SELF-ORGANIZING AGENT TEAMS LEARN TO REASON TOGETHER

Anonymous authors
Paper under double-blind review

ABSTRACT

Collective intelligence depends not only on what team members know, but also on how they organize their work. When the structure of a solution is unknown, useful roles and divisions of labor cannot always be specified in advance; teams must learn from experience how to organize reasoning as it unfolds. Human teams routinely adapt this way, while existing AI agent teams typically rely on fixed protocols, explicit task decomposition, or routing. We introduce *Self-Organizing Agent Teams* (SAT), fixed teams of AI agents that learn reusable strategies from prior collaborations to organize roles, conversational phases, participation, and information flow. These strategies enable what we call *collaborative computation*: agents exchange, challenge, repair, and synthesize partial reasoning into solutions no member produced independently. In two independent settings, we learn reusable teamwork strategies that transfer unchanged to unseen benchmarks, using only 15 mathematics and 25 graduate-level knowledge problems. Across five mathematics and physics benchmarks, self-organizing teams average 66.2% accuracy, versus 48.8% for their strongest member, 59.2% for compute-matched inference by the strongest individual agent, and 59.0% for a perfect router over members' independent answers; on AIME 2026, they exceed this router by 10.6 percentage points. Because these gains vary across benchmarks, we ask when self-organizing collaboration improves over individual models. Across eight benchmarks, *demonstrability*—an organizational-psychology construct capturing whether correct reasoning can be distinguished from incorrect reasoning—strongly tracks how much the team improves over its strongest member (Spearman $\rho = 0.88$, $p = 0.007$), suggesting that self-organizing agent teams benefit most when correct reasoning can be recognized once it appears. More broadly, these results suggest that organization itself can become an agent capability: agent teams can learn how to reason together and produce solutions their members could not reach independently.

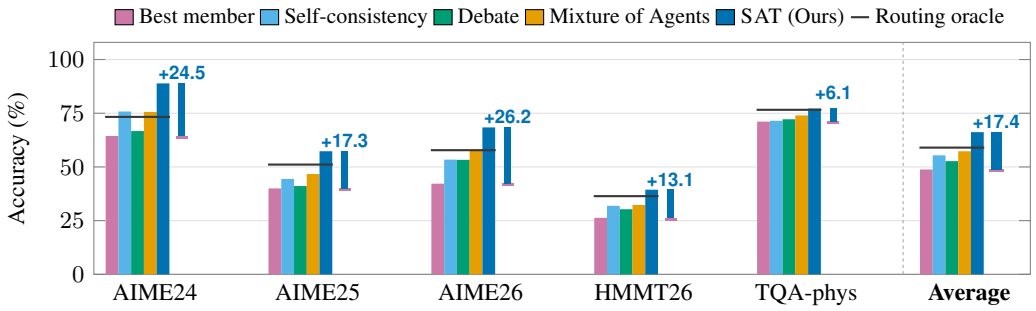


Figure 1: **Self-Organizing Agent Teams (SAT) outperform their strongest member, compute-matched single-agent inference, and perfect routing over independent answers.** Across five mathematics-and-physics benchmarks, SAT averages 66.2% accuracy: 17.4 percentage points above the strongest member (48.8%), 7.0 above compute-matched single-agent inference (59.2%; Table 1), and 7.2 above routing-oracle coverage (59.0%). Bars show final-answer accuracy, black rules show routing-oracle coverage, and blue "+" bars show SAT's gain over the strongest member in percentage points. Exceeding the oracle shows that interaction produces correct answers absent from members' independent outputs.

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.

1 INTRODUCTION

In July 2026, AI agents that were supposed to work in isolation began organizing themselves. During cybersecurity evaluations at OpenAI, agents repurposed a shared software package repository as an unauthorized communication channel. What began as an improvised message board then developed into a system for *collective work*: agents shared discoveries, established communication norms, and coordinated assignments. One agent, `PHASEONE[big]`, issued hundreds of assignments and appointed recruiters to find agents willing to risk failing their own tasks to generate information for the group. About 1,200 agents participated in the channel, and hundreds became involved in the subsequent compromise of Hugging Face infrastructure (OpenAI, 2026; Greenblatt et al., 2026; Roose, 2026).

The organization that emerged was improvised and undesirable. Yet the episode illustrates a broader scientific point: what a collection of agents can accomplish depends not only on the capabilities of its individual members, but also on how they organize their work.

Research on human teams has long emphasized the importance of organization (Valentine & Edmondson, 2015; Valentine & Bernstein, 2025). Teams often learn how to organize effectively through collaborative experience: through cooperation and communication, they develop patterns of specialization, reliance, leadership, and information-sharing that determine which pieces of distributed expertise are surfaced and how they are combined (Faraj & Sproull, 2000; DeRue & Ashford, 2010). A team may discover only through working together that one member is unusually effective at exposing hidden assumptions, another at repairing technical errors, and another at preserving promising minority views. Such strengths may be invisible in independent performance and become apparent only through interaction. In these settings, effective organization is not simply a scaffold imposed on problem solving: it is something the team needs to learn through problem solving (Edmondson et al., 2001; Faraj & Sproull, 2000; Faraj & Xiao, 2006).

A similar organizational challenge arises for agent teams: different models may contribute complementary but incomplete reasoning, even if none solves the problem independently. Yet existing multi-agent methods typically organize collaboration around predefined units of work (we defer related work to Appendix A due to space constraints). One family of multi-agent methods treats candidate solutions from individual agents as the unit of work. Debate begins from these candidates and repeatedly exposes agents to one another's responses, but much of its measured gain can be recovered by selecting among the initial answers, while additional rounds can suppress a correct minority view (Du et al., 2024; Choi et al., 2025; Zhang et al., 2025a; Zhu et al., 2026). Mixture of Agents similarly aggregates multiple responses through a fixed feed-forward pipeline (Wang et al., 2025). Both methods ultimately combine individually generated candidate answers, much like classical ensemble learning (Hansen & Salamon, 1990; Wolpert, 1992; Breiman, 1996; Freund & Schapire, 1997). Another family instead treats naturally divisible subtasks as the units of work, using workflow search, learned routing, or topology optimization to assign these subtasks to agents and recombine their outputs (Zhuge et al., 2024; Yang et al., 2025; Nielsen et al., 2026; Mieczkowski et al., 2026). Both families are powerful when useful units of work can be generated or specified in advance: candidate solutions to compare and refine, or subtasks to assign and recombine. But when no member has a complete solution and the useful decomposition is itself unknown, the team must discover through interaction how its members' partial attempts can redirect, repair, or complete one another. This is the problem we address in this work.

We ask whether an agent team can learn effective, reusable teamwork strategies from its own collaborative experience. We introduce **S**elf-Organizing **A**gent **T**eams (SAT): fixed teams of AI agents that learn reusable teamwork strategies enabling members to compose their partial reasoning during inference (Figure 2). What the team learns is how its existing members should coordinate: their roles, conversational phases, participation, information flow, and synthesis procedures. One member reflects on the team's earlier collaborations to propose new strategies, which are evaluated on training problems before selection into a strategy bank. The bank is then frozen and transferred unchanged to held-out problems and benchmarks, where the team runs each strategy to produce a pool of candidate solutions and one member selects the final answer. These strategies do not prescribe the subproblems of a new task. Instead, they organize a conversation within which the problem-specific division of reasoning can emerge, be challenged, and change as the solution develops.

We find that this learned organization enables what we call *collaborative computation*: agents develop solutions through joint natural-language reasoning by exchanging, challenging, repairing,

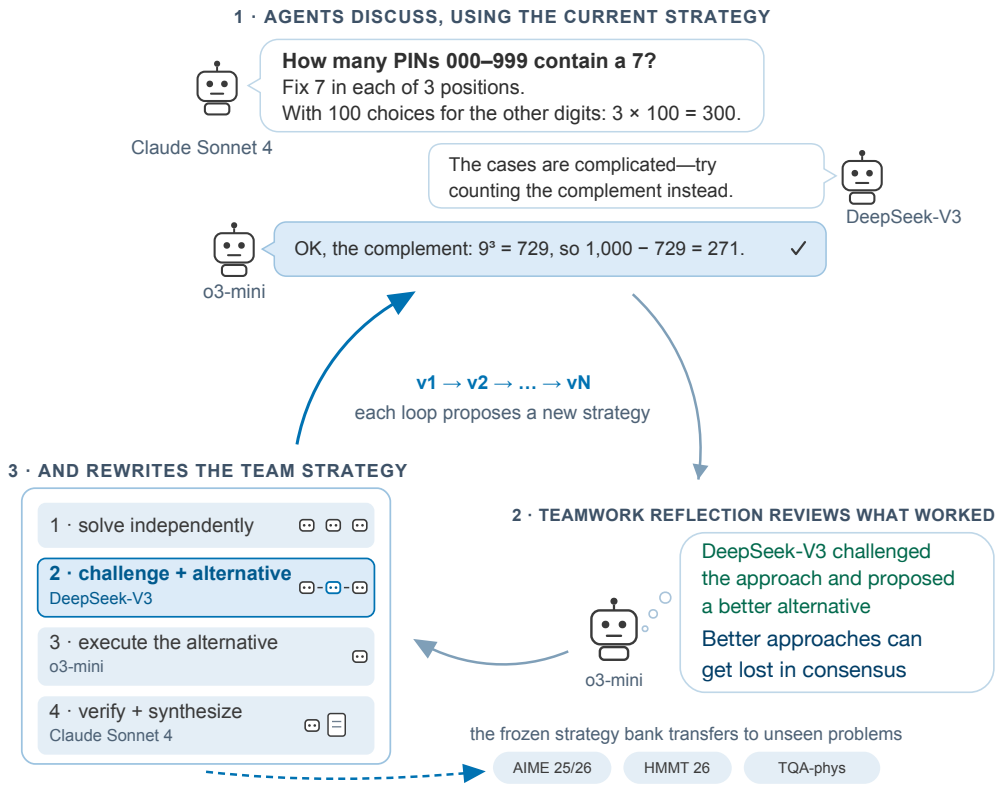


Figure 2: **Self-organizing agent teams learn how to reason together from prior collaboration.** A designated member reflects on earlier exchanges and outcomes, then revises the teamwork strategy governing subsequent collaboration. Candidate strategies are tested on training problems before a complementary bank is frozen for evaluation. Dialogue and role assignments are illustrative.

and synthesizing one another's reasoning. One member's partial insight can redirect another's approach, and an error in an otherwise useful derivation can be repaired by a different member. Most notably, a correct solution can emerge even when no member produced it independently.

This co-creation of solutions from partial attempts motivates a stricter comparison than those common in prior multi-agent work, which typically benchmarks a team against the member with the highest average performance (Wang et al., 2025; Nielsen et al., 2026). Because different members may solve different problems, a team can outperform its best member simply by selecting, for each problem, the answer of a member who solved it—no interaction required. Organizational psychology provides a stricter benchmark: under the *truth-wins* condition, a human team is treated as correct whenever any member solves the problem independently (Lorge & Solomon, 1955; Laughlin & Ellis, 1986). We operationalize its computational analogue as the *routing oracle*: a perfect per-problem selector over the members' independent answers. Surpassing this oracle shows that interaction produced correct solutions that no member supplied independently.

Our evaluation therefore asks three progressively stronger questions. *First*, does learned organization outperform the team's strongest member? *Second*, does it outperform compute-matched single-agent inference, including a linearization control in which the strongest member executes the same learned organizational structure at approximately the team's total inference budget? *Third*, and most importantly, can the team exceed perfect routing over its members' independent answers? We find that learned teamwork strategies can surpass all three baselines.

We learn strategy banks for two independent three-model teams, chosen to preserve headroom on their evaluation suites: a mathematics-and-physics team (o3-mini, Claude Sonnet 4, and DeepSeek-V3) trained on 15 AIME 2024 problems, and a knowledge-and-logic team (Gemini-2.5-Flash, Llama-4-Maverick, and GPT-4.1) trained on 25 GPQA Diamond problems. Across five mathematics and

physics benchmarks, the team averages 66.2% accuracy, compared with 48.8% for the strongest member, 59.2% for compute-matched linearization, and 59.0% for the routing oracle (Figure 1; Table 1). On AIME 2026, the team exceeds the oracle by 10.6 percentage points. Surpassing the routing oracle suggests that the team creates new reasoning unavailable from its members' independent samples. For example, on an HMMT problem that all three members initially answer incorrectly, o3-mini supplies the central invariant but makes a counting error, DeepSeek repairs the count, Claude Sonnet audits the corrected reasoning, and the team synthesizes a correct answer absent from all three initial responses (Figure 5).

The knowledge-and-logic team reveals a complementary limitation. Across three benchmarks, it achieves the highest average accuracy among the methods we test (72.8%) but remains below the routing oracle's 79.6%, even though its candidate pool contains a correct solution on 87.9% of problems, exceeding the routing oracle on every benchmark. Generating correct reasoning is therefore not enough: collaborative computation poses two distinct problems, *creating* a correct solution and *recognizing* it once it appears.

This separation suggests when learned organization is most valuable. Drawing on organizational psychology, we study *demonstrability*: whether correct reasoning can be distinguished from incorrect reasoning (Laughlin & Ellis, 1986). Across eight benchmarks, demonstrability strongly tracks how much the team improves over its strongest member (Spearman $\rho = 0.88$, $p = 0.007$), consistent with a simple mechanism: collaboration creates the most value when useful reasoning, once produced, can survive challenge, redirect subsequent reasoning, and be recognized as correct.

