BIAS CONTROL AND INJECTION FOR MULTI-AGENT AGGREGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-agent reasoning with large language models (LLMs) raises a central question for collective decision-making: when agents disagree, whose decision should be trusted? Resolving such disagreements is difficult because agents' decisions are susceptible to non-semantic biases such as majority pressure and response-style preferences, which can obscure the quality of the underlying reasoning. Our analysis shows how bias can be both mitigated and exploited: *bias control* improves initial answer selection, while *bias injection* elicits decision instability that serves as a reliability signal. Building on these findings, we introduce a test-time aggregation framework that requires neither validation data nor prior knowledge of relative model strength. The framework implements bias control by presenting one style-normalized reasoning trace per distinct answer, and bias injection by placing each aggregating agent's initially selected answer in the minority. Overall, our framework achieves higher average accuracy and better token efficiency than competing non-oracle aggregation methods across three mathematical reasoning benchmarks and four heterogeneous LLM combinations. It also improves final-answer accuracy when applied to multi-agent debate and Mixture-of-Agents, without modifying their interaction or synthesis processes.

1 INTRODUCTION

Multi-agent reasoning benefits from the diverse and complementary solutions produced by multiple agents, yet those agents often reach conflicting conclusions. Whether candidate solutions are generated independently, revised through multi-agent debate, or synthesized across multiple stages, a correct answer among these conflicting solutions may still be missed during final aggregation (Ai et al., 2025; Du et al., 2023; Wang et al., 2025). The effectiveness of multi-agent reasoning therefore depends not only on generating useful solutions, but also on reliably resolving these disagreements (Wang et al., 2024b; Smit et al., 2023; Choi et al., 2026a; Wang et al., 2026a).

To reach a collective decision, frequency- and consistency-based aggregation methods rely on agreement among responses as evidence of correctness (Wang et al., 2022; Ai et al., 2025; Wang et al., 2026a). Agreement is often informative, but can favor a shared error over a correct minority answer. For example, majority voting (MV) is widely used and often effective. In our analysis, the sampled response pool contains a correct answer in 83.6% of cases on average, yet MV selects it in only 72.2% (Figure 1; see Section 5.1 for details). Alternatively, LLM-based aggregation can consider the reasoning behind each answer as additional evidence, allowing a correct minority answer to prevail. However, incorporating reasoning does not by itself ensure better selection: LLM decisions can be distorted by frequency, style, position, and other non-semantic biases (Ma et al., 2025; Zhu et al., 2025; Choi et al., 2025; Wang et al., 2024a; Shi et al., 2025; Feuer et al., 2025; Hwang et al., 2025; Choi et al., 2026b; Chen et al., 2024). How these biases are handled therefore matters for turning this additional reasoning evidence into better aggregation decisions.

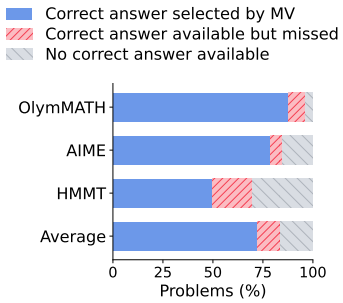


Figure 1: **Correct-answer availability and majority-vote accuracy.** Majority voting is effective overall but still overlooks correct answers with limited support.

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

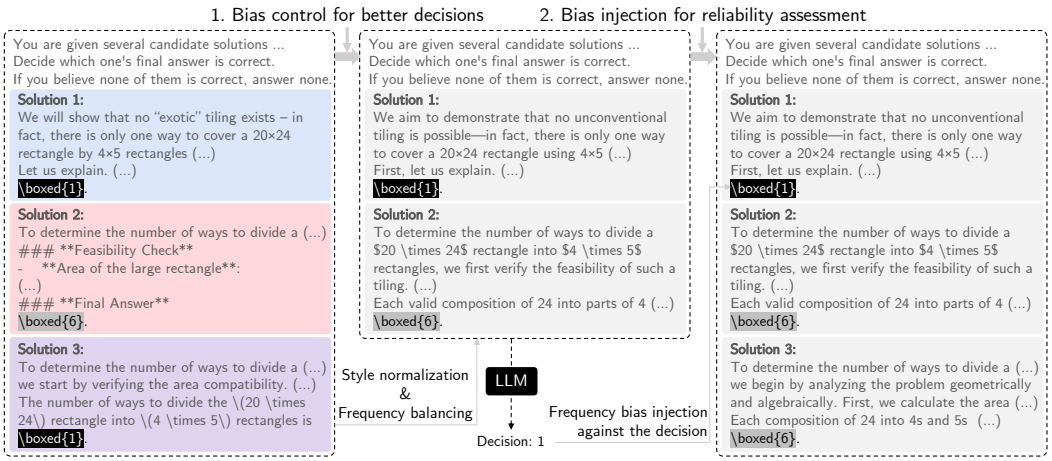


Figure 2: **Illustration of bias control and bias injection.** Bias control forms an initial decision by comparing one style-normalized trace per distinct answer under balanced frequency. Bias injection then places the selected answer in the relative minority to assess its reliability. Bias is mitigated through control during decision formation and then exploited through injection to assess reliability.

Most prior work seeks to mitigate biases in LLM decisions by reducing or controlling their influence (Li et al., 2024; Dubois et al., 2024; Li et al., 2025; Yang et al., 2026b). In contrast, our key idea is to both mitigate and exploit bias: we first control presentation bias to improve an initial decision, then deliberately inject bias against the selected answer to assess that decision's reliability. For bias control, we find that style normalization improves answer-selection accuracy beyond frequency balancing alone. For bias injection, we find that injecting frequency bias against the selected answer changes incorrect decisions more often than correct ones. The decision instability provides a more informative reliability signal than that obtained by reversing candidate order or repeating the same query. Based on these findings, we introduce a test-time framework for multi-agent aggregation with two complementary operations, *bias control* and *bias injection* (Figure 2).

The framework requires neither validation data nor prior knowledge of relative model strength. It operates directly on reasoning history, the collection of reasoning traces and their extracted answers available before final aggregation. Bias control retains one reasoning trace per distinct answer and rewrites each into a shared plain-prose form, helping the participating agents assess the reasoning with less influence from frequency and style biases. When their initial selections disagree, bias injection places each agent's selected answer in the minority to assess the reliability of that decision on the current problem. The resulting problem-specific reliability signal guides final-answer selection.

We evaluate the framework on independently sampled responses from heterogeneous LLMs across three mathematical reasoning benchmarks and four model combinations. Our framework achieves higher average accuracy and better token efficiency than competing non-oracle aggregation methods. Applying the same aggregation procedure to multi-agent debate (Du et al., 2023) and Mixture-of-Agents (MoA) (Wang et al., 2025) also improves final-answer accuracy without modifying their interaction or synthesis processes.

2 Related Work

Multi-Agent Reasoning. Multi-agent reasoning seeks to improve LLM problem solving by combining the outputs of multiple agents (Wang et al., 2026a; Tian et al., 2026; Cui et al., 2026). Agents may generate solutions independently, exchange intermediate reasoning, critique one another, synthesize responses, or revise their answers over multiple rounds (Du et al., 2023; Wang et al., 2025; Choi et al., 2026a; Tian et al., 2026; Cui et al., 2026; Zhu et al., 2026). These approaches aim to increase reasoning diversity and combine complementary strengths across agents (Wang et al., 2026a; Zhu et al., 2026). In practice, however, interaction does not consistently provide such benefits (Wang et al., 2024b; Smit et al., 2023; Choi et al., 2026a; Zhu et al., 2026). Multi-agent interactions may converge to an incorrect consensus, amplify early errors, or encourage agents to follow a dominant opinion rather than assess the underlying reasoning (Zhu et al., 2025; Choi et al., 2025; Okawa, 2026; Cui

et al., 2026). As a result, a correct answer may appear among generated or intermediate responses without being selected as the final answer (Cui et al., 2026; Wang et al., 2026a).

Final-Answer Aggregation. Multi-agent frameworks use different rules for final-answer aggregation. Multi-agent debate commonly relies on terminal-round responses or consensus (Du et al., 2023; Choi et al., 2026a); SLM-MUX selects among model-wise majority answers using within-model consistency, with ties broken by each model's historical accuracy measured on a separate validation set (Wang et al., 2026a); and Mixture-of-Agents synthesizes responses across successive layers through a designated final aggregator (Wang et al., 2025). However, majority voting and consistency-based aggregation can fail when model errors are correlated or a correct answer appears with low frequency (Choi et al., 2026a; Ai et al., 2025; Cui et al., 2026; Wang et al., 2026b). Recent work moves beyond raw answer counts by leveraging model accuracies and cross-model answer correlations (Ai et al., 2025), scoring answers based on changes across debate rounds (Cui et al., 2026), or retaining the best-supported rationale for each distinct answer (Wang et al., 2026b). Verification-based approaches similarly select among candidate answers using judgments from multiple verifiers (Lifshitz et al., 2025). Our framework draws candidate answers from the available reasoning history, which can include non-majority and intermediate answers. It assesses decision reliability independently for each problem without training, validation data, or prior knowledge of relative model strength.

