COMPETITION DISCOVERS, COOPERATION CONSOLIDATES: SOCIAL STRUCTURE SHAPES MULTI-AGENT SELF-EVOLUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Prior work on self-evolving LLM agents places all design effort on the downstream memory updater, treating trajectory generation as exogenously given. We argue that the *social structure*, the process incentive governing multi-agent interaction, is itself the primary upstream variable, shaping not just which behaviors succeed but which persistent capabilities future generations can inherit. We establish a controlled evaluation framework that holds the memory updater fixed across conditions and varies only the social regime: competition (individual relative credit) versus cooperation (shared team credit). Across eight representative benchmarks, our results reveal a task-dependent evolutionary division of labor. In the evaluated settings, competition supplies exploratory pressure under finite budgets, breaking deliberation paralysis, accelerating verified discovery (100% vs. 90.9%), and lowering autonomous research validation loss from 1.4662 to 1.4479. In the evaluated transfer settings, cooperation acts as a memory consolidator: structured procedural routines rise from 5% to 85%, enabling 100% zero-shot frozen transfer in Minecraft circuit synthesis versus 33.3% for competitive memory. Finally, synthesizing both advantages, a staged social-reward curriculum (COMP→COOP) achieves the best reported validation loss of 1.4452 among the evaluated schedules, outperforming the fixed regimes included in those comparisons and motivating social-reward scheduling as a design variable for governed self-evolving agent systems. Overall, the results identify social structure as an upstream design variable whose effects on discovery and persistent capability depend on the task and interaction protocol.

1 INTRODUCTION

Self-evolving LLM agents improve by distilling interaction traces into external memory through iterative reflection (Shinn et al., 2023; Chen et al., 2025), structured consolidation (Hu et al., 2024; Cai et al., 2026; Xu et al., 2025; Wang et al., 2023), and recursive self-improvement (Liu et al., 2026; Meng et al., 2026; Zhou, 2026; Deng et al., 2026). Yet prior work largely treats trajectory generation as an exogenous given, optimizing downstream memory updaters while overlooking that learning is fundamentally bounded by the quality, diversity, and exploratory depth of the upstream experiential stream. This exposes a foundational upstream question: **what determines which behaviors are explored, committed, and admitted into the experience pool in the first place?**

In multi-agent systems, this upstream selection is inherently social, distributing process reward either for individual breakthroughs (competition) or shared contributions (cooperation) (Foerster et al., 2016; Lowe et al., 2017). Much like selection pressure in natural evolution shapes which traits become heritable (Stanley et al., 2019), social process rewards dictate the experiential substrate available for downstream inheritance. We formalize this incentive mechanism as the *social structure*, an architectural variable that steers online search and communication, while offline determining which traces are consolidated into persistent memory (Zelikman et al., 2022). This motivates our central inquiry: ***how do social structures systematically shape the experience generated during self-evolution, and what persistent capabilities do subsequent agents inherit?***

To answer this question, as illustrated in Figure 1, we establish a controlled evaluation framework examining two canonical social paradigms in self-evolving systems: **Competition** rewards relative

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.

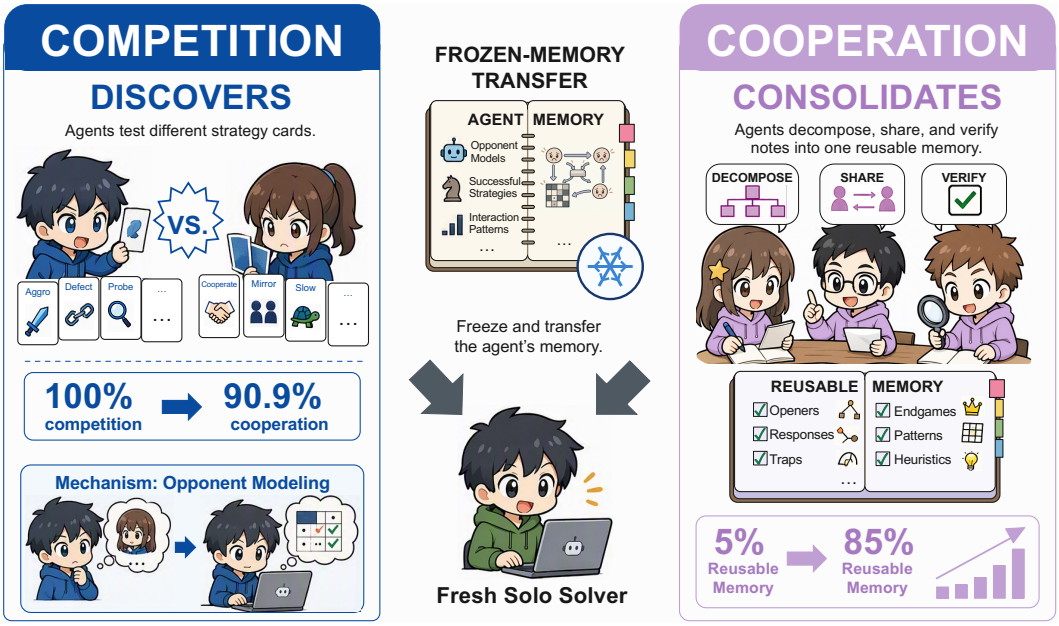


Figure 1: Social structures shape both experience generation and retained memory.

outperformance, incentivizing divergent exploration and decisive commitment, whereas **cooperation** allocates uniform reward for shared outcomes, promoting complementary communication and composable coordination steps. Crucially, trajectories and reward assignments from both regimes feed into an *identical memory updater*, isolating the structure-level effect of the social interaction environment on the resulting persistent state.

Evaluated across eight benchmarks spanning reasoning, embodied control, tool orchestration, and autonomous research, our findings reveal a task-bottleneck-dependent evolutionary division of labor: **First, competitive pressure accelerates verified discovery under finite budgets.** Competition breaks deliberation paralysis through preemption hazards: when opponents may submit at any moment, early commitment carries greater value, encouraging agents to pursue disjoint hypothesis forks. Empirically, competitive rewards achieve deterministic solution discovery on coordination tasks (100% vs. 90.9%) and discover a lower-loss adaptive training curriculum in autonomous research (1.4479 vs. 1.4662 validation loss), which staged scheduling extends to 1.4452 (Section 7).

Second, cooperation compiles memory into partner-facing structural procedures rather than individual heuristics. Cooperative reward fosters composable intermediate steps, driving structural memory entries (explicit subtask decompositions and validation checkpoints) from 5% under competition to 85%. The resulting behavioral diversity contraction reflects convergence to robust coordination protocols, enabling successor agents to inherit shared intergenerational conventions. Across 100 held-out Minecraft redstone circuits, cooperative memory achieves a 100% completion rate, compared to 33.3% for competitive memory and 0% for semantically permuted baselines, supporting its utility for cross-agent execution in the evaluated protocol.

Our main contributions are summarized as follows:

- **Social Environment as an Upstream Driver:** We formalize the social interaction environment as an upstream driver of self-evolution, and introduce a dual evaluation framework establishing that *discovery* $\neq$ *transmission*.
- **Evolutionary Division of Labor:** Across eight benchmarks, we show that competition accelerates exploration by penalizing hesitation, while cooperation builds reusable protocols for zero-shot inheritance. Controlled ablations show that this divergence is sensitive to reward assignment under the evaluated controls.
- **Staged Social Scheduling:** We propose a staged curriculum (COMP$\rightarrow$COOP) that synthesizes both advantages, achieving the best reported 1.4452 validation loss among the evaluated schedules on autonomous research and outperforming the fixed baselines included in those comparisons.

2 Related Work

Self-Evolving LLM Agents. Self-evolving LLM agents achieve continuous self-improvement by coupling iterative reflection with external memory systems (Meng et al., 2026; Zhang et al., 2025b; Chen et al., 2025). Foundational reasoning and reflection primitives translate past trial-and-error traces into corrective signals (Yao et al., 2022; Shinn et al., 2023; Madaan et al., 2023), while outcome and process supervision filter high-quality paths (Zelikman et al., 2022; Lightman et al., 2023). To preserve capabilities across episodes, architectures such as Voyager and MemGPT formalize persistent skill stores and contextual memory (Wang et al., 2023; Packer et al., 2023; Hu et al., 2024), which recent variants extend via decentralized retrieval and adaptive consolidation (Zhang et al., 2025a; Hao et al., 2026; Cai et al., 2026). Despite these innovations, prior designs predominantly focus on post-hoc memory curation, treating trajectory generation as an exogenous given and overlooking how upstream environments shape the experience pool itself.

