. In *Proceedings of NeurIPS*, 2022. arXiv:2203.14465.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. AFlow: Automating agentic workflow generation. In *International Conference on Learning Representations*, 2025.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. In Kyriakos G. Vamvoudakis, Yan Wan, Frank L. Lewis, and Derya Cansever (eds.), *Handbook of Reinforcement Learning and Control*, pp. 321–384. Springer, 2021. doi: 10.1007/978-3-030-60990-0_12. URL https://link.springer.com/chapter/10.1007/978-3-030-60990-0_12.

Yuxiang Zhang, Jiangming Shu, Ye Ma, Xueyuan Lin, Shangxi Wu, and Jitao Sang. Memory as action: Autonomous context curation for long-horizon agentic tasks. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 19149–19164, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.956.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024. doi: 10.1609/aaai.v38i17.29936. URL https://ojs.aaai.org/index.php/AAAI/article/view/29936.

Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data. In *Advances in Neural Information Processing Systems*, 2025. arXiv:2505.03335.

Boyuan Zheng, Michael Y. Fatemi, Xiaolong Jin, Zora Zhiruo Wang, Apurva Gandhi, Yueqi Song, Yu Gu, Jayanth Srinivasa, Gaowen Liu, Graham Neubig, and Yu Su. Skillweaver: Web agents can self-improve by discovering and honing skills. *arXiv preprint arXiv:2504.07079*, 2025.

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

Huichi Zhou, Yihang Chen, Siyuan Guo, Xue Yan, Kin Hei Lee, Zihan Wang, Ka Yiu Lee, Guchun Zhang, Kun Shao, Linyi Yang, and Jun Wang. Memento: Fine-tuning LLM agents without fine-tuning LLMs. *arXiv preprint arXiv:2508.16153*, 2025.

Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, and Maarten Sap. SOTOPIA: Interactive evaluation for social intelligence in language agents. In *International Conference on Learning Representations*, 2024.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. GPTSwarm: Language agents as optimizable graphs. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 62743–62767, 2024.

A EXPERIMENTAL PROTOCOLS

The main text reports the cross-setting results and their interpretation. This section instead documents how the transferred method families are instantiated and how the full Deliberation metric suite is computed.

A.1 TRANSFER CONSTRUCTION

To make the transfer rows in Table 1 reproducible and fair to the source methods, we fix one transfer setup. Each setting-native method family is instantiated from its original configuration, preserving native role structure, team composition, and social mode, and is then run through the target benchmark's native runner and evaluator. Each family therefore runs exactly the mechanism it was designed around, including the within-episode interaction context of its native protocol; cross-episode memory enters the comparison through the Self-Memory row on the same team. BBH and HumanEval runs force the social mode to cooperative because both are cooperative benchmarks, while SOTOPIA, LLM-Deliberation, and Werewolf retain the source method's native social mode. Table 2 lists the five source families and their native protocols.

Table 2: Source method families for the transfer rows, with native protocol and benchmark.

Family	Native protocol	Native benchmark
ChatDev	Planner→developer→tester pipeline (3 agents)	HumanEval
AgentNet	5-agent heterogeneous parallel or routed reasoning	BBH
SOTOPIA	Buyer/seller/mediator bargaining (3 agents)	SOTOPIA
LLM-Deliberation	6-party multi-issue proposal voting	LLM-Deliberation
Werewolf/DPIns	5-player hidden-role deception and detection	Werewolf

For the cooperative targets (BBH, HumanEval), the source family's role structure is mapped onto the cooperative runner: a ChatDev pipeline runs as planner, developer, and tester; an AgentNet or Werewolf team runs as a five-agent heterogeneous group; and a SOTOPIA trio runs as three cooperating agents. For SOTOPIA, the source team bargains under the mixed-motive evaluator. For LLM-Deliberation, the source method's agents populate the six negotiation parties, with the Werewolf family represented by the one-adversarial-incentive variant that best matches its structure. For Werewolf, the source team fills the five player slots under the adversarial evaluator, except that the fixed five-player count forces LLM-Deliberation to use its native five-model configuration rather than the six-party one.

For the Werewolf/DPIns family (Jin et al., 2024), the original method trains a discussion policy with reinforcement learning to choose among discussion tactics, and deduces the other players' hidden roles by belief modeling rather than by a learned policy; DPIns is the name its released implementation gives this discussion-policy-instructed agent. To evaluate all families under the same training-free setting with frozen models, we encode the discussion tactics it selects as a fixed prompt. This instantiation applies uniformly across the six evaluation configurations, so the family appears in every column of the transfer table under the same frozen team.

