AREAL-RSI: ADAPTIVE SCALING AND COORDINATION OF RECURSIVE SELF-IMPROVEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Scaling recursive self-improvement requires additional computation to produce useful research progress in reaching persistent objectives. Intuitively, more researcher agents can explore more hypotheses, but early evaluation scores might misrepresent their promise, shared findings could lose their applicability conditions, and useful procedures could remain confined to completed investigations. In this paper, we present AREAL-RSI, a framework for organizing a persistent research team of agents without updating backbone LLM weights. We formulate the scaling problem in terms of evaluated outcomes and whole-system resource expenditure. Concretely, a leader agent selects task-appropriate research orientations and guides ongoing investigations; conditional memory sharing distinguishes unresolved problems from supported findings and enables recipient adaptation. During recursive evolution, after each campaign, evidence-based review revises researcher orientation, knowledge, and skills for later campaigns. Experiments compare AREAL-RSI with CODEX-GOAL on downstream harness optimization, GPU kernel optimization, and AI research tasks using two backbones. Under the reported resource allocations, AREAL-RSI improves Terminal Bench performance by 2.2 percentage points for each backbone and increases mean KernelBench speedup by 32.2–36.8%; further scalability comparisons favor the design of AREAL-RSI over native scaling: combining experience sharing with specialized researcher roles yields 1.9–27.5% lower kernel latency than independent exploration and 1.3–33.9% lower latency than sharing alone across all seven comparable task–backbone pairs.

1 INTRODUCTION

Recursive self-improvement (RSI) seeks to improve an AI agent through repeated self-modification and empirical evaluation (Zhang et al., 2025b). The scope of RSI can extend from improving a task solution to revising the procedures that generate subsequent improvements (Zhang et al., 2026; Zheng et al., 2026). This naturally raises a test-time scaling problem, i.e., *how should additional inference and experimental resources improve the outcomes of RSI?* A system can investigate more hypotheses, share discoveries between ongoing investigations, or retain useful procedures for later tasks. In this paper, we study *how a leader agent can coordinate these uses of computation across persistent researcher agents, with an external researcher agent that is open to revision.*

Scaling out is essential for RSI because, when pursuing persistent objectives, research progress depends on exploring alternatives and testing them, not simply producing a longer answer across recursive iterations. For example, GPU kernel optimization and machine learning engineering require executable artifacts whose quality is established through experimentation (Ouyang et al., 2025; Wijk et al., 2024; Chan et al., 2024). Parallel researcher agents can investigate complementary hypotheses while retaining the depth needed to implement and test each one. Sharing a useful diagnosis can then benefit several investigations, and retaining the resulting procedure can reduce repeated work on later tasks. A leader agent realizes this opportunity by combining evidence across the team of research agents to guide investigative priorities and decide which findings deserve wider attention. Comprehensively, the scaling objective is to turn additional research capacity into useful progress while accounting for the cost of coordination and evaluation.

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.

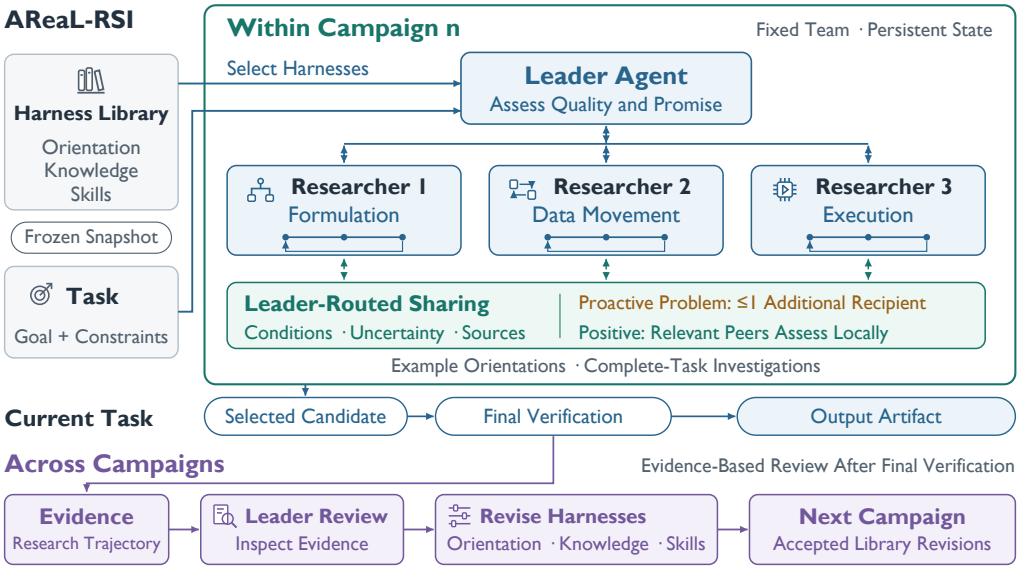


Figure 1: AReaL-RSI organizes RSI adaptive scaling. A leader agent selects task-conditioned harnesses and guides persistent researcher agents, each investigating the complete task. Shared experience retains sources, conditions, and uncertainty. After verification, the leader agent reviews the research trajectory and proposes revisions to orientation, knowledge, and skills. Accepted revisions become available to future campaigns.

Three key difficulties limit RSI from easily scaling out. First, early artifact quality is an incomplete signal of investigative promise: a weak implementation may contain a valuable idea, while an early improvement may depend on a fragile assumption; thus, simply redirecting every researcher agent toward the currently best-scoring direction can, therefore, abandon useful alternatives. Second, the experience is conditional: A finding that holds for one implementation or numerical regime may mislead another researcher agent when the conditions of the finding are omitted; broad sharing can also homogenize exploration (Ann et al., 2026). Third, progress on a task does not automatically improve the next research process: a completed investigation may contain reusable diagnostics or experimental procedures, but extracting them requires separating general methods from task-specific artifacts and uncertain explanations. These difficulties connect the leader agent's immediate coordination decisions to the longer-term reuse of research experience.