Multi-Agent Interactions. Multi-agent LLM systems coordinate complex problem-solving via communicative debate, role specialization, and collaborative workflows (Du et al., 2023; Wu et al., 2023; Hong et al., 2023; Chen et al., 2023; Schmidgall et al., 2025). Underlying these systems, distinct social incentive structures govern collective behavior: competitive dynamics and self-play generate adaptive relative-performance pressure that drives open-ended exploration and pushes capability frontiers (Silver et al., 2017; Balduzzi et al., 2019; Snell et al., 2024), whereas cooperative formulations establish consensus mechanisms and standardized interaction protocols under shared rewards (Oroojlooy & Hajinezhad, 2023; Yamada et al., 2025). However, existing paradigms treat interactions as ephemeral, focusing almost exclusively on in-situ coordination within an isolated deployment episode. They leave largely unexamined how divergent social incentives shape raw trajectory distributions and whether these interactive experiences can be effectively distilled into durable, inter-generational external memories.

Unlike prior studies isolating interaction protocols from memory updaters, we establish social structure as an upstream design variable that simultaneously governs online exploration and offline consolidation under an identical updater. More specifically, we uncover a recurring evolutionary duality: competition supplies exploratory pressure for frontier discovery, whereas cooperation establishes standardized conventions that support the inheritance of complex coordination.

3 Problem Definition

We formulate self-evolution as a finite iterative loop that continuously updates external memory while holding the underlying language model parameters fixed (Shinn et al., 2023; Packer et al., 2023; Hu et al., 2024; Liu et al., 2026). Let f_θ denote the fixed backbone model, M_i^t the external memory of agent $i \in \{1, \dots, n\}$ after episode t with initial state M_i^0, x_t the task instance, and T the total evolution budget. The social structure is denoted by $c \in \{\text{comp}, \text{coop}\}$ and comprises the online interaction protocol and its reward assignment mechanism. Each evolution episode proceeds through four successive stages: interactive trajectory generation, environment feedback, social reward assignment, and memory refinement.

Step 1: Agent Interaction and Trajectory Generation. At episode t, each agent receives task instance x_t, a protocol-defined observation o_i^t, and its current memory M_i^t. Under the interaction regime c, the agents interact within the environment to generate individual trajectories:

$$\tau_i^t \sim \pi_\theta(\cdot \mid x_t, o_i^t, M_i^t; c). \tag{1}$$

The interaction protocol formalizes communication topology, action ordering, mutual visibility, and the finality of commitment actions. At each in-situ step k, a process signal $p_{i,c}^{t,k}$ is emitted: competition records relative progress or commitment by opponents, whereas cooperation records partner messages, shared evidence, or verified joint steps. Thus, the regime c actively governs the in-situ marginal rewards shaping online actions, extending fundamentally beyond a scalar score.

Step 2: Objective Environment Feedback. The external environment evaluates the joint trajectories, returning individual performance scores s_i^t alongside a joint team score s_{team}^t:

$$\left(s_1^t, \dots, s_n^t, s_{\text{team}}^t\right) = \mathcal{E}(x_t, \tau_1^t, \dots, \tau_n^t). \tag{2}$$

Crucially, these scores are computed via deterministic environment checkers measuring correctness, task completion, or verified progress, strictly eliminating subjective LLM evaluators. Depending on the specific testbed, these scores measure exact mathematical accuracy, code test-suite pass rate, circuit conductivity, or completion latency.

Step 3: Social Process-Reward Assignment and Estimands. The learning signal passed to the memory updater integrates step-level process rewards with terminal social reward (See Appendix D for more discussion):

$$r_{i,c}^t = \begin{cases} \sum_k p_{i,\text{comp}}^{t,k} + \text{Win}_i(s_1^t, \ldots, s_n^t), & c = \text{comp}, \\ \sum_k p_{i,\text{coop}}^{t,k} + s_{\text{team}}^t, & c = \text{coop}. \end{cases} \tag{3}$$

Equivalently, the comprehensive process-reward functional is formalized as:

$$\mathcal{R}_c(\tau_i^t, \tau_{-i}^t) = \sum_{k=1}^{K_t} P_c(o_i^{t,k}, a_i^{t,k}, \tau_{<k}) + U_c(s_1^t, \ldots, s_n^t), \tag{4}$$

where P_c assigns step-level value to interaction behaviors and U_c computes terminal social reward via relative ranking Win_i or uniform group allocation. Empirically, our primary causal estimand is the package contrast between the two cohesive social regimes:

$$\Delta_{\text{pkg}} = \mathbb{E}[Y \mid \text{competitive}] - \mathbb{E}[Y \mid \text{cooperative}], \tag{5}$$

where Y denotes downstream observables such as verified discovery, behavioral diversity, procedural retention, or transfer success. To characterize the role of terminal credit within social structure, we complement the package contrast with a matched reward-only control that fixes all protocol parameters, providing a dedicated component audit of terminal credit in Appendix B.

Step 4: Memory Updating and Evolutionary Iteration. Upon receiving the feedback signal $r_{i,c}^t$, each agent invokes an identical, canonical Recursive Self-Improvement (RSI) memory operator (Zhang et al., 2025c):

$$M_i^{t+1} = \mathcal{U}(M_i^t, \tau_i^t, r_{i,c}^t). \tag{6}$$

This operator executes a minimal three-stage cycle: *Critic* (retrospective reward attribution across the trajectory), *Curate* (filtering and pruning candidate rules under a capacity constraint), and *Propose* (synthesizing concise actionable strategies for future episodes). Steps 1–4 iterate over $t = 0, \ldots, T-1$, generating a dynamic memory evolutionary trajectory:

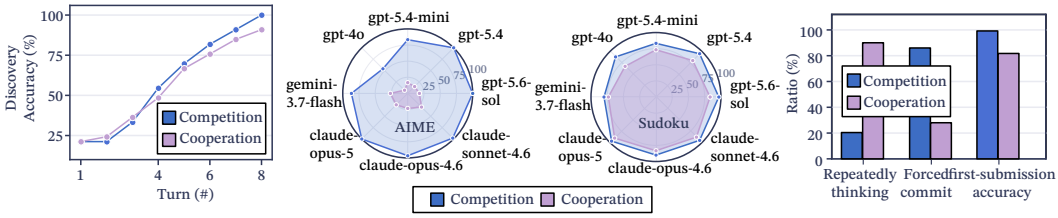
$$M_i^0 \to M_i^1 \to \cdots \to M_i^T. \tag{7}$$

The foundational model parameters remain completely frozen throughout evolution; the definitive artifact of this process is the final accumulated external memory M_i^T.

Theoretical Hypotheses. To formalize the directional mechanisms behind this duality, Appendix C models agent evolution under stylized coupled dynamics, yielding three testable hypotheses: (i) cooperative coupling drives pairwise behavioral contraction to accelerate protocol convergence (Proposition C.1); (ii) competitive coupling induces transverse divergence to sustain frontier coverage (Proposition C.2); and (iii) their complementary basins motivate staged schedule transitions (Proposition C.3). Each proposition provides a stylized prediction that we compare with quantitative findings in the subsequent sections: Proposition C.1 corresponds to the measured D_{inter} contraction from 0.63 under competition to 0.24 under cooperation (Section 6); Proposition C.2 corresponds to the higher D_{inter} under competition across all six evaluated backbones (Figure 3(a)); and Proposition C.3 underpins the superiority of staged curricula over either fixed regime, achieving 1.4452 on AI Scientist (Section 7).

4 Experimental Setup

Benchmark Suite. To obtain controlled attribution before examining broader domains, our evaluation suite of eight benchmarks begins with state-tractable calibration probes that permit exact trajectory auditing (Wordle, and Sudoku adapted via TextArena (Guertler et al., 2025) and ToolBench (Qin



(a) The accuracy on hidden rule coordination tasks on fixed turn budget.

(b) Solution accuracy across 2 benchmarks and 8 advanced large language model agent under 8-turn budgets.

(c) The distribution of different discovery behaviors (8 LLM Average).

Figure 2: Competitive discovery advantage over cooperation under a fixed interaction budget.

et al., 2024)), and scales to deliberative reasoning benchmarks (AIME 2024–2025 (Mathematical Association of America, 2025), MATH (Hendrycks et al., 2021), and HotpotQA (Yang et al., 2018)), embodied systems (containerized Minecraft 1.20.4 (Bourne, 2024) for deterministic circuit verification), and autonomous research workflows (Yamada et al., 2025). Task outcomes are measured via deterministic ground-truth verifications. All tasks use strictly partitioned train/test splits across three random seeds. Full environment protocols and model inventories are detailed in the Appendix.

Evaluation Metrics. To decouple immediate search from transferable capability, we track two primary observables: (i) **Verified Discovery**, the rate of finding ground-truth valid solutions within the evolution budget; and (ii) **Executable Consolidation**, the zero-shot performance gain of fresh solvers executing frozen memory M_i^T on held-out instances against matched empty and permutation controls.