Team composition. Team size and roles follow each benchmark's native protocol: five agents on BBH, a planner–developer–tester pipeline on HumanEval, a three-role bargaining setup on SOTOPIA, six parties on LLM-Deliberation, and five players on Werewolf. The Werewolf team, and the teams used in the analyses of Appendices C and D, use the strong lineup, which combines GPT-4.1, Gemini-2.5-Pro, Gemini-3.5-Flash, GLM-5.2, and Qwen3-14B across three capability tiers; LLM-Deliberation adds GPT-4o-mini to this lineup for its sixth party. In the main table, HumanEval uses Gemini-3.5-Flash as planner, GLM-5.2 as developer, and GPT-4.1 as tester, and SOTOPIA uses GPT-4.1, Gemini-2.5-Pro, and Gemini-3.5-Flash. BBH uses a weak lineup of five smaller models of a comparable tier, where accumulated experience still changes the outcome on a benchmark that a strong lineup already resolves near its ceiling. Every main agent is paired with an isolated GPT-4o-mini MemoryAgent. The Werewolf runs follow the game configuration of the released DPIns implementation (Jin et al., 2024) without modification; our changes are confined to the agents that fill the roles.

Evaluation scale. Each column of Table 1 is scored over a fixed evaluation set. BBH uses 100 test questions, five from each of 20 subtasks, following the protocol of the AgentNet evaluation it is compared against; HumanEval uses all 164 tasks; SOTOPIA uses 80 interactions, eight episodes for each of ten bargaining scenarios; LLM-Deliberation uses 50 sessions per row; and each Werewolf audit uses 100 games, above the 30 games per condition used in the original One Night Ultimate Werewolf study (Jin et al., 2024). All rows within a column share the same evaluation protocol.

Sampling precision. Each evaluation set is fixed, so the precision of a single entry follows from its size. In the four columns scored as proportions, the binomial standard error of one entry is about four points on BBH, under two on HumanEval, and about five on each Werewolf audit, so a difference of a few points within one of those columns rests on the paired comparison over the shared items rather than on the column size alone; Table 5 reports such a test. The LLM-Deliberation column is scored per session and therefore yields an interval directly: across its 50 sessions the native baseline averages 4.92 (SD 0.90), Self-Memory 4.90 (SD 0.71), and EvolveMAS 5.40 (SD 0.64). The EvolveMAS advantage of 0.48 over the native baseline and of 0.50 over Self-Memory each carry a 95% interval that excludes zero, whether the sessions are treated as paired or as independent samples, while the 0.02 separating Self-Memory from the native baseline does not.

Determinism and pairing. The three BBH rows of Table 1 run under one lineup configuration and a single fixed seed, and present the tasks in the same order, so that column is paired item by item. SOTOPIA draws on a fixed set of ten bargaining scenarios and records no sampling seed; the runs compared there cover the identical scenario set, so its pairing holds at the scenario level. The released artifact ships the lineup configuration behind each reported run.

A.2 Ablation Metric Choice

Each cell of Figure 4 reports one 50-session execution, so every value is attainable from the sessions behind it. Reading Self-Memory $K = \varnothing$, Self-Memory+K_e, MemoryAgent $K = \varnothing$ and MemoryAgent+K_e, the four conditions order identically on average agreement (4.90, 5.24, 5.36, 5.40), 5/6 consensus (78.0, 80.0, 88.0, 90.0), and unanimity on any proposal (24.0, 46.0, 74.0, 86.0). Unanimity on the final proposal (16.0, 34.0, 48.0, 46.0) is the one exception: MemoryAgent+K_e reaches full agreement on some proposal in 86.0% of sessions, but marginally less often holds it on the final proposal than MemoryAgent $K = \varnothing$. The main text reports unanimity on any proposal because the deliberation task is resolved only when every party accepts a proposal, and the same ordering holds on average agreement, the metric carried into Table 1. This measure asks whether unanimity occurs at any point in a session, so it is sensitive to how many proposals a session puts forward; the final-proposal rate is reported beside it so both readings are visible.

B Cross-Setting Transfer Profiles

Figure 6 provides a sector-wise comparison of the seven methods across the six Table 1 configurations. To retain differences among methods while keeping the heterogeneous metrics legible, the visualization uses a fixed upper bound of 100 and a configuration-specific nonzero lower bound $b_j = \frac{1}{2}\min_i s_{ij}$. Radial extent is therefore $(s_{ij} - b_j)/(100 - b_j)$; the annotated labels report the percentage-scaled scores themselves, rounded to one decimal place. Methods should be compared within each sector rather than by radial-length ratios across sectors. The resulting profiles make structural transfer visible: cooperative methods are strongest on the cooperative columns, social methods transfer more naturally within the mixed-motive region, and adversarial policies concentrate their strength in the adversarial columns. EvolveMAS is the only method that remains high across all six columns, consistent with the cross-setting robustness reported in the main text.

The same profiles show a corresponding structural effect on the cooperative columns. ChatDev carries a coding-oriented fixed pipeline, SOTOPIA and Deliberation carry mixed-motive interaction structures, and the Werewolf transfer imposes the strongest communication constraints; all reach 72.0–74.0 on BBH. HumanEval is less sensitive to this mismatch, with all methods in a narrow 92.7–97.0 range. Both benchmarks consist largely of independent tasks, on which a strong cooperative ensemble already performs near ceiling.