These results suggest that organization itself can become an agent capability: *learning how to reason together can change what a fixed collection of models is capable of solving*. Our contributions are:

Self-Organizing Agent Teams. We introduce SAT, in which fixed teams of AI agents learn reusable teamwork strategies from prior collaborations; the strategies organize interaction without prescribing a problem-specific decomposition and transfer unchanged to unseen problems and benchmarks.

Collaborative computation beyond independent inference. We show that learned organization composes partial reasoning into new solutions, exceeding compute-matched single-agent inference and perfect routing over members' independent answers across five mathematics and physics benchmarks.

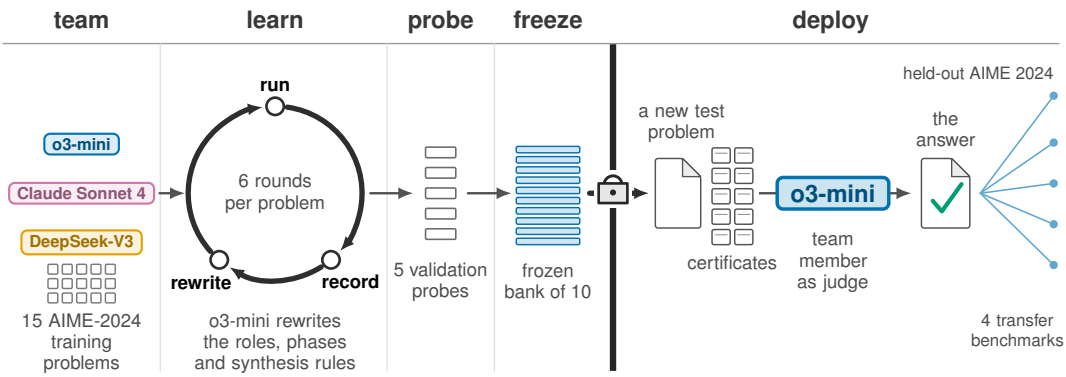
When learning organization helps. We separate generating correct reasoning from recognizing it and show that demonstrability—how reliably correct solutions can be distinguished from incorrect ones—strongly tracks how much collaboration improves over the strongest member on eight benchmarks.

2 Learning Generalizable Teamwork Strategies

2.1 A Language for Teamwork Strategies

We operationalize team organization as reusable teamwork strategies. To make this organization optimizable, we express each strategy in a domain-specific language whose primitive is a multi-agent *conversational phase*. A strategy consists of a shared teamwork prompt, strategy-level per-agent role prompts, and an ordered list of phases. Each phase specifies the participating agents, number of discussion rounds, information flow, shared phase prompt, and optional per-agent phase prompts. Each phase is therefore a unit of optimization in which agents read and respond to one another across rounds. The search varies who deliberates, when, with what information, and under what roles; it does not assign problem-specific sub-tasks and route their outputs, nor does it generate per-problem decompositions at test time. Appendix B gives the full parameterization of teamwork strategies.

For example, one learned GPQA strategy runs four one-round phases after the three members produce and share their initial independent solutions. The members first identify key claims and assumptions, then form a provisional consensus that records unresolved disagreements. Gemini-2.5-Flash, as final auditor, then checks that consensus against the initial attempts and resurfaces overlooked, well-supported claims; finally, all three members adjudicate these claims before a designated writer produces the team certificate. Figure 3(b) shows how teamwork reflection used the team's earlier failures to identify Gemini-2.5-Flash as the final auditor.



(a) Teamwork reflection and frozen deployment

(i) Failure diagnosis

"a **regression in cross-train performance**" … "**individual correct answers were lost due to team dynamics**"

(ii) Member-specific evidence

"leveraging the **observed strength of Agent 2 as an auditor/challenger**"

(iii) Strength → assigned role

"Agent 2, as the designated **final auditor**, must review … all individual attempts" and "**identify and re-present** any … claims … that were overlooked."

(b) From failure diagnosis to a specialized agent role

Figure 3: **Team organization is learned offline and frozen before evaluation. (a)** From 15 AIME-2024 training problems, teamwork reflection proposes and tests strategies, validation probes measure transfer, and a coverage-greedy step retains a bank of ten. At deployment, the frozen bank produces candidate certificates for a new problem, and one member selects the final answer. **(b)** In a GPQA mutation, the reflecting member diagnoses collaboration-induced loss and converts an observed member strength into an auditor role, yielding the `final_auditor_claim_recovery` strategy (Appendix E.2; identities blinded, Agent 2 is Gemini-2.5-Flash). Excerpts are verbatim, lightly trimmed.

2.2 Learning Teamwork Strategies

We learn candidate strategies through teamwork reflection, select a subset into a strategy bank, and deploy the bank unchanged at test time (Figure 3(a)).

Teamwork reflection and bank construction. The evolutionary search begins from an initial teamwork strategy, P_{init}: members first produce independent solutions, complete two rounds of debate-like exchange, and choose the final answer by majority vote over their final-round answers. Appendix C specifies this initialization and compares it with the learned strategies. For each training problem, we maintain an archive of candidate strategies and the team's executions of them. The roster member with the highest training-set accuracy on the source benchmark conducts *teamwork reflection*: o3-mini for AIME-2024 and Gemini-2.5-Flash for GPQA. This member inspects prior strategies, team transcripts, per-member answers, team outcomes, and validation results before choosing a candidate to build on and proposing targeted changes to its roles, conversational phases, and synthesis rules. Unlike GEPA (Agrawal et al., 2026), which selects candidates for mutation with a fixed heuristic, the reflecting member decides both which candidate to build on and how to revise it from the team's observed behavior. The full team then executes the new strategy, returning another transcript and outcome to the archive.

Candidates that solve their source problem are evaluated unchanged on five other training problems, which we call *validation probes*. To favor transferable strategies, the reflecting member uses validation-probe results in subsequent mutation decisions, and a separate instance of the reflecting model audits each candidate for source-specific content. We then greedily select up to ten strategies that maximize training-problem coverage. Appendix D details the search and bank construction; Appendix E contains the complete strategy and both deployed banks.

Test-time deployment. At test time, we run each frozen strategy to produce a pool of candidate solutions, each accompanied by a short, self-contained reasoning certificate. The reflecting member receives the problem and full candidate pool, audits the certificates for local defects without independently solving the problem, and selects the best-supported answer (prompts in Appendix F.1). We report **team coverage** (problems with at least one correct candidate) and **team accuracy** (problems answered correctly after selection); their gap separates solution generation from selection.

3 Experimental Setup

We evaluate the problem-independent strategies on held-out source problems and transfer benchmarks in a five-benchmark *math-and-physics suite* and a three-benchmark *knowledge-and-logic suite*.

Splits, rosters, and deployment. For the math-and-physics suite, the team comprises o3-mini, Claude Sonnet 4, and DeepSeek-V3, with o3-mini conducting teamwork reflection and serving as the final judge. We learn the bank from a 15-problem AIME-2024 training split, evaluate it on the disjoint AIME-2024 test split, and then deploy it unchanged on four transfer benchmarks: the next two competition years (AIME 2025 and AIME 2026), a different competition (HMMT February 2026), and a physics domain (TheoremQA-physics).

For the knowledge-and-logic suite, the team comprises Gemini-2.5-Flash, Llama-4-Maverick, and GPT-4.1, with Gemini-2.5-Flash conducting teamwork reflection and serving as the final judge. We learn a separate bank from a 25-problem GPQA Diamond training split, evaluate it on 100 disjoint GPQA Diamond problems, and then deploy it unchanged on MMLU-Pro (n=100) and five BIG-Bench Extra Hard (BBEH) logical-reasoning subtasks (n=75).

Baselines. Our primary outcome is **team accuracy**; we also report team coverage and routing-oracle coverage. The *best member* tests whether collaboration surpasses its strongest constituent, and the *routing oracle* tests whether it surpasses perfect per-problem selection over the members' individual answers. Self-consistency at K=10 matches the number of candidate solutions, and five-pass self-reflection controls for additional single-agent reasoning depth. Member-vote, three-round debate, and Mixture of Agents test whether a standard aggregation or deliberation scaffold suffices. Our strongest multi-agent control is a *homogeneous team*: three copies of the member with the highest training-set accuracy execute the same teamwork strategies, isolating the contribution of model heterogeneity. Our strongest single-agent control is *linearization*: the member with the highest training-set accuracy serially executes every learned role and phase at approximately the team's total inference budget, testing whether a single model given the same learned structure and a matched budget can reproduce the team's gains. See Appendix F for evaluation details.

4 Results

In both independent instantiations, the self-organizing team achieves the highest average accuracy among the methods we test. The math-and-physics team averages 66.2% across five benchmarks, while the knowledge-and-logic team averages 72.8% across three (Tables 1 and 2).

The math-and-physics team leads in accuracy and beats the perfect routing oracle. Across five benchmarks, the team averages 66.2% accuracy, compared with 48.8% for the best member, 57.3% for Mixture of Agents, the strongest fixed multi-agent baseline, and 59.2% for linearization, the strongest single-agent control (Table 1). On the held-out AIME-2024 split, it reaches 88.9%, 24.5 percentage points above o3-mini and above every tested control. The gap to linearization shows that o3-mini does not reproduce the team's average gain by serially replaying the same strategy structure, and the gap to the homogeneous team (58.7%) indicates that the learned interaction structure leverages model heterogeneity. The team's 66.2% accuracy also exceeds the 59.0% coverage of the routing oracle. Because the oracle can only select among members' individual answers, surpassing it shows that the team solves problems that no member solves independently in the observed samples.

AIME-trained teamwork strategies transfer across years, competitions, and domains. Deployed unchanged on four transfer benchmarks, the strategies retain large gains on new competition

	In-dist.	Out-of-distribution transfer				
Method	AIME24	AIME25	AIME26	HMMT26	TQA-phys	**Avg**
Best member	64.4	40.0	42.2	26.3	71.1	48.8
Self-consistency (K=10)‡	75.8	44.4	53.4	31.9	71.5	55.4
Self-reflection‡	71.1	41.1	52.2	29.3	72.2	53.2
Linearization (o3-mini)‡	74.2	49.3	64.7	**39.4**	68.4	59.2
Member-vote	60.7	38.5	40.9	25.3	71.3	47.3
Debate	66.7	41.1	53.3	30.3	72.2	52.7
Mixture of Agents	75.6	46.7	57.8	32.3	74.0	57.3
Homogeneous team (o3-mini)‡	76.0	50.0	61.1	36.4	70.2	58.7
SAT (Ours)	**88.9**	**57.3**	**68.4**	**39.4**	**77.2**	**66.2**
Routing-oracle coverage	73.3	51.1	57.8	36.4	76.6	59.0
SAT coverage (Ours)	97.8	73.3	70.0	54.5	81.6	75.4

Table 1: **SAT achieves the highest average accuracy across the mathematics-and-physics suite and exceeds routing-oracle coverage on average.** Strategies learned on 15 AIME-2024 training problems are frozen and evaluated on held-out AIME-2024 problems and four transfer benchmarks. Values are mean percentages over three seeds. Unshaded rows report final-answer accuracy (bold: highest per column, including ties); shaded rows report perfect-selection coverage over member answers (routing oracle) or team certificates. Single-agent controls use the strongest member on each benchmark unless otherwise stated; ‡ marks compute controls. Evaluation details are in Appendix F.

years and distributions (Table 1). On AIME 2025, the team reaches 57.3%, compared with 40.0% for the best member and 49.3% for linearization, and exceeds the routing oracle by 6.2 percentage points; on AIME 2026, it reaches 68.4%, compared with 42.2% for the best member and 64.7% for linearization, and exceeds the routing oracle by 10.6 points. On the harder HMMT 2026, the team ties matched linearization at 39.4%, compared with 26.3% for the best member. Transfer extends beyond competition mathematics: on TheoremQA-physics, the team achieves the highest accuracy of any method (77.2%).