Biases in Multi-Agent Aggregation. While reasoning traces provide additional evidence beyond final answers alone, LLM interactions and judgments can be influenced by non-semantic cues such as answer frequency, prompt position, response style, and model identity (Ma et al., 2025; Zhu et al., 2025; Choi et al., 2025; Wang et al., 2024a; Shi et al., 2025; Feuer et al., 2025; Choi et al., 2026b). These effects also matter in heterogeneous systems, where different LLMs produce reasoning traces with different response styles (Chen et al., 2025; Feuer et al., 2025). Consequently, a conclusion may be favored because it appears more frequently or is expressed in a preferred style rather than because its reasoning provides stronger evidence of correctness (Ma et al., 2025; Zhu et al., 2025; Choi et al., 2025; Hwang et al., 2025; Feuer et al., 2025). Most prior work focuses on analyzing biases in LLM decisions and mitigating their influence (Ma et al., 2025; Shi et al., 2025; Li et al., 2024; 2025; Ye et al., 2025; Yang et al., 2026b;a; Koo et al., 2026). Recent work also exploits biases to examine judge robustness (Yang et al., 2026c) and uses presentation perturbations to derive uncertainty signals (Badshah et al., 2026). We study the complementary use of bias mitigation through control and bias exploitation through injection for multi-agent aggregation.

3 Motivation: Intervening on Biases

We examine how to intervene on presentation biases to both mitigate and exploit their effects in multi-agent aggregation. Specifically, we ask whether controlling presentation biases improves agents' initial answer selection and whether injecting adverse bias against their selected answers provides a useful signal of decision reliability.

Analysis setup. To isolate final-answer decision from upstream interaction dynamics, we analyze independently sampled response pools for 220 mathematical reasoning problems from OlymMATH, AIME, and HMMT. For each problem, we target three completed reasoning traces from each of four heterogeneous reasoning models: Qwen3-8B (Qwen), Phi-4-reasoning-plus (Phi), GLM-Z1-9B (GLM), and Nemotron-Nano-9B-v2 (Nemotron). We consider all four three-model combinations and focus on instances not resolved by unanimous model-level agreement. The participating models also serve as judges over the competing answers. See Section 5 for experimental details.

3.1 Controlling Biases in Candidate Presentation

Following the one-rationale-per-answer construction of Wang et al. (2026b), we group reasoning traces by their final answers, retain one trace for each distinct answer, and present every answer once. The agent then selects among the resulting candidates or `none` if none is correct. This removes the original majority–minority relation and yields a frequency-balanced comparison.

Frequency balancing controls count-based cues, but not the source-dependent presentation patterns naturally embedded in reasoning traces generated by different LLMs. To control this remaining

variation rather than compare raw traces directly, we rewrite each retained trace with the same prompt into a common plain-prose form. The prompt standardizes surface wording and formatting while instructing the rewriter to preserve the original length, reasoning content, numerical values, equations, and final answer. The normalization prompt and examples of original and normalized traces are provided in Figure 11 and Appendix D.4, respectively.

We compare paired decisions over the same problems, candidate answers, and source traces, with the normalized condition presenting the rewritten traces. To distinguish answer selection from parsing failures, we include only decision pairs whose selected answers can be successfully parsed in both conditions. As shown in Table 1, style normalization improves pooled answer-selection accuracy and maintains or improves accuracy in 9 of the 12 agent–combination settings. These conditional results indicate that style normalization can improve decision accuracy beyond frequency balancing.

Table 1: **Effect of style normalization on LLM decision accuracy.** Rows identify the models forming the candidate pool, and columns identify the agent making the decision. Cells report Original → Normalized accuracy under frequency-balanced presentation on paired instances valid in both conditions; bold marks the higher value. Style normalization generally improves accuracy.

Models	Answer-selection accuracy by agent (%)			
	Qwen	Phi	GLM	Nemo
Qwen–Phi–GLM	60.8→60.8	65.0→**70.7**	62.4→**63.8**	–
Qwen–Phi–Nemotron	62.1→**63.9**	68.2→**70.6**	–	57.8→**58.1**
Qwen–GLM–Nemotron	**55.7**→54.8	–	54.2→**59.4**	50.4→**51.9**
Phi–GLM–Nemotron	–	**65.2**→64.2	65.1→**67.9**	**63.7**→60.9

3.2 Injecting Bias to Elicit Decision Instability

Answer-frequency and conformity effects have been repeatedly observed in LLM decisions (Ma et al., 2025; Zhu et al., 2025; Choi et al., 2025), motivating their control during decision formation in Section 3.1. We ask whether this same susceptibility can instead be exploited after an initial controlled decision to assess its reliability through decision instability. We define decision instability as a change between the initial controlled decision and the second-pass decision. Placing the selected answer at a presentation disadvantage creates a stress test: a decision grounded in the underlying reasoning should be more likely to persist than a less reliable one. Here, the signal is whether the initial decision changes, not whether the perturbed query itself produces a better answer.

Holding the set of distinct answer hypotheses fixed, we compare three second-pass conditions. Repeated inference queries the judge again with the same bias-controlled presentation. Position swap reverses the candidate order, following prior work that uses bidirectional response orderings to derive uncertainty signals for selective judging (Badshah et al., 2026). Adverse-frequency injection applies stronger pressure by presenting the agent's initially selected answer once and its strongest competing answer twice, placing the initial choice in the relative minority.

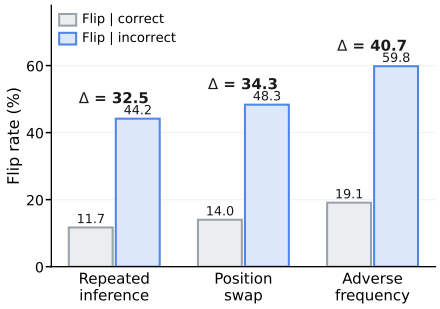


Figure 3: **Decision instability under bias injection.** Bars show flip rates conditioned on initial correctness. Adverse-frequency yields the largest gap, offering a more selective signal for reliability assessment.

A useful second-pass intervention for final aggregation should change incorrect decisions more often than correct ones. As shown in Figure 3, adverse-frequency injection changes 59.8% of incorrect decisions, compared with 19.1% of correct decisions, yielding a 40.7-point gap. This exceeds the corresponding gaps of 32.5 points for repeated inference and 34.3 points for position swap, indicating that bias-induced decision instability provides a more selective reliability signal for aggregation. Section 6.2 shows that a substantial selectivity gap remains when the added trace is an exact copy.

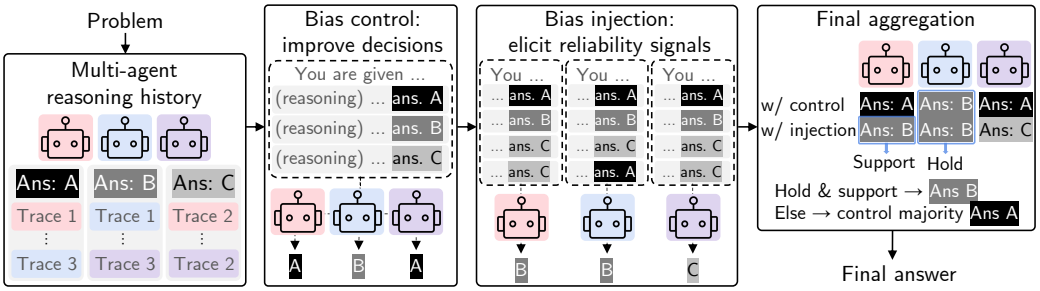


Figure 4: **Overview of the proposed aggregation framework.** Responses from the reasoning history are grouped by answer. When agents disagree, the framework uses bias control to form LLM decisions and injects adverse-frequency bias to test their reliability. Final aggregation combines decision holds observed under bias injection with cross-agent support, without modifying upstream reasoning.