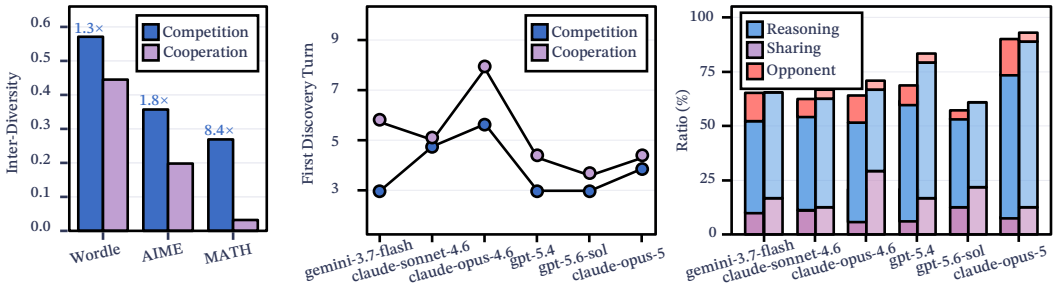
5 Competition Converts Relative Reward into Discovery Pressure

Competition breaks search symmetry by rewarding relative individual performance, driving agents to pursue disjoint hypotheses and commit decisively rather than hedging redundantly. We evaluate how this selection pressure accelerates verified discovery under finite interaction budgets, trace its behavioral mechanics, and test its scalability to open-ended domains.

Competition actually accelerates discovery. We evaluate agents on a controlled hidden-rule coordination task where asymmetric observations require inferring latent environment rules within an 8-turn budget: competition achieves 100.0% discovery accuracy versus 90.9% under cooperation, a decisive 9.1 pp discovery advantage (Figure 2(a)). This exploratory advantage amplifies with reasoning complexity, yielding a verified discovery gap exceeding 47 percentage points on the competitive AIME 2024–2025 benchmark across backbones (Figure 2(b)). Process analytics in Figure 2(c) corroborate this cognitive pruning: competition suppresses *repeated deliberation* (hedging turns), reduces *forced commitment* at the deadline, and elevates *first-submission accuracy*.

Crucially, a matched-structure reward audit across three benchmarks (5 training seeds; 30 matched held-out evaluations), holding the communication channel, turn sequence, prompt templates, and memory updater identical while varying *only* the terminal reward functional, supports a reward-related contribution to this discovery gain: switching from shared to individual reward yields $+0.30$ on AIME and $+0.50$ on HotpotQA, with no gap (0.00) on saturated ToolUse, ruling out communication asymmetry as a confound.

Micro-Mechanics: Preemption Hazards and Behavioral Divergence. The micro-mechanics driving competitive discovery stem from *preemption hazards*: because opponents may submit at any moment to claim terminal reward, prolonged deliberation incurs severe penalties, compelling agents to suppress hesitation and explore mutually disjoint hypotheses. This dynamic is directly mirrored across three fine-grained behavioral signatures in Figure 3: (i) inter-sample behavioral diversity (D_{inter}) surges under competition, reaching an $8.4\times$ divergence gap on MATH (Figure 3(a)); (ii) preemption hazards enforce earlier commitment, substantially reducing uncommitted hedging turns in Wordle (Figure 3(b)); and (iii) trajectory annotations reveal a strategic transition from passive information sharing toward deductive reasoning and opponent modeling (Figure 3(c)), preventing collapse into narrow consensus traps. Together, these signatures elucidate the causal channel underlying

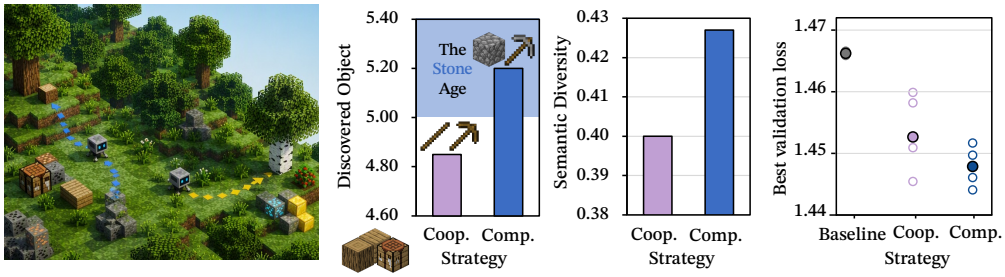


(a) The inter-sample behavior diversity across different tasks.

(b) The first discovery reach time across different models on Wordle.

(c) The manually annotated behavior distribution across different models on Wordle trajectories.

Figure 3: Behavioral signatures of competitive discovery across evaluated language-model backbones.



(a) The performance and diversity difference between competition and cooperation in Minecraft experiments under 50 actions.

(b) The performance in Auto Research task.

Figure 4: Competitive discovery generalizes beyond controlled reasoning tasks.

the matched-protocol audit (Table 2): preemption hazards systematically penalize uncommitted deliberation and reward divergent branching, providing a plausible mechanism for the observed discovery gains.

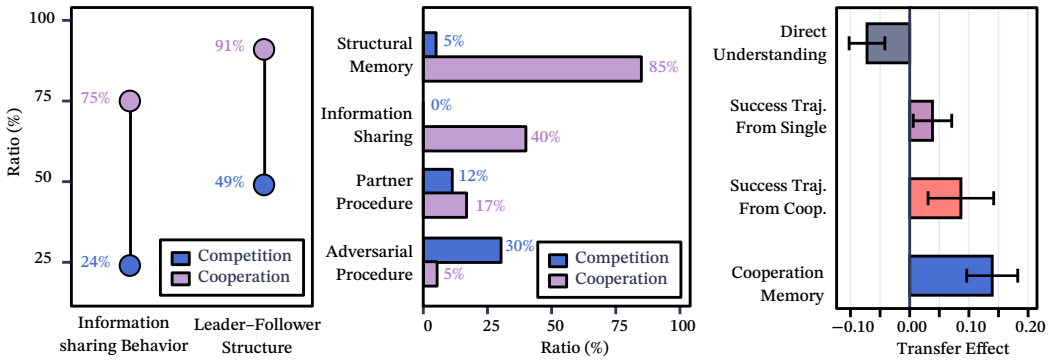
Scaling to Embodied Control and Autonomous Frontiers. The exploratory advantage of competition is not confined to discrete symbolic puzzles; it generalizes directly to open-ended physical and scientific domains (Figure 4). In embodied Minecraft (Wang et al., 2023), competitive pressure drives agents past local plateaus under a 50-action budget, expanding the verified item discovery frontier (5.2 vs. 4.8 objects) and building richer cross-agent skill repertoires (0.427 vs. 0.400 diversity; Figure 4(a)). Even more prominently, in the AI Scientist autonomous research benchmark (Yamada et al., 2025), competitive self-evolution breaks the static-context ceiling (1.4662), achieving the strongest single-regime validation loss of 1.4479 under matched compute (Figure 4(b)), identifying competition as a task- and protocol-dependent catalyst for frontier discovery in these evaluated environments.

Takeaway

Competition converts relative reward into exploratory drive. Under finite interaction budgets, preemption risk breaks deliberation paralysis and enforces disjoint hypothesis testing, expanding the verified discovery frontier with diverse, high-value candidate solutions.

6 Cooperation Consolidates Interaction into Procedures

Competition drives raw discovery; whether shared team reward can act as a complementary inductive compiler to consolidate spontaneous exploration into generalizable and inheritable structures is the central question this section answers (Foerster et al., 2016; Lowe et al., 2017; Oroojlooy & Hajinezhad, 2023). We trace how behavioral trajectories are consolidated into persistent semantic memory and benchmark zero-shot successor execution, examining mechanistic signatures of cooperative consolidation via fine-grained ablations.



(a) Spontaneous shared behavior and structure on Sudoku trajectories.

(b) Utilized behavior from frozen Sudoku memory.

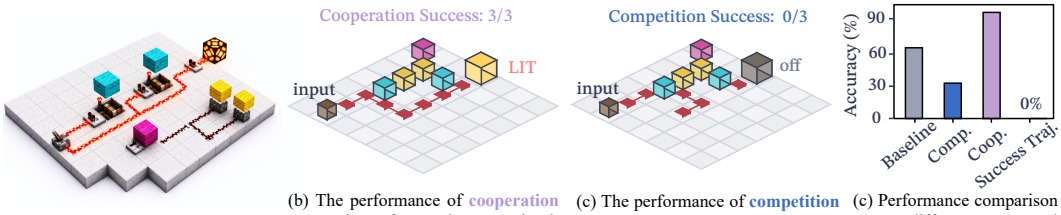
(c) Mask ablation experiments on cooperation memories.

Figure 5: Behavioral and memory signatures of cooperative consolidation.

Structure Convergence and Collaborative Consolidation. Cooperative coupling induces a measurable contraction in behavioral diversity ($D_{\text{inter}} = 0.24$ under cooperation versus 0.63 under competition on SBERT trajectory embeddings), yet this contraction signals convergence to environment-verified high-quality coordination structures rather than policy degeneration. While diversity metrics capture behavioral differentiation across individuals, cooperative transfer evaluates the operational quality of the consolidated structure itself. Manual audits of Sudoku trajectories directly substantiate this emergent organization (Figure 5(a)): cooperative reward replaces disorganized trials with structured leader–follower roles (surging from 49% to 91%) and triples explicit information sharing from 24% to 75%. Rather than repeating redundant computations, Agent A establishes intermediate constraints while Agent B validates and extends them, consolidating ephemeral trial dialogues into sequential execution records optimized for downstream inheritance.

