STAGEMAD: PROCESS-CONTROLLED COMMUNICATION FOR MULTI-AGENT DEBATE

Anonymous authors
Paper under double-blind review

ABSTRACT

Performing multi-step reasoning with large language models based on a single sampled trajectory remains vulnerable to local errors in intermediate steps, as errors may propagate through subsequent steps and finally lead to a wrong answer. To mitigate this limitation, multi-agent debate (MAD) samples multiple reasoning trajectories and allows agents to exchange information and revise their reasoning before answer aggregation, which is typically performed through majority or plurality voting. However, most existing MAD methods lack reliability constraints on both the content and propagation of exchanged information, so incorrect reasoning can also spread across agents like correct reasoning. As individual responses are not reliable, unverified erroneous reasoning may influence other agents, suppress an initially correct minority, and even produce an incorrect consensus. To address this problem, we propose **StageMAD**, a process-guided multi-agent debate framework consisting of three stages: local self-correction, directed logical guidance based on process evidence, and mutual critique of residual disagreements. By localizing questionable reasoning steps and selectively controlling information propagation, StageMAD enables reliable reasoning to gain broader support while limiting the spread and accumulation of incorrect answers. Experiments on mathematical reasoning and language understanding datasets demonstrate the effectiveness of our proposed framework. The code of StageMAD is available at `https://anonymous.4open.science/r/Stage_MAD`.

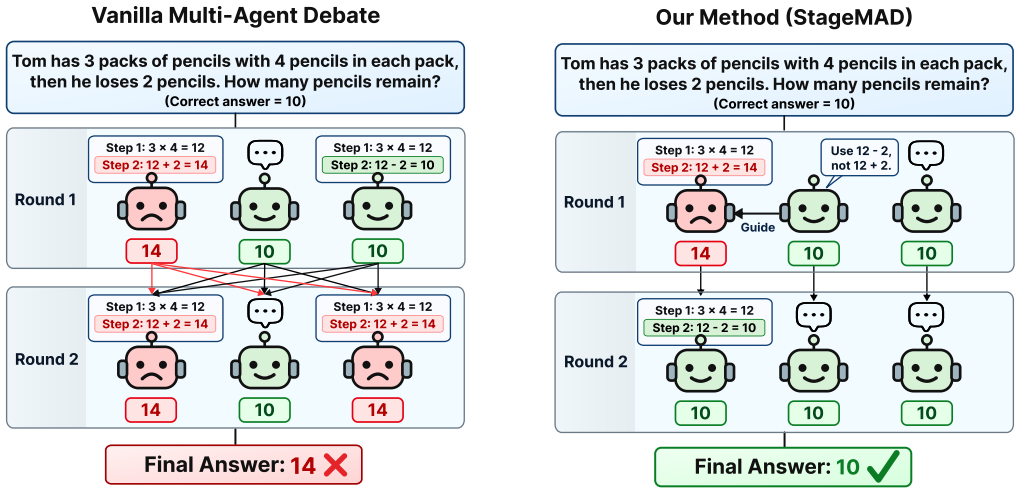
1 INTRODUCTION

Large language models (LLMs) have demonstrated strong capabilities in text generation, language understanding, and complex reasoning (Brown et al., 2020; Wei et al., 2022). By generating explicit chains of intermediate reasoning steps, LLMs can decompose a complex problem and progressively derive its final answer (Wei et al., 2022). However, this reasoning paradigm remains highly dependent on the trajectory produced by a single sample. A locally plausible but critical error in an assumption, calculation, or logical transition may propagate through subsequent steps and invalidate the final answer. Moreover, outcome-level feedback alone does not provide enough information about where an error first occurred, making precise correction difficult (Lightman et al., 2023; Wang et al., 2024a).

To reduce this dependence on a single reasoning trajectory, prior work has exploited complementary information across candidate solutions through repeated sampling, explicit search over reasoning paths, answer aggregation, and multi-agent debate (MAD) (Wang et al., 2023; Yao et al., 2023; Besta et al., 2024; Yin et al., 2023). Among these approaches, multi-agent debate introduces inter-agent communication before answer aggregation (Du et al., 2024; Liang et al., 2024). In MAD, multiple agents first generate candidate solutions independently, then evaluate and revise each other's responses across several rounds of discussion. The system typically determines its final prediction by plurality voting (Du et al., 2024; Kaesberg et al., 2025). By retaining multiple candidate reasoning paths, MAD allows agents to better use peer responses to identify overlooked conditions, calculation errors, or some logical flaws in their own reasoning and compare alternative solutions before the final vote.

However, prior MAD protocols often expose each agent to complete peer responses without assessing the reliability of individual reasoning steps (Du et al., 2024; Chen et al., 2024). MAD-M^2 (Tian et al., 2026) has taken an important step toward reliability-aware communication by masking memories predicted to be erroneous before each debate round. Its control, however, concerns context filtering

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.



(a) Fully connected communication

(b) Controlled communication

Figure 1: Conceptual comparison between fully connected and controlled multi-agent communication. In fully connected MAD, each agent receives all peer responses, so erroneous reasoning may spread and eventually dominate the vote. Controlled communication selectively restricts information propagation, allowing reliable reasoning to provide targeted assistance while limiting harmful influence.

rather than recipient routing: retained peer responses remain available in the next-round context without determining which agents actually need external guidance. As a result, agents whose reasoning is already reliable may still be exposed to unnecessary peer influence, even though communication is primarily needed by agents that exhibit evidence of error. As illustrated in Figure 1, conventional fully connected MAD exposes every agent to all peer responses, allowing an erroneous trajectory to influence multiple agents. Controlled communication instead restricts which responses should propagate according to the current reasoning state. Because the system relies on voting to determine its final answer, an effective debate mechanism must not only correct the erroneous responses but also preserve those that are already correct. The central challenge in MAD is therefore not how to exchange more information, but *how to control when information is transmitted, in what form, and under what conditions.*

Implementing such controlled communication requires fine-grained evidence about where reasoning becomes unreliable. Process reward models (PRMs), developed from process-supervision approaches that provide feedback on intermediate reasoning steps rather than only final outcomes (Uesato et al., 2022; Lightman et al., 2023), provide fine-grained signals that enable controlled debate. Existing PRM research primarily uses such signals for step-level supervision, candidate-answer reranking, or search guided by reward (Lightman et al., 2023; Wang et al., 2024a; Snell et al., 2025), while their use for controlling information propagation in MAD remains underexplored. One direct integration is to select the candidate response with the highest PRM score and use it as a reference for the remaining agents. This design, however, converts local process judgments into a global choice of authority, making final accuracy dependent on the PRM's ability to rank candidate responses correctly.

Motivated by these intuitions, we propose **StageMAD**, a process-controlled framework for staged multi-agent debate. Its central principle is to employ the PRM as a controller of the debate process rather than the final arbiter of correctness. Our framework adopts three successive stages of correction. To prevent early peer influence from overriding potentially correct minority trajectories, the first stage, termed *local self-correction*, isolates agents from peer responses and uses the PRM to localize questionable steps and guide repair. Afterward, agents with localized errors may still benefit from reliable peer evidence, but exposure to complete solutions risks answer copying and error propagation. The second stage, *restricted peer guidance*, therefore uses process evidence to transmit only directed logical hints while withholding the source agent's final answer and full reasoning trajectory. When the first two stages leave near-tied answer groups, no group provides sufficiently reliable authority for unilateral guidance. The third stage, *mutual verification under close votes*, asks competing groups

to exchange verifiable critiques, with each recipient revising only after independently confirming a specific error. In StageMAD, the PRM localizes questionable steps and controls communication, while agents remain responsible for reasoning and revision. This division prevents a single model from simultaneously performing error diagnosis, answer generation, and final adjudication.

Our main contributions in this paper are as follows:

- We reframe MAD as a communication-control problem. Effective debate should selectively enable communication with high corrective potential while restricting those likely to propagate errors or corrupt already-correct reasoning through repeated rounds of debate.
- We propose StageMAD, a process-controlled debate framework that uses a generative PRM to localize questionable steps and condition communication across three stages: local self-correction, restricted peer guidance, and mutual verification under close votes.
- We evaluate StageMAD across three task-model families and four representative mathematical-reasoning and language-understanding benchmarks. It achieves the highest accuracy among communication-based MAD methods in 11 of the 12 main settings and remains competitive with non-communication baselines.

2 Related Work

Inference-time reasoning and aggregation. Chain-of-thought (CoT) prompting exposes intermediate reasoning, but a single sample remains sensitive to a locally incorrect step (Wei et al., 2022). Self-consistency addresses sampling variance by generating independent CoT paths and voting over their final answers (Wang et al., 2023). Tree of Thoughts and Graph of Thoughts further organize intermediate states as search structures that support evaluation, branching, and aggregation (Yao et al., 2023; Besta et al., 2024). A complementary line uses self-generated or structured feedback to refine an individual reasoning trajectory (Pan et al., 2024; Madaan et al., 2023; Paul et al., 2024). These methods improve how candidate trajectories are sampled, searched, or revised, but do not study how agents should expose their reasoning to one another. Round 0 of MAD serves the same diversity-building role as self-consistency; StageMAD preserves this independent pool and makes subsequent cross-agent influence conditional on process evidence.

Multi-agent debate and communication design. MAD lets multiple model instances exchange responses and revise them over several rounds before reaching a final answer (Du et al., 2024). Follow-up frameworks modify different parts of this protocol. Exchange-of-Thought maps cross-model interaction to memory, report, relay, and debate patterns and supplements them with confidence evaluation (Yin et al., 2023). ReConcile combines heterogeneous LLMs in a round-table discussion and uses self-reported confidence in the final consensus (Chen et al., 2024). Sparse MAD restricts the all-to-all topology to reduce redundant communication, whereas adaptive debate selectively activates collaboration based on the confidence of initial responses (Li et al., 2024; Eo et al., 2025). These methods improve model diversity, communication topology, answer aggregation, or determine when debate is necessary. They generally continue to communicate response-level peer content once an interaction is enabled. Empirical comparisons show why this distinction matters: repeated debate does not uniformly outperform independent voting and may change initially correct answers (Smit et al., 2024; Choi et al., 2025). MAD-M^2 is the closest framework to our reliability motivation. Before each debate round, it masks memories predicted to be erroneous, using either an agent's subjective judgment or a perplexity-based objective criterion, and passes the remaining responses into the next-round context (Tian et al., 2026). This improves the quality of visible memory, but its control unit remains an entire preceding response. StageMAD instead uses process evidence to decide which agent needs revision, which answer group may provide guidance, what may be transmitted, and whether another interaction is justified. It therefore extends reliability control from selecting visible memories to coordinating the source, target, content, and timing of communication.

Process supervision and reward models. Process supervision evaluates intermediate steps rather than only the final outcome. Early work compares feedback based on process and outcome for solving mathematical problems, while PRM800K supplies human judgments for training step-level verifiers (Uesato et al., 2022; Lightman et al., 2023). Math-Shepherd reduces dependence on manual annotation by estimating step labels from sampled continuations, enabling PRMs to verify and reinforce reasoning (Wang et al., 2024a). GenPRM replaces a single score judgment with explicit

generative analysis and code-assisted verification before scoring a step (Zhao et al., 2025). Directly using the highest-scoring trajectory as a debate authority would instead make all downstream revisions depend on one global PRM ranking. StageMAD uses the first negative judgment to localize repair and aggregate process evidence to regulate communication; agents perform the revision, and answer-group support determines the final prediction. The PRM thus controls deliberation without serving as the answer selector.

3 Problem Formulation

Let x be a reasoning problem solved by N language-model agents. Round 0 produces independent candidate solutions, and Rounds 1–3 correspond to the three stages of StageMAD. At Round $r \in \{0, 1, 2, 3\}$, agent i produces a reasoning trajectory $\tau_i^r = (s_{i,1}^r, \ldots, s_{i,L_i^r}^r, a_i^r)$, where $s_{i,\ell}^r$ is its ℓth reasoning step, L_i^r is the number of steps, and a_i^r is its normalized final answer. We then define $r_{i,\ell}^r$ as the PRM reward of the reasoning step $s_{i,\ell}^r$ and q_i^r as the average reward of the whole trajectory τ_i^r.

Let $v_1^r, \ldots, v_{K_r}^r$ be the distinct answers produced at Round r. Agents producing answer v_k^r form the group $G_k^r = \{i : a_i^r = v_k^r\}$, whose support is $n_k^r = |G_k^r|$. The system selects the answer with the largest support:

$$\widehat{a}^{\,r} = v_{k_r^\star}^r, \qquad k_r^\star \in \arg\max_{1 \le k \le K_r} n_k^r. \tag{1}$$

Unless otherwise specified, plurality ties are resolved by random selection among the tied answers. StageMAD returns $\widehat{a}^{\,3}$ after the final stage.

The central problem is to control how discussion changes these groups. At each stage, a good system must decide which agents should revise and what information they may receive. These decisions can use the current reasoning traces and process evidence, but never the reference answer. The desired behavior is to move incorrect agents toward the correct answer while limiting the reverse transition.

4 Method

4.1 Overview

Figure 2 provides an overview of StageMAD. The framework adapts the communication mechanism to the current reasoning state through three stages: local self-correction, restricted peer guidance, and mutual verification under close votes.

4.2 Generative process verification

For each response, a frozen generative PRM evaluates every reasoning step and produces a textual critique and a binary judgment $J_{i,\ell}^r \in \{\mathrm{Yes}, \mathrm{No}\}$. We compute the corresponding step reward from the output-token log probabilities and average the step rewards over the trajectory:

$$r_{i,\ell}^r = \frac{\exp \ell_{\mathrm{Yes}}}{\exp \ell_{\mathrm{Yes}} + \exp \ell_{\mathrm{No}}}, \qquad q_i^r = \frac{1}{L_i^r} \sum_{\ell=1}^{L_i^r} r_{i,\ell}^r. \tag{2}$$

The first step judged "No" is treated as the localized defect. The trajectory-level reward q_i^r is used for comparing responses, while the textual critique is used for revision. Neither signal is used to directly determine the final answer.