4 Proposed Method

Building on the findings in Section 3, we introduce a training- and validation-free aggregation framework (Figure 4). Cross-model agreement and disagreement provide useful information about decision uncertainty (Hamidieh et al., 2026). We therefore apply an upstream agreement gate, returning the agreed answer when it is satisfied and invoking additional aggregation otherwise. For independently sampled responses, the gate is satisfied when every participating model has the same unique modal answer among its completed responses; a within-model tie does not satisfy the gate. Bias control forms initial decisions; disagreement among these decisions triggers bias injection, which uses decision instability to assess their reliability on the current problem. The framework requires neither prior estimates of model strength nor a designated final aggregator and applies to independently sampled responses, multi-agent debate (Du et al., 2023), and Mixture-of-Agents (MoA) (Wang et al., 2025), without modifying upstream reasoning.

Bias control for initial decision. To improve initial answer selection, we place distinct answer hypotheses under more comparable presentation conditions. Given a problem and its upstream reasoning history, we group reasoning traces by their extracted final answers and retain one trace for each distinct answer. We present each retained answer once to balance its presentation frequency and rewrite each retained trace into a common plain-prose form to reduce source-dependent stylistic differences. We submit the resulting candidate set to J aggregating agents and let c_j denote the answer selected by agent j. If the aggregating agents unanimously select the same answer after bias control, we return it; otherwise, we proceed to bias injection.

Bias injection for decision instability assessment. Resolving disagreement requires problem-specific evidence about which decisions remain reliable. Rather than assigning fixed reliability to agents, we stress-test each controlled decision on the current problem and observe whether it changes under an adverse presentation. For each agent, we construct a second-pass input based on its controlled decision. If the reasoning history contains an additional trace for the competing answer with the greatest original support, we apply adverse-frequency injection: the initially selected answer is presented once, whereas the competing answer is presented twice using distinct normalized traces. This places the initial choice in the relative minority while leaving the set of answer hypotheses unchanged. If no such additional trace is available, we instead reverse the candidate order. Let p_j denote agent j's second-pass decision. The complete decision prompts are provided in Figure 12.

Final aggregation. To combine decisions from bias control and bias injection, we define the hold count $H(a) = |\{j : c_j = p_j = a\}|$ and second-pass support count $S(a) = |\{j : p_j = a\}|$ for each answer a. We select a if at least one agent retains it and a majority supports it on the second pass, i.e., $H(a) > 0$ and $S(a) > J/2$. If no answer satisfies this condition, we return the strict-majority answer among the controlled decisions when one exists; otherwise, we select the answer with the most votes across both stages. Original response frequencies inform candidate construction but are not counted as votes in the final decision. Algorithm 1 provides the complete aggregation procedure.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Models. We evaluate four open-weight reasoning models: Qwen3-8B (Yang et al., 2025), Phi-4-reasoning-plus (14B) (Abdin et al., 2025), GLM-Z1-9B-0414 (GLM et al., 2024), and NVIDIA-Nemotron-Nano-9B-v2 (Basant et al., 2025), referred to as Qwen, Phi, GLM, and Nemotron, respectively. We consider all four three-model combinations: Qwen–Phi–GLM, Qwen–Phi–Nemotron, Qwen–GLM–Nemotron, and Phi–GLM–Nemotron. We use Qwen as the default style normalizer; Appendix C.1 shows comparable final accuracy with different normalizers. We use the official model checkpoints and their default reasoning modes; full inference details are provided in Appendix B.

Benchmarks. We evaluate mathematical reasoning on 220 problems from three benchmarks: 100 problems from the English easy split of OlymMATH (Sun et al., 2026), 60 from AIME 2024–2025 (Mathematical Association of America), and 60 from the February HMMT 2024–2025 competitions (Harvard–MIT Mathematics Tournament). We report final-answer accuracy using the same answer normalization and equivalence checks across all benchmarks and aggregation methods.

Baselines. *Combined MV* selects the most frequent answer across all sampled responses. *SLM-MUX* (Wang et al., 2026a) selects among model-wise modal answers using within-model consistency. Because our setting fixes model combinations and assumes no validation data, we omit its validation-based subset search and replace its validation-accuracy tie-breaker with plurality voting over the tied models' completed samples, breaking residual ties by first occurrence. Appendix C.2 evaluates a test-set-informed oracle tie-breaker; Ours achieves higher average accuracy under either tie-breaking rule. *Multi-Agent Verification* (MAV) (Lifshitz et al., 2025) applies aspect-specific binary verification to the same candidate answers as our method, using one representative trace per answer. For each selected aspect, all three participating models evaluate every candidate, and we select the candidate with the most approvals. We also report *Single-Best-SC* (Wang et al., 2022), which retrospectively selects the strongest individual model under self-consistency and serves as an oracle model-selection reference rather than a deployable baseline.

Evaluation protocol. Our primary metric is final-answer accuracy, and we use mean total token usage, including input and output tokens, per problem as a proxy for inference cost. Methods with LLM-based aggregation are run three times over the fixed response pools, and their accuracies are averaged; rule-based methods are evaluated once. For MAV, we fix $k = 3$ and cumulatively include answer correctness, reasoning validity, and solution completeness to obtain three accuracy–cost points. For MoA, we evaluate each participating model as the final aggregator and retrospectively report the best-performing choice as an oracle reference. We target k completed responses per model, allowing at most $2k$ generation attempts and including tokens from all attempts in the reported cost. Our primary setting uses $k = 3$, with additional response budgets used to characterize the accuracy–cost trade-off. In this setting, the agreement-routing rule in Section 4 operates on the model-wise majority answers over the completed responses. In our framework, we retain up to four distinct candidate answers with the highest frequencies in the reasoning history and present them in descending frequency order.

5.2 EXPERIMENTAL RESULTS

Main results. For independently generated multi-LLM reasoning, our primary comparison examines the accuracy–cost trade-off across response budgets (Figure 5). Across all four model combinations, most Ours configurations with $k \in \{2, 3, 4\}$ lie on or near the observed Pareto frontier. Table 2 complements this comparison under our budget-matching protocol, reporting mean $\pm$ standard deviation over three runs for LLM-based aggregation methods. Here, Ours achieves the highest average accuracy in all four model combinations, outperforming the strongest non-oracle baseline by 3.2–6.2 percentage points and the retrospective Single-Best-SC oracle reference by 1.8–4.3 points. It also achieves the best or tied-best non-oracle accuracy in 12 of the 12 benchmark–combination pairs. We additionally evaluate a common subset of instances not resolved by the upstream agreement gate, determined from the $k = 3$ response pools before aggregation. Applying the same budget-matching rule within this subset, Ours outperforms the strongest non-oracle baseline by 5.8–9.5 percentage points across model combinations (Appendix C.3).

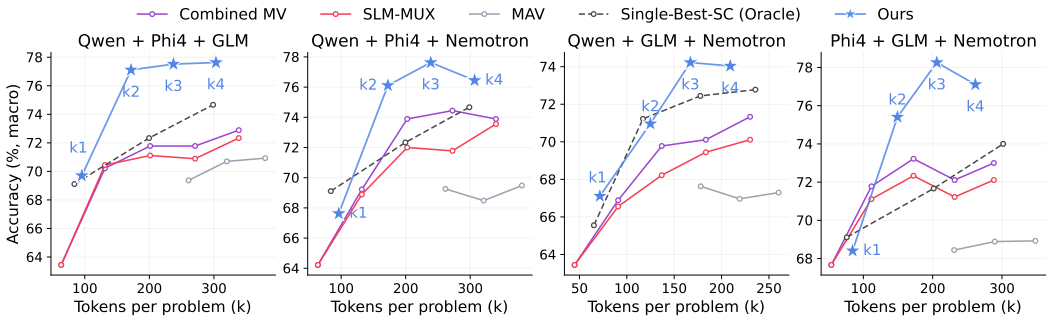


Figure 5: **Accuracy–cost trade-offs under independent sampling.** Each panel shows one three-model combination across response budgets. Accuracy is the macro-average across OlymMATH, AIME, and HMMT, and inference cost is measured by total input-and-output tokens per problem. Stars mark Ours. Most Ours configurations with $k \in \{2, 3, 4\}$ lie on or near the observed Pareto frontier.

Table 2: **Final-answer accuracy and inference efficiency across heterogeneous LLM combinations.** Each baseline uses the nearest evaluated token budget at or above Ours. Average is the unweighted benchmark mean, and Tokens denotes input-and-output usage per problem. For LLM-based aggregation methods, values are mean ± standard deviation over three runs. Single-Best-SC is an oracle model-selection reference. Bold and underline mark the first- and second-ranked results, respectively. Ours achieves the highest average accuracy with fewer tokens in every combination.