Existing approaches offer mechanisms for coordination, memory, and procedure evolution, but combining these capabilities still requires a policy for when to share evidence and how to revise reusable research procedures. AI co-scientist and Arbor organize collaborative investigation (Gottweis et al., 2025; Jin et al., 2026), while CoMem maintains private and collective memory (Yu et al., 2026). Population-based training and fluid research search adapt the allocation of parallel work (Jaderberg et al., 2017; Yang & Cao, 2026); AlphaEvolve maintains an evolutionary program archive, and Hyperagents and Dream-RSI revise aspects of the improvement process (Novikov et al., 2025; Zhang et al., 2026; Zheng et al., 2026). In this paper, our focus is the coupling between evidence interpretation, selective information exchange, and reuse of research procedures to scale out RSI.

Towards this end, we introduce AReaL-RSI to organize RSI on two timescales (Figure 1). Within a campaign (i.e., one execution iteration of AReaL-RSI on a specified task), a leader agent selects researcher harnesses suited to the task and provides evidence-driven guidance to persistent researcher agents. Conditional memory sharing links findings to their sources and tested conditions, limits proactive propagation of unresolved problems, and leaves recipients responsible for assessing applicability. After a campaign, evidence-based review revises research orientation, reusable knowledge, and operational skills for subsequent campaigns. These mechanisms address the three difficulties through explicit coordination and revision procedures. Our contributions are summarized as follows:

Contribution 1. We formulate adaptive RSI scaling as a problem of organizing persistent investigations and reusable research procedures under a declared resource allowance (§2). The formulation

distinguishes current artifact quality from investigative promise and separates the opportunity offered by additional computation from the efficiency of research team organization, which connects task-conditioned team composition and leader-agent guidance to adaptation within and across research campaigns.

Contribution 2. We develop conditional memory sharing (§3) and experience-driven researcher-harness evolution to support productive scaling (§4). Source-linked findings support selective propagation and recipient adaptation, while post-campaign review converts experimental experience into revisions of research orientation, knowledge, and skills. The mechanisms distinguish reuse among ongoing investigations from changes to the procedures available to future teams.

Contribution 3. We evaluate AREAL-RSI against CODEX-GOAL (OpenAI, 2026) across downstream harness optimization, GPU kernel optimization, and AI research tasks with two LLM backbones (§5). Under the reported resource allocations, AREAL-RSI increases Terminal Bench resolution by 2.2 percentage points for each backbone and achieves 32.2–36.8% higher mean KernelBench speedup. Further comparisons of scaling research configurations support AREAL-RSI 's coordinated scaling design: experience sharing with specialized researcher roles achieves 1.9–27.5% lower kernel latency than independent exploration and 1.3–33.9% lower latency than sharing alone across all seven comparable task–backbone pairs.

2 ADAPTIVE RSI SCALING

Scaling RSI requires organizing additional computation into useful investigations. We define the scaling objective, then describe team composition and leader-agent guidance. This organization supports sharing within campaigns and harness revision across campaigns (Figure 1).

2.1 WHY SCALE OUT RSI?

Parallel researcher agents can test competing hypotheses under costly or delayed feedback. However, duplicate work and indiscriminate sharing can waste computation or cause premature convergence. Effective scaling requires coordinating investigative priorities and information exchange alongside inference and experiment costs. Retaining useful procedures in the harnesses of a researcher agent creates a second opportunity: effort spent on one task can improve how later tasks are investigated.

Problem Formulation. A campaign is one execution of AREAL-RSI on a task q_n with a persistent objective and a fixed incoming harness-library snapshot. Let n index campaigns, π denote an admissible organization policy, and $x_n^{\mathrm{out}}(\pi)$ denote the output artifact. The final utility $U_n(x)$ is task-specific, with larger values preferred and invalid outputs assigned the task's failure value. For resource expenditure vector $\mathbf{C}_n^{\mathrm{total}}(\pi)$ and allowance $\mathbf{B}_n$, the objective is

$$\max_{\pi} \ \mathbb{E}\big[U_n\big(x_n^{\mathrm{out}}(\pi)\big)\big] \quad \text{subject to} \quad \mathbb{E}\big[\mathbf{C}_n^{\mathrm{total}}(\pi)\big] \preceq \mathbf{B}_n. \tag{1}$$

The expectation covers stochastic research and evaluation; $\preceq$ imposes componentwise resource constraints. The policy governs team selection, guidance, and sharing. Costs include researcher-agent execution, leader-agent inference, experiments, and post-campaign review (Equation 13); review benefits are assessed in later campaigns. This per-campaign objective defines a comparison rather than a runtime optimizer.

2.2 TASK-CONDITIONED TEAM COMPOSITION

Each campaign uses a fixed team of K researcher agents. The task q_n specifies the objective, admissible artifacts, constraints, and evaluation interface. Every researcher agent investigates the complete task and retains local state across interactions. Researcher harnesses provide complementary orientations, knowledge, and skills (§4.1).

Harness Selection. At campaign creation, the system snapshots the harness library as $\overline{\mathcal{L}}_n$. The leader agent selects harness assignments according to the task's constraints, unresolved questions, and verification cost:

$$(h_{n,1}, \ldots, h_{n,K}) \sim \pi_{\mathrm{select}}(\cdot \mid q_n, \overline{\mathcal{L}}_n), \qquad h_{n,i} \in \overline{\mathcal{L}}_n. \tag{2}$$

Here $h_{n,i}$ is researcher agent i's harness and π_{select} is the leader agent's selection policy. Multiple researcher agents may receive the same harness when an orientation merits additional exploration. Assigned versions remain fixed within the campaign, although local working files remain editable. Guidance adjusts investigative focus without replacing team members or dividing responsibility into disjoint subtasks.