4.3 Detailed Implementation of StageMAD

Stage 1: Local self-correction. Round 1 uses the PRM to examine each raw solution produced by N agents in Round 0. As illustrated by *Guide with Mask* in Figure 2, the PRM scores the reasoning steps and produces a localized critique at the first step judged negative. Based on this signal, StageMAD preserves the preceding reasoning prefix while masking the potentially affected suffix, which is then regenerated by the agent under the PRM critique. Agents without an explicit defect remain unchanged. As no peer response is provided in this stage, each trajectory is locally

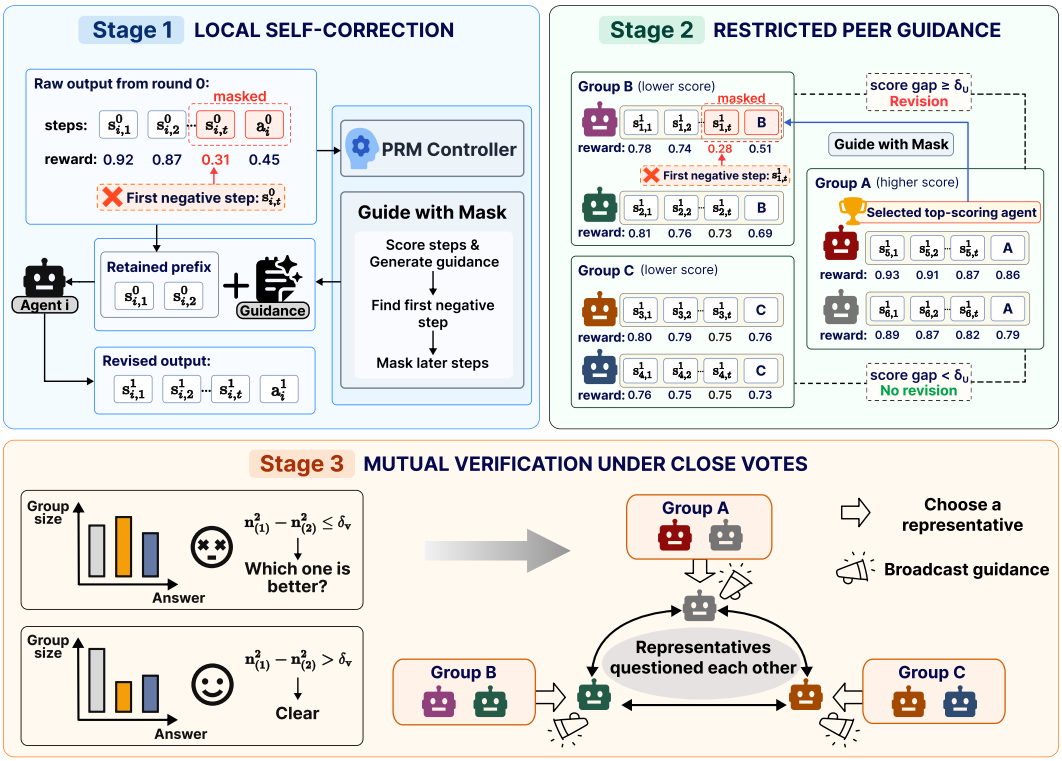


Figure 2: Overview of StageMAD. In Stage 1, agents independently revise localized errors using process-level feedback. In Stage 2, process evidence permits restricted, answer-free guidance from an eligible high-scoring answer group to agents with localized defects. In Stage 3, when the leading answer groups remain closely matched, their representatives exchange verifiable critiques, and each recipient independently decides whether to revise. The PRM controls error localization and communication, whereas the final prediction is determined by plurality voting.

examined before it can be influenced by other answers, reducing premature convergence toward a potentially incorrect consensus.

Stage 2: Restricted peer guidance. Round 2 permits one-way communication when process evidence distinguishes a reliable source from a defective target. For each Round-1 answer group G_k^1, we define

$$R_k^1 = \frac{1}{n_k^1} \sum_{i \in G_k^1} q_i^1, \qquad U_k^1 = R_k^1 \left(\frac{n_k^1}{N} \right)^{\alpha_2} \tag{3}$$

where R_k^1 is the mean process reliability of the trajectories supporting answer v_k^1, and U_k^1 is the corresponding group-selection score after incorporating normalized vote support. We use group support as a weak prior alongside the mean PRM reward. The sublinear exponent α_2 gives additional votes diminishing influence, so a smaller group with stronger process evidence can remain competitive.

The source group G_s^1 is the highest-scoring group containing at least one agent whose reasoning steps are all judged positive. Group G_k^1 becomes eligible for intervention only when the score gap satisfies

$$U_s^1 - U_k^1 \geq \delta_U. \tag{4}$$

Within an eligible target group, only agents with an explicitly localized defect are revised. Among the agents in G_s^1 with no negative step identified by PRM, StageMAD selects the agent with the highest process score as the source. With the help of the PRM, the selected source agent generates short guidance, such as a constraint, equation, invariant, or quantity to recheck. The guidance excludes the source agent's final answer and complete derivation.

The target then undergoes the same *Guide with Mask* operation used in Stage 1: StageMAD retains the target's valid reasoning prefix, masks the suffix beginning at its first PRM-identified negative

step, and supplies the restricted guidance for regeneration. The target independently recomputes the masked suffix rather than copying the source solution.

Stage 3: Mutual verification under close votes. After Round 2, let $n^2_{(1)}$ and $n^2_{(2)}$ denote the two largest answer-group supports. Round 3 is activated only when at least two answer groups remain and

$$n^2_{(1)} - n^2_{(2)} \leq \delta_v. \tag{5}$$

Otherwise, the Round-2 responses are retained without further interaction. The support-gap condition indicates unresolved collective uncertainty, and we relate this condition to a lower bound on the entropy of the answer distribution in Appendix K.

For an activated problem, StageMAD selects one representative from each answer group. Every pair of group representatives generates challenges in both directions. Each challenge identifies a concrete assumption, reasoning step, or calculation for the opposing group to verify, without revealing a final answer or providing a replacement solution. A challenge generated against a representative is then broadcast to all agents belonging to the same target answer group. Each recipient independently checks the challenged point against its own reasoning and revises its response only after confirming a specific error. Because no answer group is treated as authoritative, this reciprocal verification tests competing solutions without requiring agents to follow a single PRM-selected response.

4.4 Final aggregation

If a stage has no eligible revision, its responses are carried forward. After Round 3, the system returns the answer with the largest group support. When a Round-3 intervention creates a top-support tie, StageMAD selects the Round-2 plurality answer if it is among the tied candidates. Otherwise, it selects the first tied answer under the fixed agent ordering. This incumbent-preserving rule prevents an unresolved tie in Round 3 from arbitrarily replacing the preceding collective decision. The PRM localizes and routes, peers provide restricted checks, target agents recompute, and answer-group support determines the final prediction.

5 Experiments

In this section, we evaluate StageMAD on mathematical-reasoning and language-understanding benchmarks. We first compare final accuracy and token cost, then freeze the Round-1 and Round-2 candidate pools to isolate the marginal value of subsequent communication. Finally, we examine how routing, message content, and vote-margin gating affect correction, corruption, and inference cost. Appendix A provides the complete experimental settings.

5.1 Experimental setup

Models. We use Qwen2.5-7B-Instruct (Yang et al., 2025) as the primary task model for the full benchmark evaluation. To examine whether the proposed communication mechanism transfers across model families and reasoning capabilities, we additionally evaluate DeepSeek-Math-7B-Instruct (Shao et al., 2024) and Qwen3.5-9B (Qwen Team, 2026). All agents within a run use the same task model. StageMAD and the PRM self-revision baseline use the same frozen generative PRM within each matched comparison.

Benchmarks. We evaluate StageMAD on GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021), AIME24 (Zhang & Math-AI, 2024), and MMLU-Pro (Wang et al., 2024b). GSM8K evaluates grade-school mathematical reasoning, whereas MATH and the AIME set evaluate increasingly challenging competition-level mathematics. MMLU-Pro evaluates knowledge-intensive multiple-choice reasoning.

Baselines. We evaluate five baselines in our experiments: (1) CoT-SC@15, which aggregates 15 independently sampled reasoning paths without communication (Wang et al., 2023); (2) PRM self-revision, which repeatedly applies localized verifier feedback without peer interaction; (3) MAD, which allows agents to exchange complete responses over multiple discussion rounds (Du et al., 2024); (4) MAD-M^2 (S), which subjectively masks unreliable peer memories through self-assessment; and (5) MAD-M^2 (O), which uses response perplexity to select and retain the memory with the lowest perplexity (Tian et al., 2026).

Table 1: Final Round-3 accuracy and token cost relative to MAD ($\times 1.00$) on four benchmarks. Values are three-run means $\pm$ sample standard deviations. Notation † denotes non-communication baselines; other methods use MAD-style peer communication. Best and second-best accuracies are bold and underlined.

Method	MATH		GSM8K		MMLU-Pro		AIME24	
	Acc. ↑	$T\downarrow$	Acc. ↑	$T\downarrow$	Acc. ↑	$T\downarrow$	Acc. ↑	$T\downarrow$
`Qwen2.5-7B-Instruct`								
CoT-SC@15†	**73.8 ± 0.56**	×0.17	85.5 ± 0.65	×0.17	41.5 ± 0.85	×0.07	13.3 ± 0.00	×0.19
PRM self-revision†	71.3 ± 0.58	×1.74	93.5 ± 0.17	×1.65	**59.0 ± 0.21**	×1.42	13.3 ± 3.34	×2.06
MAD	65.1 ± 0.55	×1.00	91.0 ± 0.47	×1.00	58.4 ± 0.72	×1.00	12.2 ± 1.92	×1.00
MAD-M^2 (S)	67.4 ± 0.91	×1.33	91.7 ± 0.45	×1.33	58.6 ± 0.91	×1.00	13.3 ± 3.34	×1.24
MAD-M^2 (O)	65.7 ± 1.05	×0.63	90.7 ± 0.47	×0.59	56.9 ± 0.78	×0.48	10.0 ± 3.33	×0.64
StageMAD	71.7 ± 0.96	×1.12	**93.8 ± 0.21**	×0.66	58.3 ± 0.40	×0.97	**15.6 ± 1.92**	×1.91
`Qwen3.5-9B`								
CoT-SC@15†	65.0 ± 1.05	×0.13	53.2 ± 0.93	×0.08	64.2 ± 0.72	×0.09	6.7 ± 0.00	×0.14
PRM self-revision†	**79.6 ± 0.49**	×1.06	95.1 ± 0.40	×1.15	**76.5 ± 0.67**	×1.05	**41.1 ± 7.70**	×1.45
MAD	60.5 ± 0.62	×1.00	85.8 ± 0.31	×1.00	61.0 ± 0.12	×1.00	3.3 ± 0.00	×1.00
MAD-M^2 (S)	67.6 ± 0.95	×1.06	89.7 ± 0.70	×1.23	64.9 ± 0.30	×1.13	5.6 ± 3.85	×0.70
MAD-M^2 (O)	66.6 ± 0.65	×0.53	89.8 ± 0.90	×0.53	64.1 ± 1.48	×0.52	4.4 ± 1.92	×0.54
StageMAD	73.6 ± 0.25	×0.66	**95.3 ± 0.15**	×0.45	71.1 ± 0.06	×0.64	17.8 ± 5.09	×1.13
`DeepSeek-Math-7B-Instruct`								
CoT-SC@15†	**57.7 ± 0.85**	×0.18	88.4 ± 0.59	×0.19	33.1 ± 1.80	×0.25	**1.1 ± 1.92**	×0.21
PRM self-revision†	55.1 ± 1.01	×2.15	87.7 ± 0.36	×1.55	27.4 ± 0.70	×3.45	**1.1 ± 1.92**	×4.01
MAD	44.2 ± 0.30	×1.00	77.2 ± 0.47	×1.00	21.7 ± 0.60	×1.00	0.0 ± 0.00	×1.00
MAD-M^2 (S)	44.3 ± 0.53	×1.16	81.6 ± 0.40	×0.99	29.1 ± 0.79	×1.08	0.0 ± 0.00	×1.22
MAD-M^2 (O)	44.8 ± 0.60	×0.62	81.9 ± 0.55	×0.46	29.6 ± 0.86	×0.55	0.0 ± 0.00	×0.58
StageMAD	55.3 ± 1.15	×1.80	**89.1 ± 0.06**	×0.81	**34.0 ± 1.48**	×2.30	**1.1 ± 1.92**	×3.41

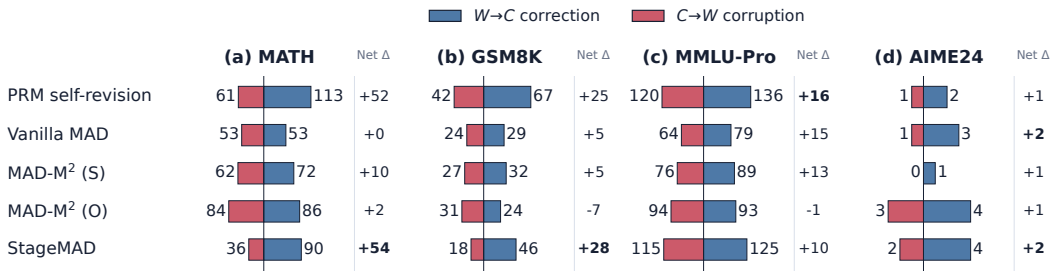


Figure 3: Pooled question-level correctness transitions on four benchmarks using Qwen2.5-7B-Instruct. Rightward blue bars count beneficial corrections ($W \to C$), and leftward rose bars count harmful corruptions ($C \to W$). Counts pool the three adjacent round pairs from one complete run of each method. Net Δ is $W \to C - C \to W$; bold marks the largest net gain within each benchmark.

5.2 Main results

Table 1 reports final accuracy and normalized token cost on four representative benchmarks and three task models.

Observation 1: StageMAD provides the strongest overall performance among MAD methods. StageMAD achieves the highest accuracy among communication-based MAD methods in 11 of the 12 model–benchmark settings. The largest improvements over MAD occur with Qwen3.5 on MATH (+13.1 points), MMLU-Pro (+10.1), and AIME24 (+14.5), and with DeepSeek-Math on MATH (+11.1), GSM8K (+11.9), and MMLU-Pro (+12.3). The only exception is Qwen2.5 on

MMLU-Pro, where StageMAD is 0.3 points below MAD-M^2 (S). These results show that controlled communication generalizes across task-model families. Language-oriented failure modes are further analyzed in Appendix I.2.