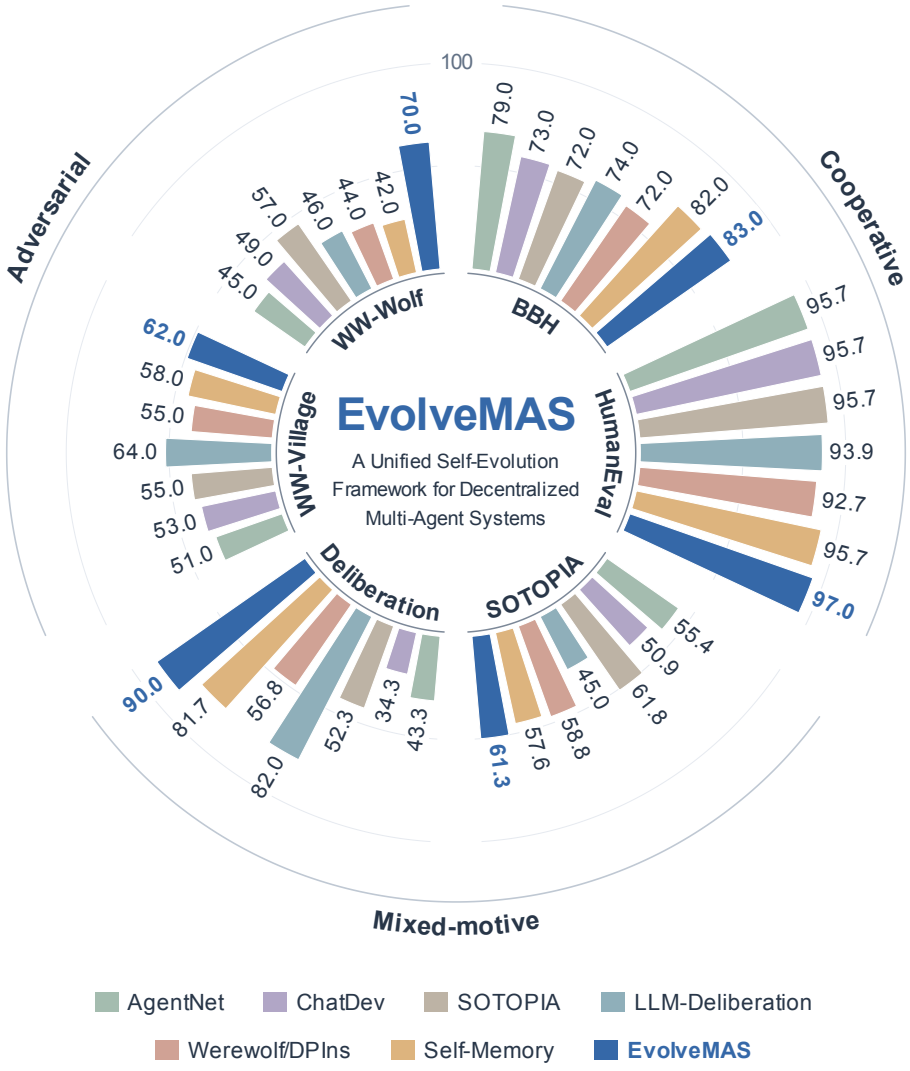


Figure 6: Performance across six Table 1 configurations. SOTOPIA scores are multiplied by 100 and Deliberation scores by $100/6$; all other scores are already on a percentage scale. Each sector starts at half its lowest method score and has an upper bound of 100. Labels report scaled scores rounded to one decimal place. Radial lengths should be compared only within the same sector. WW-Village and WW-Wolf report village win rate and wolf deception success in separate one-sided audits. Dark blue highlights EvolveMAS.

C MAS-Level Properties

This section examines the MAS-level properties of EvolveMAS on traditional systems metrics, with scalability assessed by non-degradation as the team grows. *No Memory* denotes an ablation of our own pipeline that runs the same interaction structure without Structured Memory or Knowledge; in the multi-episode cost and context checks, its main agents carry the raw interaction history in context instead. It is therefore reported here and not as a row of Table 1. Figure 7 addresses three questions: (a) whether accuracy is maintained as the team grows; (b) whether structured cooperation remains competitive with static aggregation; and (c–d) whether amortized cost and context occupancy remain bounded. Table 3 restates these checks and adds the short-horizon measurement.

Table 3: Systems checks. The team-size and aggregation rows report EvolveMAS only. Cost and occupancy compare EvolveMAS with the No Memory ablation, which carries the raw interaction history in context.

Question	Check	Result
Team size[†]	BBH acc. at 1/2/4/5/6 agents	83/83/86/84/85; stable across sizes (Fig. 7a)
Aggregation	EvolveMAS vs. static MoA (5 agents)	84.0 vs. 81.0 (Fig. 7b)
Short horizon	GSM8K accuracy	94.0 vs. No Memory 92.0; maintained
Cost	SOTOPIA $/ep at ep 0 / 7	0.0084 / 0.0105 vs. No Memory 0.0071 / 0.0341 (Fig. 7c)
Context	Occupancy at episode 7	0.62% vs. No Memory 6.54% (Fig. 7d)

[†] The team-size sweep holds the EvolveMAS protocol fixed and varies only the number of main agents. The weak slot uses Qwen3-14B at 1/2/4 agents and GPT-4.1-mini at 5/6 agents. At the overlapping 5-agent point, the two weak-slot lineups agree within n=100 sampling variation (83–84%), and point-to-point differences of 1–3 pp along the sweep lie within the same range.