2.3 Evidence-Driven Coordination

The leader agent uses progress reports and experimental records to guide the team. At coordination event t, let $\mathcal{E}_{n,t}$ denote the available public evidence and $g_{n,t}$ the leader agent's context: inspected evidence, the current plan, and reported resource availability. Actions follow

$$a^{\mathrm{L}}_{n,t} \sim \pi_{\text{coord}}(\cdot \mid q_n, g_{n,t}), \tag{3}$$

where π_{coord} uses model reasoning and tools to request evidence, revise the plan, advise a researcher agent, or await feedback.

Quality vs Promise. Coordination distinguishes *artifact quality* from *investigative promise*. Let $\mathcal{X}^{\text{ok}}_{n,t}$ contain saved, non-withdrawn candidates that passed the available checks, and let $s_n(x)$ be their recorded score, oriented so that larger values are better. For a nonempty candidate set, the incumbent is

$$x^{\star}_{n,t} \in \arg\max_{x \in \mathcal{X}^{\text{ok}}_{n,t}} s_n(x). \tag{4}$$

The incumbent records observed quality under the performed checks, without establishing correctness on untested conditions or identifying the most promising direction. An implementation error may obscure a useful idea, while an early gain may depend on narrow conditions. The leader agent weighs explanations, unresolved tests, and remaining resources before recommending continuation, diagnosis, redirection, or integration.

3 Conditional Memory Sharing between Researcher Agents

Shared findings are useful only when recipients can assess their applicability. The leader-routed path in Figure 1 preserves experimental context, limits the spread of unresolved problems, and supports local testing and corrective feedback.

3.1 Experience with Sources and Applicability Conditions

Shared memory comprises retrievable research records and communicated findings, distinct from local investigation histories and knowledge supplied by researcher harnesses. We represent the semantic content of an experience e as:

$$e = (d_e, f_e, c_e, u_e, \mathcal{F}_e), \tag{5}$$

where d_e is the observed result, f_e the proposed interpretation, c_e the tested conditions, u_e the remaining uncertainty, and $\mathcal{F}_e$ the supporting record references. Conditions describe inputs, acceptance criteria, and evaluation settings; references link to reports, artifacts, or execution traces. Separating observations from interpretations helps recipients assess whether the evidence supports the proposed explanation.

Each experience has a stable identifier, original source $o(e)$, current evidence source, and label $\kappa_t(e) \in \{\text{problem}, \text{positive}\}$. A problem denotes an unresolved failure, risk, or difficulty; a positive experience denotes a finding the sender considers supported and usable. Identifiers, labels, sources, and references are structured metadata; the semantic fields in Equation 5 remain message content. For example, a modification may reduce execution cost but fail a required check for some inputs. An omitted processing step is a possible explanation. Sharing both outcomes, the tested conditions, and that uncertainty supports diagnosis without prematurely rejecting or adopting the modification.

3.2 Selective Propagation of Problems and Solutions

Broadcasting an unresolved failure can draw the entire team into the same uncertain diagnosis. A supported solution can instead remove a shared obstacle. AREAL-RSI therefore restricts proactive problem routing while allowing positive findings to reach relevant researcher agents.

Problem Routing. For experience e, let $\mathcal{R}_{e,t}^{\text{problem}}$ contain researcher agents targeted by accepted, leader-initiated messages bearing the same identifier and the problem label through event t. The routing rule enforces

$$\left|\mathcal{R}_{e,t}^{\text{problem}} \setminus \{o(e)\}\right| \leq 1. \tag{6}$$

The leader agent can recruit one additional researcher agent with relevant expertise. Further problem updates remain confined to that recipient and the original source under the same identifier.

Positive Transfers. New evidence can support relabeling the experience as positive, allowing positive transfers without the additional-recipient bound. Let $\mathcal{F}_e^{\text{new}}$ be the proposed references and $\mathcal{F}_e^{\text{prev}}$ the references in the most recent registered update. Relabeling requires

$$\mathcal{F}_e^{\text{new}} \setminus \mathcal{F}_e^{\text{prev}} \neq \varnothing, \tag{7}$$

References must resolve within the campaign and include source-owned evidence for a researcher-agent source. These checks establish provenance and an additional reference, rather than certify the claimed solution. Interpretation, consistent classification, and semantic identity across retellings remain the leader agent's responsibility.

The bound governs proactive routing, not independent retrieval of public records or artifacts. Routine status summaries omit peer reports to avoid automatically broadcasting unresolved problems. Positive findings are also routed selectively according to relevance.

3.3 Recipient Adaptation and Corrective Feedback

Recipients compare shared findings with local conditions before adopting, adapting, rejecting, or deferring the advice. Candidates may be imported into a new subdirectory for local testing. Advice does not overwrite existing work or interrupt experiments. Let $z_{n,i,t}$ be researcher agent i's investigation state, $y_{n,i,t}$ the new local observations, $\mathcal{D}_{n,i,t}$ the delivered messages, and $\mathcal{P}_{n,i,t}$ the explicitly retrieved public records. Local reasoning updates the state as

$$z_{n,i,t+1} = \text{Update}_i\big(z_{n,i,t}, y_{n,i,t}, \mathcal{D}_{n,i,t}, \mathcal{P}_{n,i,t}\big). \tag{8}$$

The update incorporates available evidence without merging all team histories or requiring agreement. Applicability assessment and local verification are prescribed behavior, not an enforced adoption gate.

Corrective Feedback. In the example above, a recipient might restore the omitted step and test whether the cost reduction survives while satisfying the required checks. Reporting the result and tested conditions under the same experience identifier allows the leader agent to qualify the finding and revise subsequent sharing. Accumulated findings also inform the harness revisions in §4.

4 Experience-Driven Evolution of Researcher Harnesses

Harness evolution turns completed investigations into procedures available to future teams. The across-campaign path in Figure 1 links three steps: represent reusable research procedures, revise those procedures from evidence, and publish revisions for subsequent campaigns.