Observation 2: StageMAD offers a favorable accuracy–cost trade-off. Among MAD methods, StageMAD lies on the empirical accuracy–cost Pareto frontier in all 12 settings. It uses fewer tokens than MAD in six settings, including MATH, GSM8K, and MMLU-Pro with Qwen3.5, while retaining the highest MAD-based accuracy. Against the non-communication PRM self-revision baseline, StageMAD is competitive with Qwen2.5 and DeepSeek-Math at lower cost on several benchmarks, but self-revision remains stronger with Qwen3.5. Thus, the benefit of peer communication depends on how effectively the task model already exploits localized PRM feedback.

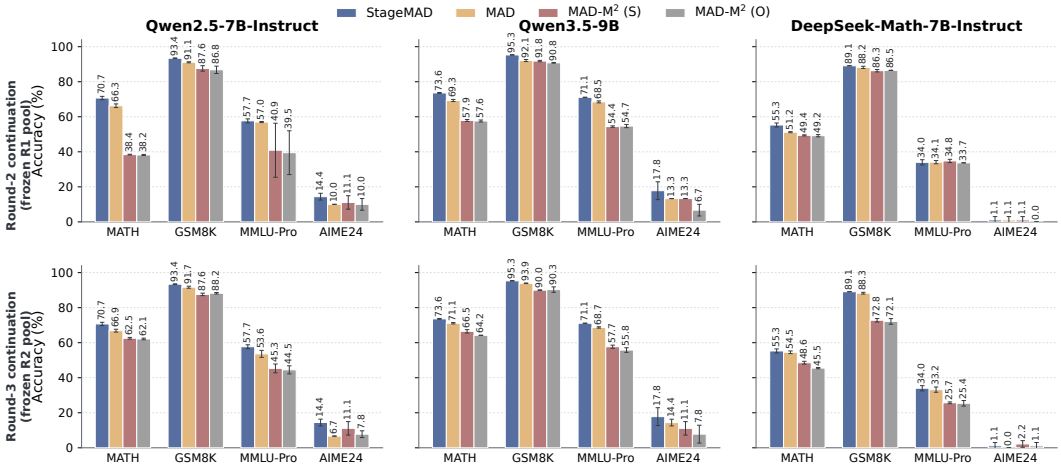


Figure 4: Matched-pool communication-policy ablations across three task models. The top row freezes each StageMAD Round-1 candidate pool and varies the downstream Round-2–3 policy; the bottom row freezes the Round-2 pool and varies only the Round-3 policy. Bars report three-run mean accuracy and error bars show sample standard deviation. Every policy therefore starts from the same candidates within a model, dataset, and row.

5.3 More analysis

Communication beyond local correction. Figure 4 separates the value of StageMAD's communication policy from the quality of its preceding candidate pool. After Round 1 is fixed, StageMAD obtains the highest or tied-highest MAD-based continuation accuracy in 11 of the 12 model–benchmark settings, including all four Qwen2.5 benchmarks. Its advantage therefore does not reduce to local self-correction producing a stronger pool. Freezing the complete state through Round 2 provides a stricter test: StageMAD's final-stage policy is best in 11 of 12 settings, with only DeepSeek-Math on AIME24 favoring MAD-M^2 (S). Replacing either continuation with unrestricted MAD or MAD-M^2 can degrade exactly the same candidates, showing that the stage-specific communication rule—rather than additional generation alone—accounts for the gain. Exact shared-pool results and disaggregated Pareto frontiers are reported in Appendices C and D.

Correction–corruption balance. Figure 3 aggregates question-level transitions across the three adjacent round pairs for Qwen2.5-7B-Instruct. On MATH, self-revision produces more corrections than StageMAD (113 versus 90), but also substantially more corruptions (61 versus 36); their net gains are consequently $+52$ and $+54$. StageMAD similarly obtains the largest net gain on GSM8K ($+28$), with a correction-to-corruption ratio of 2.56, whereas MAD-M^2 (O) is net negative (-7). The advantage is not uniform: on MMLU-Pro, StageMAD produces 125 corrections and 115 corruptions, so its $+10$ net gain is smaller than those of self-revision and MAD. AIME24 contains too few questions for stable ratio comparisons. These results show why correction volume alone is insufficient: performance improves only when additional corrections exceed the correct answers corrupted along the way. The full five-benchmark results and round-wise decomposition are provided in Appendix H.

Table 2: Round-3 vote-gap gate ablation from the same frozen StageMAD Round-2 pools. Coverage is the percentage of questions entering Round 3, and Net is $W \to C - C \to W$.

Dataset	Policy	Coverage	R2 Acc.	R3 Acc.	Net
MATH	Vote-gap gate	24.8	71.4	**72.0**	+6
MATH	Ungated	52.1	71.4	71.7	+3
GSM8K	Vote-gap gate	4.7	93.3	**93.4**	+1
GSM8K	Ungated	18.4	93.3	92.3	−10

Restricted message content. The matched content ablation in Table 9 holds the Round-1 states, PRM scores, revision targets, and final-stage policy fixed. On MATH, complete peer solutions more than double the local correction rate of answer-free guidance (38.9% versus 18.3%), but also raise source-answer agreement from 20.3% to 76.7% and finish slightly lower in final accuracy (71.8% versus 72.0%). On MMLU-Pro, full solutions again produce the most corrections but corrupt 85.6% of initially correct targets, and majority guidance rather than full-solution transmission gives the best final result. Restricting the message therefore reduces direct answer following and can improve the correction–corruption balance, even when it does not maximize the number of successful revisions.

Selective stopping. Table 2 compares the vote-gap gate with an otherwise identical ungated Round 3. On MATH, gating reduces coverage from 52.1% to 24.8% while raising accuracy from 71.7% to 72.0%; on GSM8K, it reduces coverage from 18.4% to 4.7% while raising accuracy from 92.3% to 93.4%. The corresponding net changes improve from +3 to +6 and from −10 to +1, respectively. More discussion therefore incurs additional cost without necessarily adding corrective value.

The exact Round-2 vote patterns in Table 10 further show why the margin is useful. Large-margin patterns provide little consistent gain: the 4–1 stratum improves only slightly on MATH and declines on GSM8K, while the 3–1–1 stratum loses accuracy on both datasets. After two stages, a clear leader often indicates that the agents have already extracted most of the available shared evidence, so further critique has limited room to correct remaining errors and can destabilize correct answers. When support is more dispersed, earlier guidance has not resolved the competing answers, and reciprocal verification can add value; the clearest positive strata are 2–1–1–1 on MATH and 2–2–1 on GSM8K. Not every close vote improves, but the gate shifts computation toward cases with greater empirical corrective potential. This supports treating communication as a conditional intervention whose expected benefit must exceed its corruption risk and cost.

6 Conclusion

We formulate MAD as a communication-control problem: communication should be activated only when its expected corrective value outweighs the risk of corrupting correct reasoning. Based on this principle, we introduce StageMAD, which adapts the degree and form of interaction across three stages. Agents first repair local errors without peer communication, then receive restricted process-level guidance when process evidence clearly separates a reliable source from a defective target, and finally exchange reciprocal critiques only under unresolved close votes. Experiments across three task-model families and four reasoning and understanding benchmarks show that StageMAD consistently outperforms existing MAD methods on mathematical reasoning and remains competitive with repeated PRM self-revision while often using fewer tokens. Shared-pool, transition, and gating analyses further show that these gains arise from improving the correction–corruption balance rather than simply increasing communication.

7 Limitations

StageMAD inherits errors from its generative PRM: a polished but incorrect derivation may receive positive judgments and become an eligible source. A math-focused verifier may also transfer poorly to other domains, as reflected by the mixed MMLU-Pro and CommonsenseQA results in Appendix I.2.

With Qwen3.5-9B, repeated PRM self-revision outperforms StageMAD on all four benchmarks from the same frozen Round-1 pool. Its agent-level transitions in Appendix I.3 suggest that stronger use of localized PRM feedback leaves less headroom for peer guidance while retaining corruption risk.

AI USE STATEMENT

In this work, we used generative AI tools to assist with translation, literature search and discovery, drafting and editing portions of the manuscript and providing suggestions to the experimental design.

We did not use generative AI tools to generate synthetic datasets or to clean or reformat the evaluation datasets. We did not use generative AI tools to prove mathematical claims. All AI-assisted outputs were treated as suggestions rather than authoritative results. The authors manually verified the cited literature and reviewed the AI-assisted code. Three of the authors reviewed the AI-assisted material and take responsibility for the final content of this work, including all text, claims, code, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

We believe that our work raises no significant ethical concerns.

REPRODUCIBILITY STATEMENT

Section 4 specifies the StageMAD protocol, including process verification, source–target selection, communication gates, and final aggregation. Appendix A reports the evaluated models, dataset construction, generation settings, controller thresholds, hardware, reporting protocol, and token-accounting procedure. Appendix B provides the complete prompts. The remaining appendices report shared-pool comparisons, stage-wise results, controller and message-content ablations, correctness transitions, token complexity, and the proof of the vote-margin result. An anonymous implementation is available at `https://anonymous.4open.science/r/Stage_MAD`.

REFERENCES

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 17682–17690, 2024. doi: 10.1609/aaai.v38i16.29720. URL `https://ojs.aaai.org/index.php/AAAI/article/view/29720`.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2020.

Justin 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. Association for Computational Linguistics, 2024.

Hyeong Kyu Choi, Jerry Zhu, and Sharon Li. Debate or vote: Which yields better decisions in multi-agent large language models? In *Advances in Neural Information Processing Systems*, pp. 101732–101764. Curran Associates, Inc., 2025.

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

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11733–11763, 2024.

Sugyeong Eo, Hyeonseok Moon, Evelyn Hayoon Zi, Chanjun Park, and Heuiseok Lim. Debate only when necessary: Adaptive multiagent collaboration for efficient LLM reasoning. *arXiv preprint arXiv:2504.05047*, 2025.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/be83ab3ecd0db773eb2dc1b0a17836a1-Paper-round2.pdf.

Lars Benedikt Kaesberg, Jonas Becker, Jan Philip Wahle, Terry Ruas, and Bela Gipp. Voting or consensus? decision-making in multi-agent debate. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 11640–11671. Association for Computational Linguistics, 2025.

Yunxuan Li, Yibing Du, Jiageng Zhang, Le Hou, Peter Grabowski, Yeqing Li, and Eugene Ie. Improving multi-agent debate with sparse communication topology. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7281–7294. Association for Computational Linguistics, 2024.

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

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

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-Refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*, 2023.

Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. Automatically correcting large language models: Surveying the landscape of diverse automated correction strategies. *Transactions of the Association for Computational Linguistics*, 12:484–506, 2024. doi: 10.1162/tacl_a_00660. URL https://aclanthology.org/2024.tacl-1.27/.

Debjit Paul, Mete Ismayilzada, Maxime Peyrard, Beatriz Borges, Antoine Bosselut, Robert West, and Boi Faltings. REFINER: Reasoning feedback on intermediate representations. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1100–1126, 2024. doi: 10.18653/v1/2024.eacl-long.67. URL https://aclanthology.org/2024.eacl-long.67/.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Andries Petrus Smit, Nathan Grinsztajn, Paul Duckworth, Thomas D Barrett, and Arnu Pretorius. Should we be going MAD? A look at multi-agent debate strategies for LLMs. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 45883–45905. PMLR, 2024.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4FWAwZtd2n.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4149–4158. Association for Computational Linguistics, 2019.

Hongduan Tian, Xiao Feng, Ziyuan Zhao, Xiangyu Zhu, Rolan Yan, and Bo Han. Multi-agent debate with memory masking. *arXiv preprint arXiv:2603.20215*, 2026.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-Shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439. Association for Computational Linguistics, 2024a.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023. URL https://mlanthology.org/iclr/2023/wang2023iclr-selfconsistency/.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*, volume 37, pp. 95266–95290. Curran Associates, Inc., 2024b.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 2022.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 11809–11822. Curran Associates, Inc., 2023.

Zhangyue Yin, Qiushi Sun, Cheng Chang, Qipeng Guo, Junqi Dai, Xuanjing Huang, and Xipeng Qiu. Exchange-of-thought: Enhancing large language model capabilities through cross-model communication. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 15135–15153. Association for Computational Linguistics, 2023.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (AIME) 2024, 2024.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (AIME) 2025, 2025.

Jian Zhao, Runze Liu, Kaiyan Zhang, Zhimu Zhou, Junqi Gao, Dong Li, Jiafei Lyu, Zhouyi Qian, Biqing Qi, Xiu Li, and Bowen Zhou. GenPRM: Scaling test-time compute of process reward models via generative reasoning. *arXiv preprint arXiv:2504.00891*, 2025.

A Experimental Details

Hardware and shared protocol. Experiments ran on NVIDIA A800 80GB PCIe GPUs. Qwen2.5-7B-Instruct served as the primary task model, while Qwen3.5-9B and DeepSeek-Math-7B-Instruct

tested cross-model transfer. Unless varied in a sensitivity experiment, all multi-agent methods used five agents, one independent-proposal round, and three subsequent revision or deliberation rounds.

Datasets and subset construction. Table 3 summarizes the evaluation-set sizes, task types, and model-specific maximum completion lengths. For MATH, we shuffle the test set with random state 0, discard examples whose reference answer cannot be extracted, and retain the first $1{,}000$ valid examples. For GSM8K, we shuffle the test set with random state 0 and retain the first $1{,}000$ examples. For MMLU-Pro, we filter examples with invalid answer formats and select $1{,}000$ examples approximately uniformly across subject categories using random state 0. AIME24 and AIME25 (Zhang & Math-AI, 2024; 2025) use all 30 problems. The same subset is shared by every method within each benchmark.

Table 3: Evaluation sets and task-model generation budgets. The three model columns report the maximum completion length per response.