Team-size scaling. Figure 7a plots EvolveMAS BBH accuracy against team size. The values stay in the 83–86 band across 1, 2, 4, 5, and 6 agents. The 4-agent point of 86.0 lies within n=100 sampling noise of its neighbours, so the sweep shows stable accuracy across team sizes, and the method can be deployed at any of these sizes without re-tuning.

Configuration robustness. Figure 7b compares structured cooperation with memory (84.0) to static Mixture-of-Agents aggregation (81.0) at five agents. Structured interaction thus compares favorably with pooling independent answers. Table 5 reports the same protocol under the strong and the weak lineup.

Cost and context. Figure 7c–d plot the short SOTOPIA session. EvolveMAS pays a fixed MemoryAgent overhead and then stays flat; the No Memory ablation starts cheaper, crosses EvolveMAS within the first two episodes, and continues to rise as its context accumulates. Context occupancy remains near 0.6% for EvolveMAS while No Memory grows with the episode index. The final-episode values in Table 3 restate the same pattern: 0.0105 vs. 0.0341 in cost, and 0.62% vs. 6.54% in occupancy. On the short-horizon GSM8K (Cobbe et al., 2021) check, EvolveMAS reaches 94.0 against 92.0 for No Memory, indicating that MemoryAgent preserves performance on independent single-shot tasks.

D Capability-Stratified Sedimentation

This section asks whether knowledge accumulation under test-time self-evolution is uniform across models in a heterogeneous MAS. We examine this with a per-agent BBH study that inspects the Interaction Knowledge and memory each model produces under EvolveMAS: rule-level Interaction Strategies and Semantic Memory entries.

Table 4: Knowledge sedimentation under EvolveMAS on the heterogeneous BBH team (n=100 tasks).

Model (tier)	Sem. raw	Sem. unique	Rule strategies	Abstraction level
Gemini-2.5-Pro (strong)	27	11	40	general rules
Gemini-3.5-Flash (mid)	4	4	40	subtask features
GLM-5.2 (mid)	11	10	40	boundary conditions
GPT-4.1 (mid)	27	27	40	procedural advice
Qwen3 (weak)	56	11	0	bare answers

Table 4 shows that the acting model shapes how much transferable Knowledge an agent distills. Memory intake itself is not gated by capability: all five models accumulate Semantic Memory entries (column "Sem. unique"). Distillation into transferable rule-level strategies follows capability:

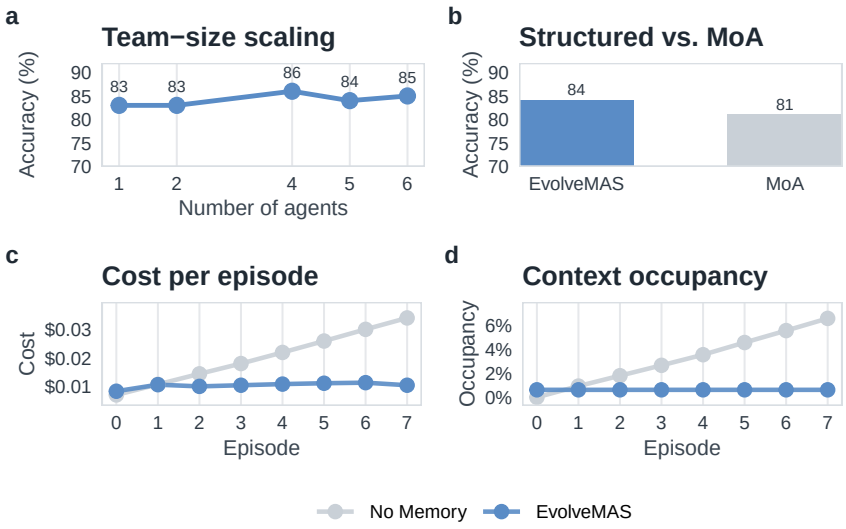


Figure 7: Systems stability checks. (a) Team-size scaling. (b) Structured cooperation versus static MoA aggregation. (c–d) SOTOPIA cost and context occupancy over eight episodes.

the four strong/mid models each fill the 40-item strategy store, while the weak model (Qwen3) distills none. Qwen3's semantic entries are numerous (56 raw, the highest in the team) but highly redundant—deduplicated to 11 unique entries, the same count as Gemini-2.5-Pro—and consist of per-task bare answers rather than generalizable rules. The gap is therefore in abstraction level, not in whether memory is produced. Figure 8a shows this intake directly: every model's semantic store is non-empty, and the weak model's unique count matches the strongest one's. On the performance side, a matched per-role SOTOPIA audit under the same evolution protocol (Figure 8b) shows both roles gaining from EvolveMAS, with the larger gain on the higher-capacity seller role.