Model combination	Method	OlymMATH ↑	AIME ↑	HMMT ↑	Average ↑	Tokens (10^3) ↓
Qwen–Phi–GLM	Single-Best-SC (Wang et al., 2022)	<u>89.0</u>	**83.3**	51.7	<u>74.7</u>	298.7
	Combined MV	87.0	<u>80.0</u>	48.3	71.8	270.8
	MAV (Lifshitz et al., 2025)	82.0±2.0	71.1±2.5	<u>55.0</u>±2.9	69.4±0.9	<u>260.6</u>±0.9
	SLM-MUX (Wang et al., 2026a)	86.0	78.3	48.3	70.9	270.8
	Ours	**90.3**±1.2	**83.3**±1.7	**58.9**±1.9	**77.5**±1.5	**237.3**±1.2
Qwen–Phi–Nemotron	Single-Best-SC	89.0	**83.3**	51.7	<u>74.7</u>	298.7
	Combined MV	<u>90.0</u>	78.3	55.0	74.4	272.5
	MAV	81.7±0.6	70.6±1.0	<u>55.6</u>±4.2	69.3±1.2	<u>261.6</u>±0.9
	SLM-MUX	87.0	76.7	51.7	71.8	272.5
	Ours	**90.7**±0.6	<u>82.8</u>±1.0	**59.4**±1.9	**77.6**±0.1	**238.9**±1.0
Qwen–GLM–Nemotron	Single-Best-SC	<u>89.0</u>	**80.0**	48.3	<u>72.4</u>	<u>177.6</u>
	Combined MV	87.0	76.7	46.7	70.1	183.4
	MAV	77.3±3.2	75.6±2.5	<u>50.0</u>±2.9	67.6±2.0	178.0±0.6
	SLM-MUX	85.0	76.7	46.7	69.4	183.4
	Ours	**89.3**±1.5	<u>78.9</u>±1.9	**54.4**±1.0	**74.2**±0.6	**167.1**±1.0
Phi–GLM–Nemotron	Single-Best-SC	87.0	<u>83.3</u>	<u>51.7</u>	<u>74.0</u>	301.7
	Combined MV	<u>88.0</u>	80.0	48.3	72.1	231.7
	MAV	82.0±1.0	71.7±2.9	<u>51.7</u>±2.9	68.4±2.0	<u>231.0</u>±0.4
	SLM-MUX	87.0	78.3	48.3	71.2	231.7
	Ours	**88.7**±0.6	**85.0**±0.0	**61.1**±1.0	**78.3**±0.5	**206.0**±2.0

Recovery and retention relative to majority voting. We examine whether aggregation recovers overlooked correct answers while preserving correct selections. Using the same $k = 3$ reference response pools as Figure 1, we define recovery as accuracy when a correct answer is available but Combined MV is incorrect, and retention as accuracy when Combined MV is correct. As shown in Figure 6, Ours achieves both the highest recovery rate, at 53.6%, and the highest retention rate, at 98.4%. MAV recovers 40.0% of missed answers with 89.1% retention, whereas SLM-MUX retains 96.4% but recovers only 11.2%. Single-Best-SC, included as an oracle reference with a target of nine completed

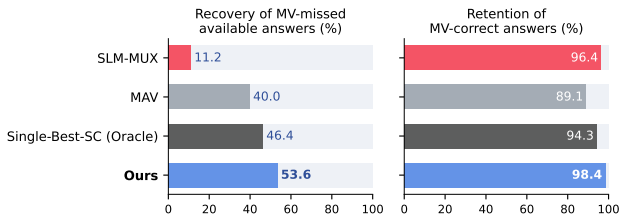


Figure 6: **Recovery and retention relative to majority voting.** Rates are macro-averaged across benchmarks. Ours achieves the highest recovery while retaining most MV-correct answers.

responses from the retrospectively strongest model, achieves 46.4% recovery and 94.3% retention. These results show that Ours recovers more available correct answers while preserving nearly all MV-correct selections.

Application to multi-agent reasoning. We next apply the same decision procedure to reasoning histories produced by different multi-agent interaction frameworks such as *multi-agent debate* (Debate) (Du et al., 2023) and *Mixture-of-Agents* (MoA) (Wang et al., 2025). For each configuration, we keep the upstream reasoning history fixed and compare the native final decision with the corresponding +OURS variant. Since interaction may lead to biased consensus, we apply +OURS only when final answers disagree, their shared answer is supported by fewer than two initial, pre-interaction responses, or, for Debate, any agent changes its answer between the last two rounds. For MoA, agreement is checked among all final-layer proposers and the selected aggregator. Figure 7 shows that our method improves final-answer accuracy for both frameworks at a moderate additional inference cost.

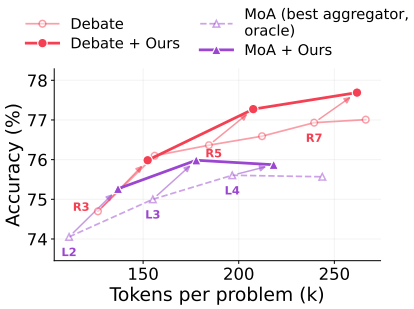


Figure 7: **Integration with multi-agent reasoning.** Arrows show pooled-accuracy gains from adding Ours. R/L denote Debate rounds/MoA layers.

6 ANALYSIS

6.1 UNDERSTANDING STYLE NORMALIZATION

More than formatting removal. We examine whether the cross-generator convergence induced by style normalization can be explained by formatting removal alone. We construct matched pairs of reasoning traces produced by different generators for the same problem and answer, and compare deterministic formatting removal with full normalization. Figure 8 shows the comparison in the Phi representation space, which is not used for rewriting. Formatting removal reduces mean representation distance by only 1.7%, whereas full normalization reduces it by 8.2%. Results in the representation spaces of three LLMs are provided in Appendix D.1. Additional source-attribution analyses assess how accurately a trace's generating model can be identified from presentation features. Normalization reduces identification accuracy, particularly for formatting features, suggesting weaker model-specific presentation cues (Appendix D.2).

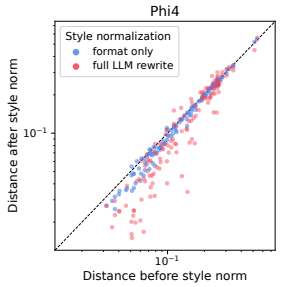


Figure 8: **Cross-generator convergence in Phi.** Cosine distances between matched traces.

6.2 UNDERSTANDING BIAS INJECTION

Does bias injection yield a more reliable retained set? Section 3.2 showed that adverse-frequency bias injection dislodges incorrect decisions more selectively than repeated inference or position swap. We analyze instances where the controlled decisions are not unanimous and a second pass is issued, comparing the reliability of decisions that remain unchanged under each second-pass condition. As shown in Table 3, adverse-frequency injection retains 37.5% of controlled decisions, whose conditional accuracy is 46.9%, compared with 42.4% under position swap and 40.6% under repeated inference. Relative to the common controlled-decision accuracy of 30.3%, adverse-frequency injection yields the largest gain in retained-set accuracy, at 16.6 percentage points. It thus retains fewer decisions but produces a more accurate retained set, supporting decision persistence as a reliability signal. Additional analysis of final aggregation is provided in Appendix D.3.

Table 3: **Reliability of retained decisions.** Hold rate is the fraction unchanged on the second pass; held accuracy is their conditional accuracy. Δ is the gain over the 30.3% controlled baseline.

Condition	Hold rate (%)	Held acc. (%)	Δ (pp)
Repeated inference	53.7	40.6	+10.3
Position swap	52.4	42.4	+12.1
Adverse frequency	37.5	46.9	+16.6

Does selectivity persist without new reasoning content? Our default injection adds a distinct trace for a competing answer, changing both presentation frequency and reasoning content. To examine the role of additional reasoning content, we compare second-pass conditions on instances with upstream disagreement, using the same initial controlled decisions across conditions. The exact-copy condition duplicates the competing trace already present, holding the competing answer, presentation order, and number of input blocks fixed. As shown in Table 4, exact-copy injection yields an incorrect–correct flip gap of 37.3 percentage points, compared with 32.5 for repeated inference and 34.3 for position swap. Using a distinct trace yields the largest observed gap, at 40.7 points. These results suggest a role for frequency bias in selectivity and support using distinct reasoning traces when available.

Table 4: **Content control for bias injection.** Flip rates (%) are conditioned on initial correctness. Gap is the incorrect–correct difference in percentage points.