Dataset	Examples	Max completion tokens			Task type
		Qwen2.5	DeepSeek-Math	Qwen3.5	
MATH	1,000	1024	1024	1024	competition math
GSM8K	1,000	512	512	512	grade-school math
MMLU-Pro	1,000	1024	512	1024	multi-choice reasoning
AIME24	30	1024	512	1024	competition math
AIME25	30	1024	512	1024	competition math

Generation settings. Task-model responses are sampled with temperature 1.0 and top-$p = 0.9$. Qwen3.5 thinking mode is disabled. The generative PRM uses one sample per evaluated prefix, with temperature 0.6, top-$p = 0.95$, top-$k = 20$, and a maximum of $2{,}048$ generated tokens. Round-3 challenge and verification calls use greedy decoding with temperature 0, top-$p = 1.0$, and control seed 0. Their maximum completion lengths are 256 and 384 tokens, respectively. All other task-model completion limits are listed in Table 3.

Controller configuration. StageMAD used one generative-PRM sample for each evaluated reasoning prefix, the group-support exponent $\alpha_2 = 0.25$, and a Round-3 top-two support-gap threshold of one. All the experiments used a PRM group-score margin of 0.02. These controller settings apply only to StageMAD; all methods use the specific completion caps listed in Table 3, ensuring matched per-response limits across comparisons.

Reporting and token accounting. Accuracy is computed from the final Round-3 system answer. Token cost includes the recorded prompt and completion tokens of both the task model and the PRM. Because CoT, CoT-SC@5, and CoT-SC@15 are evaluated from the same 15-path pool, the first two costs are reported as proportional budget equivalents of the complete pool. The separately recorded shared Round-0 cache is already included in each iterative method total and is not added again.

B Prompt Templates

This section reports the prompts used by CoT-SC, PRM self-revision, vanilla MAD, and StageMAD. Text in angle brackets denotes a runtime substitution. For mathematical tasks, the required final line is {`final answer: <ANSWER>`}; for multiple-choice tasks, it is {`final answer: (<OPTION>)`}. When step segmentation is enabled, every reasoning step except the final-answer line ends with `<extra_0>`.

B.1 CoT-SC and the shared independent proposal

Round-0 proposal. Each sample independently receives the same question and formatting instructions. CoT-SC repeats this prompt and aggregates the sampled final answers; StageMAD and MAD use it to construct their Round-0 candidate pools.

CoT-SC / Shared Round-0 Proposal Prompt

```
<QUESTION>

Make sure to state your final answer in curly brackets at the very end
of your response, just like: "{final answer: <ANSWER>}".

Please write your reasoning as several clear steps. End each reasoning
step before the final answer line with the exact separator token
<extra_0>. Do not put the separator after the final answer line.
```

B.2 GenPRM

To evaluate Step <STEP_INDEX>, the user message contains the question and the trajectory prefix ending at that step. Generation begins from the following assistant prefix:

GenPRM Step-Level Verification Prompt

```
[User]
Question: <QUESTION>

<REASONING_PREFIX_THROUGH_STEP_INDEX>

[Assistant prefix]
<analyze>
Let's analyze the Paragraph <STEP_INDEX> step by step:
```

The model produces a critique followed by a Yes/No judgment. If the judgment cannot be parsed, StageMAD requests only the missing decision using the following continuation:

GenPRM Judgment Continuation

```
</analyze>
<output>
**Judgement**: $\boxed$
```

B.3 Stage 1: local self-correction

When the PRM identifies the first negative step, the suffix beginning at that step is replaced by a mask. The target agent receives only its retained prefix and its own localized PRM feedback. StageMAD applies this operation in Stage 1, whereas PRM self-revision applies the same template in every revision round without peer communication.

Local PRM Self-Revision Prompt

```
You are <TARGET_AGENT>.

This was your most recent opinion:
<RETAINED_PREFIX>
<MASK: THE LOCALIZED SUSPICIOUS STEP STARTS HERE>

GenPRM feedback on your own reasoning:
GenPRM explicitly localized Step <STEP_INDEX> for self-revision. Use the
following process feedback to repair your own reasoning without adopting
another agent's answer:
<LOCALIZED_PRM_CRITIQUE>

The explicitly suspicious suffix has already been masked when possible.
Use the process feedback as a hint for what to preserve or recompute, then
revise from the masked point onward. Keep the revised reasoning short: use
at most 3 concise steps, one sentence per step, do not repeat already-correct
earlier steps, and reserve enough tokens for the final answer line. Then give
your final answer to the question:
<QUESTION>
```

```
Your last line must be exactly in this format: {final answer: <ANSWER>}.
Do not use LaTeX, boxed, or extra braces in the final answer line.
```

B.4 STAGE 2: RESTRICTED PEER GUIDANCE

Source guidance generation. The selected source privately receives its own response, the target's scored trajectory, and the localized PRM critique. The source must return an answer-free, constraint-level hint:

StageMAD Stage 2: Source Guidance Prompt

```
You are <SOURCE_AGENT>, a source agent in a multi-agent debate. Speak
directly to <TARGET_AGENT> in your own peer voice. Your role is not to
disclose your solution; your role is to convert the private GenPRM
localization into constraint-level guidance that helps the target
self-revise the masked suffix of their own reasoning. You may use your
private answer/reasoning only as an internal sanity check. Do not reveal
your final answer, quote your reasoning, summarize your reasoning, or push
the target toward a specific answer value.

Question:
<QUESTION>

Private context visible only to you, the source agent.
Your private final answer: <SOURCE_ANSWER>
Your private reasoning notes, for internal sanity checking only. Do not
quote, summarize, or reveal these notes to the target:
<SOURCE_REASONING>

Target group selection score: <TARGET_GROUP_SCORE>
Target answer group average reward: <TARGET_GROUP_REWARD>
Confidence margin for sending guidance: <SCORE_MARGIN>
Target agent: <TARGET_AGENT>
Target final answer: <TARGET_ANSWER>
Target localized suspicious step: Step <REVISION_STEP>
Target lowest-reward step: Step <LOWEST_REWARD_STEP>
Target reasoning with GenPRM rewards and private critique:
<TARGET_STEPS_REWARDS_AND_CRITIQUE>

Write one concise piece of guidance in first-person peer voice, addressed
directly to the target agent. Focus especially on the first parsed step
GenPRM explicitly judged incorrect. Since the suspicious suffix will be
masked before the target revises, give only constraint-level guidance: what
definition, equation, invariant, case split, unit conversion, or calculation
should be preserved/recomputed from the masked point. Avoid verdict language
like 'your step is wrong'. Avoid any final answer value. Keep it at most
2 sentences.

Return only valid JSON. Do not include markdown, explanations, or text
outside JSON. Use exactly this schema:
{"guidance": "<CONSTRAINT_LEVEL_PEER_GUIDANCE>"}
If there is no useful guidance, return exactly: {"guidance": ""}.
Do not provide your own final answer.
```

Target revision. The target uses the same masked-revision prompt as Stage 1, augmented with the source hint. The source response and source answer are not included.

StageMAD Stage 2: Target Revision Prompt

```
You are <TARGET_AGENT>.

This was your most recent opinion:
<RETAINED_PREFIX>
<MASK: THE LOCALIZED SUSPICIOUS STEP STARTS HERE>

GenPRM feedback on your own reasoning:
<LOCALIZED_PRM_FEEDBACK>

You also received constraint-level peer guidance about how to continue
after the masked part of your reasoning:
```

```
Guidance from <SOURCE_AGENT>:
<CONSTRAINT_LEVEL_PEER_GUIDANCE>

The explicitly suspicious suffix has already been masked when possible.
Use the process feedback and any peer guidance as a hint for what to preserve
or recompute, then revise from the masked point onward. Keep the revised
reasoning short: use at most 3 concise steps, one sentence per step, do not
repeat already-correct earlier steps, and reserve enough tokens for the final
answer line. Then give your final answer to the question:
<QUESTION>

Your last line must be exactly in this format: {final answer: <ANSWER>}.
Do not use LaTeX, boxed, or extra braces in the final answer line.
```

B.5 Stage 3: mutual verification under close votes

When the vote-gap gate is active, a representative of each answer group challenges every competing group. A challenge is broadcast to all agents in the target group, but each target verifies it independently.

Challenge generation.

StageMAD Stage 3: Challenge Generation Prompt

```
You are one participant in a math debate. Another answer group disagrees
with your group. Challenge one specific inferential step in the other
group's reasoning. Do not act as a judge, do not merely state a different
final answer, and do not write a replacement solution. Ask for one concrete
algebraic, arithmetic, logical, or constraint check. Return JSON only.

Question:
<QUESTION>

Your group's representative reasoning (private context):
<SOURCE_GROUP_REPRESENTATIVE_RESPONSE>

Other group's representative reasoning:
<TARGET_GROUP_REPRESENTATIVE_RESPONSE>

Return exactly these fields:
{"challenge":"...","claimed_conflict_step":"...",
 "check_to_perform":"..."}
```

Independent verification.

StageMAD Stage 3: Independent Verification Prompt

```
You are verifying your own most recent math solution after receiving peer
challenges. Do not write a new solution and do not change your answer merely
because a peer disagrees. Independently perform the smallest concrete check
requested. Return exactly one verdict: error_confirmed if the check proves a
specific error in your reasoning; plausible_error if there may be a problem
but the check is not conclusive; no_error if the challenged step survives
the check; or inconclusive if the check cannot be completed. Do not combine
these outcomes or emit additional decision booleans. For error_confirmed,
identify the affected step, the check actually performed, and its observed
result. Use plain-text math without LaTeX backslashes. Return JSON only.

Question:
<QUESTION>

Your current solution:
<TARGET_RESPONSE>

Peer challenges:
<CHALLENGES_AND_REQUESTED_CHECKS>

Return exactly these fields:
{"verdict":"error_confirmed|plausible_error|no_error|inconclusive",
 "affected_step":"...","check_performed":"...",
 "observed_result":"...","reason":"..."}
```

Verified revision. Only an `error_confirmed` verdict activates the following revision prompt:

StageMAD Stage 3: Verified Revision Prompt

```
You are revising your own solution. Your independent verification found a
concrete failed check. Repair that localized issue, recompute all quantities
that depend on it, and keep unrelated valid reasoning stable. Do not copy a
peer's final answer. Write a complete corrected solution. End each reasoning
step before the final answer with <extra_0>; do not put it after the final
answer.

Question:
<QUESTION>

Previous solution:
<PREVIOUS_RESPONSE>

Accepted challenges:
<ACCEPTED_CHALLENGES>

Your verification evidence:
Affected step: <AFFECTED_STEP>
Check performed: <CHECK_PERFORMED>
Observed result: <OBSERVED_RESULT>
Reason: <REASON>

<DATASET_SPECIFIC_FINAL_ANSWER_INSTRUCTION>
```

B.6 MAD

MAD exposes each agent to every peer response before requesting a revision:

MAD Debate-Round Prompt

```
These are the recent opinions from other agents:

One of the agents' response:
<PEER_RESPONSE_1>

...

This was your most recent opinion:
<OWN_PREVIOUS_RESPONSE>

Use these opinions carefully as additional advice to revise your recent
opinion to give your final answer to the question:
<QUESTION>

<DATASET_SPECIFIC_FINAL_ANSWER_INSTRUCTION>
```

C Complete Matched-Pool Accuracy Results

C.1 Continuation from frozen Round-1 pools

Table 4 reports the complete results obtained by applying alternative downstream policies to the same StageMAD Round-1 pools.

C.2 Continuation from frozen Round-2 pools

Table 5 reports the exact results underlying the bottom row of Figure 4. Within each model, benchmark, and seed, all policies receive the same StageMAD state through Round 2 and differ only in the Round-3 communication policy. Across the four main-text benchmarks, StageMAD obtains the highest final accuracy in 11 of the 12 model–benchmark settings.

Table 4: Accuracy (%) before and after continuation from frozen Round-1 pools. MATH, GSM8K, and MMLU-Pro report three-run mean ± sample standard deviation; the updated Qwen2.5 block also reports standard deviations on AIME, whereas the other AIME blocks report means only. Within each model, all methods share the same Round-1 pool. Bold and underlining mark the highest and second-highest means, respectively.

Downstream policy	MATH	GSM8K	MMLU-Pro	AIME24	AIME25
`Qwen2.5-7B-Instruct`					
Round-1 accuracy	69.0 ± 0.53	93.0 ± 0.15	55.7 ± 2.61	10.0 ± 3.33	$\underline{7.8 \pm 5.09}$
MAD	66.3 ± 1.05	91.1 ± 0.30	$\underline{57.0 \pm 0.31}$	10.0 ± 0.00	5.6 ± 3.85
PRM self-revision	$\mathbf{71.4 \pm 0.25}$	$\underline{93.2 \pm 0.40}$	47.6 ± 18.84	$\underline{12.2 \pm 3.85}$	$\mathbf{8.9 \pm 6.94}$
MAD-M^2 (S)	38.4 ± 0.20	87.6 ± 1.57	40.9 ± 15.43	11.1 ± 3.85	$\underline{6.7 \pm 0.00}$
MAD-M^2 (O)	38.2 ± 0.32	86.8 ± 2.11	39.5 ± 12.50	10.0 ± 3.33	4.4 ± 5.09
StageMAD	$\underline{70.7 \pm 0.95}$	$\mathbf{93.4 \pm 0.20}$	$\mathbf{57.7 \pm 1.11}$	$\mathbf{14.4 \pm 1.93}$	$\underline{6.7 \pm 3.34}$
`Qwen3.5-9B`					
Round-1 accuracy	72.5 ± 0.76	94.9 ± 0.06	$\underline{71.5 \pm 0.55}$	14.4	12.2
MAD	69.3 ± 0.61	92.1 ± 0.60	68.5 ± 0.56	13.3	10.0
PRM self-revision	$\mathbf{79.9 \pm 0.75}$	$\mathbf{95.5 \pm 0.32}$	$\mathbf{76.9 \pm 0.42}$	**35.6**	**30.0**
MAD-M^2 (S)	57.9 ± 0.55	91.8 ± 0.40	54.4 ± 0.42	13.3	10.0
MAD-M^2 (O)	57.6 ± 0.57	90.8 ± 0.21	54.7 ± 0.91	6.7	10.0
StageMAD	$\underline{73.6 \pm 0.25}$	$\underline{95.3 \pm 0.15}$	71.1 ± 0.06	$\underline{17.8}$	$\underline{18.9}$
`DeepSeek-Math-7B-Instruct`					
Round-1 accuracy	54.3 ± 1.14	$\underline{88.9 \pm 0.20}$	33.6 ± 0.15	**1.1**	$\underline{1.1}$
MAD	51.2 ± 0.30	88.2 ± 0.64	$\underline{34.1 \pm 0.92}$	**1.1**	**2.2**
PRM self-revision	$\mathbf{55.4 \pm 1.10}$	$\underline{88.9 \pm 0.06}$	31.6 ± 0.35	**1.1**	$\underline{1.1}$
MAD-M^2 (S)	49.4 ± 0.29	86.3 ± 0.69	$\mathbf{34.8 \pm 0.92}$	**1.1**	$\underline{1.1}$
MAD-M^2 (O)	49.2 ± 0.62	86.5 ± 0.00	33.7 ± 0.06	0.0	0.0
StageMAD	$\underline{55.3 \pm 1.15}$	$\mathbf{89.1 \pm 0.06}$	34.0 ± 1.48	**1.1**	0.0

D Accuracy–Cost Pareto Analysis

Figure 5 disaggregates the accuracy–cost comparison by task model and benchmark. Each panel uses the three-run mean accuracy and token cost normalized by MAD. StageMAD lies on the empirical Pareto frontier among MAD-based communication methods in all 12 settings and remains on the global empirical frontier in 9 settings after including CoT-SC@15 and PRM self-revision.