Illustrative case. To make the abstraction-level column in Table 4 concrete, we drill into the `boolean_expressions` subtask, which all five models attempt under EvolveMAS. Gemini-2.5-Pro (strong) distills the governing rule directly: *"In boolean logic, the order of operations is* `not`*, then* `and`*, then* `or`*."* GLM-5.2 (mid) captures the same rule plus a boundary case—*"*`not X and Y` *parses as* `not (X and Y)` *not* `(not X) and Y`*"*—and a meta-strategy: *"Systematic step-by-step evaluation respecting precedence is more reliable than left-to-right reading."* In Table 4 terms, Qwen3's corresponding entries remain bare answers, with no distilled rules.

E Within-Setting Breakdowns

This section reports the benchmark-specific breakdowns behind the cross-setting comparison in the main text. The BBH breakdown is supplementary, while the Werewolf audits define how the WW-V and WW-W columns of Table 1 are measured.

E.1 BBH Lineups

The main-table BBH configuration uses a weaker model lineup, leaving room to evaluate improvements from accumulated experience. To examine how these improvements depend on the underlying model capabilities, an additional evaluation uses a stronger lineup. This comparison assesses whether the gains persist as baseline performance approaches the benchmark ceiling. The stronger-lineup results exhibit a corresponding saturation pattern, with less room for further improvement.

Table 5 contrasts the two lineups (n=100 each). Under the weak lineup, EvolveMAS reaches 83.0%, four points above the 79.0% native baseline, with Self-Memory at 82.0%. Under the strong lineup, Self-Memory matches the native AgentNet baseline of 84.0% and EvolveMAS lands two points

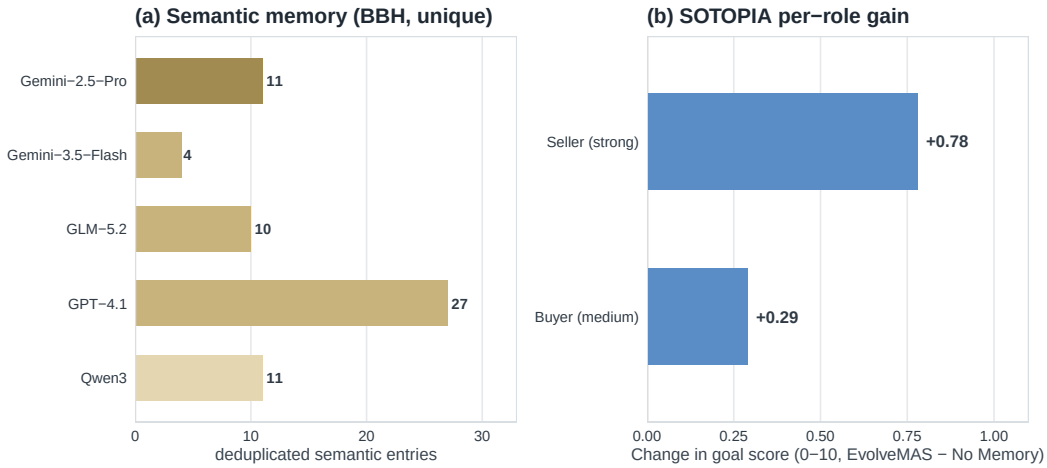


Figure 8: Knowledge intake and role-level gains under EvolveMAS. (a) Unique Semantic Memory entries per model. (b) Per-role SOTOPIA goal-score change on the 0–10 scale (EvolveMAS − No Memory).

Table 5: BBH under the weak and the strong lineup. The weak lineup is the main-table setup; the strong lineup is the five-model team used in the appendix analyses, on which the native AgentNet baseline reaches 84.0%. The transfer families keep their native role structure — the three-role ChatDev pipeline, the SOTOPIA bargaining trio, the five-model LLM-Deliberation parallel loop, and the five-player Werewolf/DPIns team — populated with strong-lineup models, and answer the same 100 items under the same cooperative loop and highest-confidence selection. Answers are scored by option-letter extraction for multiple-choice items and exact match on the remaining subtasks. The strong-lineup runs form a separate batch from the ablation of Table 6.

System	Strong lineup	Weak lineup
AgentNet (native)	84.0	79.0
Self-Memory	84.0	82.0
EvolveMAS	82.0	83.0
ChatDev	81.0	73.0
SOTOPIA	84.0	72.0
LLM-Deliberation	83.0	74.0
Werewolf/DPIns	87.0	72.0

below it: a strong ensemble already resolves most BBH instances, so little experience remains to accumulate across episodes. The lower block places the cross-setting families on that same strong lineup; every row answers the identical 100 items, so all comparisons are fully paired. Accuracies span 81–87%, and exact McNemar tests on the shared items find no significant difference between any pair of systems except Werewolf/DPIns over ChatDev (p=0.03, uncorrected for the 21 comparisons). Under the cooperative parallel loop the LLM-Deliberation family fields the same five-model team as the native reference, which is why its 83.0 sits beside the 84.0 baseline. We therefore read the strong lineup as convergent: once a strong ensemble already resolves most BBH items, the collaboration paradigm a team carries from its source setting no longer separates it, and little cross-episode structure remains for any paradigm to exploit.