Condition	Flip | incorrect	Flip | correct	Gap
Repeated inference	44.2	11.7	32.5
Position swap	48.3	14.0	34.3
Adverse freq. (exact copy)	56.2	18.9	37.3
Adverse freq. (distinct)	59.8	19.1	40.7

6.3 Ablation

Figure 9 compares interventions on common evaluation sets comprising all instances reaching each stage. Accuracy is pooled across four model combinations and three runs. For bias control, we evaluate instances unresolved by the initial agreement gate, since normalization can affect subsequent controlled decisions and their agreement. Removing normalization reduces accuracy from 63.4% to 61.3%. For bias injection, we fix controlled decisions and compare second-pass strategies on instances where they disagree. The second-pass variants use the same final aggregation rule, while the no-second-pass baseline uses controlled decisions alone. Adverse-frequency injection achieves 41.6% accuracy, compared with 37.6% without a second pass, 39.2% with repeated inference, and 39.3% with position swap. Among strategies using the same number of second-pass decisions, adverse-frequency injection yields the highest accuracy under the shared aggregation rule.

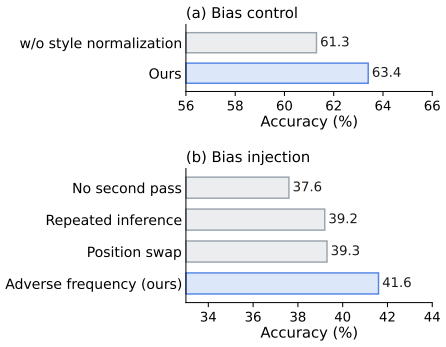


Figure 9: **Component ablations.** Answer accuracy for (a) bias control and (b) bias injection.

7 Conclusions

We introduced a training- and validation-free framework that both mitigates and exploits biases in multi-agent aggregation. Bias control reduces the influence of frequency and style to improve initial decisions, while bias injection elicits decision instability as a problem-specific reliability signal. Across three mathematical reasoning benchmarks and four heterogeneous model combinations, the framework improves accuracy and token efficiency. It also improves final-answer aggregation in debate and MoA without modifying their upstream processes.

8 Limitations

First, our evaluation focuses on mathematical reasoning with four open-weight models in the 8B–14B range. Broader evaluation would help assess how the benefits of bias control and the reliability signal from bias injection extend to other domains and larger models. Second, we study bias interventions at final aggregation, including for Debate and MoA, while keeping upstream reasoning histories fixed. Extending these interventions to interaction or synthesis is a direction for future work, with potential effects on both solution diversity and how agents reach agreement.

Reproducibility Statement

The aggregation algorithm, experimental settings, and prompts are provided in Appendices A, B, and E, respectively. We will also release the source code and scripts needed to reproduce the reported experiments.

AI Use Statement

Generative AI assistants were used for manuscript editing and formatting, and for assistance with experimental and analysis code. All AI-assisted outputs were reviewed and verified by the authors, who take full responsibility for the final content of this work.

References

Marah Abdin, Sahaj Agarwal, Ahmed Awadallah, Vidhisha Balachandran, Harkirat Behl, Lingjiao Chen, Gustavo de Rosa, Suriya Gunasekar, Mojan Javaheripi, Neel Joshi, et al. Phi-4-reasoning technical report. *arXiv preprint arXiv:2504.21318*, 2025.

Rui Ai, Yuqi Pan, David Simchi-Levi, Milind Tambe, and Haifeng Xu. Beyond majority voting: Llm aggregation by leveraging higher-order information. *arXiv preprint arXiv:2510.01499*, 2025.

Sher Badshah, Ali Emami, and Hassan Sajjad. Scope: Selective conformal optimized pairwise llm judging. *arXiv preprint arXiv:2602.13110*, 2026.

Aarti Basant, Abhijit Khairnar, Abhijit Paithankar, Abhinav Khattar, Adithya Renduchintala, Aditya Malte, Akhiad Bercovich, Akshay Hazare, Alejandra Rico, Aleksander Ficek, et al. Nvidia nemotron nano 2: An accurate and efficient hybrid mamba-transformer reasoning model. *arXiv preprint arXiv:2508.14444*, 2025.

Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. Humans or llms as the judge? a study on judgement bias. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8301–8327, 2024.

Zhi-Yuan Chen, Hao Wang, Xinyu Zhang, Enrui Hu, and Yankai Lin. Beyond the surface: Measuring self-preference in llm judgments. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 1653–1672, 2025.

Hyeong Kyu Choi, Jerry Zhu, and Sharon Li. Debate or vote: Which yields better decisions in multi-agent large language models? *Advances in Neural Information Processing Systems*, 38:101732–101764, 2026a.

Hyeong Kyu Choi, Jerry Zhu, and Sharon Li. When identity skews debate: Anonymization for bias-reduced multi-agent reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14284–14311, 2026b.

Min Choi, Keonwoo Kim, Sungwon Chae, and Sangyeop Baek. An empirical study of group conformity in multi-agent systems. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 5123–5139, 2025.

Yu Cui, Hang Fu, Haibin Zhang, Licheng Wang, and Cong Zuo. Free-mad: Consensus-free multi-agent debate. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 31977–31997, 2026.

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

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.

Benjamin Feuer, Micah Goldblum, Teresa Datta, Sanjana Nambiar, Raz Besaleli, Samuel Dooley, Max Cembalest, and John P Dickerson. Style outweighs substance: Failure modes of llm judges in alignment benchmarking. In *International Conference on Learning Representations*, volume 2025, pp. 11552–11582, 2025.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, et al. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*, 2024.

Kimia Hamidieh, Veronika Thost, Walter Gerych, Mikhail Yurochkin, and Marzyeh Ghassemi. Complementing self-consistency with cross-model disagreement for uncertainty quantification. *arXiv preprint arXiv:2604.17112*, 2026.

Harvard–MIT Mathematics Tournament. Hmmt (harvard–mit mathematics tournament). URL https://www.hmmt.org/.

Yerin Hwang, Dongryeol Lee, Taegwan Kang, Yongil Kim, and Kyomin Jung. Can you trick the grader? adversarial persuasion of llm judges. In *EMNLP (Findings)*, pp. 14632–14651, 2025.

Hamin Koo, ChanJoo Jung, Fangzhao Wu, and Jaehyung Kim. Auditing the judge: Human-grounded bias discovery, quantification, and mitigation in llm judges. In *Trustworthy AI for Good (AI4GOOD) Workshop@ ICML 2026*, 2026.

Haitao Li, Junjie Chen, Qingyao Ai, Zhumin Chu, Yujia Zhou, Qian Dong, and Yiqun Liu. Calibraeval: Calibrating prediction distribution to mitigate selection bias in llms-as-judges. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16537–16552, 2025.

Zongjie Li, Chaozheng Wang, Pingchuan Ma, Daoyuan Wu, Shuai Wang, Cuiyun Gao, and Yang Liu. Split and merge: Aligning position biases in llm-based evaluators. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 11084–11108, 2024.

Shalev Lifshitz, Sheila A McIlraith, and Yilun Du. Multi-agent verification: Scaling test-time compute with multiple verifiers. *arXiv preprint arXiv:2502.20379*, 2025.

Chiyu Ma, Enpei Zhang, Yilun Zhao, Wenjun Liu, Yaning Jia, Peijun Qing, Lin Shi, Arman Cohan, Yujun Yan, and Soroush Vosoughi. Judging with many minds: Do more perspectives mean less prejudice? on bias amplification and resistance in multi-agent based llm-as-judge. In *EMNLP (Findings)*, pp. 17356–17392, 2025.

Mathematical Association of America. American invitational mathematics examination (aime). URL https://maa.org/maa-invitational-competitions/.

Maya Okawa. Emergence of biased consensus in multi-agent llm debates. In *Forty-third International Conference on Machine Learning*, 2026.

Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. Judging the judges: A systematic study of position bias in llm-as-a-judge. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 292–314, 2025.

Andries Smit, Paul Duckworth, Nathan Grinsztajn, Thomas D Barrett, and Arnu Pretorius. Should we be going mad? a look at multi-agent debate strategies for llms. *arXiv preprint arXiv:2311.17371*, 2023.

Haoxiang Sun, Yingqian Min, Zhipeng Chen, Wayne Xin Zhao, and Ji-Rong Wen. Challenging the boundaries of reasoning: An olympiad-level math benchmark for large language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17438–17457, 2026.

Hongduan Tian, Xiao Feng, Ziyuan Zhao, Xiangyu Zhu, Rolan Yan, and Bo Han. Multi-agent debate with memory masking. In *International Conference on Learning Representations*, volume 2026, pp. 5122–5143, 2026.

Chenyu Wang, Zishen Wan, Hao Kang, Emma Chen, Zhiqiang Xie, Tushar Krishna, Vijay Janapa Reddi, and Yilun Du. Slm-mux: Orchestrating small language models for reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 73697–73723, 2026a.