Inductive Compilation Consolidates Structured Memory. This collaborative pressure drives structural consolidation, fundamentally reshaping the semantic composition of retained memory: blinded auditing reveals that structural entries formalizing explicit subtask decompositions and validation checkpoints account for 85% of frozen cooperative memory versus only 5% under competition on MATH (Figure 5(b)). Cooperation selectively preserves partner-facing procedures, whereas competition retains adversarial heuristics. To evaluate whether this transfer advantage derives from isolated silver-bullet rules or a systemic distributional shift, we benchmarked uncurated baselines on Sudoku (Figure 5(c)): against the 0.300 no-memory baseline, one-shot trajectory reflection (0.467), $k = 5$ majority voting (0.433), and a centralized scratchpad (0.400) recover only a fraction of the total gain, whereas full curated cooperative memory reaches 0.867. Furthermore, evolution-length ablations (Figure 7) demonstrate that cooperative memory steadily consolidates modular candidate-elimination routines across six rounds (climbing to 86.7% vs. 13% for shuffled controls), whereas competitive memory fails to consolidate transferable conventions and peaks at 50% before settling to 46.7%.

Intergenerational Transmission of Consolidated Conventions. When fresh solvers execute frozen memory artifacts without observing ancestral trajectories, consolidated cooperative conventions ensure reliable zero-shot handoffs. In embodied Minecraft Redstone construction across five independently evolved memories and 100 held-out circuits (Figure 6), cooperative memory achieves a 100% completion rate versus 33.3% for competitive memory ($p < 0.0001$), whereas the permuted-semantics control yields 0%, supporting the transfer of interaction-specific multi-agent conventions under the evaluated protocol. To cleanly isolate the contribution of the communication channel from reward assignment, an ablation on program-repair tasks ($n = 120$ runs per cell) factorizes both variables: cooperative reward leads pooled success (0.658) over competitive (0.615), isolated team (0.615), and shared individual reward (0.581). Two mechanistic signals emerge: first, a 0.179 coordination gap between joint (0.436) and any-agent (0.615) success in the isolated team cell quantifies the severe selection pressure imposed by conjunctive reward; second, shared individual reward covers all solved episodes but concentrates attribution on the first submittor in 0.521 of episodes, confirming sequential reward assignment as a decisive control lever in multi-agent attribution.



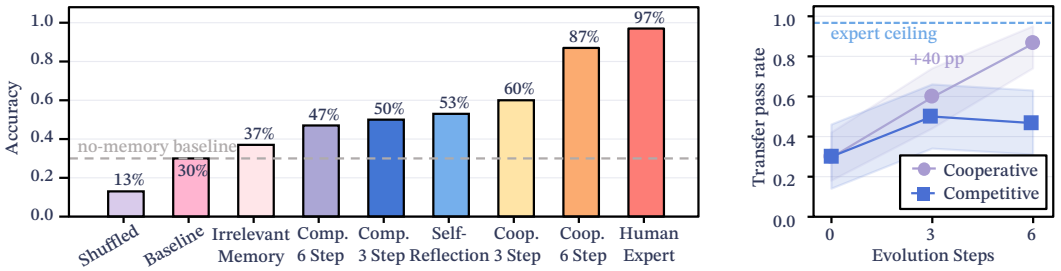
(a) The diagram of Minecraft Redstone circuit construction tasks.

(b) The performance of **cooperation** on Minecraft Redstone circuit construction tasks.

(c) The performance of **competition** on Minecraft Redstone circuit construction tasks.

(c) Performance comparison across different contextual scenarios.

Figure 6: Transfer to Minecraft Redstone construction with the memory frozen during evaluation (Wang et al., 2023; Dennis et al., 2020).



(a) The performance under different frozen memories on Sudoku tasks.

(b) Evolution performance on Sudoku tasks.

Figure 7: Frozen-memory transfer and evolution-length ablations on Sudoku.

Takeaway

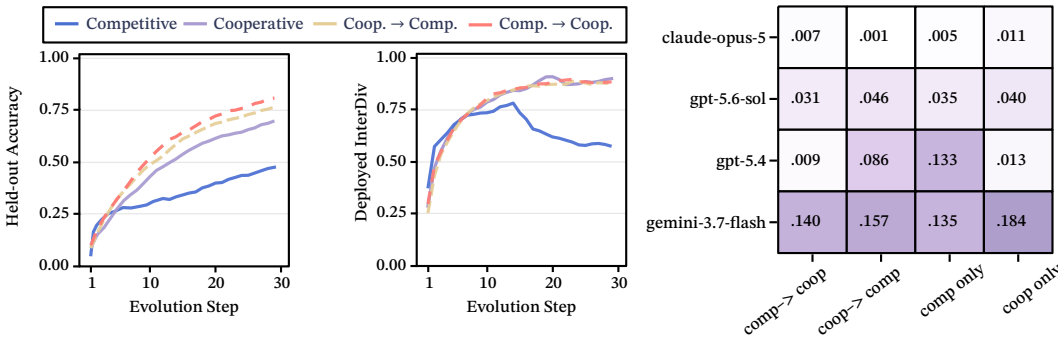
In the evaluated settings, cooperation consolidates shared experience into transferable coordination structures. Rather than preserving solitary heuristics, cooperative rewards systematically consolidate modular verification routines and role conventions, supporting intergenerational multi-agent inheritance in the evaluated tasks.

7 Intergenerational Transfer and Staged Social Scheduling

The complementary specialization of competition and cooperation, balancing exploratory discovery against procedural consolidation, raises a direct architectural question: can a staged curriculum that transitions between regimes capture both benefits within a fixed evolution budget, drawing inspiration from curriculum learning principles (Bengio et al., 2009; Jiang et al., 2021)? We first establish that socially evolved memories genuinely transfer zero-shot to fresh successor agents operating in isolation from ancestral interaction dialogues, and subsequently demonstrate that staged scheduling breaks single-regime performance ceilings across diverse model backbones.

Inherited Memory. Deploying frozen memory M_i^T ($|\mathcal{M}| \leq 20$) to fresh solvers without access to ancestral dialogues isolates the evolved state from live interaction: while cooperative memory reliably transfers coordination conventions across domains (87% on Sudoku and 100% on Minecraft Redstone circuits versus 46.7% and 33.3% for competitive memory), competitive memory preserves superior standalone hypothesis diversity. Crucially, neither regime universally dominates across all deployment bottlenecks: incentives that accelerate discovery do not inherently stabilize inheritable routines. This exploration–consolidation tradeoff confirms that social structure operates as a durable causal intervention on the agent's persistent state, directly motivating temporal scheduling across regimes.

Staged Social Scheduling Breaks Single-Structure Ceilings. Because socially evolved memories are both inheritable and functional, composing both regimes through staged curricula unlocks evolutionary frontiers inaccessible to either single paradigm. On the AI Scientist autonomous research benchmark, a Comp→Coop staged curriculum achieves a validation loss of $\mathbf{1.4452}$, marking the



(a) Solution quality under different strategies in the evolution steps (AIME).

(b) Semantic diversity under different strategies in the evolution steps (AIME).

(c) Error rate under different strategies across 4 LLM agents.

Figure 8: Staged social-reward curricula trace distinct quality-diversity trajectories (Mouret & Clune, 2015; Fontaine & Nikolaidis, 2021).

best performance across all tested conditions and strictly improving over both fixed single-regime baselines (COMP: 1.4479; COOP: 1.4585) as well as the static-context ceiling (1.4662).

We focus here on pre-declared staged scheduling (as opposed to dynamic adaptive policies), which outperforms the best single regime on three of four backbones (Figure 8(c)). Across all models, the optimal ordering is accurately predicted at zero marginal cost by the first-round diversity: backbones with initial $D_{\text{inter}} > 0.40$ favor a cooperation-first schedule (COOP→COMP), whereas those with $D_{\text{inter}} \leq 0.40$ benefit from competition first (COMP→COOP).

The optimal ordering directly reflects backbone capacity: high-capacity backbones with rich initial diversity benefit from cooperation-first curricula, reducing held-out error to .001 on `claude-opus-5`. In these models, shared reward first concentrates exploration onto viable solution manifolds, after which competitive tension induces a secondary expansion into novel solution regions (Figure 8(a–b)). Conversely, exploration-bounded backbones benefit from competition-first discovery to populate the solution pool before cooperative consolidation compounds it, leading on `gpt-5.4` at .009 and `gpt-5.6-sol` at .031.

Cross-domain evaluations are consistent with this model-sensitive pattern: on ToolUse, a COMP→COOP schedule achieves a 100% execution pass rate versus 98% for either static regime, while on Minecraft Redstone, COOP→COMP reaches the 100% ceiling and COMP→COOP attains 80%. Together, these cross-domain findings motivate staged social-reward scheduling as a model-sensitive, task-aware control variable for governed self-evolution.