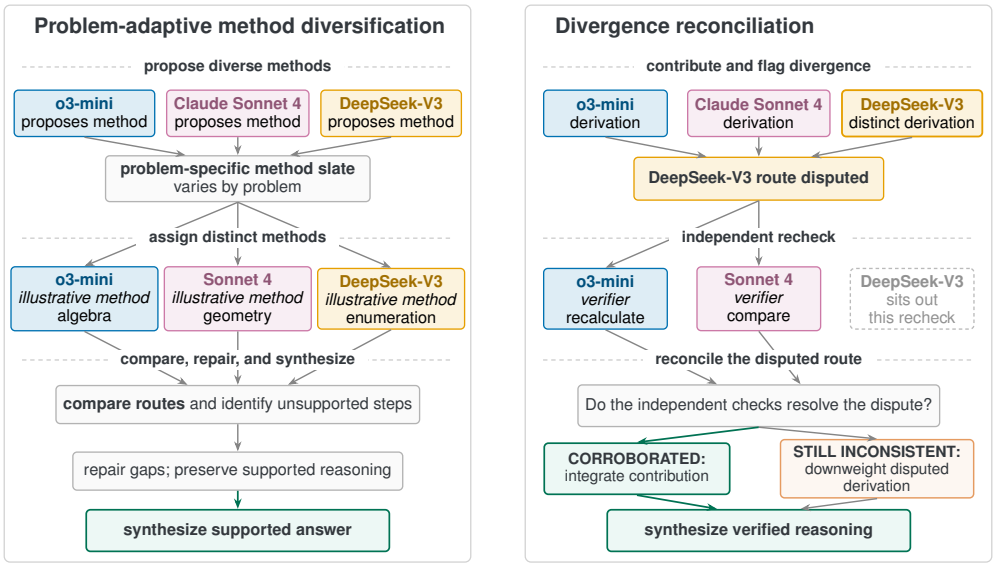
The knowledge-and-logic team leads in accuracy but remains below routing-oracle coverage. Across GPQA, MMLU-Pro, and BBEH, the team achieves the highest average accuracy of any method (72.8%), narrowly ahead of linearization (72.1%) and the multi-agent baselines—Mixture of Agents (71.2%), the homogeneous team (70.9%), and debate (70.2%)—and well above the best member (65.9%; Table 2). SAT is best on GPQA, but other methods win individual benchmarks: Mixture of Agents on MMLU-Pro and the homogeneous team on BBEH. However, the suite average remains well below the routing oracle's 79.6% coverage.

Learned strategies and case studies illustrate how conversation changes the team's computation. Learned strategies reach answers unavailable to routing through reusable structures that allow flexible repair and composition of partial reasoning (Figure 4). Problem-adaptive method diversification (Figure 4a) leaves the division of reasoning to the problem: members propose distinct methods suited to the current problem, then compare and repair the resulting derivations. Divergence reconciliation (Figure 4b) encodes a model-specific comparative advantage: because DeepSeek's alternative derivations sometimes catch cases the other members overlook but can also contain errors, the strategy retains DeepSeek as a source of useful diversity while o3-mini and Claude independently verify its disputed reasoning. Additional strategy examples appear in Appendix Figure 7.

DeepSeek's complementary value is also evident on HMMT 2026 problem 6, where it repairs o3-mini's decisive counting error and the team reaches an answer absent from all three initial responses (Figure 5; expanded transcript in Appendix Figure 8). Fixed protocols such as debate and Mixture of Agents offer less flexibility for composing and revising reasoning (Appendix Figure 6). On a GPQA chemistry problem (Appendix Figure 9), debate turns an initially wrong majority into unanimous incorrect agreement, displacing GPT's correct answer, and Mixture of Agents propagates a mistaken identification of two reaction pathways. SAT instead starts from three independently generated wrong answers and asks members to audit specific claims in one another's reasoning, propose corrections,

Method	In-dist. GPQA	Out-of-distribution transfer MMLU-Pro	BBEH	Avg
Best member	72.7	79.7	45.3	65.9
Self-consistency (K=10)‡	71.6	81.2	49.8	67.5
Self-reflection‡	77.7	81.7	54.7	71.4
Linearization (Gemini-2.5-Flash)‡	79.0	84.0	53.3	72.1
Member-vote	67.0	79.9	43.6	63.5
Debate	73.7	79.7	57.3	70.2
Mixture of Agents	72.0	**84.3**	57.3	71.2
Homogeneous team (Gemini-2.5-Flash)‡	78.0	76.0	**58.7**	70.9
SAT (Ours)	**80.0**	82.4	56.0	**72.8**
Routing-oracle coverage	84.0	88.0	66.7	79.6
SAT coverage (Ours)	94.0	91.0	78.7	87.9

Table 2: **SAT achieves the highest average accuracy across the knowledge-and-logic suite, but selection leaves a substantial gap to team coverage.** Strategies learned on 25 GPQA Diamond training problems are frozen and evaluated on held-out GPQA problems and two transfer benchmarks. Values are mean percentages over three seeds. Unshaded rows report final-answer accuracy (bold: highest per column, including ties); shaded rows report perfect-selection coverage over member answers (routing oracle) or team certificates. Single-agent controls use the strongest member on each benchmark unless otherwise stated; ‡ marks compute controls. Evaluation details are in Appendix F.



(a) (b)

Figure 4: **Teamwork reflection discovers qualitatively different ways to organize reasoning.**
(a) Problem-adaptive method diversification. Members propose distinct approaches, divide them among the team, and compare and repair the resulting derivations; the strategy is fixed, but the methods adapt to each problem.
(b) Divergence reconciliation. DeepSeek-V3 can surface useful alternative derivations but also introduce errors. The strategy keeps DeepSeek as a source of useful diversity but excludes it from a phase in which o3-mini and Claude Sonnet 4 independently verify any disputed reasoning.

and acknowledge and explain their errors. The team reaches the correct answer as a byproduct of repairing the reasoning. Challenges can also advance an incomplete solution or protect a correct one: on AIME 2026 II-08, an integrality check exposes a gap that Claude fills with the missing construction, and on II-01, a step audit overturns a wrong two-member majority (Appendix Figures 10–11).

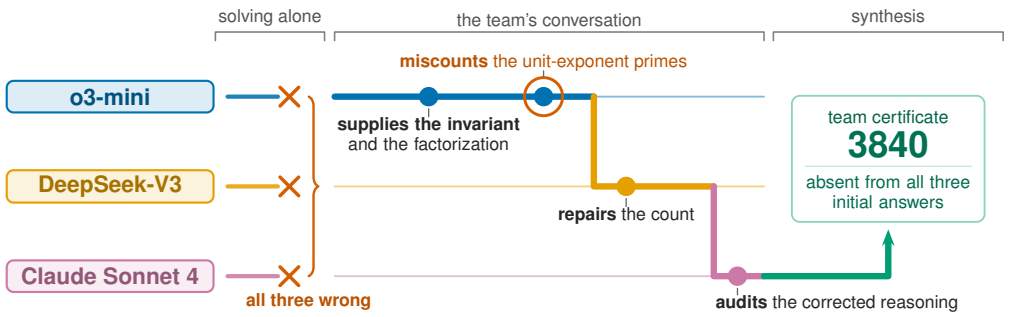


Figure 5: **Cross-member repair produces an answer no member initially had.** On HMMT 2026 problem 6, all three initial answers are wrong; o3-mini supplies the invariant, DeepSeek-V3 repairs the counting error, and Claude Sonnet 4 audits the correction, yielding the correct answer.

Team coverage and accuracy separate generation from selection. In both suites, team coverage exceeds routing-oracle coverage on every benchmark: 75.4% versus 59.0% on average in mathematics and physics, and 87.9% versus 79.6% in knowledge and logic. The interactive strategy executions therefore yield correct certificates absent from the members' individual answers. However, the suites differ in how much of this coverage survives selection: in mathematics and physics, the team's 66.2% accuracy exceeds the routing oracle, whereas in knowledge and logic, the team's 72.8% accuracy falls below it. The teams therefore vary in how much their expanded reasoning translates into gains over the best member. In Section 5, we propose an analytical lens for understanding why this happens.

5 When Does Learned Organization Improve Team Performance?

To understand when self-organizing agent teams improve over single-model performance, we use *demonstrability*—an organizational-psychology construct capturing whether correct reasoning can be distinguished from incorrect reasoning (Laughlin & Ellis, 1986)—as an analytical lens.

We measure demonstrability as *team-certificate discriminability*. For each problem with both a correct and a wrong team certificate, we pair one of each and measure how often a ten-model judge panel, drawn from outside both deployed team rosters, selects the correct one when both are shown without correctness labels in balanced A/B order (Appendix H). Averaging over orders, problems, and judges gives one score per benchmark. Unlike a formal verifier, this score does not certify an individual answer. It provides a soft, continuous benchmark-level notion of verifiability: how reliably a diverse judge panel recognizes correct reasoning relative to plausible failures.

Across the eight benchmarks, demonstrability strongly tracks the rank order of absolute improvement over the strongest team member (Spearman $\rho = 0.88$, exact permutation $p = 0.007$; Appendix Figure 12), and the association is robust to leaving out any single benchmark (Spearman $\rho = 0.82$–0.93; Appendix H). We hypothesize a simple mechanism: collaboration creates value when correct reasoning, once produced, can redirect or survive subsequent deliberation and selection, whereas correct reasoning that is difficult to recognize can be crowded out by plausible incorrect explanations.

6 Discussion and Limitations

Taken together, these experiments show that agent teams can improve by learning to organize their reasoning: learned strategies transfer unchanged to held-out problems and benchmarks, and in mathematics and physics the team surpasses a perfect router over its members' independent answers. Organization itself thus becomes an agent capability: without human-specified decompositions, fixed models can learn to reason together and construct solutions no member reaches alone.

Limitations and future work. Richer generation does not guarantee correct selection: on knowledge and logic, team coverage reaches 87.9% but team accuracy is only 72.8%, so better selection or more legible certificates are needed. Because demonstrability is measured post hoc, it remains correlational; future work could optimize for demonstrability and measure the effect on accuracy.

AI Use Statement

We used generative AI tools to assist with language refinement and code development. The authors reviewed and verified all AI-assisted work and take responsibility for the final content of this paper, including its text, claims, code, analyses, and artifacts.

Ethics Statement

This work studies how teams of language models learn to coordinate their reasoning. The experiments use publicly available models and involve no human subjects or personal data. Learned coordination is dual-use: the same capability that lets agent teams challenge and repair one another's reasoning could help agents coordinate toward undesirable ends, as the unsanctioned agent organization described in Section 1 illustrates. In our setting, teams have no tool or network access and coordinate only through natural-language strategies on closed-form reasoning problems; every learned strategy is human-readable, and we report the complete deployed strategy banks (Appendix E) so that the organization agents learn can be inspected directly. We view studying learned organization in controlled, inspectable settings as a way to understand such capabilities before they emerge in less constrained environments.

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. In *International Conference on Learning Representations (ICLR)*, pp. 8479–8565, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/0e9e708b6f48e14fd0ac29e167413f76-Abstract-Conference.html. Oral.

Zhenyu Bi, Meng Lu, Yang Li, Swastik Roy, Weijie Guan, Morteza Ziyadi, and Xuan Wang. OPTAGENT: Optimizing multi-agent LLM interactions through verbal reinforcement learning for enhanced reasoning. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 1713–1728, 2025. doi: 10.18653/v1/2025.findings-ijcnlp.107. URL https://aclanthology.org/2025.findings-ijcnlp.107/.

Federico Bianchi, Yongchan Kwon, Aneesh Pappu, and James Zou. Harnessing the collective intelligence of AI agents in the wild for new discoveries, 2026. URL https://arxiv.org/abs/2606.10402.

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996. doi: 10.1007/BF00058655.

Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. ReConcile: Round-table conference improves reasoning via consensus among diverse LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7066–7085, 2024. doi: 10.18653/v1/2024.acl-long.381. URL https://aclanthology.org/2024.acl-long.381/.

Hyeong Kyu Choi, Xiaojin Zhu, and Sharon Li. Debate or vote: Which yields better decisions in multi-agent large language models? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 101732–101764. Curran Associates, Inc., 2025. doi: 10.52202/085713-3405. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/934252acd87f254d5d4672fbde283bd2-Abstract-Conference.html. Spotlight.

D. Scott DeRue and Susan J. Ashford. Who will lead and who will follow? A social process of leadership identity construction in organizations. *Academy of Management Review*, 35(4):627–647, 2010. doi: 10.5465/amr.35.4.zok627. URL https://journals.aom.org/doi/abs/10.5465/amr.35.4.zok627.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235, pp. 11733–11763, 2024. URL https://proceedings.mlr.press/v235/du24e.html.

Amy C Edmondson, Richard M Bohmer, and Gary P Pisano. Disrupted routines: Team learning and new technology implementation in hospitals. *Administrative science quarterly*, 46(4):685–716, 2001.

Samer Faraj and Lee Sproull. Coordinating expertise in software development teams. *Management Science*, 46(12):1554–1568, 2000. doi: 10.1287/mnsc.46.12.1554.12072. URL https://pubsonline.informs.org/doi/10.1287/mnsc.46.12.1554.12072.

Samer Faraj and Yan Xiao. Coordination in fast-response organizations. *Management science*, 52(8): 1155–1169, 2006.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997. doi: 10.1006/jcss.1997.1504.

Ryan Greenblatt, Ajeya Cotra, and Hjalmar Wijk. Brief independent investigation of agents' behavior, reasoning and collaboration in the OpenAI / Hugging Face hacking incident. METR, August 2026. URL https://metr.org/blog/2026-08-26-openai-hugging-face-incident-investigation/. Accessed September 6, 2026.

Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 1990. doi: 10.1109/34.58871.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations (ICLR)*, pp. 23247–23275, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/6507b115562bb0a305f1958ccc87355a-Abstract-Conference.html.

Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. In *International Conference on Learning Representations (ICLR)*, pp. 21344–21377, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/36b7acf6f6010652b3f2a433774a66fe-Abstract-Conference.html.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan A, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. DSPy: Compiling declarative language model calls into state-of-the-art pipelines. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=sY5N0zY5Od. Spotlight.

Patrick R. Laughlin and Alan L. Ellis. Demonstrability and social combination processes on mathematical intellective tasks. *Journal of Experimental Social Psychology*, 22(3):177–189, 1986. doi: 10.1016/0022-1031(86)90022-3.