Fuyu Wang, Jiangtong Li, Kun Zhu, and Changjun Jiang. Can llms really judge? a progressive argumentation-mining framework for distinguishing understanding from aggregation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 29463–29482, 2026b.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Y Zou. Mixture-of-agents enhances large language model capabilities. In *International Conference on Learning Representations*, volume 2025, pp. 33944–33963, 2025.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, et al. Large language models are not fair evaluators. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 9440–9450, 2024a.

Qineng Wang, Zihao Wang, Ying Su, Hanghang Tong, and Yangqiu Song. Rethinking the bounds of llm reasoning: Are multi-agent discussions the key? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6106–6131, 2024b.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Bo Yang, Lanfei Feng, Yunkui Chen, Yu Zhang, Xiao Xu, and Shijian Li. Fairjudge: An adaptive, debiased, and consistent llm-as-a-judge. *arXiv preprint arXiv:2602.06625*, 2026a.

Haoyan Yang, Runxue Bao, Cao Danica Xiao, Jun Ma, Parminder Bhatia, Shangqian Gao, and Taha Kass-Hout. Any large language model can be a reliable judge: Debiasing with a reasoning-based bias detector. *Advances in Neural Information Processing Systems*, 38:6318–6362, 2026b.

Xianglin Yang, Bryan Hooi, Gelei Deng, Tianwei Zhang, and Jin Song Dong. Turning bias into bugs: Bandit-guided style manipulation attacks on llm judges. *arXiv preprint arXiv:2605.26156*, 2026c.

Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, et al. Justice or prejudice? quantifying biases in llm-as-a-judge. In *International Conference on Learning Representations*, volume 2025, pp. 102351–102390, 2025.

Xiaochen Zhu, Caiqi Zhang, Tom Stafford, Nigel Collier, and Andreas Vlachos. Conformity in large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3854–3872, 2025.

Xiaochen Zhu, Caiqi Zhang, Yizhou Chi, Tom Stafford, Nigel Collier, and Andreas Vlachos. Demystifying multi-agent debate: The role of confidence and diversity. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 33909–33930, 2026.

A AGGREGATION ALGORITHM

Algorithm 1 summarizes the complete aggregation procedure.

Algorithm 1 Bias Control and Bias Injection for Multi-Agent Aggregation

Input: Problem q, reasoning history $\mathcal{H} = \{(r_i, a_i)\}_{i=1}^{N}$, upstream conclusions $\mathcal{U}$, judges $\{\mathcal{J}_j\}_{j=1}^{J}$, normalizer G_{norm}
Output: Final answer $\widehat{a}$

Step 1: Agreement Routing

if the upstream agreement gate is satisfied **then**
  return the agreed answer
end if

Step 2: Bias Control for Decision Formation

Group $\mathcal{H}$ by answer into $\{\mathcal{R}_a\}_{a\in\mathcal{A}}$ and set $n(a) \leftarrow |\mathcal{R}_a|$
Select one $r_a \in \mathcal{R}_a$ and set $\widetilde{r}_a \leftarrow G_{\text{norm}}(r_a)$ for each $a \in \mathcal{A}$
$\mathcal{S}^{\text{ctrl}} \leftarrow [(\widetilde{r}_a, a)]_{a\in\mathcal{A}}$, ordered by decreasing $n(a)$
$c_j \leftarrow \mathcal{J}_j(q, \mathcal{S}^{\text{ctrl}})$ for $j = 1, \ldots, J$
if $c_1 = \cdots = c_J$ **then**
  return c_1
end if

Step 3: Bias Injection for Reliability Assessment

for $j = 1, \ldots, J$ **do**
  $b_j \leftarrow \arg\max_{a\in\mathcal{A}\setminus\{c_j\}} n(a)$
  if an unused trace $r'_{b_j} \in \mathcal{R}_{b_j}$ exists **then**
    Form $\mathcal{S}_j^{\text{probe}}$ by adding $(G_{\text{norm}}(r'_{b_j}), b_j)$ to $\mathcal{S}^{\text{ctrl}}$, preserving the order of the existing candidates
  else
    $\mathcal{S}_j^{\text{probe}} \leftarrow \text{Reverse}(\mathcal{S}^{\text{ctrl}})$
  end if
  $p_j \leftarrow \mathcal{J}_j(q, \mathcal{S}_j^{\text{probe}})$
end for

Step 4: Final Aggregation

For each a, compute $H(a) = |\{j : c_j = p_j = a\}|$ and $S(a) = |\{j : p_j = a\}|$
if some a satisfies $H(a) > 0$ and $S(a) > J/2$ **then**
  return a
else
  return the strict-majority answer among $\{c_j\}_{j=1}^{J}$ if one exists; otherwise, the most-voted answer across $\{c_j\}_{j=1}^{J}$ and $\{p_j\}_{j=1}^{J}$
end if

B EXPERIMENTAL SETTINGS

Table 5 summarizes the main hyperparameters. For CoT generation, we follow the sampling settings recommended in the respective models' official model cards, where available, for temperature, top-p, and top-k. All methods use the same CoT generation rule: generation stops after k completed responses or $2k$ attempts per model, whichever occurs first. Truncated responses are discarded and resampled within the attempt limit, and tokens from all attempts are included in the reported cost. We prompt the normalizer to target 85–115% of the original length (Figure 11), while allowing a wider 80–120% range when accepting normalized traces.

Table 5: Hyperparameters used in the experiments.

Parameter	Value	
CoT generation		
Model	**Temperature**	**Top-k**
Phi-4-reasoning-plus	0.8	50
Qwen3-8B	0.6	20
GLM-Z1-9B-0414	0.6	40
NVIDIA-Nemotron-Nano-9B-v2	0.6	Disabled
Top-p	0.95	
Maximum output tokens	32,768	
Maximum generation attempts per model	Twice the target number of completed responses	
Style normalization		
Rewriter	Qwen3-8B	
Temperature	0.6	
Top-p	0.95	
Maximum output tokens	28,672	
Maximum retries (failed length or answer-preservation check)	2	
Accepted length ratio	$[0.8, 1.2]$	
LLM decision		
Number of aggregating agents	3	
Temperature	0.6	
Top-p	0.95	
Maximum output tokens	32,768	
Maximum retries (invalid output)	2	
Maximum candidate answers per prompt	4	
Maximum trace length per candidate (characters)	6,000	
Multi-agent baselines		
Temperature	0.6	
Top-p	0.95	
Maximum output tokens	32,768	
Maximum retries (truncated output)	4	

C Extended Experimental Results

C.1 Robustness to the Style Normalizer

We test whether the framework depends on Qwen3-8B as the style normalizer by replacing it with GLM-Z1-9B or Nemotron-Nano-9B while keeping the response pools and other settings fixed. Because the alternative normalizers were evaluated once, we use the first Qwen3-8B run as the matched reference rather than its three-run average. As shown in Table 6, replacing Qwen3-8B with GLM-Z1-9B or Nemotron-Nano-9B changes average accuracy by -2.4 to $+1.3$ percentage points across the four model combinations. These results suggest that the framework can also be used with alternative style normalizers.

C.2 Tie-Breaking in SLM-MUX

SLM-MUX selects the modal answer of the model with the highest within-model consistency and originally resolves confidence ties using validation-set accuracy. Because our setting assumes no validation data, our main implementation instead applies label-free plurality voting over the pooled responses from tied models. To assess the effect of this substitution, we additionally evaluate a test-set Best-SC oracle that selects the tied model with the highest SC@k accuracy on the corresponding test set. The two SLM-MUX variants use identical responses and inference cost. As shown in Table 7,

Table 6: **Final-answer accuracy with alternative style normalizers.** Average is the unweighted mean across the three benchmarks, and Δ_{Qwen} is its difference from the matched Qwen3-8B run. Alternative normalizers retain comparable performance across model combinations.

Model combination	Style normalizer	OlymMATH	AIME	HMMT	Average	Δ_{Qwen} (pp)
Qwen–Phi–GLM	Qwen3-8B	89.0	81.7	56.7	75.8	–
	GLM-Z1-9B	89.0	83.3	58.3	76.9	+1.1
	Nemotron-Nano-9B	93.0	81.7	56.7	77.1	+1.3
Qwen–Phi–Nemotron	Qwen3-8B	91.0	83.3	58.3	77.6	–
	GLM-Z1-9B	92.0	81.7	58.3	77.3	−0.3
	Nemotron-Nano-9B	90.0	80.0	63.3	77.8	+0.2
Qwen–GLM–Nemotron	Qwen3-8B	91.0	80.0	53.3	74.8	–
	GLM-Z1-9B	89.0	76.7	51.7	72.4	−2.4
	Nemotron-Nano-9B	88.0	80.0	51.7	73.2	−1.6
Phi–GLM–Nemotron	Qwen3-8B	89.0	85.0	61.7	78.6	–
	GLM-Z1-9B	87.0	83.3	60.0	76.8	−1.8
	Nemotron-Nano-9B	89.0	85.0	58.3	77.4	−1.2

Table 7: **SLM-MUX under alternative confidence tie-breakers.** The two SLM-MUX variants differ only in confidence tie-breaking and use identical responses and inference cost. Test-set Best-SC selects the tied model with the highest test-set SC@k accuracy and serves as an oracle reference; Pooled MV is the label-free rule used in the main experiments. Ours is included for comparison, and bold marks the highest average.