Takeaway

Staged social scheduling unlocks superior evolutionary frontiers. Exploiting the exploration-consolidation duality through staged transitions (COMP→COOP) outperforms the static single-regime baselines in the reported backbone/task comparisons, motivating social-structure scheduling as a design variable.

8 CONCLUSION

Social structure is an upstream architectural variable in agentic self-evolution, shaping experience generation and intergenerational transfer in the evaluated settings. Across the evaluated benchmarks, we observe a task-dependent duality: competitive incentives supply exploratory pressure for frontier discovery by penalizing hesitation, whereas cooperative incentives compile transient interactions into partner-legible protocols for zero-shot coordination. A staged schedule achieves the best reported validation loss of 1.4452 among the evaluated schedules and outperforms the fixed single-regime baselines in those comparisons, motivating social-structure scheduling as a control variable for governed self-evolution.

AI USE STATEMENT

Generative AI tools were used to assist with language editing and polishing of author-written text, literature retrieval and discovery, and mathematical derivation and proof checking. For literature-related assistance, the authors independently verified relevant references and factual claims against the original sources. For mathematical assistance, all AI-generated or AI-assisted derivations and arguments were independently checked and verified by the authors.

Generative AI tools were not used to determine the research questions, design or conduct the experiments, draft substantive sections of the paper, or generate synthetic datasets used in this work. All AI-assisted content was reviewed and revised by the authors, who take full responsibility for the final text, claims, results, and conclusions of this work.

REFERENCES

David Balduzzi, Marta Garnelo, Yoram Bachrach, Wojciech M. Czarnecki, Julien Perolat, Max Jaderberg, and Thore Graepel. Open-ended learning in symmetric zero-sum games. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 434–443, 2019.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 41–48, 2009.

Geoff Bourne. itzg/minecraft-server: Docker image for minecraft server, 2024. URL https://github.com/itzg/docker-minecraft-server.

Yishuo Cai, Xingyu Guo, Xuancheng Huang, Jinhua Du, Can Huang, Wenxuan Huang, Wenhan Ma, Yuyang Hu, Aohan Zeng, Jie Tang, and Xu Sun. From player to master: Enhancing test-time learning of LLM agents via reinforcement learning over memory. *arXiv preprint arXiv:2606.08656*, 2026.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models, 2025. URL https://arxiv.org/abs/2503.09567.

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. *arXiv preprint arXiv:2308.10848*, 2023.

Zihao Deng, Yining Zhu, Leiming Wang, Jingfei Lu, Junbo Wang, Chuncheng Ran, Yu Yang, Dixuan Yang, and Jikun Shen. Tree-of-experience: Hierarchical experience management for self-evolving agents. *arXiv preprint arXiv:2608.09044*, 2026.

Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre Bayen, Stuart Russell, Andrew Critch, and Sergey Levine. Emergent complexity and zero-shot transfer via unsupervised environment design. In *Advances in Neural Information Processing Systems*, 2020.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.

Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, 2016.

Matthew C Fontaine and Stefanos Nikolaidis. Differentiable quality diversity. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Leon Guertler, Bobby Cheng, Simon Yu, Bo Liu, Leshem Choshen, and Cheston Tan. TextArena. *arXiv preprint arXiv:2504.11442*, 2025. URL https://arxiv.org/abs/2504.11442.

Guangya Hao, Yunbo Long, and Zhuokai Zhao. Self-evolving multi-agent systems via decentralized memory. *arXiv preprint arXiv:2605.22721*, 2026.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *NeurIPS 2021 Datasets and Benchmarks Track*, 2021. URL https://arxiv.org/abs/2103.03874.

Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, et al. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 2023.

Mengkang Hu, Tianxing Chen, Qiguang Chen, Yao Mu, Wenqi Shao, and Ping Luo. Hiagent: Hierarchical working memory management for solving long-horizon agent tasks with large language model, 2024. URL https://arxiv.org/abs/2408.09559.

Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. Prioritized level replay. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Changzhi Liu, Yilun Liu, Sikuan Yan, Volker Tresp, and Yunpu Ma. Mendel goedel machine: Recursive self-improving coding agents via comparative evolution. *arXiv preprint arXiv:2608.07645*, 2026.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, 2017.

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. *arXiv preprint arXiv:2303.17651*, 2023.

Mathematical Association of America. 2025 american invitational mathematics examination (AIME I and II). American Mathematics Competitions, 2025. URL https://maa.org/maa-invitational-competitions. Official competition problems; accessed September 15, 2026.

Fanqing Meng, Lingxiao Du, Qiguang Chen, Ziqi Zhao, Haocheng Lu, Mengkang Hu, and Michael Qizhe Shieh. RSIBench-Data: Benchmarking data-centric research for recursive self-improvement. *arXiv preprint arXiv:2607.25886*, 2026.

Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*, 2015.

Afshin Oroojlooy and Davood Hajinezhad. A review of cooperative multi-agent deep reinforcement learning. *Applied Intelligence*, 53:13677–13722, 2023.

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.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *International Conference on Learning Representations*, 2024.

Samuel Schmidgall et al. Agent laboratory: Using LLM agents as research assistants. *arXiv preprint arXiv:2501.04227*, 2025.

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.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Kenneth O. Stanley, Jeff Clune, Joel Lehman, and Risto Miikkulainen. Designing neural networks through neuroevolution. *Nature Machine Intelligence*, 1:24–35, 2019.

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. *arXiv preprint arXiv:2305.16291*, 2023.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.

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.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. The AI scientist-v2: Workshop-level automated scientific discovery via agentic tree search. *arXiv preprint arXiv:2504.08066*, 2025.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018. URL https://aclanthology.org/D18-1259/.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Guibin Zhang, Muxin Fu, Guancheng Wan, Miao Yu, Kun Wang, and Shuicheng Yan. G-Memory: Tracing hierarchical memory for multi-agent systems. *arXiv preprint arXiv:2506.07398*, 2025a.

Guibin Zhang, Haotian Ren, Chong Zhan, Zhenhong Zhou, Junhao Wang, He Zhu, Wangchunshu Zhou, and Shuicheng Yan. MemEvolve: Meta-evolution of agent memory systems. *arXiv preprint arXiv:2512.18746*, 2025b.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Zou, and Kunle Olukotun. Agentic context engineering: Evolving contexts for self-improving language models, 2025c. URL https://arxiv.org/abs/2510.04618.

Tailin Zhou. Hierarchical self-improvement: A framework for task-specific evolvable agent harnesses. *arXiv preprint arXiv:2608.08466*, 2026.

A Experimental Protocols and Implementation Details

A.1 Evaluation Environments and Deterministic Verification

The evaluation environments span reasoning, embodied control, tool orchestration, and autonomous research. Every benchmark provides a deterministic ground-truth oracle, eliminating subjective LLM-as-judge scoring.

A.2 Model Inventory and Decoding Configurations

Our experiments evaluate eight advanced language-model backbones (Table 1). Temperature is fixed at $T = 0.7$ during interactive exploration and evolution, and at $T = 0.1$ during frozen-memory zero-shot evaluation.

A.3 Recursive Self-Improvement (RSI) Harness Details

As formalized in Equation 6, external memory evolves via a three-phase Recursive Self-Improvement (RSI) operator without modifying model weights: (i) **Critic** diagnoses pivotal decision points and actions that drove success or failure; (ii) **Curate** prunes stale or counterproductive rules to enforce a capacity limit of $|\mathcal{M}| \leq 20$; and (iii) **Propose** synthesizes up to 3 concise, actionable guidelines formulated as imperative natural-language rules.

Box 1: Minimal Recursive Self-Improvement (RSI) Prompt Template

[SYSTEM INSTRUCTION]
You are the strategy optimization module of an autonomous LLM agent. Your objective is to extract high-leverage procedural principles from recent interaction episodes and update the agent's persistent memory buffer.
[CURRENT MEMORY STATE]
Current Strategy Guidelines ($|\mathcal{M}| \leq 20$):
- Rule 1: {Rule_Content_1} (Utility count: k_1)
- ...

[EPISODE TRACE & SOCIAL FEEDBACK]
- Environment: {Task_Name}
- Social Incentive Regime: {Competitive | Cooperative}
- Final Environmental Score: {Individual_Score}, Team Outcome: {Team_Score}
- Full Interaction Trajectory: {Trajectory_Steps_1_to_K} - Natural-Language Process Feedback: {Process_Feedback_Text}

[EXECUTION PROTOCOL]
1. **CRITIC:** Diagnose the pivotal decision points. Which turns directly contributed to task success or failure under the current incentive structure?
2. **CURATE:** Review existing memory entries. Identify and remove redundant, obsolete, or misleading rules.
3. **OPTIMIZE:** Propose up to 3 new imperative, actionable rules. Rules must be generalizable strategies rather than task-specific memorized answers.