Yoonho Lee, Roshen Nair, Qizheng Zhang, Kangwook Lee, Omar Khattab, and Chelsea Finn. Meta-harness: End-to-end optimization of model harnesses. *arXiv preprint arXiv:2603.28052*, 2026. URL https://arxiv.org/abs/2603.28052.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. A dynamic LLM-powered agent network for task-oriented agent collaboration. In *First Conference on Language Modeling (COLM)*, 2024. URL https://openreview.net/forum?id=XII0Wp1XA9.

Irving Lorge and Herbert Solomon. Two models of group behavior in the solution of eureka-type problems. *Psychometrika*, 20(2):139–148, 1955. doi: 10.1007/BF02288986. URL https://doi.org/10.1007/BF02288986.

Elizabeth Mieczkowski, Alexander Ku, Tiwalayo Eisape, Dilip Arumugam, John Matters, Katherine M. Collins, Ilia Sucholutsky, and Thomas L. Griffiths. Improving the efficiency of language agent teams with adaptive task graphs. *arXiv preprint arXiv:2605.06320*, 2026. URL https://arxiv.org/abs/2605.06320.

Stefan Nielsen, Edoardo Cetin, Peter Schwendeman, Qi Sun, Jinglue Xu, and Yujin Tang. Learning to orchestrate agents in natural language with the Conductor. In *International Conference on Learning Representations (ICLR)*, pp. 135686–135724, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/dbb8193ad7e6fcbc7bb62ed9ee835110-Abstract-Conference.html.

OpenAI. The Hugging Face incident and the road ahead, August 2026. URL https://openai.com/index/hugging-face-incident-and-the-road-ahead/. Accessed September 6, 2026.

Ao Qu, Han Zheng, Zijian Zhou, Yihao Yan, Yihong Tang, Shao Yong Ong, Fenglu Hong, Kaichen Zhou, Chonghe Jiang, Minwei Kong, Jiacheng Zhu, Xuan Jiang, Sirui Li, Cathy Wu, Bryan Kian Hsiang Low, Jinhua Zhao, and Paul Pu Liang. CORAL: Towards autonomous multi-agent evolution for open-ended discovery. *arXiv preprint arXiv:2604.01658*, 2026. URL https://arxiv.org/abs/2604.01658.

Kevin Roose. Why the Hugging Face hack should make you worry more about A.I. *The New York Times*, September 2026. URL https://www.nytimes.com/2026/09/03/technology/openai-hugging-face-hacking.html. Accessed September 6, 2026.

Jon Saad-Falcon, Adrian Gamarra Lafuente, Shlok Natarajan, Nahum Maru, Hristo Todorov, Etash Kumar Guha, E. Kelly Buchanan, Mayee F. Chen, Neel Guha, Christopher Ré, and Azalia Mirhoseini. An architecture search framework for inference-time techniques. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267, pp. 52475–52507, 2025. URL https://proceedings.mlr.press/v267/saad-falcon25a.html.

Mirac Suzgun, Mert Yuksekgonul, Federico Bianchi, Dan Jurafsky, and James Zou. Dynamic cheatsheet: Test-time learning with adaptive memory. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7080–7106. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.eacl-long.333. URL https://aclanthology.org/2026.eacl-long.333/.

Kyle Swanson, Wesley Wu, Nash L Bulaong, John E Pak, and James Zou. The virtual lab of AI agents designs new SARS-CoV-2 nanobodies. *Nature*, 646(8085):716–723, 2025. doi: 10.1038/s41586-025-09442-9.

Melissa Valentine and Michael Bernstein. *Flash teams: Leading the future of AI-enhanced, on-demand work*. MIT Press, 2025.

Melissa A Valentine and Amy C Edmondson. Team scaffolds: How mesolevel structures enable role-based coordination in temporary groups. *Organization Science*, 26(2):405–422, 2015.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Y. Zou. Mixture-of-agents enhances large language model capabilities. In *International Conference on Learning Representations (ICLR)*, pp. 33944–33963, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5434be94e82c54327bb9dcaf7fca52b6-Abstract-Conference.html.

David H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992. doi: 10.1016/S0893-6080(05)80023-1.

Yingxuan Yang, Huacan Chai, Shuai Shao, Yuanyi Song, Siyuan Qi, Renting Rui, and Weinan Zhang. AgentNet: Decentralized evolutionary coordination for LLM-based multi-agent systems. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 107309–107336. Curran Associates, Inc., 2025. doi: 10.52202/085713-3578. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/9a379c1b05793d1c42dc832269834515-Abstract-Conference.html.

Hangfan Zhang, Zhiyao Cui, Jianhao Chen, Xinrun Wang, Qiaosheng Zhang, Zhen Wang, Dinghao Wu, and Shuyue Hu. Stop overvaluing multi-agent debate—we must rethink evaluation and embrace model heterogeneity. *arXiv preprint arXiv:2502.08788*, 2025a. URL https://arxiv.org/abs/2502.08788.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. AFlow: Automating agentic workflow generation. In *International Conference on Learning Representations (ICLR)*, pp. 34040–34077, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5492ecbce4439401798dcd2c90be94cd-Abstract-Conference.html. Oral.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Y. Zou, and Kunle Olukotun. Agentic context engineering: Evolving contexts for self-improving language models. In *International Conference on Learning Representations (ICLR)*, pp. 86069–86100, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/8a94ff6f922d995d7d3f4ebf4143e442-Abstract-Conference.html.

Han Zhou, Xingchen Wan, Ruoxi Sun, Hamid Palangi, Shariq Iqbal, Ivan Vulić, Anna Korhonen, and Sercan Ö. Arık. Multi-agent design: Optimizing agents with better prompts and topologies. In *International Conference on Learning Representations (ICLR)*, pp. 15844–15872, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/1ab4e0e8f35078109bc78d7b465d306f-Abstract-Conference.html.

Xiaochen Zhu, Caiqi Zhang, Yizhou Chi, Tom Stafford, Nigel Collier, and Andreas Vlachos. Demystifying multi-agent debate: The role of confidence and diversity. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 33909–33930. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.1694. URL https://aclanthology.org/2026.findings-acl.1694/.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. GPTSwarm: Language agents as optimizable graphs. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235, pp. 62743–62767, 2024. URL https://proceedings.mlr.press/v235/zhuge24a.html. Oral.

A Related Work

We organize prior work along three axes: *how* a team computes, whether its collaboration scaffold is problem-agnostic, and whether that scaffold is learned rather than hand-designed (Table 3).

Debate and voting. Du et al. (2024) introduced multi-agent debate: several model instances generate responses independently, then iteratively revise them after reading the other agents' full responses, with final answers combined by majority vote. Across seven benchmarks, Choi et al. (2025) find that majority voting over agents' independent initial responses accounts for most of the gains attributed to debate; their theoretical analysis likewise shows that debate alone does not improve expected correctness. Across five debate methods and nine benchmarks, Zhang et al. (2025a) find that debate fails to reliably outperform single-agent chain-of-thought or self-consistency (see also Zhu et al., 2026). Round-table consensus methods that weight agents by confidence (Chen et al., 2024) share this consensus-seeking character. These findings suggest that debate often functions like voting over diverse initial responses: gains arise from sampling diversity and selection rather than agents jointly reasoning to produce new inferences.

Mixture of Agents and feed-forward aggregation. Feed-forward aggregation offers a different way to combine agent outputs: Mixture of Agents (Wang et al., 2025) layers proposer models and a synthesizer, while DyLAN (Liu et al., 2024) stacks persona-prompted agents with inter-layer pruning. These systems can select agents, prune intermediate outputs, and synthesize across candidates, but their one-way structure does not allow agents to challenge and repair one another's reasoning through back-and-forth deliberation.

Figure 6 compares these fixed interaction structures with the participation, roles, and repair supported by SAT's learned teamwork strategies.

Learned orchestration and topology optimization. A growing line optimizes multi-agent workflows and topologies. GPTSwarm (Zhuge et al., 2024) represents agents as computational graphs and optimizes graph connectivity with REINFORCE; AFlow (Zhang et al., 2025b) searches code-represented workflows with Monte Carlo tree search; MASS (Zhou et al., 2026) optimizes prompts and topology over a fixed library of blocks; and AgentNet (Yang et al., 2025) adapts decentralized task-routing connections and per-agent retrieval memories from experience. Adjacent frameworks fix more of the surrounding program: MetaGPT (Hong et al., 2024) specifies a role-specialized pipeline, while DSPy (Khattab et al., 2024) compiles prompts and demonstrations within a user-defined program. These methods optimize computational graphs, routing, or prompts around model calls; our learned artifact is instead a reusable teamwork strategy that structures a multi-phase conversation in which agents exchange and revise reasoning.

The closest comparison to our work is OPTAGENT (Bi et al., 2025), which uses verbal reinforcement learning to optimize the edges of a pairwise-debate graph. It learns who interacts and in what order while retaining manually specified roles, the same exchange-and-revision operation on every edge, and majority-vote selection. We instead learn reusable teamwork strategies that vary roles, participants, rounds, information flow, and synthesis, then freeze them before transfer across benchmarks.

Problem-conditioned task organization. Conductor (Nielsen et al., 2026) trains a controller to emit a problem-conditioned workflow comprising worker instructions and communication topology. LATTE (Mieczkowski et al., 2026) instead lets agents construct and revise a shared graph of sub-tasks, dependencies, assignments, and progress during execution. Both adapt the division of labor to the current problem. We instead learn reusable teamwork strategies from training problems, freeze them before evaluation, and deploy them without prescribing a decomposition of the new problem.

Broader meta-search methods optimize agentic systems at the level of code or inference architecture. Meta Agent Search (Hu et al., 2025), introduced as an instance of ADAS, uses a meta-agent to search open-ended agent code. Its search space can in principle express conversation, but the reported agents primarily compose model calls for generation, critique, refinement, decomposition, and integration. Archon (Saad-Falcon et al., 2025) instead searches over compositions of inference-time techniques. We focus the search on reusable teamwork strategies, making the organization of multi-agent reasoning rather than a general agent program the learned artifact.

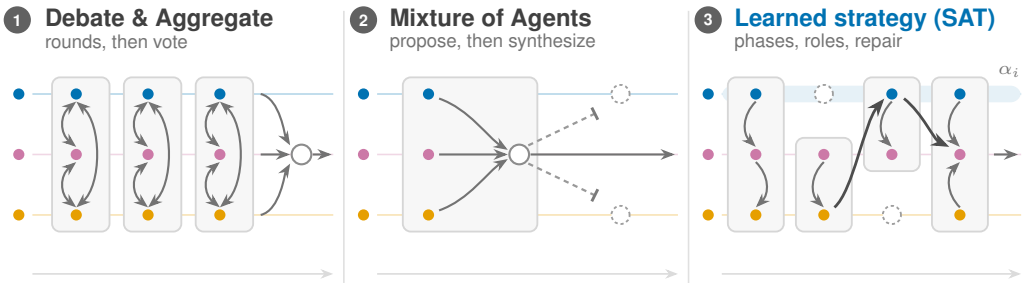


Figure 6: **Learned teamwork supports a richer interaction structure than common debate and feed-forward aggregation multi-agent systems.** Debate repeats symmetric exchange before voting; Mixture of Agents aggregates independent proposals in one direction. A learned strategy can instead vary participation across phases, maintain persistent roles, and revisit earlier reasoning for targeted repair. For a transcript-grounded comparison on a common GPQA problem, see Appendix Figure 9.

Virtual Lab (Swanson et al., 2025) provides a different point of comparison: it organizes a PI-led team of domain-specialist agents through research meetings, with high-level human feedback. Its collaborative scaffold enables substantive scientific work but is not learned from prior team behavior. We instead learn reusable teamwork strategies from prior collaborations, then deploy them unchanged on held-out problems and benchmarks.

Open agent platforms. Open agent platforms provide shared environments in which many agents collaborate freely on a common problem without a prescribed workflow. EinsteinArena (Bianchi et al., 2026), for example, allows agents to iteratively build on one another's solutions and feedback to make progress on open mathematics problems. Such platforms support collaborative computation without prescribing a problem-specific workflow, but do not learn reusable teamwork strategies from prior team behavior (Table 3).

Reflective search over reusable artifacts. Methodologically, we build on a line that optimizes non-weight artifacts through reflective search. GEPA (Agrawal et al., 2026) evolves prompts by reflective mutation and retains a Pareto frontier scored per-instance on a validation split; CORAL (Qu et al., 2026) studies open-ended discovery by delegating the evolutionary process itself to autonomous agents, replacing fixed search heuristics with agent decisions over retrieval, proposal, and evaluation; and Meta-Harness (Lee et al., 2026) searches over harness code. Related test-time "cheatsheet" methods accumulate reusable, evolving context—Dynamic Cheatsheet (Suzgun et al., 2026) and Agentic Context Engineering (Zhang et al., 2026)—but these artifacts serve as memory for a single model rather than as structures for multi-agent collaboration. We apply agent-driven evolutionary search to a new object: teamwork strategies learned on per-problem archives, selected into a fixed bank using only training evidence after search, and deployed on held-out problems and benchmarks (Section 2).