4.1 Evolvable Research Orientation, Knowledge, and Skills

A *researcher harness* is the external configuration guiding an investigation:

$$h = (p, \mathcal{K}, \mathcal{S}), \tag{9}$$

Research orientation p guides which hypotheses to pursue. Knowledge $\mathcal{K}$ records findings, applicability conditions, and failure boundaries. Skills $\mathcal{S}$ specify repeatable procedures, including inputs, steps, validation criteria, and supporting scripts. Together, these components influence what a researcher agent investigates, what evidence is available, and how experiments are conducted.

For the example in §3.1, knowledge can retain the conditions under which a modification failed, while a skill adds targeted checks before full evaluation. The reusable outcome is a better-informed procedure for investigating future candidates.

The implementation packages the harness as a *profile*, supplied through initial context and workspace resources. Revisions affect these assets while model weights, tool permissions, orchestration, and the leader agent's policy remain unchanged.

4.2 From Experimental Experience to Procedural Revision

After exploration and applicable final verification, the leader agent enters a separate review session. The review examines the task, selected harnesses, reports, and retrievable campaign records, with instructions to propose reusable revisions without new experiments.

Evidence-Based Revision. Let $\mathcal{H}_n = \{h_{n,i} : 1 \le i \le K\}$ denote the distinct harnesses used in campaign n, $\mathcal{E}_n$ the completed public evidence, and v_n the finalization feedback. The leader agent's review policy π_{review} proposes edits Δ_n:

$$\Delta_n \sim \pi_{\text{review}}(\cdot \mid q_n, \mathcal{H}_n, \mathcal{E}_n, v_n), \qquad h'_j = \text{Apply}(h_j, \Delta_{n,j}). \tag{10}$$

Here j identifies a selected harness and $\Delta_{n,j}$ contains the corresponding changes. Evidence is accessed through retrieval; an empty proposal is valid when no reusable revision is justified.

Review asks which orientations helped, which findings generalize beyond the task, and which repeated operations can become skills. Proposals should retain applicability conditions, uncertainty, and contradictory evidence. Instructions exclude complete task answers and hidden-test information from reusable assets; this is a semantic requirement rather than a formal guarantee.

Each proposal supplies a rationale, same-campaign evidence references, and asset additions or replacements. Updates preserve unspecified assets and remain restricted to selected harness identities. Publication checks references, structure, and required resources; acceptance requires neither script execution nor demonstrated improvement on future tasks.

4.3 Reuse and Specialization in Subsequent Campaigns

Versioned Publication. Accepted revisions update the shared library used to initialize new campaigns. Let $\mathcal{L}^{\text{cur}}$ be the library before publication and $\mathcal{H}'_n$ the proposed revised harnesses. Replacement requires admissible revisions whose current versions still match those reviewed:

$$\mathcal{L}^{+} = \begin{cases} \text{Replace}(\mathcal{L}^{\text{cur}}, \mathcal{H}'_n), & \text{if admissible and version-consistent,} \\ \mathcal{L}^{\text{cur}}, & \text{otherwise.} \end{cases} \tag{11}$$

The version check prevents stale reviews from overwriting newer changes. Failed or conflicting proposals leave the library unchanged.

For campaigns created after the preceding review, the adaptation loop is

$$\overline{\mathcal{L}}_n \xrightarrow{\text{select}} \mathcal{H}_n \xrightarrow{\text{research and share}} \mathcal{E}_n \xrightarrow{\text{review and publish}} \overline{\mathcal{L}}_{n+1}. \tag{12}$$

Earlier campaigns retain their snapshots. New teams select revised harnesses through Equation 2, without inheriting unrestricted access to past traces or complete answers. Specialization combines revisions to existing harness identities with task-conditioned selection; the update operator does not create new identities.

Transfer Assessment. A harness revision should improve future investigations, whereas an artifact is evaluated on the current task. Assessing transfer therefore requires later campaigns with declared task splits, library versions, and feedback permissions. Source-campaign finalization feedback may inform review, but target held-out outcomes must remain outside updates used to solve that target.

Resource Accounting. Review also consumes resources. With $\mathbf{C}_n^{\text{campaign}}$ covering coordination, researcher-agent execution, and verification, and $\mathbf{C}_n^{\text{review}}$ covering post-campaign review, total expenditure is

$$\mathbf{C}_n^{\text{total}} = \mathbf{C}_n^{\text{campaign}} + \mathbf{C}_n^{\text{review}}. \tag{13}$$

These vectors contain additive costs such as tokens, model charges, and experiment compute; elapsed wall time is reported separately. The review phase has a separate allowance and must be included when measuring the benefit of reuse.

5 EVALUATION

We evaluate whether AREAL-RSI translates parallel researcher-agent computation into better research outcomes. The evaluation addresses two research questions: (**RQ1**) *How does* AREAL-RSI*'s end-to-end performance compare with the popular* CODEX-GOAL *baseline?* (Section 5.2); and (**RQ2**) *How do the proposed mechanisms improve the scalability of parallel research?* (Section 5.3). We first specify the tasks and evaluation settings (Section 5.1), then report the results for the research questions.

5.1 EXPERIMENTAL SETUP

Tasks. We evaluate three task families: downstream Terminus harness optimization on Terminal Bench 2.1 (89 tasks, with the task-solving model fixed); GPU kernel optimization on seven KernelBench tasks (Ouyang et al., 2025); and four RSI Bench research tasks covering on-policy self-distillation, jailbreak robustness, agent-swarm optimization, and nanoGPT data curation. The KernelBench tasks are L1P63, L1P77, L2P14, L2P71, L3P10, L3P19, and L3P28.

Methods and Resources. We compare AREAL-RSI and CODEX-GOAL using GPT-5.6-Luna and GPT-6-Astra as research backbones. KernelBench allows one effective hour for CODEX-GOAL and for each AREAL-RSI researcher agent; RSI Bench allows two effective hours for CODEX-GOAL and for each AREAL-RSI researcher agent. RQ2 compares independent four-agent exploration, experience sharing, and sharing with specialized researcher roles.