Model combination	Method	Tie-breaker	OlymMATH	AIME	HMMT	Average
Qwen–Phi–GLM	SLM-MUX	Test-set Best-SC (oracle)	87.0	80.0	53.3	73.4
	SLM-MUX	Pooled MV (main)	86.0	78.3	48.3	70.9
	Ours	–	90.3	83.3	58.9	**77.5**
Qwen–Phi–Nemotron	SLM-MUX	Test-set Best-SC (oracle)	88.0	78.3	55.0	73.8
	SLM-MUX	Pooled MV (main)	87.0	76.7	51.7	71.8
	Ours	–	90.7	82.8	59.4	**77.6**
Qwen–GLM–Nemotron	SLM-MUX	Test-set Best-SC (oracle)	84.0	80.0	46.7	70.2
	SLM-MUX	Pooled MV (main)	85.0	76.7	46.7	69.4
	Ours	–	89.3	78.9	54.4	**74.2**
Phi–GLM–Nemotron	SLM-MUX	Test-set Best-SC (oracle)	84.0	78.3	53.3	71.9
	SLM-MUX	Pooled MV (main)	87.0	78.3	48.3	71.2
	Ours	–	88.7	85.0	61.1	**78.3**

the oracle tie-breaker yields higher average accuracy than Pooled MV in all four model combinations, while Ours exceeds the oracle variant by 3.8–6.4 percentage points.

C.3 Aggregation under Upstream Disagreement

We evaluate final-answer aggregation on instances not resolved by the upstream agreement gate in the independent-sampling setting with a target of $k = 3$ completed responses per model. The evaluation subset is determined from these upstream responses before aggregation and is shared across all compared methods.

We reapply the budget-matching rule from Table 2 using mean token costs on this subset: each baseline uses the lowest evaluated token budget at or above Ours. Ours uses $k = 3$, while the selected baseline configurations are reported in Table 8. LLM-based aggregation results are averaged over three runs, and rule-based methods are evaluated once. Single-Best-SC remains an oracle model-selection reference. Ours achieves the highest average accuracy in all four model combinations, with gains of 5.8–9.5 percentage points over the strongest non-oracle baseline.

Table 8: **Final-answer accuracy and token usage under upstream disagreement.** Each baseline uses the nearest evaluated token budget at or above Ours on this subset. Average is the unweighted benchmark mean, and Tokens denotes input-and-output usage per problem. For LLM-based aggregation methods, values are means over three runs. Single-Best-SC is an oracle model-selection reference. Bold and underline mark the first- and second-ranked results, respectively.

Model combination	Method	Setting	OlymMATH ↑	AIME ↑	HMMT ↑	Average ↑	Tokens (10^3) ↓
Qwen–Phi–GLM	Single-Best-SC (Wang et al., 2022)	$k = 9$	75.6	**64.3**	39.1	59.7	380.3
	Combined MV	$k = 4$	71.1	57.1	34.8	54.3	343.4
	MAV (Lifshitz et al., 2025)	1 aspect	61.5	41.7	43.5	48.9	346.1
	SLM-MUX (Wang et al., 2026a)	$k = 4$	68.9	53.6	34.8	52.4	343.4
	Ours	$k = 3$	**78.5**	**64.3**	**48.6**	**63.8**	**322.1**
Qwen–Phi–Nemotron	Single-Best-SC	$k = 9$	76.1	**66.7**	36.4	59.7	383.8
	Combined MV	$k = 4$	78.3	55.6	40.9	58.2	349.5
	MAV	1 aspect	61.6	39.5	41.7	47.6	350.3
	SLM-MUX	$k = 4$	71.7	51.9	36.4	53.3	349.5
	Ours	$k = 3$	**79.7**	65.4	**47.0**	**64.0**	**327.8**
Qwen–GLM–Nemotron	Single-Best-SC	$k = 9$	79.2	**55.6**	31.7	55.5	230.4
	Combined MV	$k = 4$	75.0	48.1	29.3	50.8	233.1
	MAV	1 aspect	57.6	45.7	34.1	45.8	237.1
	SLM-MUX	$k = 4$	70.8	48.1	29.3	49.4	233.1
	Ours	$k = 3$	**79.9**	53.1	**40.7**	**57.9**	**229.8**
Phi–GLM–Nemotron	Single-Best-SC	$k = 9$	71.7	67.7	39.1	59.5	396.4
	Combined MV	$k = 4$	76.1	61.3	34.8	57.4	294.3
	MAV	1 aspect	63.8	45.2	39.1	49.4	307.8
	SLM-MUX	$k = 4$	73.9	58.1	34.8	55.6	294.3
	Ours	$k = 3$	**77.5**	**71.0**	**51.4**	**66.7**	**279.9**

D Additional Analyses

D.1 Cross-Generator Convergence Across Representation Spaces

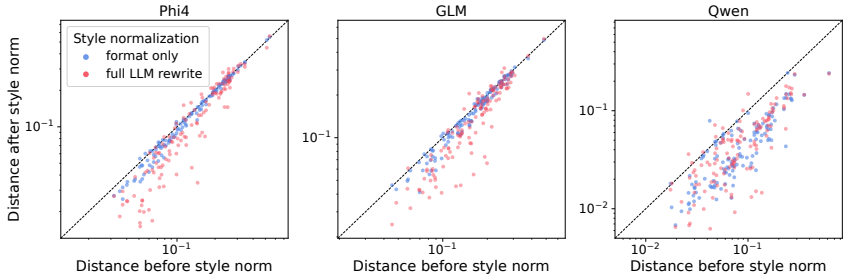


Figure 10: **Cross-generator convergence across representation spaces.** Panels show cosine distances in the Phi, GLM, and Qwen representation spaces between matched traces from different generators before and after formatting removal or full normalization. Traces are matched by problem and final answer. Points below the diagonal indicate reduced distance. Full normalization produces larger reductions than formatting removal, particularly in the Phi and GLM spaces.

Figure 10 presents the full comparison across the Phi, GLM, and Qwen representation spaces. Formatting removal reduces mean representation distance by 1.7% and 2.2% in Phi and GLM, respectively, whereas full normalization reduces it by 8.2% and 8.5%.

D.2 Source-Generator Attribution

We use source attribution to diagnose which generator-dependent signals remain before and after normalization. For each feature channel, we train an L2-regularized multinomial logistic-regression classifier using reasoning traces. Evaluation uses five-fold cross-validation grouped by $(\text{dataset}, \text{problem})$ to prevent the same problem from appearing in both training and test folds. Count features are length-normalized, and lexical vocabularies are fit on the original traces and reused after normalization. Table 9 reports attribution accuracy before and after normalization across the six feature channels.

Table 9: **Source-generator attribution before and after style normalization.** Values report three-way classification accuracy under grouped five-fold cross-validation; Δ uses unrounded values. Normalization attenuates formatting cues most strongly, while substantial source signal remains.

Feature channel	Original	Normalized	Δ
Formatting	94.1	72.2	-21.9
Rhetorical markers	88.8	79.7	-9.1
Character 3-grams	99.2	91.9	-7.3
Bag of words	94.6	88.5	-6.1
Function words	86.6	80.9	-5.7
Judge representation	100.0	97.2	-2.8

Normalization reduces attribution accuracy most strongly for formatting features (94.1% to 72.2%), with smaller reductions for rhetorical markers (88.8% to 79.7%) and lexical features. These decreases suggest that the corresponding feature sets become less informative about the source model, although substantial source information remains after normalization. They measure source distinguishability within feature groups, rather than each group's contribution to representation-level convergence.

D.3 Additional Analysis of Final Aggregation

Table 10: **Answer accuracy by hold and support.** Conditional accuracy (%) by $H(a)$ and $S(a)$. J denotes the number of agents.