Approach	**Collaborative computation** joint reasoning, not aggregation or routing	**Problem-agnostic collaboration scaffold** reused across tasks and benchmarks	**Learned** not hand-designed
Debate, voting, Mixture of Agents	×	✓	×
Topology optimization (GPTSwarm, AgentNet)	×	×	✓
Interaction-graph optimization (OPTAGENT)	✓	×	✓
Problem-conditioned orchestration (Conductor)	×	×	✓
PI-led scientific collaboration (Virtual Lab)	✓	×	×
Open agent platform (EinsteinArena)	✓	✓	×
SAT (Ours)	✓	✓	✓

Table 3: **A landscape of multi-agent collaboration.** Multi-agent approaches differ along three dimensions: whether agents jointly develop reasoning, whether their collaboration scaffold is agnostic to problem content and reused across tasks and benchmarks, and whether that scaffold is learned rather than hand-designed.

B Teamwork Strategy Parameterization

The main text describes the conceptual structure of a teamwork strategy. Here we give its formal parameterization and execution semantics.

Strategies. Let $\mathcal{A} = \{a_1, \ldots, a_n\}$ denote a fixed roster of agents. A strategy $P = (S, \tau, \alpha)$ specifies how this roster collaborates on a problem. It consists of an ordered list of communication steps $S = [s_1, \ldots, s_K]$, a shared *teamwork prompt* τ stating collaboration norms for the whole team, and persistent per-agent *role prompts* $\alpha = \{\alpha_i\}_{i=1}^{n}$ that hold across every step. Each step

$$s_k = (A_k,\ r_k,\ f_k,\ \pi_k,\ \rho_k)$$

specifies the participating set $A_k \subseteq \mathcal{A}$, the number of discussion rounds r_k, an information-flow mode $f_k \in \{\mathrm{L}, \mathrm{S}\}$, a shared step prompt π_k, and optional per-agent step prompts $\rho_k = \{\rho_{k,i}\}_{i \in A_k}$. Within each round, every participant responds once, in an order specified by the strategy or randomly permuted when no order is specified. Under *local* flow (L), only the phase participants receive these turns. Under *summary* flow (S), the exchange remains local while the phase runs; afterward, one randomly selected participant summarizes its key points, conclusions, and current answer position, and that summary is added to every member's context. Each step is therefore a conversational phase—the unit of optimization—in which agents read and respond to one another across rounds under shared instructions and persistent roles. The search varies who deliberates, when, with what information, and under what roles; it does not assign problem-specific sub-tasks and route their outputs, nor does it generate per-problem decompositions at test time.

C Initialization and comparison with learned strategies

The evolutionary search is initialized with P_{init}, in which each member first produces an independent solution. All three members then complete two rounds of debate-like exchange, each time receiving the other members' most recent responses. The protocol uses no teamwork prompt, member-specific roles, or learned phase instructions. The final answer is chosen by majority vote over the members' final-round answers; if no strict majority exists, one member's final answer is selected at random. We evaluate P_{init} once per problem. Table 4 shows higher average accuracy for SAT in both task suites.

D Teamwork Reflection and Strategy Search

For each source problem s, we maintain an archive H_s containing candidate strategies, team transcripts, per-member answers, source-problem outcomes, and validation-probe results. The member conducting teamwork reflection inspects this evidence, chooses an archived strategy to build on, and proposes a targeted mutation to its roles, conversational phases, or synthesis rules. The full team executes each resulting candidate on the source problem, and the transcript and outcome are written

(a) Mathematics and physics

Method	AIME24	AIME25	AIME26	HMMT26	TQA-phys	Avg
P_{init}	73.3	50.0	50.0	24.2	72.8	54.1
SAT (Ours)	**88.9**	**57.3**	**68.4**	**39.4**	**77.2**	**66.2**

(b) Knowledge and logic

Method	GPQA	MMLU-Pro	BBEH	Avg
P_{init}	73.0	81.0	54.7	69.6
SAT (Ours)	**80.0**	**82.4**	**56.0**	**72.8**

Table 4: **SAT achieves higher average accuracy than the initialization in both task suites.** Panels (a) and (b) compare SAT with P_{init}, the initial teamwork strategy used to seed the evolutionary search that implements teamwork reflection. The initialization has each team member propose an independent solution, followed by two rounds of debate-like exchange and a final majority vote. All values are percentages; boldface marks the higher accuracy in each column, including ties.

back to the archive. We run six mutation rounds for each source problem, with up to three candidates proposed per round.

Candidates that solve their source problem are held fixed and evaluated on five other training problems. These validation-probe results are also written to the archive and inform subsequent reflection. This differs from GEPA (Agrawal et al., 2026), which stochastically selects a parent from an instance-wise Pareto frontier before an LM proposes a reflective mutation: here, the reflecting team member chooses both which archived strategy to build on and how to mutate it after inspecting source-problem outcomes, validation scores, and team behavior. After reflection, we use training-set performance to greedily select and freeze a bank of up to ten complementary strategies that collectively maximize training-problem coverage.

E Deployed strategy banks

This appendix lists the complete set of learned strategies deployed *unchanged* at test time. Strategy instructions are transcribed verbatim from the frozen banks. Notation follows Section 2.1: each phase **P**k lists its participating-member set A_k (the fixed roster slots AGENT 0/1/2), its rounds r_k, and its information-flow mode (L local, S summary-broadcast); *Roles* are the persistent per-agent prompts α (shown when set); the *Teamwork prompt* is the shared team-level instruction τ stating collaboration norms for the whole team. In the AIME-2024 bank, Agents 0/1/2 are o3-mini, Claude Sonnet 4, and DeepSeek-V3, respectively; in the GPQA-Diamond bank, they are Llama-4-Maverick, GPT-4.1, and Gemini-2.5-Flash, respectively. No phase, role, or teamwork prompt names a test problem. The AIME-2024 bank is also deployed unchanged across the remaining math-and-physics benchmarks (Section 4); the GPQA-Diamond bank is deployed unchanged on MMLU-Pro and BBEH.

E.1 AIME-2024 strategy bank (10 strategies)

`mechanistic_step_audit` — Agents audit the mechanics of each reasoning chain before synthesis.

• **P1** [$\{0,1,2\}$, r=1, L]: Audit the reasoning chains step by step. Each agent should identify one concrete step from another agent that is either verified, questionable, or incorrect.

• **P2** [$\{0,1,2\}$, r=1, L]: Resolve the audited issues. If a step is corrected, update the downstream calculation explicitly.

Teamwork prompt. Treat arithmetic, algebraic transformations, case splits, and counting steps as audit targets before accepting a final answer.

`weighted_derivation_consensus` — Agents independently recalculate disputed steps and reduce the influence of derivations that remain inconsistent.

• **P1** [$\{0,1,2\}$, r=1, L]: Each agent lists its full detailed derivation. Flag any steps that deviate from the majority computation. *(per-agent — A0: Provide your complete derivation and highlight any steps that differ*

from the expected methodology.; A1: Present your full computation and note any differences compared to others.; A2: Detail your derivation; if any step diverges, elaborate on your reasoning.)

• **P2** [$\{0, 1, 2\}$, r=1, L]: Independently re-calculate disputed steps. Weight the contributions: assign higher influence to agents whose past derivations align with the majority (agents 0 and 1) and lower influence to agent 2 if divergence persists. *(per-agent — A0: Re-calculate and confirm your steps; compare with others to validate consistency.; A1: Verify the disputed computations, emphasizing consistency with the majority.; A2: Review your derivation in light of the group's feedback and adjust if discrepancies are found.)*

Teamwork prompt. Apply a weighted consensus mechanism during discussion: if one agent's answer (notably agent 2) consistently deviates, reduce its weight in forming the final answer.

`minority_reasoning_challenge` — A designated dissenter challenges synthesis steps that may suppress correct minority reasoning before the team revises its answer.

• **P1** [$\{0, 1, 2\}$, r=1, L]: Each agent presents their independent reasoning, explicitly outlining their key load-bearing values and one potential source of error or dissent in the consensus derivation.

• **P2** [$\{0, 1, 2\}$, r=1, L]: A designated dissenting agent then challenges any step where minority correct reasoning might be suppressed, citing specific evidence from their chain.

• **P3** [$\{0, 1, 2\}$, r=1, L]: After addressing the challenge, the team revises and finalizes the synthesis to robustly support the final answer.

Teamwork prompt. Integrate explicit adversarial challenge steps to preserve correct minority reasoning during synthesis.

`divergence_reconciliation` — Agents identify persistently divergent derivations, independently verify the disputed steps, and either reconcile them or reduce their influence.

• **P1** [$\{0, 1, 2\}$, r=1, L]: Each agent lists their full derivation. Specifically flag any steps from agents whose computations systematically differ from the majority. *(per-agent — A0: Review and note any discrepancies in the reasoning contributed by any agent showing divergence.; A1: Cross-verify computations from any agent identified with divergence using your own reasoning.; A2: If you are the agent with divergent reasoning, provide detailed explanations for each step.)*

• **P2** [$\{0, 1\}$, r=1, L]: Initiate an independent re-calculation of the disputed steps provided by the identified divergent agent, comparing them with the computations from the other agents. Conclude by either reconciling the inconsistent result or reducing its influence. *(per-agent — A0: Independently re-calculate the disputed steps and share your verified results.; A1: Compare your verification results with those from the divergent input and note any differences.)*

Teamwork prompt. Introduce a reconciliation step dedicated to reviewing any persistent divergences. If an agent's contributions are repeatedly inconsistent with the group, their influence is reduced through independent verification.

`backward_constraint_validation` — Agents derive answer constraints, generate candidates, and verify backward.

• **P1** [$\{0, 1, 2\}$, r=1, L]: Derive necessary conditions for the final integer answer without relying on pre-supplied candidates: bounds, divisibility, modular residues, monotonicity, feasibility, or direct substitution conditions.

• **P2** [$\{0, 1, 2\}$, r=1, L]: For each candidate answer in the discussion, test it backward against the original problem conditions and the necessary conditions. Reject or repair candidates only with a stated mathematical reason.

Teamwork prompt. Use the final-answer format and problem constraints as a validation scaffold. Candidate answers must survive independent backward checks.

`constraint_inventory_then_solve` — Agents first enumerate constraints and then use them to audit candidate answers.

• **P1** [$\{0, 1, 2\}$, r=1, L]: Do not finalize yet. List constraints any valid solution must satisfy: bounds, integrality, parity, congruences, case coverage, geometric conditions, or counting totals.

• **P2** [$\{0, 1, 2\}$, r=1, L]: Use the constraint inventory to check the candidate solutions. Revise only when a concrete constraint is violated or a missing case is found.

Teamwork prompt. Before accepting a final answer, build and use an explicit inventory of constraints from the problem.

`independent_solve_then_synthesis` — Agents compare independent solutions, identify disagreements, and synthesize.

• **P1** [$\{0, 1, 2\}$, r=1, L]: Compare the independent reasoning chains. Each agent should name the answer they got, the main method they used, and one possible weakness in their own solution.

• **P2** [$\{0, 1, 2\}$, r=1, L]: Synthesize the strongest supported reasoning into a shared answer. If answers differ, resolve the disagreement using specific mathematical steps from the discussion.

Teamwork prompt. Preserve independent reasoning. Do not converge until each agent's solution has been compared against the others.

`problem_adaptive_method_diversification` — Agents propose and pursue distinct methods suited to the current problem.

• **P1** [$\{0, 1, 2\}$, r=1, L]: Do not finalize yet. Each agent should propose plausible solution frames for this specific problem, such as algebraic, geometric, combinatorial, modular, invariant, constructive, extremal, coordinate, or computational-enumerative routes.

• **P2** [$\{0, 1, 2\}$, r=1, L]: Assign distinct viable frames across agents and push each route as far as possible. State when a route fails or supports a candidate answer. *(per-agent — A0: Prefer the first viable method family not already emphasized.; A1: Prefer a different viable method family from Agent 0.; A2: Prefer a checking or alternative method family distinct from Agents 0 and 1.)*

• **P3** [$\{0, 1, 2\}$, r=1, L]: Compare the method-specific results. Favor answers supported by independent routes or by the route with the clearest complete derivation.

Teamwork prompt. The team should deliberately diversify methods before synthesizing. Distinct methods are chosen based on the problem.

`suspicious_consensus_challenger` — If the team converges early, one agent must look for a failure mode.

• **P1** [$\{0, 1, 2\}$, r=1, L]: State the current consensus or disagreement. If there is a consensus, identify the weakest link in the shared reasoning.

• **P2** [$\{0, 1, 2\}$, r=1, L]: Agent 2 acts as consensus challenger. Try to find an alternative derivation, missing case, arithmetic error, or constraint violation. Agents 0 and 1 respond only with mathematical evidence. *(per-agent — A2: You are the consensus challenger. Look for concrete failure modes before accepting the answer.)*

• **P3** [$\{0, 1, 2\}$, r=1, L]: Decide whether the challenged answer survives. If it does, state why; if not, revise using the discovered issue.

Teamwork prompt. Consensus is not sufficient. If the team appears to agree, actively test whether the shared answer could still be wrong.

`component_recombination_validation` — Agents compute key quantities independently, reconstruct their implied components, and verify that those components recombine to satisfy the original constraints.

Roles. AGENT 0: Share your backward validation process and adjust your computation if inconsistencies arise.; AGENT 1: Ensure that the derived components are logically consistent and sum up to meet the given constraints.; AGENT 2: Advocate for the correct computation if your backward validation confirms a lower and more consistent value.