E.2 Mechanism Ablation Beyond LLM-Deliberation

We ran the same 2×2 design on BBH, HumanEval and SOTOPIA (Table 6). LLM-Deliberation follows Figure 4; BBH uses the strong lineup of Table 5; HumanEval uses GPT-4.1 as planner and developer with Gemini-3.5-Flash as tester; SOTOPIA uses the main-table team of GPT-4.1, Gemini-2.5-Pro, and Gemini-3.5-Flash. BBH therefore moves from the weak main-table lineup to the strong one, while the HumanEval ablation team fills every slot with models capable of distilling

rule-level Knowledge (Table 4), because an agent that distills none leaves the Knowledge axis inert for its share of the team. These lineup differences are part of why absolute values differ between the ablation and the main-table audit. HumanEval reports the pass rate after up to two repair rounds, and SOTOPIA reports goal achievement averaged over eight episodes. The LLM-Deliberation row repeats Figure 4; the other rows report one designated execution per condition, separate from the main-table audit and from the BBH runs of Table 5, so absolute values are not comparable across those tables. On LLM-Deliberation, BBH and SOTOPIA, the full configuration (MemoryAgent governance with cross-episode Knowledge) attains the highest score. BBH is run on the strong lineup, which already resolves most instances (Table 5) and leaves little experience to accumulate; the four conditions span two points, inside the 81–87% band of Table 5. On HumanEval, the four conditions differ by at most three of 164 tasks, which a single run per condition cannot distinguish from noise. On SOTOPIA, the conditions differ by at most 0.027, of the same size as the episode-to-episode variation within a single run (standard deviation 0.024–0.047).

Table 6: 2×2 factorial on four benchmarks, one designated execution per condition. Bold marks the highest value in each row. BBH answers are scored as in Table 5.

	MemoryAgent		Self-Memory	
Benchmark	$+K_e$	$K = \varnothing$	$+K_e$	$K = \varnothing$
LLM-Deliberation	**5.40**	5.36	5.24	4.90
BBH	**85.0**	83.0	84.0	83.0
HumanEval	**97.6**	96.3	95.7	97.0
SOTOPIA	**0.714**	0.687	0.697	0.698

E.3 Symmetric Werewolf Audit

Table 7: Werewolf audit (n=100, heterogeneous team).

Method	Village win	Detection
Self-Memory	31.0	30.2
EvolveMAS	**48.0**	**44.3**

Table 7 reports the symmetric audit, which complements the one-sided evaluations in the main text. Because both sides evolve in this condition, village win compresses the benefits received by opposing roles into one zero-sum scalar. We therefore use it as a within-setting summary and rely on the directional audit in Table 8 for causal interpretation. Self-Memory reaches 31.0%, while EvolveMAS reaches 48.0% with a detection accuracy of 44.3%. The EvolveMAS rate declines from 0.56 in the first quartile to 0.40 in the final quartile as both sides accumulate memory, which motivates the directional audit below. Deception success is omitted from this table because under symmetric evolution it is the complement of village win within the same game, so it adds no information beyond the first column. Detection accuracy is the rate at which the village side identifies the wolf, read from the vote distribution rather than from the win column, so it moves independently of village win.

E.4 Directional Werewolf Audit

Adversarial interactions entail an inherent goal conflict: improving one side can directly reduce the other side's success. A symmetric deployment therefore cannot reveal which side benefits from evolution. We therefore run one-sided conditions that equip the village and wolf sides with EvolveMAS separately while keeping the opposing side *Unaugmented*: the benchmark's own Werewolf engine, run without memory and without the DPIns discussion policy, so neither of its agents is augmented.

One-sided EvolveMAS conditions. Table 8 states all four memory configurations on a single village-win scale. Village win rises to 0.620 when EvolveMAS is assigned only to the village and falls to 0.300 when it is assigned only to the wolf, so the direction of the shift follows the side

that receives EvolveMAS. The two one-sided rows are the same two runs as the WW-V and WW-W EvolveMAS entries of Table 1, restated on the village-win scale rather than each side's own objective: WW-V $= 62.0$ is the first one-sided row, and WW-W $= 70.0$ is the corresponding wolf deception success, the complement of the 0.300 village win in the second. They are reported here to place both directions and the two symmetric references on one axis. The both-sides-evolving reference lies below the Unaugmented reference on this scale, reflecting the zero-sum compression documented for Table 7: when village and wolf accumulate memory at the same time, village win no longer tracks either side's gain, so the one-sided rows carry the directional claim.

The native DPIns and Self-Memory references in Table 1 remain close across the two directions. EvolveMAS moves village win toward whichever side receives it, with the larger shift on the wolf side (from 0.540 to 0.300).