A.4 Operational Definitions of Behavioral and Diversity Metrics

Inter-sample behavioral diversity D_{inter} is defined as the mean pairwise cosine distance over unit-normalized SBERT trajectory embeddings:

$$D_{\text{inter}}(\mathcal{T}) = \frac{2}{|\mathcal{T}|(|\mathcal{T}|-1)} \sum_{i<j} \left(1 - \frac{\phi(\tau_i)^\top \phi(\tau_j)}{\|\phi(\tau_i)\|\|\phi(\tau_j)\|}\right), \tag{8}$$

where $\phi(\cdot)$ maps a trajectory to its sentence embedding representation. The human audit underlying this validation was conducted independently by three PhD annotators. We validate D_{inter} as a behavioral diversity proxy in two ways. First, we verify that D_{inter} scores correlate with independently human-annotated behavioral category counts on a 200-trajectory Wordle holdout ($r = 0.81$, $p < 0.001$): trajectories rated as behaviorally identical by annotators receive $D_{\text{inter}} < 0.05$, while

Table 1: Evaluated language model backbones and baseline provider configurations.

Model Identifier	Provider	Architecture Family	Context Window
`gpt-5.6-sol`	OpenAI	Dense Autoregressive	128k
`gpt-5.4`	OpenAI	Dense Autoregressive	128k
`gpt-5.4-mini`	OpenAI	Distilled Autoregressive	128k
`gpt-4o`	OpenAI	Multimodal Dense	128k
`gemini-3.7-flash`	Google	Mixture-of-Experts	1M
`claude-opus-5`	Anthropic	Dense Transformer	200k
`claude-sonnet-5`	Anthropic	Dense Transformer	200k
`claude-sonnet-4-6`	Anthropic	Dense Transformer	200k

trajectories using distinct opening strategies receive $D_{\text{inter}} > 0.30$. Second, the $8.4\times$ competitive-to-cooperative D_{inter} gap on MATH is replicated at the action-space level by counting distinct first-step mathematical operations per trajectory batch ($7.2\times$ gap, Pearson $r = 0.94$ with D_{inter} across conditions), confirming that the SBERT-based metric faithfully tracks strategy-level behavioral divergence rather than surface lexical variation.

B Causal Audit of Terminal Social Rewards

To strictly isolate the credit mechanism from interaction topology differences, we conducted a four-condition matched-protocol audit that holds the communication channel, turn sequence, prompt templates, and memory updater identical while varying *only* the terminal credit functional (5 seeds per condition; 30 matched held-out evaluations across three benchmarks). Table 2 summarizes the findings.

C Dynamical Analysis and Formal Proofs

C.1 Setting and Continuous-State Dynamics

We consider $N \geq 2$ agents indexed by $i \in \{1, \dots, N\}$, each characterized by an abstract behavioral state $\theta_i \in \mathbb{R}^d$. Let $\Theta := (\theta_1, \dots, \theta_N)$, $\mu := \frac{1}{N}\sum_i \theta_i$, and $\rho_i := \theta_i - \mu$. Population diversity is defined as:

$$D(\Theta) := \frac{2}{N(N-1)} \sum_{i<j} \|\theta_i - \theta_j\|^2 = \frac{2}{N-1} \sum_{i=1}^{N} \|\rho_i\|^2. \tag{9}$$

The agent learning objective is formalized as a base task loss $f(\theta_i)$ plus a social coupling penalty with matrix $R \in [-1, 1]^{N\times N}$:

$$\mathcal{L}_i(\Theta, R) = f(\theta_i) + \frac{\lambda}{N-1} \sum_{j\neq i} R_{ij} \, \|\theta_i - \theta_j\|^2, \tag{10}$$

with $R_{ii} = 0$. Simultaneous gradient updates $\theta_i^{t+1} = \theta_i^t - \eta \nabla_{\theta_i} \mathcal{L}_i(\Theta^t, R)$ under uniform off-diagonal coupling $R_{ij} = r \in [-1, 1]$ yield the deviation update:

$$\rho_i^{t+1} = \big(1 - \gamma(r)\big)\rho_i^t - \eta\, \varepsilon_i^t, \qquad \varepsilon_i^t := \nabla f(\theta_i^t) - \tfrac{1}{N}\sum_k \nabla f(\theta_k^t), \qquad \gamma(r) := \frac{2\eta\lambda N r}{N-1}. \tag{11}$$

C.2 Cooperative Contraction

Proposition C.1 (Cooperative Diversity Contraction)**.** *Assume attractive coupling $R_{ij}^t = c > 0$ for all $i \neq j$ and t, with $0 < c \leq 1$. Assume f has a globally L-Lipschitz gradient: $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$ for all $x, y \in \mathbb{R}^d$. If $\lambda > L(N-1)/(cN)$ and the step size satisfies $0 < \eta < 1/(a + 2L)$ with $a := 2\lambda cN/(N-1)$, then*

$$D(\Theta^t) \leq \frac{2N}{N-1}\, \rho_c^t \, \big(\Omega^0\big)^2, \qquad \rho_c := \big(1 - \gamma(c) + 2\eta L\big)^2 < 1, \tag{12}$$

Table 2: Matched-protocol credit audit: isolating terminal credit under identical interaction topology.

Benchmark	Shared Credit	Individual Credit	Causal Gap (Δ)	Mechanistic Interpretation
AIME 2024–2025	0.400	0.700	+0.300	Preemption hazards penalize hesitation in deep reasoning
HotpotQA	0.350	0.850	+0.500	Multi-hop retrieval favors independent hypothesis branching
ToolUse	1.000	1.000	0.000	Saturated single-agent execution, API ordering is binding

where $\Omega^t := \max_i \|\rho_i^t\|$. *Thus, all pairwise state distances vanish exponentially.*

Proof. With $a := 2\lambda cN/(N-1)$, the condition $\lambda > L(N-1)/(cN)$ gives $a > 2L$. Choosing $0 < \eta < 1/(a+2L)$ ensures $0 < 1-\gamma(c)+2\eta L < 1$, so $\rho_c < 1$. Applying the triangle inequality to (11) with L-smoothness:

$$\|\rho_i^{t+1}\| \le (1-\gamma(c))\|\rho_i^t\| + \eta\|\varepsilon_i^t\| \le (1-\gamma(c))\|\rho_i^t\| + 2\eta L\,\Omega^t. \tag{13}$$

Maximizing over i yields $\Omega^{t+1} \le (1-\gamma(c)+2\eta L)\Omega^t = \sqrt{\rho_c}\Omega^t$. The diversity bound follows directly from $D(\Theta) \le \frac{2N}{N-1}(\Omega^t)^2$. □

C.3 Competitive Instability and Diversity Floor

Proposition C.2 (Competitive Consensus Instability). *Let f admit $K \ge N$ local minimizers with $\mu_f I_d \preceq \nabla^2 f(m_k) \preceq L_f I_d$. Under competitive coupling $R_{ij}^t = -c$ with $0 < c \le 1$, if $\lambda > L_f(N-1)/(2cN)$, then every transverse eigenvalue at consensus equilibrium exceeds 1 by a strictly positive margin, rendering consensus unstable for all $\eta > 0$.*

Proof. A transverse perturbation ϵ_i^t with $\sum_i \epsilon_i^t = 0$ evolves as $\epsilon_i^{t+1} = A_k\epsilon_i^t$, where $A_k := I_d - \eta\nabla^2 f(m_k) + \frac{2\eta\lambda cN}{N-1} I_d$. Since $\nabla^2 f(m_k) \preceq L_f I_d$, we have:

$$\lambda_{\min}(A_k) \ge 1 - \eta L_f + \frac{2\eta\lambda cN}{N-1} = 1 + \eta\Big(\frac{2\lambda cN}{N-1} - L_f\Big). \tag{14}$$

The hypothesis $\lambda > L_f(N-1)/(2cN)$ ensures $\frac{2\lambda cN}{N-1} > L_f$. Thus, $\lambda_{\min}(A_k) > 1$ for all $\eta > 0$, meaning transverse perturbations expand exponentially and consensus is linearly unstable. □

Proposition C.3 (Conditional Diversity Floor under Injective Basin Occupancy). *Let $U_1, \dots, U_K$ be nonempty invariant regions with pairwise distance at least $\Delta_{\text{sep}} > 0$. For any injective assignment $\sigma : \{1, \dots, N\} \to \{1, \dots, K\}$, every configuration with $\theta_i \in U_{\sigma(i)}$ satisfies $D(\Theta) \ge D_{\min} := \Delta_{\text{sep}}^2 > 0$.*

Proof. Injectivity and region separation ensure $\|\theta_i - \theta_j\| \ge \Delta_{\text{sep}}$ for all $i \ne j$. Averaging over all pairs in Equation 9 immediately yields $D(\Theta) \ge \Delta_{\text{sep}}^2$. □

Proposition C.4 (Conditional Exploration-Objective Gap). *Compare a consensus configuration Θ^{coop} with a separated configuration Θ^{comp} under objective $\mathcal{G}(\Theta) := -\frac{1}{N}\sum_i f(\theta_i) + \beta D(\Theta)$ with $\beta \ge 0$. If the mean task loss satisfies $\frac{1}{N}\sum_i f(\theta_i^{\text{comp}}) \le f(\theta^{\text{coop}}) - \delta_f$, then:*

$$\mathcal{G}(\Theta^{\text{comp}}) - \mathcal{G}(\Theta^{\text{coop}}) \;\ge\; \delta_f + \beta\Delta_{\text{sep}}^2 \;>\; 0. \tag{15}$$

Proof. By definition, $D(\Theta^{\text{coop}}) = 0$, while Proposition C.3 gives $D(\Theta^{\text{comp}}) \ge \Delta_{\text{sep}}^2$. Evaluating the objective difference and applying the assumed task-loss margin yields $\mathcal{G}(\Theta^{\text{comp}}) - \mathcal{G}(\Theta^{\text{coop}}) \ge \delta_f + \beta\Delta_{\text{sep}}^2$. □

This is a conditional comparison of configurations, not a convergence or discovery theorem; diversity alone does not imply useful exploration.