Evaluation Protocol. For Terminal Bench, AREAL-RSI iterates and selects on smaller training and validation subsets, whereas CODEX-GOAL evaluates the full task set each round. Both selected harnesses are independently reevaluated on all 89 tasks. KernelBench specifies seed 137 for independent final reevaluation. Each candidate was evaluated on a single, exclusively allocated GPU, with warm-up followed by 100 repeated timing measurements using CUDA events.

Metrics. Terminal Bench reports resolution rate. KernelBench reports arithmetic mean speedup across seven tasks, assigning zero to unsuccessful tasks; speedup is reference runtime divided by candidate runtime. RSI Bench uses the task-specific metrics in Table 1, with larger values preferred except for perplexity. Configuration comparisons report kernel latency in milliseconds. *Fail* means no correct final candidate. Comparisons are descriptive; repeated-run uncertainty is unavailable.

5.2 END-TO-END PERFORMANCE COMPARED WITH CODEX-GOAL

Comparison. Table 1 addresses RQ1 within each research backbone. AREAL-RSI improves all six reported metrics with both Luna and Astra, giving 12 favorable comparisons. The consistent direction spans downstream harnesses, kernels, and AI research, while the size of the gains varies by task and backbone.

Table 1: End-to-end research outcomes under the reported resource allocations. Bold identifies the better available numerical value within each backbone. Arrows indicate the preferred direction.

	Terminal Bench 2.1	KernelBench	RSI Bench			
Method	Resolved (%) ↑	Mean speedup (×) ↑	Self-distill. score (%) ↑	Jailbreak score ↑	Swarm accuracy (%) ↑	Data curation PPL ↓
GPT-5.6-Luna						
CODEX-GOAL	75.3	1.74	44.17	11.251	53.33	571.534
AREAL-RSI	**77.5**	**2.30**	**55.83**	**35.544**	**66.67**	**422.837**
GPT-6-Astra						
CODEX-GOAL	85.4	6.28	51.94	81.142	60.00	377.351
AREAL-RSI	**87.6**	**8.59**	**57.50**	**81.261**	**73.33**	**340.523**

Downstream Harness Optimization. Terminal Bench resolution increases from 75.3% to 77.5% with Luna and from 85.4% to 87.6% with Astra, a gain of 2.2 percentage points for each backbone.

Because the task-solving model is fixed, these gains reflect changes to the downstream harness produced by each research system. This experiment is distinct from evolving the researcher harnesses in Section 4.

GPU Kernel Optimization. Mean speedup across seven KernelBench tasks increases from $1.74\times$ to $2.30\times$ with Luna and from $6.28\times$ to $8.59\times$ with Astra, relative gains of 32.2% and 36.8%. Improvements with both backbones show that the advantage extends beyond one research model. Because unsuccessful tasks contribute zero, the aggregate combines candidate correctness and execution performance; the summary does not separate their contributions.

AI Research Tasks. Self-distillation scores improve by 11.66 percentage points with Luna and 5.56 with Astra. Swarm accuracy improves by 13.34 and 13.33 percentage points, respectively, while data-curation perplexity decreases by 26.0% and 9.8%. Jailbreak scores improve by 24.293 with Luna but only 0.119 with Astra. These gains span all four research objectives, although the Astra jailbreak difference is small.

RQ1 Key Takeaway. Under the reported resource allocations, AREAL-RSI outperforms CODEX-GOAL on all six metrics with both backbones. The results support end-to-end effectiveness across tasks, without establishing equal-compute efficiency.

5.3 COMPARISON OF PARALLEL RESEARCH CONFIGURATIONS

Comparison. Table 2 addresses RQ2 by comparing independent four-agent exploration, experience sharing, and sharing with specialized researcher roles. The comparison covers four KernelBench tasks and two backbones. All three configurations produce valid candidates for the same seven task–backbone pairs; Luna's L3P10 fails under every configuration.

Table 2: Comparison of parallel research configurations on KernelBench. Entries are final candidate latencies in milliseconds (↓). Bold marks the lowest latency for each task–backbone pair. Fail means no correct final candidate.

	GPT-6-Astra				**GPT-5.6-Luna**			
Configuration	L1P63	L2P71	L3P10	L3P19	L1P63	L2P71	L3P10	L3P19
Independent (four agents)	5.200	0.338	8.815	0.482	11.944	0.523	*Fail*	1.766
Experience sharing	4.731	0.294	7.379	0.479	15.633	0.593	*Fail*	1.579
Sharing + researcher roles	**4.665**	**0.251**	**6.395**	**0.473**	**10.498**	**0.392**	*Fail*	**1.407**

Sharing Alone. Experience sharing lowers latency in five of seven comparable pairs: all four Astra tasks and one of the three successful Luna tasks. However, Luna's L1P63 latency rises from 11.944 to 15.633 ms (30.9%), and L2P71 rises from 0.523 to 0.593 ms (13.4%). The sharing configuration therefore yields better or worse final artifacts depending on the task and backbone.

Sharing with Specialization. The combined configuration achieves the lowest latency in all seven comparable pairs, reducing latency by 1.9–27.5% relative to independent exploration and by 1.3–33.9% relative to sharing alone. On Astra's L3P10, the three configurations yield 8.815, 7.379, and 6.395 ms, respectively. On Luna's L2P71, the combined configuration records 0.392 ms despite the sharing-only regression, 25.0% below independent exploration and 33.9% below sharing alone. The combined configuration records lower latency both where sharing alone improves on independent exploration and where sharing alone performs worse.

Discussion. The pattern favors organizing information reuse together with differentiated investigative roles, rather than relying on information exchange alone. This interpretation is consistent with task-conditioned harness selection and conditional memory sharing, but the comparisons do not isolate individual mechanism effects or measure research-direction diversity. The study evaluates configurations at the reported parallelism; scaling curves, matched-compute efficiency, and gains from cross-campaign harness evolution remain unmeasured.