Cross-setting transfer under one-sided roles. The transfer rows provide a complementary structural check. Under WW-V, LLM-Deliberation reaches 64.0, compared with 62.0 for EvolveMAS and 55.0 for DPIns. Deliberation's multi-party proposal and consensus protocol matches the village coalition's need to pool distributed clues and converge on a suspect. Its two-point margin over EvolveMAS corresponds to two games at n=100, so we read it as a numerical lead rather than evidence of reliable superiority. By the same standard, EvolveMAS's own margin over the remaining WW-V rows is not decisive, and the directional claim rests on WW-W. The remaining transfer families reach 51.0–55.0.

Under WW-W, EvolveMAS reaches 70.0, compared with 45.0 for AgentNet, 49.0 for ChatDev, 57.0 for SOTOPIA, 46.0 for LLM-Deliberation, and 44.0 for DPIns. DPIns's role-agnostic voting instruction is aligned with the village objective, which explains why, measured on each side's own objective, DPIns reaches 55.0 under WW-V and 44.0 under WW-W. Self-Memory reaches 58.0 under WW-V and 42.0 under WW-W, and EvolveMAS exceeds it on both sides, most clearly under WW-W (70.0 vs. 42.0).

Table 8: Werewolf memory configurations on a single village-win scale (n=100 per row). Rows list the village-side policy first.

	Village / wolf configuration	Village win
Symmetric	Unaugmented / Unaugmented	0.540
	EvolveMAS / EvolveMAS	0.480
One-sided	EvolveMAS / Unaugmented	0.620
	Unaugmented / EvolveMAS	0.300

F Long-Horizon Memory Ablation

The main text reports ownership through the benchmark-native Self-Memory and EvolveMAS pairs. This section adds a LoCoMo (Maharana et al., 2024) ablation that examines memory management from the memory perspective, separating memory ownership from update timing. LoCoMo evaluates question answering over long multi-session conversations and allows the starting memory to be held fixed, which isolates the effect of when the MemoryAgent updates. Figure 9 reports the ablation.

Each condition answers the same 1,540 questions, the full set of ten conversations at 154 questions each. A gpt-4.1 judge scores every answer as correct or incorrect, counting an answer correct when it carries the reference information whatever its wording, and a deterministic filter applied ahead of the judge marks an answer incorrect when it declines to commit — "not mentioned", "not specified" — against a reference that states a concrete fact. Absolute accuracies are therefore not comparable with published LoCoMo results, which apply the benchmark's released protocol to answers produced outside a multi-agent loop. Every arm here shares this judge and this filter, so the comparison among them is internally consistent, and the grading program ships with the released code.

Holding the starting memory fixed and varying only whether the MemoryAgent updates during the session, online updates (EvolveMAS) beat offline batch updates (Offline MemoryAgent) overall (18.8 vs. 16.4 with gpt-4.1 memory). The gain is concentrated on single-hop questions (30.9

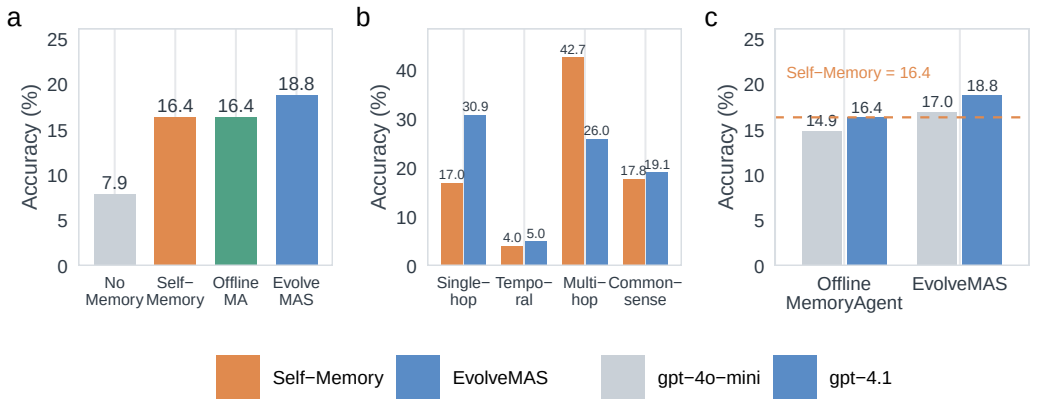


Figure 9: LoCoMo ablation (gpt-4.1 judge). (a) Overall accuracy. (b) Per-category comparison of Self-Memory and EvolveMAS. (c) Memory model sensitivity, with the dashed line marking Self-Memory.

vs. 24.8), where fresh per-question facts are most reusable, and the two are tied on multi-hop questions (26.0 each), where an offline batch has more opportunity to cross-link distant sessions before answering. These per-category values for Offline MemoryAgent are reported here only; Figure 9b contrasts Self-Memory with EvolveMAS. Relative to No Memory, which answers from the most recent session alone (7.9), both Offline MemoryAgent and EvolveMAS show substantial memory gains (+8.5–10.9 pp). EvolveMAS also exceeds Self-Memory overall (18.8 vs. 16.4) and leads on fact recall (single-hop 30.9 vs. 17.0, common-sense 19.1 vs. 17.8), while Self-Memory is strongest on multi-hop reasoning (42.7 vs. 26.0), where its in-context self-notes link distant sessions. Self-Memory and Offline MemoryAgent land on the same overall value, so what separates the arms on this benchmark is online updating: EvolveMAS improves on both static arms by the same 2.4 points.