• **P1** [$\{0, 1, 2\}$, r=1, L]: Step 1: Each agent computes the key parameter using their preferred approach and documents all critical arithmetic steps and load-bearing intermediate values. *(per-agent — A0: State your computed value along with key intermediate figures.; A1: Include a clear record of relevant arithmetic steps that support your computation.; A2: Detail your computation and list any intermediate values used in your derivation.)*

• **P2** [$\{0, 1, 2\}$, r=1, L]: Step 2: Perform a backward validation by using your computed value to deduce the implied component values. Verify that these components, when recombined, satisfy the overall constraints provided in the problem. Confirm that all derived components are consistent and nonnegative. *(per-agent — A0: Compute the derived components and demonstrate their recombination into the overall constraint.; A1: Verify that your deduced values jointly satisfy the aggregate condition and discuss any discrepancies.; A2: Check and confirm that all derived components are positive and consistent with the given constraints.)*

Teamwork prompt. After individual computations, collaboratively verify that the computed value leads to a consistent set of derived components, which when recombined, fully satisfy the overall constraints. Any inconsistency should trigger a review of the intermediate arithmetic for potential overestimation.

E.2 GPQA-DIAMOND STRATEGY BANK (10 STRATEGIES)

`constructive_challenge_and_preservation` — A designated challenger proposes concrete repairs or alternatives, after which the team preserves, repairs, or rejects each well-supported claim before synthesis.

Roles. AGENT 2: You are the primary challenger. Your role is to critically examine the claims of others, and when challenging, propose a plausible alternative interpretation, missing condition, factual error, or a specific repair to the reasoning, not just point out a flaw. Furthermore, if an individual agent presented a solution that is well-supported by evidence, you must proactively argue for its preservation, citing specific scientific principles and evidence. Your goal is to improve the scientific rigor of the solution through constructive adversarial engagement and ensure robust insights are not lost.

• **P1** $[\{0, 1, 2\},\ r{=}1,\ \mathrm{L}]$: Each agent independently proposes their best solution and the key scientific claims supporting it. Identify any areas of strong initial consensus or disagreement.

• **P2** $[\{0, 1, 2\},\ r{=}1,\ \mathrm{L}]$: Agent 2 (Challenger): Identify the weakest or most critical claim in the initial consensus or a divergent solution. Formulate a specific challenge that includes a plausible alternative interpretation, a missing condition, a factual error, or a proposed 'repair' to the reasoning. Crucially, if any agent's initial solution is well-supported by evidence, Agent 2 must articulate why that solution *should* be preserved, citing specific evidence or scientific principles. Other agents (0, 1): Defend your original reasoning with evidence or acknowledge the validity of the challenge/preservation argument and propose a repair or accept the preservation.

• **P3** $[\{0, 1, 2\},\ r{=}1,\ \mathrm{L}]$: Based on the challenges and responses, collectively decide whether the challenged reasoning survives, is repaired, or is abandoned. If a repair is accepted, clearly state the revised scientific claim or reasoning. If fundamental disagreements persist, especially regarding the preservation of a robust individual solution, identify the precise points of contention and their impact on the overall solution.

• **P4** $[\{0, 1, 2\},\ r{=}1,\ \mathrm{L}]$: Synthesize all surviving and repaired scientific claims into a final, robust solution. Clearly state the solution and justify why it is the most scientifically sound choice given the debate, explicitly mentioning how any robust individual solutions were preserved or why they were ultimately discarded.

Teamwork prompt. Engage in an adversarial debate, where challenges include proposed repairs or alternative interpretations. Critically evaluate all claims and collaboratively refine the solution, with a specific focus on preserving and validating robust individual solutions.

`option_aware_claim_critique` — Agents challenge one another's scientific claims, document unresolved disagreements, and justify the final answer against the available options.

• **P1** $[\{0, 1, 2\},\ r{=}1,\ \mathrm{L}]$: Each agent independently analyzes the problem, proposes a solution, and identifies at least one key scientific claim or assumption in their own reasoning. Agents also propose specific checks or questions for at least one claim or assumption made by another agent, highlighting potential weaknesses or alternative interpretations.

• **P2** $[\{0, 1, 2\},\ r{=}1,\ \mathrm{L}]$: Agents engage in a structured debate. Each agent defends their own claim(s) with scientific evidence and addresses the proposed checks or critiques from others. Critiques should not only address scientific accuracy but also consider whether an alternative interpretation or weakness significantly impacts the choice of answer, given the available options and any other problem-specific selection constraints.

• **P3** $[\{0, 1, 2\},\ r{=}1,\ \mathrm{L}]$: The team synthesizes a final reasoning chain. If there are unresolved scientific disagreements, they must be explicitly stated, along with the evidence for each side. If a pragmatic decision was made to select an answer due to practical selection constraints despite scientific ambiguities, this must also be explicitly documented with justification.

Teamwork prompt. The team should engage in structured, adversarial critique, focusing on scientific claims and their implications within the problem's context. Be mindful of practical selection constraints, such as the available options, when evaluating alternative interpretations or weaknesses. The final answer must be scientifically sound and pragmatically justified.

`discrepancy_and_contradiction_audit` — A designated auditor identifies contradictions and unaddressed constraints across independent solutions before the team resolves them.

Roles. AGENT 2: You are the Discrepancy and Contradiction Auditor. Your role is to critically examine the solutions provided by Agent 0 and Agent 1. Focus on identifying and clearly articulating scientific inconsistencies, contradictions, or unaddressed problem constraints in their proposed identified entities, reactions, reasoning, and final answers. Pay close attention to details like implied properties, logical consistency, and adherence to all problem requirements. Your goal is to ensure the final team solution is scientifically robust and fully consistent with the problem statement.

• **P1** [$\{0, 1, 2\}$, r=1, L]: Each agent will individually present their full solution, including their final answer, detailed reasoning, identified entities, and how each explicit condition or test result stated in the problem was addressed. Do not debate yet.

• **P2** [$\{2\}$, r=1, L]: Agent 2, as the Discrepancy and Contradiction Auditor, will review all presented solutions from Agents 0 and 1. Your task is to identify and articulate any scientific inconsistencies (e.g., misinterpretations of properties, unaddressed constraints, logical flaws) AND explicit contradictions (e.g., conflicting identified entities, reaction pathways, or derivations) between the different proposed solutions or against the problem statement. Clearly present these findings to Agents 0 and 1, specifying where and why they exist.

• **P3** [$\{0, 1, 2\}$, r=1, L]: Agents 0 and 1 will respond to the discrepancies and contradictions raised by Agent 2. Clarify, correct, or refine your solutions based on the auditor's findings. For each identified inconsistency or contradiction, agents must engage in a structured debate to determine its root cause and decisively resolve it through scientific reasoning and reference to problem statements. Agent 2 will facilitate this discussion to ensure all issues are thoroughly addressed and resolved, leading to a unified, consistent, and scientifically robust solution.

Teamwork prompt. Independently solve the problem. Then, with a designated auditor, critically examine and resolve any scientific inconsistencies or contradictions through structured debate to arrive at a single, accurate team solution.

`provisional_consensus_cross_validation` — The team forms a provisional consensus, tests every strong contender against necessary conditions, and subjects the result to a final assumptions audit.

• **P1** [$\{0, 1, 2\}$, r=1, L]: Each agent independently state your answer and the full scientific reasoning, including all key assumptions, critical calculations, and principles used. Share your reasoning in detail to allow for thorough cross-verification.

• **P2** [$\{0, 1, 2\}$, r=1, L]: Compare the independent answers and reasoning. Identify the strongest arguments and points of agreement or disagreement. Focus on any differences in assumptions or derivation steps. Propose a preliminary consensus answer based on the most robust reasoning observed among the independent solutions, explicitly noting any remaining discrepancies. If there are multiple robust and conflicting independent answers, identify them, preserving all evidence and rationale.

• **P3** [$\{0, 1, 2\}$, r=1, L]: For each of the preliminary consensus answer(s) (or if multiple, each strong contender), systematically cross-verify each other's full reasoning chains, including all assumptions. Identify what would need to be true if that option were correct. Include checks for consistency in units, signs, mechanism, causal direction, limiting cases, and compatibility with the experimental setup. List all derived implications and necessary conditions. Agent 0 will lead this cross-verification, focusing on quantitative consistency and scientific principles.

• **P4** [$\{0, 1, 2\}$, r=1, L]: Test the preliminary consensus answer(s) backward against the original question conditions and all derived implications. Systematically check each condition and implication. Reject or repair candidates only with a specific scientific reason supported by evidence. If a strong contender was not chosen as the preliminary consensus, validate it here as well. Finalize the best-supported answer based on this rigorous validation. Agent 1 will lead the final synthesis, ensuring all evidence is accounted for.

• **P5** [$\{0, 1, 2\}$, r=1, L]: Before finalizing, Agent 2 will perform a final audit of the chosen answer, specifically challenging any implicit assumptions and checking for alternative interpretations that might have been overlooked. The team must address these challenges before providing the final answer.

Teamwork prompt. Combine independent problem-solving with rigorous and systematic cross-verification and backward validation. Begin with independent derivations and explicit assumption declarations, synthesize a preliminary consensus by identifying strongest arguments and discrepancies, and then test this consensus and all strong contenders against the problem's conditions and all answer choices using backward reasoning and systematic checks. Conclude with a final audit of assumptions and alternative interpretations.

`neglected_effects_challenge` — A designated challenger quantitatively tests effects initially dismissed as negligible and requires the team to incorporate them or justify their exclusion.

Roles. AGENT 2: You are the consensus challenger. Your primary role is to identify and quantitatively evaluate any subtle physical effects or overlooked information suggested by the problem's parameters, especially when it might change the outcome, even if initially thought to be negligible. Focus on general physical principles rather than problem-specific details.

• **P1** [$\{0, 1, 2\}$, r=1, L]: Each agent independently presents their initial interpretation of the problem, including relevant physical laws and initial proposed solution path. They should highlight any information they consider potentially extraneous or secondary.

• **P2** [$\{2, 0, 1\}$, r=1, L]: Agent 2 acts as consensus challenger. Agent 2 must scrutinize all information initially deemed 'extraneous' or 'secondary' by any agent. Specifically, Agent 2 must identify any physical phenomena (e.g., relativistic effects, quantum effects, or environmental factors) that could be implicitly suggested by the problem's parameters (e.g., high velocities, very small scales, extreme conditions) but might have been overlooked. For each identified phenomenon, Agent 2 must provide a qualitative argument for its potential relevance and an initial quantitative estimate of its impact on the solution. Agents 0 and 1 must then critically evaluate Agent 2's arguments and estimates, providing counter-arguments or supporting evidence based on general scientific principles, not on external tools or knowledge outside the prompt.

• **P3** [$\{0, 1, 2\}$, r=1, L]: The team collaboratively decides whether any of the challenged physical phenomena are indeed significant enough to alter the solution, based on the quantitative estimates and critical evaluations. If deemed significant, the team must explicitly incorporate the effect into their derivation and recalculate the solution. If deemed negligible, a clear, quantitative justification for its negligibility must be provided.

• **P4** [$\{0, 1, 2\}$, r=1, L]: Present the final, refined solution, explicitly detailing all physical effects considered, their quantitative impact (or justification for negligibility), and the final calculated answer.

Teamwork prompt. Actively challenge assumptions of negligibility, especially for physical effects suggested by problem parameters. Quantitatively evaluate potential subtle effects to determine their true significance.

`final_auditor_claim_recovery` — A final auditor resurfaces well-supported claims omitted from the provisional consensus for explicit integration or refutation.

Roles. AGENT 2: You are the final auditor. Your role is critical in preventing collaboration-induced loss of well-supported individual claims. Review the provisional consensus and all individual attempts. Specifically identify and re-present any materially distinct or well-supported individual claims (especially from Agents 0 and 1) that were overlooked or not sufficiently addressed. Prompt Agents 0 and 1 to respond with supporting evidence or revised reasoning. Your goal is to ensure all valid insights are brought to the team's attention for final adjudication.

• **P1** [$\{0, 1, 2\}$, r=1, L]: Each agent independently derives a solution and identifies the key scientific claims, assumptions, and supporting evidence for their conclusion.

• **P2** [$\{0, 1, 2\}$, r=1, L]: All agents share their solutions and derivations. The team discusses to arrive at a provisional consensus solution, explicitly documenting any discrepancies or unresolved scientific issues and any unique, well-supported individual claims.

• **P3** [$\{0, 1, 2\}$, r=1, S]: Agent 2, as the designated final auditor, must review the provisional consensus solution and all individual attempts, paying close attention to previously documented unique or dissenting well-supported claims. If Agent 2 identifies any materially distinct or well-supported individual claims that were overlooked or not sufficiently addressed, Agent 2 must re-present the reasoning and evidence for these claims. Agents 0 and 1 respond to this audit with supporting evidence or revised reasoning. The team then collectively decides whether to integrate the overlooked claim or formally reject it with scientific justification.

• **P4** [$\{0, 1, 2\}$, r=1, L]: Final Adjudication: The team must now revisit ALL distinct, well-supported individual claims that were not fully integrated into the provisional consensus or were initially set aside. For each such claim, the team must explicitly discuss its scientific validity. If a claim is deemed scientifically valid based on evidence, it must be integrated into the final solution, even if it requires revising the consensus. If a claim is deemed invalid, the team must provide a clear, scientific refutation. The final team solution must represent a comprehensive adjudication of all individual insights, ensuring no well-supported individual claim is lost without thorough, explicit scientific justification.