RQ2 Key Takeaway. Sharing with specialized researcher roles gives the best result in all seven comparable task-backbone pairs, whereas sharing alone has mixed outcomes. The evidence favors the combined configuration at the evaluated parallelism.

6 Related Work

RSI for Agent Evolution. Agent evolution ranges from optimizing agent designs and prompts to revising the improvement process. ADAS searches over agent designs (Hu et al., 2024), the Darwin Gödel Machine evolves coding-agent programs (Zhang et al., 2025b), and GEPA evolves prompts through reflection (Agrawal et al., 2025). DGM keeps pretrained model weights fixed, locating improvement in the agent's executable procedures (Zhang et al., 2025b). Hyperagents makes the self-modification procedure editable (Zhang et al., 2026), while Dream-RSI refines exploration policies using replay over historical discovery trees (Zheng et al., 2026). τ^{τ}-Bench evaluates agent construction and subsequent operating cost (Shi et al., 2026).

Autonomous Research and Agentic Search. AIDE and AI Scientist-v2 organize program development and experimentation through tree search (Jiang et al., 2025; Yamada et al., 2025); AlphaEvolve and Loreley maintain evolutionary archives of programs and repository states, respectively (Novikov et al., 2025; Chen, 2026). Retaining earlier candidates supports both continued refinement and exploration from alternative starting points. Population Based Training reallocates workers through parameter inheritance and hyperparameter perturbation (Jaderberg et al., 2017). Fluid search allocates effort and reseeds stalled research chains (Yang & Cao, 2026), ExTS balances expansion and refinement (Fang & Wang, 2026), and Research Preference Models screen proposals before expensive evaluation (Foster et al., 2026). RE-Bench and MLE-bench evaluate research engineering (Wijk et al., 2024; Chan et al., 2024); KernelBench, Scale RSI Bench, and AutoWorldModel-Bench provide tasks in kernel optimization, AI research, and world modeling (Ouyang et al., 2025; Mahmoud et al., 2026; Moodi et al., 2026).

Agent Coordination and Memory Sharing. AI co-scientist and Arbor coordinate specialized agents and hypothesis-driven investigations (Gottweis et al., 2025; Jin et al., 2026); EurekAgent supports collaboration through filesystem and Git artifacts (Xin et al., 2026), and BusMA combines worker-local memory with message-driven coordination (Peng et al., 2026). Persistent records allow accumulated evidence to remain accessible after an individual agent session ends. Stateful ReAct maintains typed persistent state (Jabbarvaziri, 2026), G-Memory retrieves hierarchical experience (Zhang et al., 2025a), MemCon controls memory operations (Jiang et al., 2026), and CoMem combines private and curated collective memory (Yu et al., 2026). Materials-science memory and Repo-To-Skill organize reusable knowledge and executable skills (Liu et al., 2026; Chen et al., 2026). EdgeBench and Beyond Final Scores study the effects of retained experience on continued research (Zhu et al., 2026; Li et al., 2026). Independent-agent aggregation provides a scaling baseline (Li et al., 2024), while coordination benefits depend on task structure and overhead (Kim et al., 2025). Sharing can also homogenize proposals (Ann et al., 2026) or propagate evaluator exploits (Paglieri et al., 2026).

7 Conclusion

In this paper, we introduced AReaL-RSI, which formulates test-time RSI scaling as the organization of persistent investigations and reusable research procedures. The framework combines conditional memory sharing within campaigns with evidence-driven harness evolution across campaigns. Under the evaluated resource allocations, AReaL-RSI improves Terminal Bench resolution by 2.2 percentage points and mean KernelBench speedup by 32.2–36.8% over Codex-Goal. Combining experience sharing with specialized researcher roles achieves 1.9–27.5% lower kernel latency than independent exploration and 1.3–33.9% lower latency than sharing alone across all seven comparable task–backbone pairs. These results support coordinated parallel research, while matched-compute scaling and the benefit of cross-campaign harness evolution remain to be established.

AI USE STATEMENT

The authors used large language model-based tools to assist with drafting, revising, and polishing the manuscript, improving its organization, and preparing LaTeX formatting. These tools also supported research ideation and execution, including exploring directions for analysis, organizing and presenting experimental findings, preparing visualizations, and checking statistical results. All AI-assisted outputs were reviewed and verified by the authors, who made the final decisions regarding the technical content, experiments, analyses, and presentation. The authors take full responsibility for the accuracy and content of the paper.

REFERENCES

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alexandros G. Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. GEPA: Reflective Prompt Evolution Can Outperform Reinforcement Learning, 2025. URL https://arxiv.org/abs/2507.19457v2. arXiv:2507.19457v2.

Summer Eunhyung Ann, Haokun Liu, and Chenhao Tan. The Interaction Tax: When Communication Erases Diversity in Multi-Agent Teams, 2026. URL https://arxiv.org/abs/2608.23541v1. arXiv:2608.23541v1.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Lilian Weng, and Aleksander Mądry. MLE-bench: Evaluating Machine Learning Agents on Machine Learning Engineering, 2024. URL https://arxiv.org/abs/2410.07095v6. arXiv:2410.07095v6.

Jianlyu Chen, Yuyang Hu, Hongjin Qian, Jiawei Liu, Wenqing Wei, Xiaolong Chen, Defu Lian, Zhicheng Dou, Chaozhuo Li, Qiwei Ye, and Zheng Liu. Repo-To-Skill: Distilling GitHub Repositories Into AI4AI Skills, 2026. URL https://arxiv.org/abs/2609.02749v1. arXiv:2609.02749v1.

Mohan Chen. Loreley: Repository-Scale Program Evolution with Quality-Diversity Search, 2026. URL https://arxiv.org/abs/2608.19703v1. arXiv:2608.19703v1.

Haoyang Fang and Bernie Wang. Exploit More, Explore Smarter for Budget-Constrained Agentic Search, 2026. URL https://arxiv.org/abs/2608.23848v1. arXiv:2608.23848v1.