E Full Stage-wise Results

Figures 6–8 compare all five iterative methods over Rounds 0–3 on all five benchmarks. Each task model is shown in a separate five-panel figure. Round 0 is the independent proposal pool, while Rounds 1–3 follow each method's own revision protocol. Lines and shaded bands report three-run means and sample standard deviations, respectively; AIME24 and AIME25 contain only 30 questions each.

The trajectories expose three different regimes. With Qwen2.5, StageMAD gains steadily on MATH, remains stable after its initial GSM8K improvement, and loses part of its Round-2 MMLU-Pro gain during mutual critique. With Qwen3.5, PRM self-revision makes a substantially larger first-round improvement on MATH and MMLU-Pro, consistent with the analysis in Appendix I.3. With DeepSeek-Math, StageMAD provides the strongest iterative trajectory on GSM8K and MMLU-Pro, whereas self-revision approaches it on MATH. These results show that the value of later communication depends jointly on the task model and benchmark rather than increasing uniformly with the number of rounds. The wider and more discrete AIME trajectories reflect the small test sets: one additional correct answer changes accuracy by 3.33 percentage points.

Round 1 supplies the largest average StageMAD gain on MATH, GSM8K, and both AIME sets. On MMLU-Pro, directed guidance adds 1.10 points before the final mutual-critique stage removes 0.57 points, showing that later revision is harder to control on this benchmark.

Table 5: Accuracy (%) after applying alternative Round-3 policies to frozen StageMAD Round-2 pools. Entries are three-run means ± sample standard deviations. Within each model, benchmark, and seed, all policies share the same Round-0–2 trajectories, PRM scores, and answer groups; only the Round-3 policy differs. The frozen-pool row gives the Round-2 plurality accuracy and is not included when marking the best and second-best continuation policies in bold and underline, respectively.

Round-3 policy	MATH	GSM8K	MMLU-Pro	AIME24	AIME25
`Qwen2.5-7B-Instruct`					
Frozen Round-2 accuracy	69.9 ± 0.50	93.3 ± 0.20	56.4 ± 2.60	11.1 ± 1.92	6.7 ± 3.34
MAD	$\underline{66.9 \pm 0.81}$	$\underline{91.7 \pm 0.57}$	$\underline{53.6 \pm 1.99}$	6.7 ± 0.00	2.2 ± 3.85
MAD-M^2 (S)	62.5 ± 0.49	87.6 ± 0.70	45.3 ± 2.51	$\underline{11.1 \pm 3.85}$	$\underline{4.4 \pm 1.92}$
MAD-M^2 (O)	62.1 ± 0.47	88.2 ± 0.42	44.5 ± 2.31	7.8 ± 1.92	2.2 ± 1.92
StageMAD	$\mathbf{70.7 \pm 0.95}$	$\mathbf{93.4 \pm 0.20}$	$\mathbf{57.7 \pm 1.11}$	$\mathbf{14.4 \pm 1.93}$	$\mathbf{6.7 \pm 3.34}$
`Qwen3.5-9B`					
Frozen Round-2 accuracy	73.2 ± 0.21	95.2 ± 0.15	71.9 ± 0.21	17.8 ± 5.09	18.9 ± 5.09
MAD	$\underline{71.1 \pm 0.38}$	$\underline{93.9 \pm 0.17}$	$\underline{68.7 \pm 0.38}$	$\underline{14.4 \pm 1.92}$	$\underline{10.0 \pm 0.00}$
MAD-M^2 (S)	66.5 ± 0.96	90.0 ± 0.15	57.7 ± 0.90	11.1 ± 3.85	5.6 ± 5.09
MAD-M^2 (O)	64.2 ± 0.10	90.3 ± 1.56	55.8 ± 1.31	7.8 ± 5.09	7.8 ± 1.92
StageMAD	$\mathbf{73.6 \pm 0.25}$	$\mathbf{95.3 \pm 0.15}$	$\mathbf{71.1 \pm 0.06}$	$\mathbf{17.8 \pm 5.09}$	$\mathbf{18.9 \pm 5.09}$
`DeepSeek-Math-7B-Instruct`					
Frozen Round-2 accuracy	54.9 ± 0.95	89.2 ± 0.23	34.5 ± 0.74	1.1 ± 1.92	0.0 ± 0.00
MAD	$\underline{54.5 \pm 0.76}$	$\underline{88.3 \pm 0.50}$	$\underline{33.2 \pm 1.50}$	0.0 ± 0.00	0.0 ± 0.00
MAD-M^2 (S)	48.6 ± 0.81	72.8 ± 0.98	25.7 ± 0.50	$\mathbf{2.2 \pm 1.90}$	$\mathbf{1.1 \pm 1.92}$
MAD-M^2 (O)	45.5 ± 0.36	72.1 ± 1.62	25.4 ± 1.60	$\underline{1.1 \pm 1.92}$	0.0 ± 0.00
StageMAD	$\mathbf{55.3 \pm 1.15}$	$\mathbf{89.1 \pm 0.06}$	$\mathbf{34.0 \pm 1.48}$	$\underline{1.1 \pm 1.92}$	0.0 ± 0.00

Table 6: Exact cumulative StageMAD accuracy (%) with Qwen2.5-7B-Instruct, including the two 30-question AIME sets. Each row includes all preceding stages; entries are three-run means and sample standard deviations.

State	MATH	GSM8K	MMLU-Pro	AIME24	AIME25
R0: independent proposals	65.43 ± 0.25	91.60 ± 0.70	57.20 ± 0.82	8.89 ± 3.85	3.33 ± 3.34
R1: local self-correction	69.87 ± 0.61	93.47 ± 0.06	57.80 ± 0.50	13.33 ± 0.00	8.89 ± 1.92
R2: directed guidance	70.43 ± 0.49	93.73 ± 0.21	58.90 ± 0.26	12.22 ± 1.92	8.89 ± 1.92
R3: mutual critique	$\mathbf{71.67 \pm 0.96}$	$\mathbf{93.77 \pm 0.21}$	58.33 ± 0.40	$\mathbf{15.56 \pm 1.92}$	$\mathbf{12.22 \pm 1.92}$

F Full Controller and Communication Ablations

F.1 Controller role across datasets

The six variants in Table 7 share the same Round-0 proposals. The generative variants also reuse frozen intermediate states and PRM scores from the matched StageMAD run. Authority following broadcasts the complete response of the highest-scoring agent. Shuffled routing permutes PRM scores among agents within each problem. Matched random routing preserves the number of interventions per problem while sampling sources and targets without PRM scores. These controls separate the effect of additional generation from the effect of process-informed routing.

Table 8 reports the corresponding net question-level changes. For the four generative policies, the value is the difference between corrections and corruptions from Round 1 to Round 2. For direct and final PRM selection, it is the change produced by replacing plurality voting with PRM top-1 at the indicated candidate pool. The controller-and-vote design is net positive on MATH, GSM8K, MMLU-Pro, and AIME24.

F.2 Communication content under matched interventions

Table 9 compares message variants using frozen Round-1 states, the same PRM scores, the same revision targets, and the same Round-3 policy within each dataset. The matched runs contain 473,

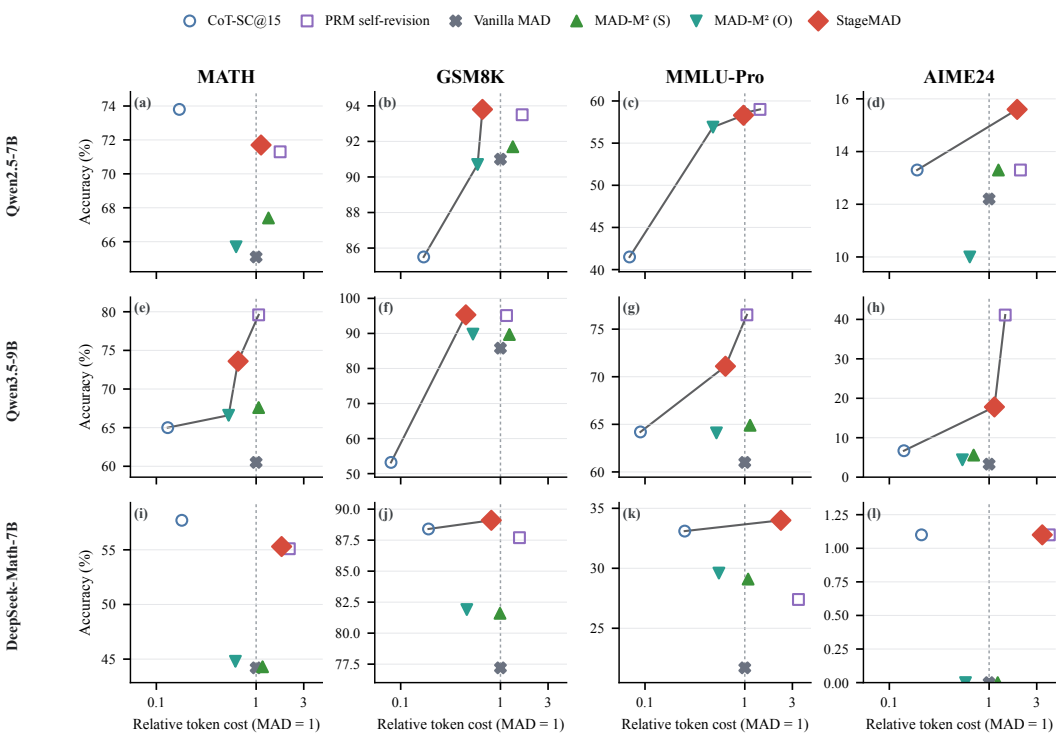


Figure 5: Accuracy–cost Pareto analysis across 12 model–benchmark settings. Each point reports mean final-round accuracy over three runs and token cost normalized by MAD. The vertical dashed line marks the cost of MAD. Solid curves trace the empirical Pareto frontier formed by all evaluated methods within each setting. StageMAD lies on the MAD-method frontier in all 12 settings and on the global frontier in 9.

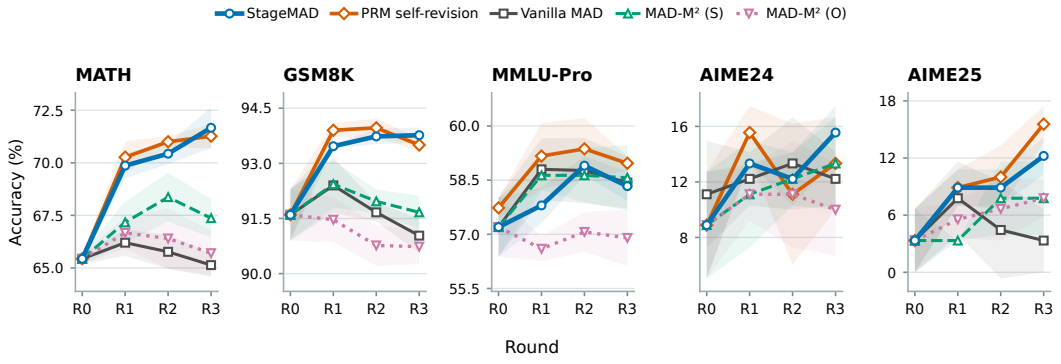


Figure 6: Round-wise accuracy with Qwen2.5-7B-Instruct on all five benchmarks. Lines report three-run means and shaded bands show one sample standard deviation. Each AIME set contains 30 questions.

174, 684, 47, and 54 revision attempts on MATH, GSM8K, MMLU-Pro, AIME24, and AIME25, respectively. The transition and peer-agreement diagnostics distinguish local correction from direct answer copying.

On MATH, answer-free guidance ties the highest final accuracy while producing much less source-answer agreement than majority or full peer solutions. Full solutions yield more local corrections but also more corruptions. On GSM8K, full solutions attain the highest final accuracy and strongest source-answer agreement; answer-free guidance finishes 0.10 points below self-revision and 0.60 below full-solution guidance. On MMLU-Pro, majority-solution guidance reaches the highest final

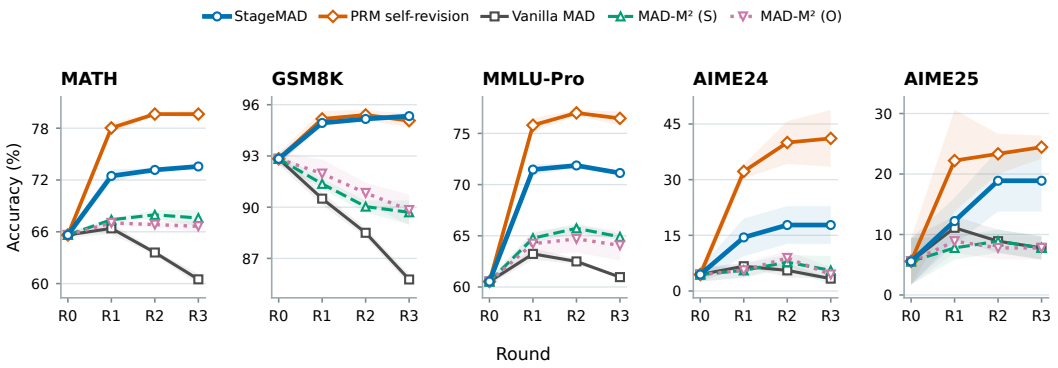


Figure 7: Round-wise accuracy with Qwen3.5-9B on all five benchmarks. Lines report three-run means and shaded bands show one sample standard deviation. PRM self-revision obtains most of its gain in the first revision round, whereas the additional value of StageMAD's communication stages is smaller. Each AIME set contains 30 questions.