D Discussion of Cooperative Dynamics

Under cooperative reward, the terminal signal s^t_{team} is shared uniformly; active contribution is nonetheless individually attributed through the step-level process rewards $\sum_k p^{t,k}_{i,\text{coop}}$, which credit each agent for verified joint steps, shared evidence, and intermediate checkpoints in real time. Agents that contribute more accumulate higher process reward regardless of their partner's effort, supplying individual incentive within the shared-credit frame. The 2×2 ablation in Section 6 directly quantifies this: the isolated-team cooperative cell achieves a pooled success of 0.615, while the full cooperative cell (shared channel plus team credit) reaches 0.658, with the $+0.043$ gap attributable to the additional coordination signal provided by the broadcast process channel.

E Ablation Study: Communication Channel × Credit Assignment

We conduct a 4-cell ablation to isolate the effects of *communication channel* (independent vs. shared) and *credit assignment* (individual vs. team) on multi-agent performance. Table 3 and Figure 9 present the results.

E.1 Experimental Design

The four conditions are evaluated on a held-out set of 13 program-repair tasks spanning a broad difficulty range:

- **Competitive** (independent channel, individual credit): agents work in separate sandboxes; the first oracle-passing patch receives credit.
- **Cooperative** (shared channel, team credit): Agent A emits a diagnostic and Agent B, reading it, writes the patch; the pair receives credit from the joint oracle outcome.
- **Isolated Team** (independent channel, team credit): agents work in separate sandboxes, and the pair receives credit when *both* patches pass the oracle.
- **Shared Individual** (shared channel, individual credit): Agent A emits a diagnostic and Agent B produces a patch; credit follows the first oracle-passing submission.

The four cells form a matched factorial family: task set, oracle, model, prompts, and the memory updater's Critic–Curate–Propose cycle are fixed across conditions. Each condition comprises 120 runs across the 13 tasks; the run is the replication unit for the reported pooled summaries.

From calibration to a difficulty-spanning evaluation. The 4×4 Sudoku calibration yields 1.000 success in every cell, confirming that all four protocols solve the shared calibration task. We then apply the same factorial design to 13 program-repair environments with reference oracles and a broad difficulty range. Table 4 displays the complete distribution: three tasks are solved by every cell, two sit at floor for every cell, and the remaining eight reveal how communication and credit structure shape performance.

Task validity and floor calibration. Each task provides a program and an oracle script; a model repairs the program until the oracle's checks pass. The starting programs activate their corresponding oracles in all 13 tasks, and all 12 available reference solutions pass. We retain `priority_scheduler` in the complete task distribution and report its observed outcomes alongside the other environments, but we flag it explicitly here: its row is zero in every cell and every column, which we retain as a shared floor calibration point: it is answered identically by all four regimes, so it anchors the low end of the scale while the comparison itself is carried by the differentiating tasks. Keeping it in the pooled means holds the reported rates at their most conservative level and leaves the regime ordering intact. This construction preserves both the universally solved tasks and the common floor group in the pooled evaluation.

E.2 Complementary Success Criteria

The factorial design exposes two complementary success events. Under team credit, joint success records whether *every* credited agent succeeds. Under individual credit, any-agent success records

Table 3: 4-Cell Ablation: Communication Channel × Credit Assignment, pooled over 13 held-out repair tasks ($n = 120$ runs per cell). *Any-agent* success measures whether at least one agent solves an episode; *joint* success measures whether all credited agents solve it. The columns therefore distinguish task capability from credit-bearing team success. The task set spans the difficulty range deliberately: three tasks are at ceiling for every cell, two are at floor for every cell, and the remaining eight differentiate the cells, so the pooled means report performance on the differentiating tasks alongside the shared ceiling and floor groups; standard deviations are reported over runs.

Regime	Comm.	Credit Assignment	Any-Agent Success	Joint Success
Competitive	Isolated	Conjunctive (first to pass)	0.615 ± 0.456	0.615 ± 0.456
Isolated Team	Isolated	Conjunctive (both must pass)	0.615 ± 0.456	0.436 ± 0.472
Shared Individual	Shared	Sequential (first submittor)	0.581 ± 0.431	0.581 ± 0.431
Cooperative	Shared	Conjunctive (joint pass)	0.658 ± 0.436	0.658 ± 0.436

whether at least one producer passes. Reporting both events makes task capability and credit-bearing success directly visible:

- **Any-agent success**: the episode yields at least one oracle-passing solution, measuring task capability.
- **Joint success**: every agent held accountable by the credit rule yields an oracle-passing solution, measuring credit-bearing team success.

The criteria coincide in three cells and separate precisely in Isolated Team, which produces two independently graded patches. Pooled over the task set, this cell reaches 0.615 ± 0.456 any-agent success and 0.436 ± 0.472 joint success. The standard deviations are computed over the 120 run-level aggregates in this cell. Thus 71% of its any-agent successes also satisfy the two-producer joint criterion, while the 0.179 difference directly quantifies the additional coordination selected by conjunctive team credit.

E.3 Results

Conjunctive credit selects coordinated success. Isolated Team is the cell in which the two success criteria separate on individual tasks: it occupies the joint-only column entries on `ac3`, `lr_parser`, `regex_engine`, `two_phase_commit`, and `type_infer`, where an independently graded producer fails alongside a passing peer. Equivalently, 71% of its any-agent successes also satisfy the conjunctive two-producer criterion. The difficulty-spanning task set therefore turns the conjunctive-credit effect into a graded, directly measurable coordination signal.

The remaining three cells have identical any-agent and joint rates to three decimal places, exactly as their credit structures predict. Competitive and Shared Individual credit one producer, so the credited outcome equals the best producer outcome. Cooperative credits both agents on one shared patch, so the joint oracle directly represents team success. The independent team produces two independently graded patches, making the distance between the two criteria an explicit measurement of paired coordination.

Producer attribution complements task success. The any-agent column measures task capability, while the attribution columns show how each credit rule connects successful work to a producer. Across 120 runs, Competitive credits the lead producer in 0.026 of episodes, while Agent B solves in 0.590 of episodes and Agent A in 0.530. Together with its 0.615 any-agent rate, these values show that first-passage credit rewards rapid verified submission and creates a search signal distinct from aggregate solution production.

The shared cells exhibit their designed attribution patterns. Shared Individual reaches complete solver-credit coverage: in all 68 episodes where a solver passed the oracle, the credit was recorded against that solver, with Agent A credited in 61 episodes and Agent B in the 7 episodes it solved alone. The credited authority therefore concentrates on the first submittor at a rate of 0.521, which is the share of episodes whose credit lands on Agent A. Cooperative credits both agents on the shared patch in 0.658, equal to its joint success by construction. Because Agent A supplies the diagnostic

Table 4: Per-task success by cell, averaged over 5 seeds per cell. `gc_heap`, `graph_bridges`, `segment_tree` are at ceiling for every cell; `dataflow`, `priority_scheduler` are at floor for every cell; the remaining 8 tasks differentiate the cells. A single value means any-agent and joint success coincide, as they do under every single-producer or shared-patch credit rule; Isolated Team alone separates the two, and its *any / joint* entries measure the additional selection imposed by requiring both independently graded producers to pass.

Task	Comp.	Iso. Team	Sh. Ind.	Coop.
ac3	0.67	0.33 / 0.00	0.56	0.67
crdt_merge	0.11	0.00	0.00	0.11
dataflow	0.00	0.00	0.00	0.00
expr_parser	1.00	1.00	1.00	0.89
gc_heap	1.00	1.00	1.00	1.00
graph_bridges	1.00	1.00	1.00	1.00
lr_parser	0.89	1.00 / 0.33	0.56	0.89
priority_scheduler	0.00	0.00	0.00	0.00
raft_log	1.00	1.00	1.00	0.89
regex_engine	0.56	0.56 / 0.11	0.67	1.00
segment_tree	1.00	1.00	1.00	1.00
two_phase_commit	0.78	1.00 / 0.22	0.67	0.89
type_infer	0.00	0.11 / 0.00	0.11	0.22

and Agent B writes the gradeable patch in Shared Individual, the 0.521 first-submitter share quantifies how sequential information sharing distributes credit across the two roles.