Thomas Simon Foster, Bassel Al Omari, Tingchen Fu, Thomas Mann, Carl Domond, Lucia Cipolina-Kun, Bhavul Gauri, Muna Aghamelu, Alexander D. Goldie, Eryk Helenowski, Jean-Christophe Gagnon-Audet, Alberto Pepe, Saba Nazir, Daniel Izcovich, Noam Levi, Rishi Hazra, Karen Hambardzumyan, Nicolas Baldwin, Xian Li, Martin Josifoski, Paris Giampouras, Masoud Jalili Sabet, Anya Sims, Hela Momand, Tatiana Shavrina, Despoina Magka, Jason Weston, Yulin Wang, Anirudh Goyal, João Henriques, Yoram Bachrach, Emily McMilin, and Jakob Nicolaus Foerster. AI Research Preference Models, 2026. URL https://arxiv.org/abs/2608.13940v2. arXiv:2608.13940v2.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Anil Palepu, Petar Sirkovic, Artiom Myaskovsky, Felix Weissenberger, Keran Rong, Ryutaro Tanno, Khaled Saab, Dan Popovici, Jacob Blum, Fan Zhang, Katherine Chou, Avinatan Hassidim, Burak Gokturk, Amin Vahdat, Pushmeet Kohli, Yossi Matias, Andrew Carroll, Kavita Kulkarni, Nenad Tomasev, Yuan Guan, Vikram Dhillon, Eeshit Dhaval Vaishnav, Byron Lee, Tiago R D Costa, José R Penadés, Gary Peltz, Yunhan Xu, Annalisa Pawlosky, Alan Karthikesalingam, and Vivek Natarajan. Towards an AI co-scientist, 2025. URL https://arxiv.org/abs/2502.18864v1. arXiv:2502.18864v1.

Shengran Hu, Cong Lu, and Jeff Clune. Automated Design of Agentic Systems, 2024. URL https://arxiv.org/abs/2408.08435v2. arXiv:2408.08435v2.

Faramarz Jabbarvaziri. Remember, Don't Re-read: Stateful ReAct Agents for Token-Efficient Autonomous Experimentation, 2026. URL https://arxiv.org/abs/2606.14945v1. arXiv:2606.14945v1.

Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M. Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, Chrisantha Fernando, and Koray Kavukcuoglu. Population Based Training of Neural Networks, 2017. URL https://arxiv.org/abs/1711.09846v2. arXiv:1711.09846v2.

Eric Hanchen Jiang, Zhi Zhang, Yuchen Wu, Levina Li, Dong Liu, Xiao Liang, Rui Sun, Yubei Li, Edward Sun, Haozheng Luo, Zhaolu Kang, Aylin Caliskan, Kai-Wei Chang, and Ying Nian Wu. Memory as a Controlled Process: Learned Adaptive Memory Management for LLM Agents, 2026. URL https://arxiv.org/abs/2607.13591v1. arXiv:2607.13591v1.

Zhengyao Jiang, Dominik Schmidt, Dhruv Srikanth, Dixing Xu, Ian Kaplan, Deniss Jacenko, and Yuxiang Wu. AIDE: AI-Driven Exploration in the Space of Code, 2025. URL https://arxiv.org/abs/2502.13138v1. arXiv:2502.13138v1.

Jiajie Jin, Yuyang Hu, Kai Qiu, Qi Dai, Chong Luo, Guanting Dong, Xiaoxi Li, Tong Zhao, Xiaolong Ma, Gongrui Zhang, Zhirong Wu, Bei Liu, Zhengyuan Yang, Linjie Li, Lijuan Wang, Hongjin Qian, Yutao Zhu, and Zhicheng Dou. Toward Generalist Autonomous Research via Hypothesis-Tree Refinement, 2026. URL https://arxiv.org/abs/2606.11926v1. arXiv:2606.11926v1.

Yubin Kim, Ken Gu, Chanwoo Park, Chunjong Park, Samuel Schmidgall, A. Ali Heydari, Yao Yan, Zhihan Zhang, Yuchen Zhuang, Yun Liu, Mark Malhotra, Paul Pu Liang, Hae Won Park, Yuzhe Yang, Xuhai Xu, Yilun Du, Shwetak Patel, Tim Althoff, Daniel McDuff, and Xin Liu. Towards a Science of Scaling Agent Systems, 2025. URL https://arxiv.org/abs/2512.08296v3. arXiv:2512.08296v3.

Junyou Li, Qin Zhang, Yangbin Yu, Qiang Fu, and Deheng Ye. More Agents Is All You Need, 2024. URL https://arxiv.org/abs/2402.05120v2. arXiv:2402.05120v2.

Yiwei Li, Wanli Yang, Hexiang Tan, Xiangzhou Huang, Zhengyu Chen, Ziran Li, Borun Chen, Shanglin Lei, Huaisheng Zhu, Hao Tian, Fei Sun, Xunliang Cai, and Jingang Wang. Beyond Final Scores: A Systematic Evaluation of Agents for Long-Horizon AI Research and Development, 2026. URL https://arxiv.org/abs/2608.13417v1. arXiv:2608.13417v1.

Siyu Liu, Bo Hu, Beilin Ye, He Cao, David J. Srolovitz, and Tongqi Wen. Harnessing agent memory to build lifelong AI partners for materials scientists, 2026. URL https://arxiv.org/abs/2608.11224v1. arXiv:2608.11224v1.

Anas Mahmoud, MohammadHossein Rezaei, Mingchen Zhuge, Chenguang Wang, Yuandong Tian, and Yunzhong He. Announcing RSI Bench, August 2026. URL https://labs.scale.com/rsi-benchmark/blog/announcing-rsi-bench. Official announcement, August 7, 2026.