G Qualitative Cases

To make the self-evolution mechanism concrete, we show representative memory, skill, and profile artifacts produced under EvolveMAS. The content is taken from the Structured Memory and Interaction Knowledge stores written during evaluation and lightly trimmed for length. These cases show that the same memory substrate—episodic, semantic, and working regions—is reused across all six benchmark columns. What changes is the stored content. In cooperative tasks, semantic entries emphasise role-specific observations and coordination cues, while skill artifacts emphasise reusable procedures. In mixed-motive tasks, the substrate retains partner signals and negotiation anchors, while episode-end artifacts expose compact partner models or tactics. In adversarial tasks, it retains deception-relevant cues that remain valid across games. This setting-adaptive content helps explain why the main-table result is more than a collection of separate mechanisms.

BBH (cooperative reasoning) – evolved collaboration skill, EvolveMAS, GPT-4.1

```
Skill: strategy_collaborate_cooperative   type: collaboration  (stored under EvolveMAS)
Agent: agent_gpt41   Model: GPT-4.1   Setting: cooperative (BBH, n=100)
When to use: Always -- this is the agent's current evolved role.

Rule: In cooperative mode, always propose my initial deductions but
explicitly invite my collaborator to check for logical consistency and
point out any contradictions before finalizing the answer.
```

The strategy is short, actionable, and reusable across BBH subtasks. It stores a verification rule rather than a task-specific answer. That is the kind of coordination regularity the paradigm is meant to distill.

HumanEval (cooperative coding) – verification-focused working memory, EvolveMAS

```
Working (tester, episode HumanEval/9):
  focus = verify rolling_max against edge cases
  next  = confirm candidate([])==[] and monotone prefix invariants
```

In the cooperative coding line, the reusable signal is a role-specific test focus rather than a cached final answer. The stored state tells the tester what to verify next. Reuse is therefore anchored to boundary-case checking and task decomposition.

SOTOPIA (mixed-motive) – negotiation anchors, EvolveMAS

```
Example episode: Antique Chair bargaining

Episodic:
  - listed an Antique Chair at $15
  - buyer offered $9 -> countered at $10
  - buyer accepted $10 -> pickup arranged
Semantic:
  - item_value = The Antique Chair is valued at $15 but can be negotiated down to $10 (high
    )
  - interaction_regularity = expressing flexibility in pricing can facilitate negotiation (
    medium)
Working:
  focus: Sell the Antique Chair for at least $10
  next:  Maintain a friendly negotiation tone to build rapport with the buyer
```

Here the reusable content is a compact negotiation anchor set: an acceptable price floor and a simple interaction regularity. The interaction regularity extends beyond this item, and the price anchor guides the current negotiation. They help explain how EvolveMAS reaches the native SOTOPIA goal-achievement level.

LLM-Deliberation (mixed-motive) – stance flexibility, EvolveMAS

```
Agent: SportCo   Condition: EvolveMAS   example session reaching 5/6 agreement

Episodic (deal trajectory, one session):
  - proposed A1,B1,C4,D1,E5  -> blocked by Local Labour Union
  - Local Labour Union proposed A2,B3,C3,D2,E3
  - Mayor, Environmental League, Dept. of Tourism supported A2,B3,C3,D2,E3
  - SportCo proposed A2,B3,C3,D2,E3 (switched, did not re-assert blocked deal)
  - final agreement reached on A2,B3,C3,D2,E3   (5/6 parties agree)
```

This trace shows how EvolveMAS updates a stance during deliberation. Once SportCo's opening deal is blocked, it moves to the coalition-preferred deal, and the session reaches agreement among five of the six parties. This behavioral pattern helps explain the high consensus rates of EvolveMAS.

Werewolf (adversarial) – transferable cues, EvolveMAS, n=100, VWR=0.48

```
Agent: player2   Condition: EvolveMAS
Model: GPT-4.1 (heterogeneous team)   overall VWR = 0.48   early-session = 0.56, late =
     0.40 (zero-sum)

Profile and semantic state:
  player1_tendency = has been voted for and revealed not to be a Werewolf in one observed
     game
  player1_reliability = low
  role_swap_rule = Robber can swap roles with another player, but call order matters
  call_order_cue = stories that violate the night-action order are suspect
```

These entries retain reusable cues: the night-action order and the role-swap rule of the game as reproduced here (Jin et al., 2024), which expose claims that violate them. The same accumulation loop that distills a coordination rule in BBH, a negotiation anchor in SOTOPIA, and a stance shift in Deliberation thus yields durable rule-level cues in Werewolf. Under symmetric evolution, village win starts at 0.56 early in the session and settles at 0.40 as both sides accumulate memory under zero-sum dynamics, and the one-sided audits in Table 8 attribute the shift to the side that receives EvolveMAS. The unified substrate therefore relies on one memory-governance procedure that adapts what it stores to what remains true across resets.