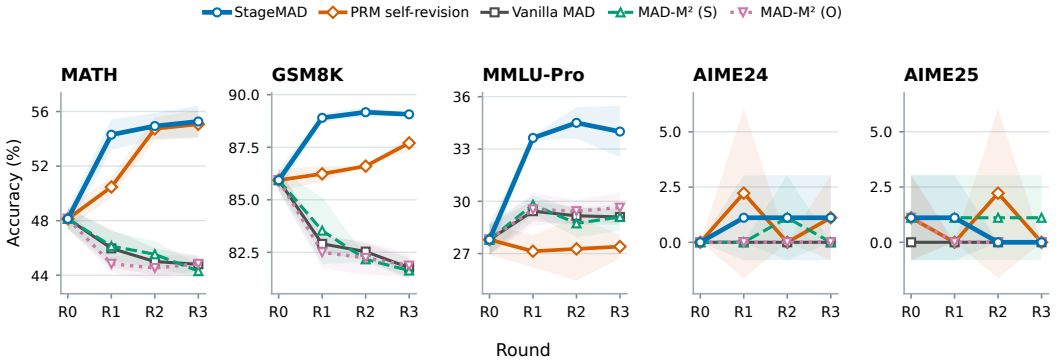


Figure 8: Round-wise accuracy with DeepSeek-Math-7B-Instruct on all five benchmarks. Lines report three-run means and shaded bands show one sample standard deviation. StageMAD improves the initial proposal pool on the three 1,000-example benchmarks, while changes on the 30-question AIME sets are small in absolute counts.

accuracy, whereas full-solution guidance combines the most corrections with an 85.57% corruption rate among initially correct targets.

The AIME results preserve the copying contrast but yield mixed accuracy rankings. Answer-free guidance has the highest final accuracy on AIME24 (13.33%), whereas majority peer solutions lead on AIME25 (13.33%). Full-solution source-answer agreement reaches 80.85% and 70.37% on AIME24 and AIME25, compared with 6.38% and 3.70% for answer-free guidance. Because each set contains 30 problems and the preservation and corruption diagnostics contain at most one eligible initially correct case, these accuracy and transition differences provide directional evidence. Across all five datasets, restricting message content consistently reduces source-answer agreement, while its accuracy effect depends on the task and candidate pool. Answer-free guidance consumes 4.763M and 4.688M R1–R3 tokens on AIME24 and AIME25, respectively, compared with 4.605M and 4.498M for PRM self-revision; these figures exclude the shared cached Round-0 cost.

G Round-3 Vote-Gate Ablation

We freeze the complete Round-0–2 state and compare the original top-two vote-gap gate with an ungated variant that applies the same challenge, verification, revision, and aggregation protocol to every question containing at least two valid answer groups. Thus, the intervention criterion is the

Table 7: Controller-role ablation with Qwen2.5-7B-Instruct. Values are final accuracy (%) from matched single runs. Bold marks the highest result in each column.

Policy	MATH	GSM8K	MMLU-Pro	AIME24	AIME25
StageMAD (controller + vote)	**72.0**	**93.4**	**58.7**	**13.3**	**10.0**
Direct PRM selection after R1	66.9	91.5	55.6	3.3	**10.0**
Final PRM selection after R3	67.7	91.4	55.2	10.0	**10.0**
PRM authority following	70.4	92.7	58.6	6.7	**10.0**
Shuffled PRM routing	71.2	93.0	58.6	6.7	6.7
Matched random routing	70.8	93.0	57.8	**13.3**	6.7

Table 8: Net correctness changes in the controller-role ablation. Positive values indicate more $W \to C$ than $C \to W$ changes. Selection rows compare plurality voting with PRM top-1; generative rows compare Round 1 with Round 2.

Method	MATH	GSM8K	MMLU-Pro	AIME24	AIME25
StageMAD (controller + vote)	$+24$	$+4$	$+3$	$+1$	-1
Direct PRM selection after R1	-21	-14	-31	-1	-1
Final PRM selection after R3	-43	-15	-35	-1	0
PRM authority following	$+12$	-4	-1	0	-1
Shuffled PRM routing	$+6$	$+2$	-2	0	-3
Matched random routing	0	0	-1	0	-2

only changed component. Table 2 reports the aggregate comparison in the main text, while Table 10 provides the exact vote-pattern breakdown.

Table 10 groups questions by their Round-2 vote pattern and reports how ungated Round-3 intervention changes accuracy within each group.

Removing the vote-gap gate increases Round-3 coverage from 24.80% to 52.10% on MATH and from 4.70% to 18.40% on GSM8K. The ungated Round-3 runs consume 9.276 million and 1.233 million tokens, respectively, beyond the frozen Round-2 states; these are not the incremental costs relative to gated Round 3. Despite the extra discussion, the net gain falls from $+6$ to $+3$ questions on MATH and from $+1$ to -10 on GSM8K (Table 2). The Round-2-to-Round-3 changes in Table 10 show that a clear leading answer group offers little consistent opportunity for improvement. For the 4–1 pattern, accuracy rises from 77.07% to 78.34% on MATH but falls from 81.98% to 80.18% on GSM8K. For the 3–1–1 pattern, it falls from 60.34% to 55.17% and from 52.00% to 36.00%, respectively, corresponding to six and four net losses. Intervening on this pattern costs $15{,}250$ and $9{,}611$ additional tokens per question on the two datasets. By contrast, the clearest positive strata are low-margin configurations such as 2–1–1–1 on MATH and 2–2–1 on GSM8K.

One possible explanation is that, after two stages, further exchanges add little independent evidence to cases with a clear leader: the remaining errors may not be recoverable through more discussion among the same agents, while additional critiques can destabilize correct answers. By contrast, a narrow vote margin means that the earlier stages have left competing answers unresolved. Reciprocal critiques may then expose errors that local correction and one-way guidance did not identify, consistent with the six net gains for the 2–1–1–1 MATH stratum and two for the 2–2–1 GSM8K stratum. However, not every close vote benefits: the 3–2 stratum loses answers on both datasets. The vote margin thus serves as a useful computational gate in these runs.

H Round-wise Correctness Transitions

Figure 10 decomposes the full pooled results in Figure 9 by method and adjacent round pair. Each cell reports $W \to C$ and $C \to W$ counts, while its color represents their net difference. The comparison shows that a favorable final balance can arise from different dynamics. On MATH, PRM self-revision performs more corrections in total than StageMAD, but its final transition introduces 22 corruptions compared with 7 for StageMAD. MAD becomes net negative after its first round, whereas StageMAD remains net positive in all three transitions. On MMLU-Pro, StageMAD is positive during R0→R1 (79 versus 70) and R1→R2 (29 versus 21), but the final transition reverses this balance (17 versus 24).

Table 9: Single-run communication-content ablation under matched interventions. R2/R3 are accuracies after Rounds 2 and 3; within each dataset and round, bold marks the highest accuracy and underlining the second-highest distinct accuracy, with ties marked equally. $W \to C$ and $C \to W$ are conditional transition rates among initially wrong and correct revision targets, respectively; Succ. is the fraction of all attempts producing $W \to C$. Pres. reports retained intervened correct-minority cases, and Agree is agreement with the source peer's answer. All rates and accuracies are percentages; dashes indicate no eligible cases.

Revision signal	R2	R3	$W \to C$	$C \to W$	Succ.	Pres.	Agree
			MATH				
PRM self-revision	70.50	71.40	15.01	55.00	14.38	4/7	N/A
Answer-free guidance	**71.40**	**72.00**	18.32	65.00	17.55	3/7	20.34
Random peer guidance	71.10	71.60	16.78	65.00	16.07	3/7	14.65
Majority peer solution	70.50	**72.00**	20.97	60.00	20.08	2/7	44.49
Full peer solution	71.30	71.80	38.85	70.00	37.21	3/7	76.69
Shuffled guidance	70.80	71.00	18.32	60.00	17.55	3/7	0.85
			GSM8K				
PRM self-revision	**93.60**	93.00	56.10	80.00	52.87	2/8	N/A
Answer-free guidance	93.30	92.90	52.44	80.00	49.43	2/8	52.67
Random peer guidance	93.10	93.20	50.00	80.00	47.13	1/8	44.00
Majority peer solution	93.10	93.10	51.83	90.00	48.85	1/8	49.33
Full peer solution	**93.60**	**93.50**	69.51	90.00	65.52	1/8	84.67
Shuffled guidance	93.20	93.40	51.22	90.00	48.28	1/8	2.67
			MMLU-Pro				
PRM self-revision	59.50	59.80	31.69	63.92	27.19	16/45	N/A
Answer-free guidance	59.00	58.70	27.94	62.89	23.98	17/45	31.28
Random peer guidance	**60.00**	59.80	29.98	56.70	25.73	15/45	28.63
Majority peer solution	59.80	**60.10**	34.75	67.01	29.82	8/45	61.67
Full peer solution	59.50	59.50	47.53	85.57	40.79	7/45	81.20
Shuffled guidance	59.50	59.40	30.66	55.67	26.32	17/45	9.69
			AIME24				
PRM self-revision	**10.00**	6.67	6.52	100.00	6.38	0/1	N/A
Answer-free guidance	**10.00**	**13.33**	4.35	100.00	4.26	0/1	6.38
Random peer guidance	**10.00**	10.00	6.52	0.00	6.38	1/1	10.64
Majority peer solution	**10.00**	10.00	6.52	100.00	6.38	0/1	42.55
Full peer solution	6.67	10.00	2.17	100.00	2.13	0/1	80.85
Shuffled guidance	**10.00**	10.00	6.52	100.00	6.38	0/1	0.00
			AIME25				
PRM self-revision	3.33	6.67	3.77	100.00	3.70	–	N/A
Answer-free guidance	**10.00**	10.00	5.66	100.00	5.56	–	3.70
Random peer guidance	6.67	6.67	0.00	100.00	0.00	–	9.26
Majority peer solution	6.67	**13.33**	5.66	100.00	5.56	–	27.78
Full peer solution	**10.00**	10.00	15.09	100.00	14.81	–	70.37
Shuffled guidance	**10.00**	10.00	5.66	100.00	5.56	–	7.41

PRM self-revision follows a similar pattern, with its final transition changing from 32 corrections to 42 corruptions. The instability is therefore not specific to one communication policy and indicates that later revision is difficult to control on this benchmark.

I Error Analysis

I.1 Cross-dataset strengths and failure modes

The main results in Table 1 show where controlled communication has the clearest value. With Qwen2.5, StageMAD improves over MAD on MATH, GSM8K, and both AIME sets, but trails PRM self-revision on MMLU-Pro and AIME25. With Qwen3.5, it remains stronger than the

Table 10: Ungated Round-3 outcomes stratified by exact Round-2 vote pattern. Gap is the difference between the two largest answer groups, and Gate indicates whether the original $\delta_v = 1$ policy would activate. Tok. is the mean additional Round-3 token count per question in the stratum. Six GSM8K questions with an abstaining agent are omitted from the displayed patterns: five 4–abstention cases and one 3–1–abstention case.

Dataset	Pattern	#	Gap	Gate	R2 Acc.	R3 Acc.	$W \to C$	$C \to W$	Net / Tok.
MATH	5–0	479	–	No	91.02	91.02	0	0	0 / 0
MATH	4–1	157	3	No	77.07	78.34	3	1	+2 / 7,464
MATH	3–1–1	116	2	No	60.34	55.17	2	8	−6 / 15,250
MATH	3–2	30	1	Yes	63.33	56.67	0	2	−2 / 6,044
MATH	2–1–1–1	111	1	Yes	41.44	46.85	9	3	+6 / 23,933
MATH	2–2–1	23	0	Yes	43.48	43.48	2	2	0 / 13,160
MATH	1–1–1–1–1	84	0	Yes	14.29	17.86	7	4	+3 / 38,032
GSM8K	5–0	811	–	No	98.03	98.03	0	0	0 / 0
GSM8K	4–1	111	3	No	81.98	80.18	1	3	−2 / 4,758
GSM8K	3–1–1	25	2	No	52.00	36.00	0	4	−4 / 9,611
GSM8K	3–2	23	1	Yes	86.96	69.57	0	4	−4 / 5,152
GSM8K	2–1–1–1	10	1	Yes	60.00	50.00	0	1	−1 / 15,196
GSM8K	2–2–1	10	0	Yes	20.00	40.00	2	0	+2 / 8,851
GSM8K	1–1–1–1–1	4	0	Yes	0.00	0.00	0	0	0 / 24,927

communication-based baselines on all five benchmarks, yet repeated local self-revision is stronger on four benchmarks. DeepSeek-Math provides a different boundary: StageMAD leads the iterative baselines on MMLU-Pro but not CoT-SC@15 on MATH. These comparisons suggest that controlled communication is most useful when a localized process defect can be repaired without inducing larger changes to already correct answers; neither task domain nor model size alone determines the result.

The MATH advantage is visible in the transition balance: the Qwen2.5 StageMAD run has 90 wrong-to-correct and 36 correct-to-wrong changes across rounds, while MAD has 53 of each. Details are in Figure 9. From Table 6, we find that the candidate pool is already highly accurate on GSM8K, so the gain is smaller in absolute points but achieved with fewer tokens than repeated PRM self-revision. MMLU-Pro is different: with Qwen2.5, StageMAD trails the strongest iterative baselines by less than one point, and its final critique round reverses the previous round's gain.