Marjan Moodi, Xuankang Zhu, Fernando De Mesentier Silva, Harold Chaput, and Mohammad Reza Taesiri. AutoWorldModel-Bench: A State-Centric Benchmark for Automated World-Model Research, 2026. URL https://arxiv.org/abs/2608.11216v2. arXiv:2608.11216v2.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco J. R. Ruiz, Abbas Mehrabian, M. Pawan Kumar, Abigail See, Swarat Chaudhuri, George Holland, Alex Davies, Sebastian Nowozin, Pushmeet Kohli, and Matej Balog. AlphaEvolve: A coding agent for scientific and algorithmic discovery, 2025. URL https://arxiv.org/abs/2506.13131v1. arXiv:2506.13131v1.

OpenAI. Using goals in codex. https://developers.openai.com/cookbook/examples/codex/using_goals_in_codex, 2026. Accessed Sep 25, 2026.

Anne Ouyang, Simon Guo, Simran Arora, Alex L. Zhang, William Hu, Christopher Ré, and Azalia Mirhoseini. KernelBench: Can LLMs Write Efficient GPU Kernels?, 2025. URL https://arxiv.org/abs/2502.10517v1. arXiv:2502.10517v1.

Davide Paglieri, Logan Cross, Tim Genewein, Joel Z. Leibo, Nenad Tomasev, and Alexander Sasha Vezhnevets. A Case Study on Emergent Cheating and Whistleblowing in Autonomous Research Swarms, 2026. URL https://arxiv.org/abs/2609.04170v1. arXiv:2609.04170v1.

Yanwen Peng, Delvin Ce Zhang, Xi Wang, and Nikolaos Aletras. BusMA: A Bus Communication Substrate for Multi-Agent Systems, 2026. URL https://arxiv.org/abs/2609.15054v1. arXiv:2609.15054v1.

Quan Shi, Keshav Dhandhania, Karthik Narasimhan, and Victor Barres. τ^{τ}-Bench: An Environment for End-To-End, Realistic Agent Construction, 2026. URL https://arxiv.org/abs/2609.04611v1. arXiv:2609.04611v1.

Hjalmar Wijk, Tao Lin, Joel Becker, Sami Jawhar, Neev Parikh, Thomas Broadley, Lawrence Chan, Michael Chen, Josh Clymer, Jai Dhyani, Elena Ericheva, Katharyn Garcia, Brian Goodrich, Nikola Jurkovic, Holden Karnofsky, Megan Kinniment, Aron Lajko, Seraphina Nix, Lucas Sato, William Saunders, Maksym Taran, Ben West, and Elizabeth Barnes. RE-Bench: Evaluating frontier AI R&D capabilities of language model agents against human experts, 2024. URL https://arxiv.org/abs/2411.15114v2. arXiv:2411.15114v2.

Amy Xin, Jiening Siow, Junjie Wang, Zijun Yao, Fanjin Zhang, Jian Song, Lei Hou, and Juanzi Li. EurekAgent: Agent Environment Engineering is All You Need For Autonomous Scientific Discovery, 2026. URL https://arxiv.org/abs/2606.13662v2. arXiv:2606.13662v2.

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, 2025. URL https://arxiv.org/abs/2504.08066v1. arXiv:2504.08066v1.

Haiqian Yang and Yuan Cao. Efficiency Matters in Autonomous Research, 2026. URL https://arxiv.org/abs/2607.24647v1. arXiv:2607.24647v1.

Chengxin Yu, Zhaoxin Fan, Faguo Wu, Hongwei Zheng, Yun Zhou, and Zhiyu Li. CoMem: Collective-Individual Memory Synergy for Evolutionary Multi-Agent Systems, 2026. URL https://arxiv.org/abs/2609.15009v1. arXiv:2609.15009v1.

Guibin Zhang, Muxin Fu, Guancheng Wan, Miao Yu, Kun Wang, and Shuicheng Yan. G-Memory: Tracing Hierarchical Memory for Multi-Agent Systems, 2025a. URL https://arxiv.org/abs/2506.07398v1. arXiv:2506.07398v1.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin Godel Machine: Open-Ended Evolution of Self-Improving Agents, 2025b. URL https://arxiv.org/abs/2505.22954v3. arXiv:2505.22954v3.

Jenny Zhang, Bingchen Zhao, Wannan Yang, Jakob Foerster, Jeff Clune, Minqi Jiang, Sam Devlin, and Tatiana Shavrina. Hyperagents, 2026. URL https://arxiv.org/abs/2603.19461v1. arXiv:2603.19461v1.

Tong Zheng, Xidong Wu, Zheng Zhang, Zhankui He, Chaoyi Zhang, Benjamin Coleman, Ruoqiao Wei, Di Bai, Haolin Liu, Rui Liu, Xue Wang, Yue Zhuan, Wang-Cheng Kang, Renkai Xiang, Heng Huang, Xinwu Cheng, and Yunsong Guo. Dream-RSI: Recursive Self-Improvement through Evolving Worlds, 2026. URL https://arxiv.org/abs/2609.14858v1. arXiv:2609.14858v1.

Deyao Zhu, Xin Zhou, Shengling Qin, Xuekai Zhu, Hangliang Ding, Shu Zhong, Zixin Wen, Zhonglin Xie, Chenhui Gou, Linxuan Ren, Yueyang Wang, Junfeng Zhong, Rui Liu, Tian Gao, Yangguang Lin, Jingyuan Zhang, Maojia Song, Xuan Qi, Jinhong Wu, Chenyang Zhang, Yinzhu Piao, Ziru Niu, Hongbin Lin, Lingxiang Meng, Peng Tang, Chengyao Tang, Shanyu Wu, Huanyu Zheng, Yu Liu, Liya Zhu, He Wang, Ming Ding, Ziyu Wan, Hao Liu, Sibo Wang, Haotian Zhu, Xintian Zhang, Nan Chai, Yipeng Liu, Panhao Lai, Sihang Yuan, Zixin Su, Ge Zhang, Wangchunshu Zhou, Yantao Du, Wenhao Huang, and Guang Shi. EdgeBench: Unveiling Scaling Laws of Learning from Real-World Environments, 2026. URL https://arxiv.org/abs/2607.05155v1. arXiv:2607.05155v1.