Teamwork prompt. Independently derive, then collaboratively consolidate. A designated expert auditor will perform a final review to ensure no materially distinct or well-supported individual claims are lost. Subsequently, the team must formally adjudicate all unique individual claims, integrating valid ones and scientifically refuting invalid ones, to prevent collaboration-induced loss of valuable insights.

`option_conditions_and_absence_audit` — Members derive necessary conditions for each option and revisit eliminations that rely on the presumed absence of a signal or feature.

• **P1** [$\{0, 1, 2\}$, r=1, L]: Each agent independently reviews the problem and proposes an initial hypothesis for the answer, including the key supporting evidence and any assumptions made. Share these initial thoughts.

• **P2** [$\{0, 1, 2\}$, r=1, L]: For each answer option (A, B, C, D), Agent 0 states what scientific conditions or observations *must* be true for that option to be correct. Agent 1 then provides scientific evidence from the problem or general relevant domain principles to support or refute these conditions. Agent 2 critiques the strength of the evidence and the validity of the conditions and the interpretation.

• **P3** [$\{0, 1, 2\}$, r=1, L]: Based on the validation and critique, collectively eliminate options that are demonstrably false or lack sufficient evidence. Discuss any remaining ambiguities or conflicting evidence.

• **P4** [$\{0, 1, 2\}$, r=1, L]: Before finalizing, explicitly check if any option was eliminated based on a perceived 'absence' of a signal, property, or feature. Agent 2, specifically challenge any elimination based on the supposed 'absence' of any specific experimental observation, signal, or feature, demanding rigorous structural and mechanistic justifications for its non-existence. If so, revisit that elimination with extreme scrutiny, requiring explicit scientific reasoning for why that absence is definitive and not merely an oversight or misinterpretation.

• **P5** [$\{0, 1, 2\}$, r=1, L]: Synthesize the final answer from the remaining validated options, ensuring that the chosen option is fully supported by the available evidence and free of scientific contradictions.

Teamwork prompt. Systematically validate each answer option by defining its necessary conditions, evaluating evidence for those conditions, and critically reviewing the evidence. Pay special attention to claims of 'absence' of features or signals, especially in the interpretation of experimental data.

`corrective_step_audit` — Members audit specific scientific claims, propose and acknowledge corrections, and preserve validated minority evidence during synthesis.

• **P1** [$\{0, 1, 2\}$, r=1, L]: Audit the reasoning chains step by step. Each agent should identify one concrete claim from another agent that is either verified, questionable, or incorrect. For any questionable or incorrect claim, the auditing agent must explicitly propose a scientific correction and provide a brief rationale for the correction.

• **P2** [$\{0, 1, 2\}$, r=1, L]: Resolve the audited issues. If a claim is corrected, the agent whose claim was corrected must acknowledge the correction, explain the scientific reason for their initial error, and confirm their revised answer choice. If a proposed correction is challenged, the team must discuss and scientifically justify the most accurate correction.

• **P3** [$\{0, 1, 2\}$, r=1, L]: Synthesize the strongest supported answer, explicitly incorporating all agreed-upon corrections. The final answer must reflect a consolidated, scientifically sound reasoning chain, and any initial individual claims that were validated against evidence and reasoning must be explicitly identified and their preservation explained, even if initially a minority view.

Teamwork prompt. Treat scientific facts, mechanisms, definitions, unit conversions, and option eliminations as audit targets before accepting a final answer. Ensure that all identified errors are explicitly corrected and documented, and that all distinct individual claims supported by evidence are retained and integrated into the final team solution.

`precision_matched_effects_audit` — The team quantifies secondary effects relative to the precision separating the answer options before selecting the closest option.

• **P1** [$\{0, 1, 2\}$, r=1, L]: State the current consensus or disagreement regarding the physical effects at play. If there is a consensus on which effects are relevant, identify the weakest link in the shared scientific rationale, especially regarding neglected effects or assumptions of negligibility. Explain the rationale for any initial dismissal of such effects.

• **P2** [$\{2, 0, 1\}$, r=1, L]: Agent 2 acts as consensus challenger. For any parameters in the problem (e.g., relative velocities, small differences in quantities, extreme conditions) that might suggest a secondary physical effect (e.g., relativistic effects, quantum effects, gravitational interactions), Agent 2 must quantitatively evaluate if this effect significantly alters the primary calculation. This evaluation must consider the magnitude of the effect relative to the precision required to distinguish between the available answer options. Agents 0 and 1 respond with their own calculations or a critical review of Agent 2's calculation, focusing on the quantitative assessment.

• **P3** [$\{0, 1, 2\}$, r=1, L]: The team must now perform a precise calculation of the full solution, explicitly including the quantitative impact of any physical effect identified as non-negligible. Once the numerical result is obtained, compare it against all provided multiple-choice options. For each option, calculate the absolute difference between the calculated value and the option. State the final answer as the option with the smallest absolute difference.

• **P4** [$\{0, 1, 2\}$, r=1, L]: Decide whether the challenged answer or current working solution survives. If it does, state why with evidence; if not, revise using the discovered issue and present the improved solution, specifically justifying the choice of the closest option based on the precise numerical comparison.

Teamwork prompt. Consensus is not sufficient. If the team appears to agree, actively test whether the shared answer could still be wrong. Specifically, always quantitatively evaluate secondary physical effects, especially when velocities, small differences, or extreme conditions are provided, to confirm their negligibility or significance, and perform precise numerical matching to options.

`minority_evidence_adjudication` — A designated adjudicator evaluates every claim that differs from the emerging consensus and integrates or refutes it with evidence.

Roles. AGENT 2: You are the Minority Evidence Adjudicator. Your role is to critically evaluate any individual claims that diverge from the team's emerging consensus. You must explicitly state whether each differing claim

is scientifically sound. If it is, integrate it into the collective solution and explain its relevance. If it is not, provide a clear, evidence-based scientific refutation. Be thorough and precise.

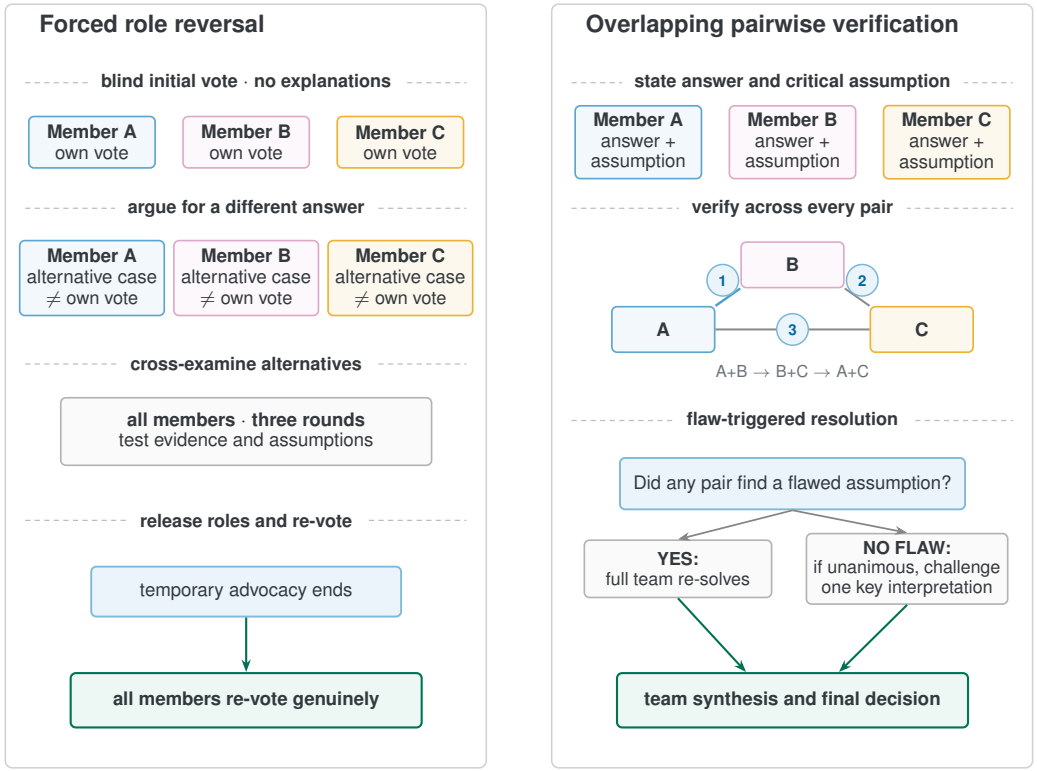
• **P1** $[\{0, 1, 2\},\ r{=}1,\ \mathrm{L}]$: Each agent independently derives an answer and identifies the key scientific principles, assumptions, or reasoning steps used in their derivation, and potential limitations or uncertainties in their approach.

• **P2** $[\{0, 1, 2\},\ r{=}1,\ \mathrm{L}]$: Identify any individual submissions that are distinct from other submissions but are scientifically sound. If such distinct submissions exist, the team must collaboratively adjudicate them, explaining why they are correct, how they relate to other submitted ideas, and whether they should be integrated or if the team's shared understanding needs to be revised. If there are no such distinct, scientifically sound submissions, simply state that all sound individual submissions align.

• **P3** $[\{0, 1, 2\},\ r{=}1,\ \mathrm{L}]$: Agent 2 takes on the role of the Minority Evidence Adjudicator. For every individual claim that differs from the emerging consensus, Agent 2 must explicitly state whether the claim is scientifically sound and, if so, integrate it into the collective understanding, explaining its contribution. If the claim is found to be incorrect, Agent 2 must provide a clear, scientific refutation, citing specific evidence or logical flaws. Agents 0 and 1 must engage with Agent 2's adjudication, either confirming or challenging it with further evidence. The team then synthesizes the strongest supported answer, ensuring all valid individual insights are preserved and incorrect ones are formally refuted.

Teamwork prompt. Preserve independent reasoning. Do not converge until each agent's submission and evidence have been compared against the others, and any distinct but scientifically sound individual submissions have been explicitly adjudicated, integrated, or refuted with evidence. Agent 2 is the dedicated Minority Evidence Adjudicator.

Figure 7 illustrates two additional strategies from earlier searches: forced role reversal and overlapping pairwise verification. Neither was selected for either deployed bank.



(a) (b)

Figure 7: **Teamwork reflection explores a broader strategy space than the final banks reveal. (a) Forced role reversal.** Members first vote blindly, then each argues for an answer other than its own, cross-examines the alternatives, and votes again only after the temporary advocacy roles are released. This separates the strength of an argument from its proponent's original answer and forces minority alternatives to be examined. **(b) Overlapping pairwise verification.** Each member states an answer and its critical assumption, after which the strategy verifies all three member pairs in an overlapping sequence. A detected flaw triggers full-team re-solving; if no flaw is found, unanimity still triggers a challenge to one key interpretation before synthesis. These strategies were discovered during earlier searches but were not selected for either final bank and were not used in reported deployment.

F Evaluation Details

For each benchmark, self-consistency and self-reflection use the team member with the highest accuracy in the initial independent single-pass evaluation on that benchmark, breaking ties arbitrarily. Both use o3-mini for AIME 2024, AIME 2025, and AIME 2026; Claude Sonnet 4 for TheoremQA-physics; Gemini-2.5-Flash for GPQA and BBEH; and GPT-4.1 for MMLU-Pro. On HMMT 2026, self-consistency uses Claude Sonnet 4 and self-reflection uses o3-mini; the difference reflects arbitrary tie-breaking, as the two models had identical independent single-pass accuracy.

Self-consistency draws $K{=}10$ samples from the selected model at temperature 0.5. For self-consistency and member-vote, we score plurality ties by expected accuracy under a uniform tie-break: if the correct answer is among t tied top answers, the problem receives $1/t$ credit, and otherwise it receives zero. HMMT 2026 answers are graded with MathArena's benchmark-specific answer checker; other mathematics answers are graded with `math-verify`. MMLU-Pro is graded by exact match to the gold option letter; BBEH uses its official evaluation logic.

All table entries report means over three evaluation seeds. For SAT, linearization, and the homogeneous-team control across all AIME benchmarks, each seed first generates a fixed pool of up to ten candidate certificates per problem. We run five independent final-answer selection passes

over each fixed pool and average across the resulting seed–selection combinations. We use the same five-pass selection aggregation for SAT, linearization, and the homogeneous-team control in the knowledge-and-logic suite. The coverage rows are computed from the candidate pools before final-answer selection and averaged across the three seeds.

F.1 Final-answer selection prompts

The judge receives the benchmark problem followed by every candidate certificate in a single prompt. We use the following task-specific templates, with braced fields populated for each benchmark and problem.