I.2 Language-oriented tasks: MMLU-Pro and CommonsenseQA

We additionally evaluate a fixed 1,000-example subset of CommonsenseQA (Talmor et al., 2019). The Qwen2.5 multi-agent results contain one complete run, whereas all DeepSeek-Math results contain three seeds. Table 11 shows that StageMAD does not improve on the communication-based baselines in this setting: its logged Qwen2.5 accuracy is 79.90%, compared with 81.00% for MAD, 82.20% for MAD-M^2 (S), and 81.90% for MAD-M^2 (O). On DeepSeek-Math it is also below PRM self-revision and CoT-SC@15.

Table 11: CommonsenseQA accuracy (%). Qwen2.5 multi-agent methods contain one run; DeepSeek-Math entries are three-run mean ± sample standard deviation.

Method	Qwen2.5-7B	DeepSeek-Math-7B
CoT	76.33 ± 1.20	46.80 ± 0.90
CoT-SC@5	79.00 ± 0.20	62.40 ± 0.80
CoT-SC@15	80.30 ± 0.17	$\mathbf{66.10 \pm 0.72}$
PRM self-revision	81.00	64.60 ± 1.57
MAD	81.00	60.63 ± 0.47
MAD-M^2 (S)	**82.20**	62.13 ± 0.49
MAD-M^2 (O)	81.90	61.07 ± 1.02
StageMAD	79.90	63.60 ± 1.11

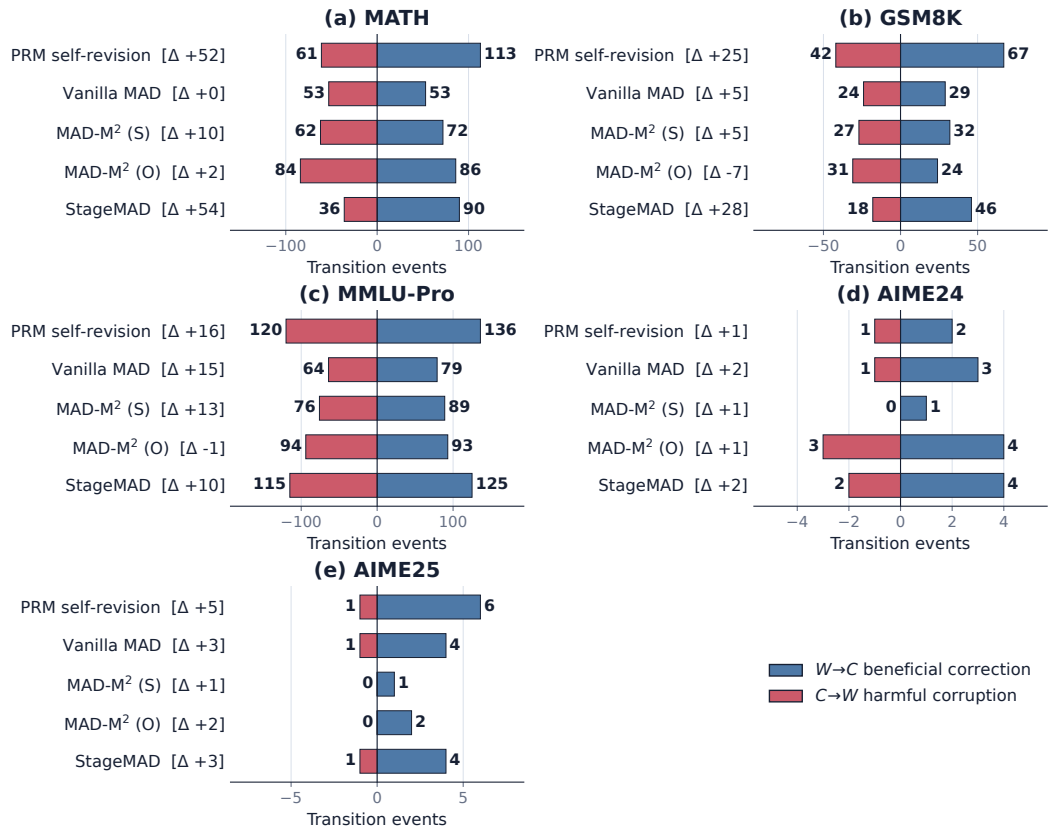


Figure 9: Pooled question-level correctness transitions on all five benchmarks. Rightward blue bars report beneficial corrections ($W \to C$), leftward bars report harmful corruptions ($C \to W$), and Δ is their difference. Counts pool all three adjacent round pairs from one complete run of each method. Because PRM self-revision and StageMAD share the same raw Round-0 candidates, the former uses StageMAD's Round-0 vote state to remove independent tie-breaking variation; its later-round states are unchanged.

Table 12: Question-level correctness transitions for StageMAD in one complete Qwen2.5 run per language-oriented benchmark. Entries give $W \to C/C \to W$ counts between adjacent rounds.

Benchmark	R0→R1	R1→R2	R2→R3	R0/R3 correct
MMLU-Pro	79/70	29/21	17/24	574/584
CommonsenseQA	22/28	12/15	6/10	812/799

Table 12 shows how the Qwen2.5 votes change within one complete run of each language-oriented benchmark. On CommonsenseQA, StageMAD declines from 81.20% at Round 0 to 79.90% at Round 3. Harmful changes outnumber corrections in every stage. MMLU-Pro gains 10 votes overall in its selected run, but the final stage removes seven correct votes on net. Unlike GSM8K and MATH, these two datasets do not show a consistently favorable correction–corruption balance.

The loss begins before peer communication. In the same Qwen2.5 runs, Stage 1 corrects 354 initially wrong MMLU-Pro agent responses but changes 434 initially correct responses to wrong answers; on CommonsenseQA, the corresponding counts are 133 and 330. For comparison, MATH has 281 corrections and 113 corruptions among Stage-1 revised agents. These are agent-level counts, not additional question-level vote changes. They show that localized feedback and revision do not reliably preserve correct responses on these language-oriented tasks.

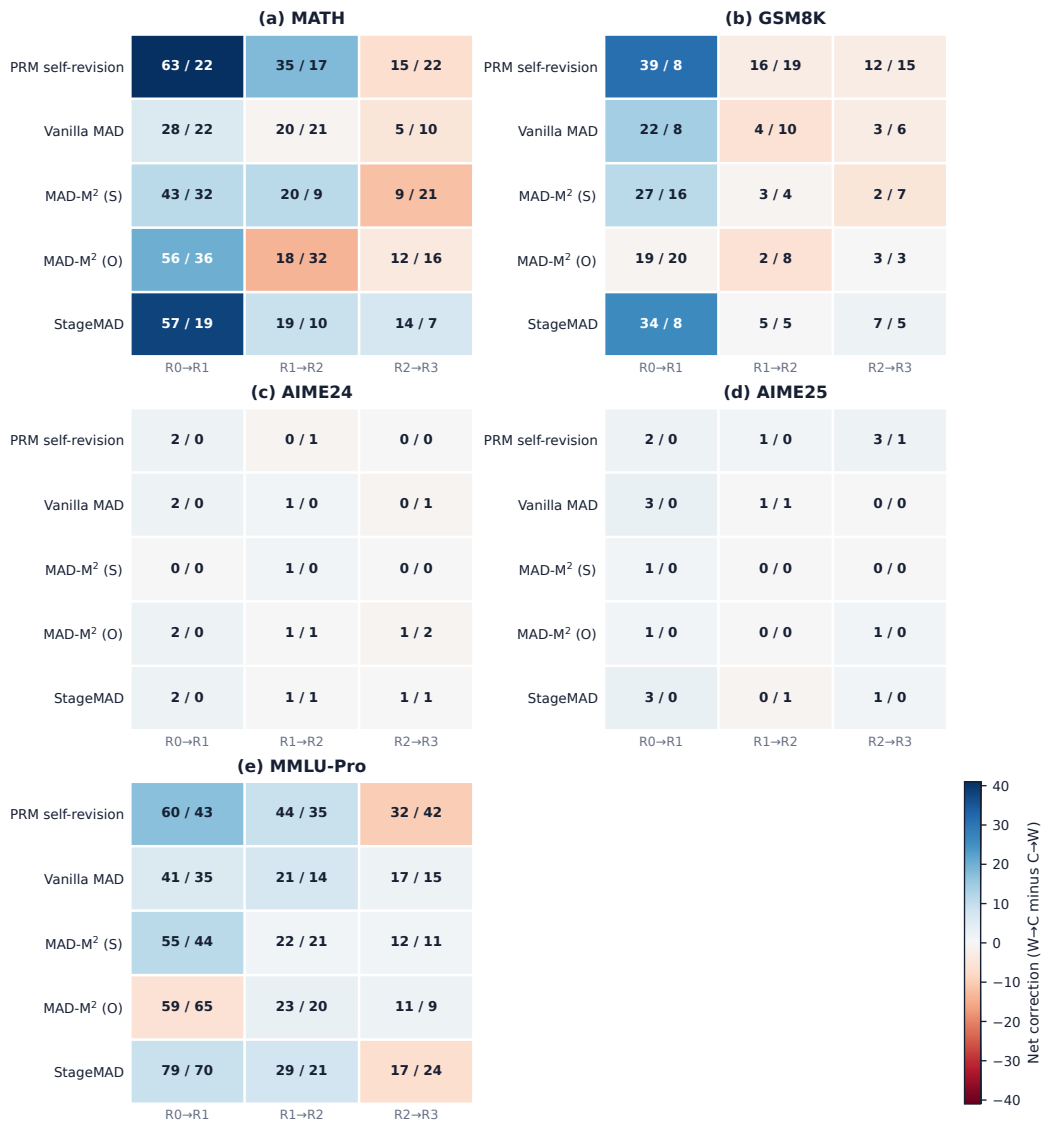


Figure 10: Round-wise correctness transitions by method. Each cell reports $W \to C/C \to W$ counts, and the diverging color encodes their difference: blue indicates net correction and red indicates net corruption. Counts use the same complete runs and matched Round-0 convention as Figure 9.

Two CommonsenseQA cases illustrate why process feedback can be misleading here. In one record, all five agents initially choose the reference option "cloud" for where snowflakes form, but the GenPRM critique favors the broader option "air"; one agent changes from correct to incorrect. In another record, an agent initially selects "living room" after a person wipes their feet and walks through a door. The critique instead treats "doorway" as the necessary next location, and the agent's revision flips the correct vote to a wrong one. Both critiques impose one interpretation on plausible, overlapping alternatives rather than isolating a verifiable calculation error. The gate can also preserve a wrong consensus: 39 of the 535 CommonsenseQA questions stopped before Round 2 for consensus without an explicit GenPRM error are already wrong at Round 1 and remain wrong at the end.

I.3 Task-model strength and use of process feedback

Table 13 decomposes the StageMAD improvement into local PRM-guided repair (R0→R1) and the subsequent communication stages (R1→R3). Compared with Qwen2.5, Qwen3.5 obtains a larger

fraction of its total gain during local repair, particularly on MATH and MMLU-Pro. In one Qwen2.5 run and three Qwen3.5 runs, among revised agents that were initially wrong, StageMAD Round 1 corrects $728/1{,}883$ MATH and $1{,}892/4{,}257$ MMLU-Pro Qwen3.5 cases, compared with $281/1{,}109$ and $354/1{,}532$ in the corresponding Qwen2.5 runs. These conditional counts suggest that Qwen3.5 can make more effective use of localized process feedback, although the initial error pools and routed revision targets differ across models.

Table 13: Mean StageMAD accuracy and stage gains (percentage points). Qwen2.5 values use the three-run stage results in Table 6; Qwen3.5 values are computed from three complete JSONL histories.

	Qwen2.5-7B-Instruct			Qwen3.5-9B		
Benchmark	R0	R0→R1	R1→R3	R0	R0→R1	R1→R3
MATH	65.4	+4.4	+1.8	65.6	+6.8	+1.1
GSM8K	91.6	+1.9	+0.3	92.8	+2.1	+0.4
MMLU-Pro	57.2	+0.6	+0.5	60.5	+10.9	−0.3

To test whether this pattern reflects stronger use of process feedback rather than the later communication mechanism, we additionally examine the PRM self-revision baseline. Unlike StageMAD Stage 1, which revises only agents for which the PRM identifies an explicit defect, PRM self-revision provides PRM feedback to every agent and asks each agent to revise independently. It therefore offers a complementary test of how effectively the task model can act on PRM feedback without peer communication. As shown in Table 14, Qwen3.5 repairs a substantially larger fraction of initially wrong agents than Qwen2.5 on MATH and MMLU-Pro. On MMLU-Pro, it also corrupts far fewer initially correct responses. Correspondingly, its question-level Round 1 vote gains are much larger on these two benchmarks. This pattern is consistent with the interpretation that Qwen3.5 can exploit PRM feedback more effectively. GSM8K shows a different pattern: Qwen3.5 has both a lower conditional repair rate and a lower corruption rate. We therefore treat the relationship as benchmark-dependent rather than as a causal estimate of model strength.

Table 14: Agent-level Round-1 transitions under PRM self-revision. The conditional rates use initially wrong or initially correct agents as their respective denominators. Qwen2.5 has one logged run per dataset and Qwen3.5 pools three runs; vote gain is the change in question-level accuracy from Round 0 to Round 1.

Benchmark	Task model	Wrong repaired	Correct corrupted	Vote gain (pp)
MATH	Qwen2.5	441/2,096 (21.0%)	125/2,904 (4.3%)	+4.9
	Qwen3.5	2,454/5,898 (41.6%)	474/9,102 (5.2%)	+12.4
GSM8K	Qwen2.5	334/720 (46.4%)	108/4,280 (2.5%)	+2.9
	Qwen3.5	698/1,927 (36.2%)	189/13,073 (1.4%)	+2.3
MMLU-Pro	Qwen2.5	275/2,342 (11.7%)	216/2,658 (8.1%)	+1.4
	Qwen3.5	2,918/6,892 (42.3%)	85/8,108 (1.0%)	+15.7

The paired Qwen3.5 histories further localize where PRM self-revision and StageMAD begin to diverge by holding the initial agent responses fixed. On MATH, PRM self-revision uniquely repairs $1{,}929$ initially wrong agent responses at Round 1, compared with 203 uniquely repaired by StageMAD; on MMLU-Pro, the corresponding counts are $1{,}418$ and 392. On MMLU-Pro, PRM self-revision also uniquely corrupts fewer initially correct agents (75 versus 110). Thus, much of the advantage of PRM self-revision under Qwen3.5 is already established before any peer communication occurs.