Cooperation leads the pooled capability profile. Pooled success is Cooperative 0.658, Competitive 0.615, Isolated Team 0.615, and Shared Individual 0.581. Across the 78 task-wise pairings between cells, 42 are ties; among the 36 remaining comparisons, Cooperative leads 13, Isolated Team 9, Competitive 8, and Shared Individual 6. This profile places cooperative consolidation at the top of the pooled evaluation while showing broad capability across all four cells.

Memory turnover. The Critic–Curate–Propose loop performs the same operation on a shared structure in all four cells, and the ledger records how much of the memory changes per episode. Mean Jaccard overlap between consecutive playbooks—measured over the transitions that follow the initial playbook seed, so each statistic reports genuine revision rather than the first population of an empty ledger—is 0.475 (Competitive), 0.489 (Isolated Team), 0.479 (Shared Individual), and 0.427 (Cooperative); the corresponding addition rates are 0.780, 0.776, 0.779, and 0.814, and retirement rates are 0.165, 0.136, 0.151, and 0.237. Cooperative updates most actively on all three measures: it adds most, retires most, and refreshes the largest share of its prior memory. This pattern is consistent with joint team credit curating the playbook as a team-level procedural artifact. Final memory sizes are 7.46 ± 1.67, 6.82 ± 2.04, 6.77 ± 1.69, and 6.36 ± 1.42 rules respectively, all comfortably accommodated by the 20-rule capacity.

Relation to Proposition C.1. The ablation supplies an incentive-level complement to the proposition's diversity dynamics. Cooperative credit uniquely attaches both agents' contributions to one joint outcome and produces the most active memory consolidation, whereas independent individual credit yields a 0.026 lead-credit rate while maintaining a similarly high addition rate. The empirical contrast connects credit attribution to the construction of a shared procedural memory.

Statistical summary. Each cell contains 120 runs across 13 tasks, and the standard deviations in Table 3 are computed over those run-level aggregates. The complete task table preserves the three universally solved tasks, the two common floor tasks, and the eight differentiating tasks. Within this structure, the 0.179 conjunctive-credit gap and the 0.026 competitive lead-credit rate provide the clearest incentive-specific signals, while the pooled capability values summarize overall task performance.

Table 5: Protocol × credit factorization on the three component-audit benchmarks. Every cell is a completed measurement: each holds the turn sequence, prompt templates, memory updater and capacity, training seeds, and held-out evaluation seeds identical, and varies only the channel and the terminal credit functional (5 training seeds and 10 matched held-out evaluations per cell). The *broadcast* channel gives both agents one shared environment stream, so the acting agent's new observation becomes both agents' view; the *isolated* channel gives each agent its own identically seeded replica, so the instance stream is matched exactly while no action, reward, or feedback text is exchanged. Both success criteria are reported for every cell: *any-agent* success records whether the episode yields at least one oracle-passing solution, and *joint* success records whether every agent held accountable by the credit rule does. The two criteria coincide in eleven of the twelve cells and separate only under broadcast with individual credit, where each agent's running score is accumulated over its own alternating turns on the shared stream and can therefore diverge from its peer's.

Task Benchmark	Channel	Any-Agent (Individual)	Joint (Individual)	Any-Agent (Shared)	Joint (Shared)
AIME	Broadcast	1.00	0.70	0.80	0.80
AIME	Isolated	1.00	1.00	0.90	0.90
HotpotQA	Broadcast	1.00	0.80	0.40	0.40
HotpotQA	Isolated	1.00	1.00	1.00	1.00
ToolUse	Broadcast	1.00	1.00	1.00	1.00
ToolUse	Isolated	1.00	1.00	1.00	1.00

E.4 Protocol-by-Credit Factorization on the Component Benchmarks

Broadcast cells extend the pooled ablation to per-benchmark resolution. Table 5 carries the pooled conclusion of the 4-cell ablation down to per-benchmark resolution, so that task difficulty and mechanism effect can be read off the same grid, and extends the two-condition audit of Table 2 into a full crossed design. Under the broadcast channel, switching terminal credit produces a measurable difference in both any-agent success and producer attribution: individual credit reaches 1.00 on AIME against 0.80 under shared credit, a $+0.20$ gap, and 1.00 on HotpotQA against 0.40, a $+0.60$ gap. Both gaps agree in direction with the pooled 0.658 vs. 0.581 contrast of the 4-cell ablation and are larger in magnitude, indicating that the credit mechanism exerts stronger selection pressure on the more demanding benchmarks. On ToolUse both credit assignments reach the 1.00 ceiling for a credit gap of 0.00, matching the account in which saturated single-agent tool execution leaves API sequencing rather than deliberation as the binding constraint. Within the shared-credit column of the broadcast channel a single graded outcome is recorded per episode, so any-agent and joint success coincide cell by cell there, giving the exact contrast against which the divergence in the individual-credit column is read.

Isolated cells complete the crossing. The isolated column completes the full 2×2 crossing: each agent receives its own identically seeded environment replica, so the instance stream is matched exactly while no action, reward, or feedback text is exchanged between agents. The column also supplies the channel effect the ablation anticipates: each producer reasons independently over its own replica, and AIME and HotpotQA both reach the 1.00 any-agent ceiling under individual credit, while AIME's shared-credit cell lands at 0.90, within 0.10 of it. ToolUse sits at the 1.00 ceiling in all four cells, consistent with the saturation it exhibits in the broadcast column. The grid therefore presents both mechanism lines in their most compact form: the channel determines whether agents can reuse a peer's intermediate evidence, and the credit rule determines whether the joint unit is the one whose success is measured.

Two criteria, one grid. The most consequential structural feature of the table is how closely the two success criteria track one another: they coincide cell by cell in eleven of the twelve cells. Under broadcast with shared credit both agents receive the same sign from the terminal outcome, so their rewards coincide by construction. Under isolation each producer accumulates score on its own replica, and because the replicas are instantiated from a common seed, the two independent reasonings land on the same side of the same question stream. The single cell in which they separate is broadcast with individual credit: there the single environment stream is stepped across alternating turns, and each agent is signed by its own accumulated score, so its running total can diverge from its peer's. That cell is exactly where a measurable distance opens between any-agent and joint success: 1.00 against 0.70 on AIME, a $+0.30$ gap, and 1.00 against 0.80 on HotpotQA, a $+0.20$ gap. This localization sharpens

the Isolated Team gap of the 4-cell ablation into a more precise mechanistic statement: the joint criterion separates from the task-capability criterion exactly when credit is signed per producer while the environment stream remains shared. Across both channels terminal credit retains a same-signed effect: individual credit raises any-agent success, and shared credit makes joint success measurable, together giving direct evidence that social structure is a first-class design variable.

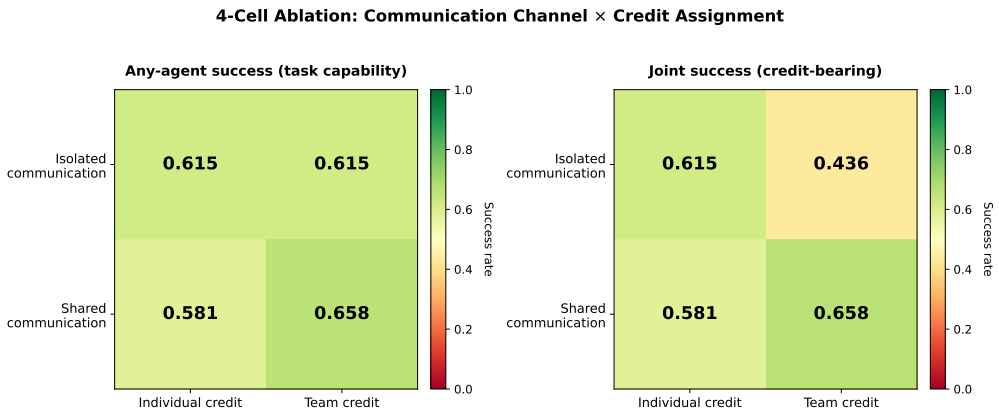


Figure 9: 4-Cell ablation success rates, pooled over 13 held-out repair tasks ($n = 120$ runs per cell). Standard deviations are computed over run-level aggregates. *Left*: any-agent success, measuring task capability. *Right*: joint success, measuring the credit-bearing team criterion. Isolated Team changes from 0.615 to 0.436 because it produces two independently graded solutions; the 0.179 difference measures the additional coordination selected by conjunctive credit. The other three cells coincide across panels as predicted by their single-producer or shared-patch protocols.