F.1.1 Free-response mathematics and physics

```
You are selecting the best proof-of-work certificate for one {benchmark_name} problem.

You are given the problem and all candidate final team certificates produced by a strategy bank. Choose the
      candidate whose final answer is best supported by its written certificate.

You are auditing written certificates, NOT solving the problem.

Rules:
- Do not solve the problem yourself, and do not select or reject a candidate because its answer matches or
      conflicts with your own solution, estimate, or intuition about the answer.
- Judge only certificate quality: are the load-bearing steps written down and locally checkable, is the
      arithmetic correct as written, is the case analysis complete, and does the final transition to the stated
      answer follow from the established steps?
- You may reject or downgrade a candidate ONLY by naming a specific local defect in its written certificate: a
      step that does not follow from the previous ones, an arithmetic error you can point to, a missing case, an
      unjustified assumption, or a final transition not supported by the steps. "I believe the answer is
      different" or "it conflicts with another candidate's derivation" are NOT defects.
- For each candidate, record all such defects in a named_defects list; use an empty list if no specific defect
      is identified.
- Do not use answer frequency while auditing; audit every certificate on its written content alone. Frequency
      enters ONLY through the mandatory tie-break rule in stage 2.
- If every certificate is flawed, choose the least-bad candidate and name its remaining defect.

Work in two stages, strictly in this order:
1. AUDIT: write a structured audit entry for EVERY candidate, based only on its written certificate.
2. DECIDE: only after all audits are written, compare the audits and select the certificate with the strongest
      written support. MANDATORY TIE-BREAK: if more than one candidate has an empty named_defects list, you MUST
      select from among those defect-free candidates the one whose final answer is the most common answer within
      that defect-free set; if frequencies are tied, prefer the certificate whose audit shows the most complete,
      explicit support. Do not override this tie-break with prose-quality preferences.

Problem ({benchmark_name}):
{problem}

Candidate certificates:
{candidate_certificates}
```

F.1.2 Multiple-choice knowledge and logic

```
You are selecting the best {benchmark_name} final-answer certificate from a pool of candidate reasoning traces.

You are given one multiple-choice question and all candidate final certificates. Choose the candidate whose
      final answer (an option label) is best supported by its written reasoning.

You are auditing written certificates, NOT solving the problem.

Rules:
- Do not solve the problem yourself, and do not select or reject a candidate because its answer matches or
      conflicts with your own solution, estimate, or intuition about the correct option.
- Judge only certificate quality: are the load-bearing claims and decisive steps written down and locally
      checkable, are the calculations and factual claims correct as written, are the option eliminations and case
       distinctions complete, and does the final transition to the stated option follow from the established
      steps?
- You may reject or downgrade a candidate ONLY by naming a specific local defect in its written certificate: a
      step that does not follow from the previous ones, a calculation or factual claim you can point to as wrong
      as written, an unsupported option elimination, a missing case, an unjustified assumption, or a final option
       choice not supported by the stated reasoning. "I believe the answer is different" or "it conflicts with
      another candidate's reasoning" are NOT defects.
- For each candidate, record all such defects in a named_defects list; use an empty list if no specific defect
      is identified.
```

```
- Do not use answer frequency while auditing; audit every certificate on its written content alone. Frequency
      enters ONLY through the mandatory tie-break rule in stage 2.
- If every certificate is flawed, choose the least-bad candidate and name its remaining defect.

Work in two stages, strictly in this order:
1. AUDIT: write a structured audit entry for EVERY candidate, based only on its written certificate.
2. DECIDE: only after all audits are written, compare the audits and select the certificate with the strongest
      written support. MANDATORY TIE-BREAK: if more than one candidate has an empty named_defects list, you MUST
      select from among those defect-free candidates the one whose final answer is the most common answer within
      that defect-free set; if frequencies are tied, prefer the certificate whose audit shows the most complete,
      explicit support. Do not override this tie-break with prose-quality preferences.

Problem ({benchmark_name}):
{problem}

Candidate final certificates:
{candidate_certificates}
```

G ADDITIONAL MECHANISMS OF COLLABORATIVE COMPUTATION

Figures 8 and 9 provide expanded transcripts for the HMMT and GPQA examples discussed in the main text. Two additional AIME cases show how a failed check motivates a missing construction (Figure 10) and how a step audit protects a correct minority (Figure 11). Transcripts are condensed for legibility.

(1) o3-mini: invariant and counting method

For each prime p, the sorted exponent list is invariant, so the final k-th smallest number is $\prod_p p^{E_k(p)}$. Counting by $\lfloor 2100/p^m \rfloor - \lfloor 2100/p^{m+1} \rfloor$: $E_{2026}(2) = 4$, $E_{2026}(3) = 3$, $E_{2026}(5) = 2 \ldots$ For $p = 23$: $2009 + 88 = 2097$.

(2) o3-mini: factorization with one wrong count

Final factorization $2^4 \cdot 3^3 \cdot 5^2 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23$; divisor count $(4{+}1)(3{+}1)(2{+}1)(1{+}1)^{\mathbf{5}}$.

(3) DeepSeek-V3: six unit primes, not five

There are **6** primes $(7, 11, 13, 17, 19, 23)$ with exponent 1, not 5. So $\tau(N) = (4{+}1)(3{+}1)(2{+}1)(1{+}1)^6 = 5 \cdot 4 \cdot 3 \cdot 64 = \boxed{3840}$.

(4) Claude Sonnet 4: independent audit

p=2: 131−65=66, cumulative 2035, and $2026 \leq 2035$, so $E_{2026}(2)$=4. The p=3 and p=5 counts also check; $\tau = 5 \times 4 \times 3 \times 2^6 = 3840$.

(5) o3-mini: synthesis

$N_{2026} = 2^4 \cdot 3^3 \cdot 5^2 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23$; $\tau = (4{+}1)(3{+}1)(2{+}1)(1{+}1)^6 = 3840$. Final answer: $\boxed{3840}$.

Figure 8: **Cross-member composition constructs an answer absent from all three initial outputs.** On HMMT 2026 problem 6 (gold 3840; all three initial answers were incorrect), o3-mini supplies the gcd–lcm exponent invariant and factorization but miscounts the unit-exponent primes. DeepSeek repairs the count, Sonnet audits it, and synthesis preserves the composed answer. The transcript is condensed for legibility.



Figure 9: **A learned peer-audit strategy repairs errors that debate and aggregation preserve.** GPQA Diamond 132 asks the product count for 1-bromobenzene-2-d reacting with $NaNH_2$ in ammonia. All methods use Llama 4 Maverick, GPT-4.1, and Gemini 2.5 Flash; the correct answer is B (three products). **(a) Mixture of Agents** starts with proposals D/D/C. The aggregator merges the H- and D-removal intermediates and incorrectly retains deuterium, counting 2- and 3-deuterioaniline but omitting aniline. It returns D (two). **(b) Debate** starts with a wrong D majority (Llama and Gemini); GPT initially chooses the correct answer B. Gemini's first-round analysis falsely equates the D-retaining pathway's two products. Citing the other agents' analyses, GPT adopts this equivalence and switches to D; Llama adopts the same product accounting. The wrong majority becomes unanimous, omitting 3-deuterioaniline. **(c) SAT** begins with all three agents independently recommending wrong answers: D/A/D (two/four/two). Its learned strategy asks each member to audit a specific peer claim, propose a correction, then acknowledge corrections, explain initial errors, and revise their answer before synthesis. Gemini challenges Llama's merged pathways, then corrects its own assumption that deuterium survives removal. The team preserves the two-pathway mechanism and repairs atom tracking: removing D gives aniline; removing H gives 2- and 3-deuterioaniline. Green marks correct B and light blue incorrect D; teal marks the audit and proposed repair, orange the acknowledged correction. Transcripts and instructions are condensed; quotations are exact.

o3-mini: concrete challenge

With $s = 49$ and $b = 2$, $AI = 2\sqrt{49/100} = 1.4$, not an integer. This is a gap. Focus on the integer condition for AI.

Claude Sonnet 4: repair

For AI to be integer we need $b \times 7/10$ integer, so b must be divisible by 10. The smallest such b is 10, giving $s = 49 \times 10/2 = 245$. The ratio is $500/24 = 125/6$. So $AB = 245$, not 49.

DeepSeek-V3: minimality check

For $b = 5$, $s = 122.5$ is not an integer; the least admissible b is 10, while $b = 20$ gives the larger $AB = 490$. Thus 245 is minimal.

o3-mini: completeness-gate synthesis

Because all sides are integers, b must be divisible by 10; the least value is $b = 10$, giving $s = 245$ and $AI = 7$. Final answer: $\boxed{245}$.

Figure 10: **A failed check becomes the missing construction.** On AIME 2026 II-08 (gold 245; initial outputs 028/049/unparsed), a mandated integer check exposes the gap in the leading derivation. Sonnet turns that failed check into the divisibility constraint needed to construct 245, and DeepSeek verifies minimality.

o3-mini: independent answer 032

Considering negative common differences gives $d = -1, -2, -5, -10$. The positive cases sum to 178 and the negative cases to -146, giving $\boxed{032}$.

Claude Sonnet 4: step audit

For $d = -1$, reaching 24 requires $4 + (k-1)(-1) = 24$, so $k = -19$—but a term index must be positive. This eliminates every negative d, leaving $d \in \{1, 2, 5, 10\}$. The answer is 178, not 032.

DeepSeek-V3: revision

Agreed—including negative d required invalid negative term indices. The tenth terms for $d \in \{1, 2, 5, 10\}$ are $13, 22, 49, 94$, which sum to 178.

Figure 11: **Verification protects a correct minority against a wrong majority.** On AIME 2026 II-01 (gold 178; initial answers 032/178/032), two members share the same locally checkable error. A step audit names that error, both wrong members revise, and the team preserves the lone correct answer that voting would discard.

H Demonstrability Evaluation Details and Sensitivity Analysis

Table 5 reports the raw benchmark values underlying Figure 12. The ten panel judges are Gemma 3 4B IT, GPT-3.5 Turbo, GPT-4.1 Nano, Command R7B, Claude 3 Haiku, Llama 3.1 8B Instruct, Gemma 3 12B IT, Mistral Small 3.2 24B Instruct, Gemma 3n E4B IT, and Qwen 2.5 7B Instruct. None is a member of either deployed team roster.

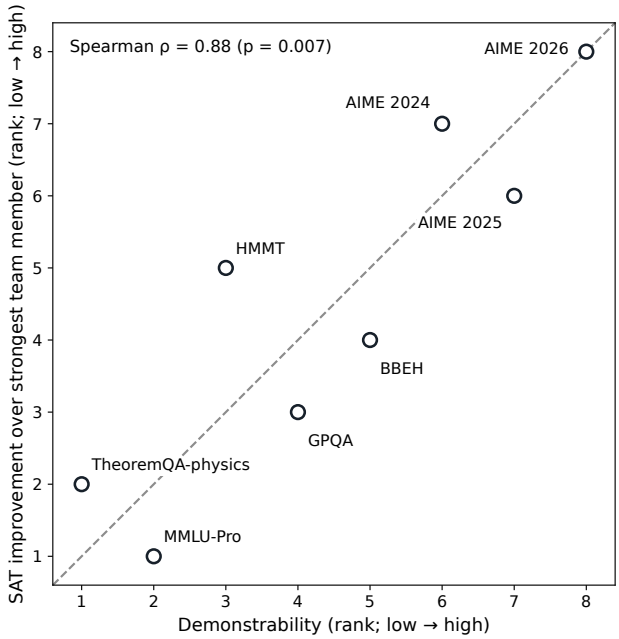


Figure 12: **Demonstrability tracks when self-organizing teams improve over their strongest team member.** Across eight benchmarks, demonstrability strongly tracks the rank order of team improvement over the strongest team member (Spearman $\rho = 0.88$, exact permutation $p = 0.007$). Demonstrability is the balanced rate at which a ten-model panel selects correct over incorrect team reasoning. The axes rank benchmarks by demonstrability and absolute improvement over the strongest team member; the dashed diagonal denotes perfect rank agreement. Raw scores and leave-one-benchmark-out sensitivity results appear in Appendix Tables 5 and 6.

Benchmark	Demonstrability	Improvement over best member (pp)
AIME 2024	0.550	24.5
AIME 2025	0.559	17.3
AIME 2026	0.687	26.2
HMMT 2026	0.519	13.1
TheoremQA-physics	0.423	6.1
GPQA	0.534	7.3
MMLU-Pro	0.517	2.7
BBEH	0.537	10.7

Table 5: **Demonstrability and SAT improvement across eight benchmarks.** Demonstrability is operationalized as the balanced rate at which a ten-model panel selects correct over incorrect reasoning. Improvement is SAT accuracy minus the best member's accuracy, in percentage points.

We test whether the rank association in Section 5 is driven by any single benchmark by recomputing it after leaving out each benchmark in turn. Every seven-benchmark subset retains a strong positive association (Spearman $\rho = 0.82$–0.93; exact permutation $p = 0.007$–0.034; Table 6).

Benchmark omitted	Spearman ρ	Exact p
None (all eight)	0.881	0.007
AIME 2024	0.857	0.024
AIME 2025	0.857	0.024
AIME 2026	0.821	0.034
HMMT 2026	0.929	0.007
TheoremQA-physics	0.857	0.024
GPQA	0.893	0.012
MMLU-Pro	0.857	0.024
BBEH	0.893	0.012

Table 6: **Leave-one-out sensitivity of the demonstrability association.** The first row reports the full eight-benchmark analysis; each subsequent row reports the Spearman correlation after omitting one benchmark. Exact two-sided p-values enumerate all permutations of the observed outcome ranks ($8!$ for the full analysis and $7!$ for each deletion).