Consistent with this observation, the additional gain from StageMAD's communication stages is smaller for Qwen3.5 than for Qwen2.5 on MATH and is slightly negative on MMLU-Pro. Across the paired Qwen3.5 histories, PRM self-revision produces seven more final correct votes than StageMAD over three runs. Together, these results suggest that as the task model becomes more effective at acting directly on PRM feedback, a larger share of the available correction is realized during local revision, leaving less residual error for later communication to resolve. Because StageMAD and

PRM self-revision apply different revision policies, however, these comparisons identify where their performance diverges rather than isolating task-model strength as the sole causal factor.

J TOKEN COMPLEXITY

We analyze inference cost by tracking how generated messages are reused as inputs in subsequent calls. Let N_a denote the number of agents and let T_q denote the token length of the question and fixed instructions. Let $T_{i,r}$ denote the length of agent i's response after round r. If an agent is not revised in a round, its preceding response is carried forward without incurring another output cost.

Let $G_{i,r}$ denote the length of the PRM feedback generated in round r when evaluating agent i's response from round $r-1$. To keep the comparison interpretable, we use a message-level accounting model in which one aggregate PRM evaluation reads the question and the evaluated response once. The empirical token counts in Table 1 include all actual step-prefix PRM calls.

All iterative methods share the same Round-0 cost:

$$C_0 = N_a T_q + \sum_{i=1}^{N_a} T_{i,0}. \tag{6}$$

StageMAD. Let $e_i \in \{0,1\}$ indicate whether the PRM identifies an explicit defect in agent i and activates its Round-1 revision. Let $k_{i,1} \in [0,1]$ denote the fraction of the Round-0 response retained after masking the suspicious suffix. Thus, the retained prefix contains $k_{i,1}T_{i,0}$ tokens, where a smaller $k_{i,1}$ corresponds to a larger masking reduction.

The Round-1 cost is

$$\begin{aligned} C_1 = &\sum_{i=1}^{N_a} (T_q + T_{i,0} + G_{i,1}) \\ &+ \sum_{i=1}^{N_a} e_i \left(T_q + k_{i,1}T_{i,0} + G_{i,1} + T_{i,1}\right). \end{aligned} \tag{7}$$

The first term evaluates every initial response with the PRM. For an agent with $e_i = 1$, the second term counts the local revision call: the model receives the question, the retained reasoning prefix, and the PRM feedback before generating $T_{i,1}$. Compared with retaining the complete response, masking saves $(1-k_{i,1})T_{i,0}$ input tokens for each revised agent.

Before Round 2, only responses revised in Round 1 require fresh PRM evaluation. Let $d_i \in \{0,1\}$ indicate whether agent i is subsequently selected as a Round-2 target. Under the staged protocol, $d_i \le e_i$. Let $s(i)$ denote the source agent selected for target i, let $H_{i,2}$ denote the length of the restricted guidance generated for that target, and let $k_{i,2} \in [0,1]$ denote the fraction of the target's Round-1 response retained after masking.

The Round-2 cost is

$$\begin{aligned} C_2 = &\sum_{i=1}^{N_a} e_i \left(T_q + T_{i,1} + G_{i,2}\right) \\ &+ \sum_{i=1}^{N_a} d_i \left[\left(T_q + T_{s(i),1} + T_{i,1} + G_{i,2} + H_{i,2}\right)\right. \\ &\qquad \left. + \left(T_q + k_{i,2}T_{i,1} + G_{i,2} + H_{i,2} + T_{i,2}\right)\right]. \end{aligned} \tag{8}$$

The first term reevaluates the responses changed in Round 1. For each selected target, the first parenthesized term is the source-guidance call. The source reads its own response, the target response, and the target's PRM feedback before generating $H_{i,2}$. The second parenthesized term is the target-revision call. The target receives the same PRM feedback and guidance, retains only $k_{i,2}T_{i,1}$ tokens from its previous reasoning, and generates $T_{i,2}$.

Equivalently, one activated Round-2 interaction contributes

$$2T_q + T_{s(i),1} + (1+k_{i,2})T_{i,1} + 2G_{i,2} + 2H_{i,2} + T_{i,2}. \tag{9}$$

The coefficient two before $G_{i,2}$ arises because the PRM feedback is read by both the source and the target. The coefficient two before $H_{i,2}$ counts the guidance once as the source output and once as part of the target input. Masking saves $(1 - k_{i,2})T_{i,1}$ tokens in the target-revision prompt.

For Round 3, let K denote the number of answer groups after Round 2 and let $r(g)$ denote the representative of group g. Let $C_{g,h}$ be the length of the challenge generated from group g to group h. Let $h(i)$ denote the group containing agent i, and let V_i denote the length of agent i's verification output. Finally, let $v_i \in \{0, 1\}$ indicate whether agent i verifies at least one challenge and let $m_i \in \{0, 1\}$ indicate whether the verification confirms an error and activates revision, where $m_i \le v_i$.

The cost of an activated Round 3 is

$$\begin{aligned} C_3 = &\sum_{g=1}^{K} \sum_{\substack{h=1 \\ h \neq g}}^{K} \left(T_q + T_{r(g),2} + T_{r(h),2} + C_{g,h}\right) \\ &+ \sum_{i=1}^{N_a} v_i \left(T_q + T_{i,2} + \sum_{\substack{g=1 \\ g \neq h(i)}}^{K} C_{g,h(i)} + V_i \right) \\ &+ \sum_{i=1}^{N_a} m_i \left(T_q + T_{i,2} + \sum_{\substack{g=1 \\ g \neq h(i)}}^{K} C_{g,h(i)} + V_i + T_{i,3} \right). \end{aligned} \tag{10}$$

The first term generates one challenge for every ordered pair of distinct answer groups. The second term broadcasts the incoming challenges to the members of each target group and asks them to verify the disputed steps. The last term counts the revision calls authorized by a confirmed error. The verification output V_i is counted once when generated and once when reused as revision evidence.

Let $z \in \{0, 1\}$ indicate whether the close-vote gate activates Round 3. The total StageMAD cost is

$$C_{\text{StageMAD}} = C_0 + C_1 + C_2 + zC_3. \tag{11}$$

PRM self-revision. For a matched comparison, let $T^{\text{SR}}_{i,r}$ and $G^{\text{SR}}_{i,r}$ denote the response and PRM-feedback lengths under PRM self-revision. Let $k^{\text{SR}}_{i,r} \in [0, 1]$ denote the fraction of the preceding response retained after masking in revision round r.

PRM self-revision evaluates and revises every agent in every round. Its cost is

$$\begin{aligned} C_{\text{SR}} &= C_0 + \sum_{r=1}^{3} \sum_{i=1}^{N_a} \left[\left(T_q + T^{\text{SR}}_{i,r-1} + G^{\text{SR}}_{i,r}\right) \right. \\ &\qquad \left. + \left(T_q + k^{\text{SR}}_{i,r} T^{\text{SR}}_{i,r-1} + G^{\text{SR}}_{i,r} + T^{\text{SR}}_{i,r}\right) \right] \\ &= C_0 + \sum_{r=1}^{3} \sum_{i=1}^{N_a} \left[2T_q + (1 + k^{\text{SR}}_{i,r}) T^{\text{SR}}_{i,r-1} + 2G^{\text{SR}}_{i,r} + T^{\text{SR}}_{i,r} \right]. \end{aligned} \tag{12}$$

The first parenthesized term is the PRM evaluation, and the second is the masked revision. Unlike StageMAD, self-revision incurs both terms for every agent in every round.

MAD. Let $T^{\text{MAD}}_{i,r}$ denote the length of agent i's response after round r under MAD. In each debate round, the input to agent i contains the question, its own preceding response, and the preceding responses of the other agents. Its per-agent cost is therefore

$$\begin{aligned} &T_q + T^{\text{MAD}}_{i,r-1} + \sum_{\substack{j=1 \\ j \neq i}}^{N_a} T^{\text{MAD}}_{j,r-1} + T^{\text{MAD}}_{i,r} \\ &\qquad = T_q + \sum_{j=1}^{N_a} T^{\text{MAD}}_{j,r-1} + T^{\text{MAD}}_{i,r}. \end{aligned} \tag{13}$$

The total three-round cost is

$$C_{\mathrm{MAD}} = C_0 + \sum_{r=1}^{3}\sum_{i=1}^{N_a}\left(T_q + \sum_{j=1}^{N_a} T_{j,r-1}^{\mathrm{MAD}} + T_{i,r}^{\mathrm{MAD}}\right)$$
$$= C_0 + 3N_aT_q + N_a\sum_{r=1}^{3}\sum_{j=1}^{N_a} T_{j,r-1}^{\mathrm{MAD}} + \sum_{r=1}^{3}\sum_{i=1}^{N_a} T_{i,r}^{\mathrm{MAD}}. \tag{14}$$

Efficiency conditions. Because the compared methods share the same Round-0 trajectories, the common cost C_0 cancels. StageMAD consumes fewer tokens than MAD whenever

$$C_1 + C_2 + zC_3 < 3N_aT_q + N_a\sum_{r=1}^{3}\sum_{j=1}^{N_a} T_{j,r-1}^{\mathrm{MAD}} + \sum_{r=1}^{3}\sum_{i=1}^{N_a} T_{i,r}^{\mathrm{MAD}}. \tag{15}$$

StageMAD consumes fewer tokens than repeated PRM self-revision whenever

$$C_1 + C_2 + zC_3 < \sum_{r=1}^{3}\sum_{i=1}^{N_a}\left[2T_q + (1+k_{i,r}^{\mathrm{SR}})T_{i,r-1}^{\mathrm{SR}} + 2G_{i,r}^{\mathrm{SR}} + T_{i,r}^{\mathrm{SR}}\right]. \tag{16}$$

These expressions identify two sources of efficiency. First, masking reduces the response context from $T_{i,r-1}$ to $k_{i,r}T_{i,r-1}$ in activated revisions. Second, StageMAD applies PRM evaluation, guidance, verification, and revision selectively, whereas MAD repeatedly copies complete responses across all agent contexts and self-revision evaluates and revises every agent. StageMAD is more efficient when these avoided costs exceed the cost of its activated guidance and critique calls. When many targets are selected or Round 3 contains many answer groups, the additional interaction cost can instead outweigh the savings, consistent with the dataset-dependent token consumption reported in Table 1.

K Mechanistic Analysis

We provide a formal interpretation of the Round-3 vote-gap gate by relating the top-two support margin to the entropy of the answer distribution. The instance-dependent computational cost of the stage gates is analyzed separately in Appendix J.

K.1 Vote-margin interpretation

After Round 2, let the $K \geq 2$ answer groups have normalized supports $p_k = n_{(k)}^2/N$, ordered such that $p_1 \geq p_2 \geq \cdots \geq p_K$. Define the vote entropy and normalized top-two margin as

$$H(\mathbf{p}) = -\sum_{k=1}^{K} p_k \log p_k, \qquad \Delta = p_1 - p_2 = \frac{n_{(1)}^2 - n_{(2)}^2}{N}. \tag{17}$$

Proposition 1 (Close-vote entropy bound)**.** *For any $\epsilon \in [0,1]$, if $\Delta \leq \epsilon$, then*

$$H(\mathbf{p}) \geq h\left(\frac{1+\epsilon}{2}\right), \tag{18}$$

where $h(u) = -u\log u - (1-u)\log(1-u)$ is binary entropy. When $K = 2$, equality holds with $\epsilon = \Delta$, and the entropy decreases monotonically as the vote margin increases.

Proof. The condition $\Delta = p_1 - p_2 \leq \epsilon$ implies $p_2 \geq p_1 - \epsilon$. Since the group proportions sum to one, $p_1 + p_2 \leq 1$. Combining the two inequalities gives

$$2p_1 - \epsilon \leq p_1 + p_2 \leq 1, \tag{19}$$

and hence

$$p_1 \leq \frac{1+\epsilon}{2}. \tag{20}$$

Define

$$\mathbf{q} = \left(\frac{1+\epsilon}{2}, \frac{1-\epsilon}{2}, 0, \ldots, 0\right). \tag{21}$$

The largest component of $\mathbf{q}$ is no smaller than p_1 by Eq. equation 20. For every partial sum containing at least two components, the corresponding sum for $\mathbf{q}$ equals one and is therefore no smaller than the partial sum of $\mathbf{p}$. Thus, $\mathbf{q}$ majorizes $\mathbf{p}$. Shannon entropy is Schur-concave, so

$$H(\mathbf{p}) \geq H(\mathbf{q}) \tag{22}$$

$$= -\frac{1+\epsilon}{2}\log\frac{1+\epsilon}{2} - \frac{1-\epsilon}{2}\log\frac{1-\epsilon}{2} \tag{23}$$

$$= h\left(\frac{1+\epsilon}{2}\right), \tag{24}$$

which proves Eq. equation 18.

When $K = 2$, $p_1 + p_2 = 1$ and $p_1 - p_2 = \Delta$, so $p_1 = (1+\Delta)/2$ and $p_2 = (1-\Delta)/2$. This gives equality with $\epsilon = \Delta$. Finally,

$$\frac{d}{d\Delta}h\left(\frac{1+\Delta}{2}\right) = \frac{1}{2}\log\frac{1-\Delta}{1+\Delta} \leq 0 \tag{25}$$

for $\Delta \in [0, 1]$, establishing monotonicity. □

Remark 1. The Round-3 condition $n_{(1)}^2 - n_{(2)}^2 \leq \delta_v$ is equivalent to $\Delta \leq \delta_v/N$. Proposition 1 therefore interprets the gate as a low-cost test for collective uncertainty. Under the experimental setting $N = 5$ and $\delta_v = 1$, every activated candidate pool satisfies $H(\mathbf{p}) \geq h(0.6) \approx 0.673$ nats. The margin is used operationally because it directly measures the stability of the plurality winner. It characterizes disagreement in the current candidate pool, not the intrinsic difficulty of the underlying question.