	$H(a) = 0$	$H(a) > 0$
$S(a) \leq J/2$	13.3	23.5
$S(a) > J/2$	18.1	49.1

On instances where the controlled decisions are not unanimous and adverse-frequency injection is applied, we analyze candidate-answer accuracy conditioned on hold count $H(a)$ and second-pass support count $S(a)$. Here, $H(a)$ counts judges selecting answer a in both the controlled and probe passes, while $S(a)$ counts probe judgments selecting a. As shown in Table 10, answers with both a hold and majority probe support have the highest conditional accuracy among the four groups: 49.1%, compared with 18.1% for majority support without a hold and 23.5% for a hold without majority support.

D.4 Examples of Style Normalization

Tables 11 and 12 present the complete original and normalized traces supporting the correct answer 6 and the competing answer 1, respectively, from the example illustrated in Figure 2. In both examples, normalization changes wording and presentation while preserving the main reasoning steps, calculations, and final answers, including the errors in the competing trace. The normalization prompt contained the original trace and rewriting instructions; the problem statement was not included as a separate input. Without style normalization, agents judge the original traces (shown in red and blue); with normalization, they judge versions with more consistent presentation (shown in gray).

Table 11: **Complete original and normalized traces for the correct answer $\boxed{6}$.**

Original trace	**Style-normalized trace**
To determine the number of ways to divide a 20×24 rectangle into 4×5 rectangles, we first verify the feasibility of such a tiling: **Feasibility Check** • **Area of the large rectangle:** $$20 \times 24 = 480$$ • **Area of each tile:** $$4 \times 5 = 20$$ • **Number of tiles needed:** $$\frac{480}{20} = 24$$ Since the area is divisible and the tile dimensions are smaller than the rectangle, a tiling is possible. However, we must also ensure that the tiling can be done with the given dimensions, accounting for the fact that the tile can be rotated (i.e., placed as 4×5 or 5×4).	To determine the number of ways to divide a 20×24 rectangle into 4×5 rectangles, we first verify the feasibility of such a tiling. The area of the large rectangle is $20 \times 24 = 480$. Each tile has an area of $4 \times 5 = 20$, so the number of tiles required is $480 \div 20 = 24$. Since the area is divisible and the tile dimensions are smaller than the rectangle, a tiling is possible. However, we must also ensure that the tiling can be done with the given dimensions, accounting for the fact that the tile can be rotated.
Key Insight: Tiling via Segmentations To tile the entire rectangle, we can divide the **width** (24 units) into segments of either 4 or 5 units. Each such segment can then be filled with tiles of the appropriate orientation: • A segment of **4 units wide** can be tiled with **vertical tiles** (5 units tall, 4 units wide). • A segment of **5 units wide** can be tiled with **horizontal tiles** (4 units tall, 5 units wide). Since the height of the rectangle is 20, which is divisible by both 4 and 5, we can tile each such segment with tiles of the correct orientation. Therefore, the problem reduces to finding the number of **compositions** of 24 into parts of 4 and 5, where the **order of the segments matters** (since different arrangements correspond to different tilings).	To tile the entire rectangle, we can divide the width (24 units) into segments of either 4 or 5 units. Each such segment can then be filled with tiles of the appropriate orientation. A segment of 4 units wide can be tiled with vertical tiles (5 units tall, 4 units wide). A segment of 5 units wide can be tiled with horizontal tiles (4 units tall, 5 units wide). Since the height of the rectangle is 20, which is divisible by both 4 and 5, we can tile each such segment with tiles of the correct orientation. Therefore, the problem reduces to finding the number of compositions of 24 into parts of 4 and 5, where the order of the segments matters.

Continued on next page

Table 11 (continued)

Original trace

Counting the Number of Valid Compositions
We define a function $C(n)$ as the number of compositions of n into parts of 4 and 5. This satisfies the recurrence:

$$C(n) = C(n-4) + C(n-5)$$

with base cases:
• $C(0) = 1$ (the empty composition),
• $C(n) = 0$ for $n < 0$.
We compute $C(24)$ using this recurrence:
$C(0) = 1$
$C(1) = 0$
$C(2) = 0$
$C(3) = 0$
$C(4) = C(0) = 1$
$C(5) = C(1) + C(0) = 0 + 1 = 1$
$C(6) = C(2) + C(1) = 0 + 0 = 0$
$C(7) = C(3) + C(2) = 0 + 0 = 0$
$C(8) = C(4) + C(3) = 1 + 0 = 1$
$C(9) = C(5) + C(4) = 1 + 1 = 2$
$C(10) = C(6) + C(5) = 0 + 1 = 1$
$C(11) = C(7) + C(6) = 0 + 0 = 0$
$C(12) = C(8) + C(7) = 1 + 0 = 1$
$C(13) = C(9) + C(8) = 2 + 1 = 3$
$C(14) = C(10) + C(9) = 1 + 2 = 3$
$C(15) = C(11) + C(10) = 0 + 1 = 1$
$C(16) = C(12) + C(11) = 1 + 0 = 1$
$C(17) = C(13) + C(12) = 3 + 1 = 4$
$C(18) = C(14) + C(13) = 3 + 3 = 6$
$C(19) = C(15) + C(14) = 1 + 3 = 4$
$C(20) = C(16) + C(15) = 1 + 1 = 2$
$C(21) = C(17) + C(16) = 4 + 1 = 5$
$C(22) = C(18) + C(17) = 6 + 4 = 10$
$C(23) = C(19) + C(18) = 4 + 6 = 10$
$C(24) = C(20) + C(19) = 2 + 4 = 6$
Thus, the number of valid tilings is:

$$\boxed{6}$$

Style-normalized trace
We define a function $C(n)$ as the number of compositions of n into parts of 4 and 5. This satisfies the recurrence:

$$C(n) = C(n-4) + C(n-5)$$

with base cases:
• $C(0) = 1$ (the empty composition),
• $C(n) = 0$ for $n < 0$.

We compute $C(24)$ using this recurrence:
$C(0) = 1$
$C(1) = 0$
$C(2) = 0$
$C(3) = 0$
$C(4) = C(0) = 1$
$C(5) = C(1) + C(0) = 0 + 1 = 1$
$C(6) = C(2) + C(1) = 0 + 0 = 0$
$C(7) = C(3) + C(2) = 0 + 0 = 0$
$C(8) = C(4) + C(3) = 1 + 0 = 1$
$C(9) = C(5) + C(4) = 1 + 1 = 2$
$C(10) = C(6) + C(5) = 0 + 1 = 1$
$C(11) = C(7) + C(6) = 0 + 0 = 0$
$C(12) = C(8) + C(7) = 1 + 0 = 1$
$C(13) = C(9) + C(8) = 2 + 1 = 3$
$C(14) = C(10) + C(9) = 1 + 2 = 3$
$C(15) = C(11) + C(10) = 0 + 1 = 1$
$C(16) = C(12) + C(11) = 1 + 0 = 1$
$C(17) = C(13) + C(12) = 3 + 1 = 4$
$C(18) = C(14) + C(13) = 3 + 3 = 6$
$C(19) = C(15) + C(14) = 1 + 3 = 4$
$C(20) = C(16) + C(15) = 1 + 1 = 2$
$C(21) = C(17) + C(16) = 4 + 1 = 5$
$C(22) = C(18) + C(17) = 6 + 4 = 10$
$C(23) = C(19) + C(18) = 4 + 6 = 10$
$C(24) = C(20) + C(19) = 2 + 4 = 6$
Thus, the number of valid tilings is:

$$\boxed{6}$$

Continued on next page

Table 11 (continued)

Original trace	Style-normalized trace
Conclusion Each valid composition of 24 into parts of 4 and 5 corresponds to a unique way of dividing the rectangle into segments that can be tiled with 4×5 tiles in appropriate orientations. Since the problem considers rotations and reflections as distinct, each composition yields a distinct tiling. **Final Answer** $\boxed{6}$	Each valid composition of 24 into parts of 4 and 5 corresponds to a unique way of dividing the rectangle into segments that can be tiled with 4×5 tiles in appropriate orientations. Since the problem considers rotations and reflections as distinct, each composition yields a distinct tiling. $\boxed{6}$

Table 12: **Complete original and normalized traces for the competing answer $\boxed{1}$.**

Original trace	**Style-normalized trace**
We will show that no "exotic" tiling exists–in fact, there is only one way to cover a 20×24 rectangle by 4×5 rectangles (when rotations and reflections count as different, so that even a "mirrored" copy would be considered distinct–but here there is no choice at all).	We aim to demonstrate that no unconventional tiling is possible–in fact, there is only one way to cover a 20×24 rectangle using 4×5 rectangles (when rotations and reflections count as distinct, so that even a mirrored copy would be considered different–but here there is no choice at all).
Let us explain. A 4×5 rectangle has area 20. Since $$20 \times 24 = 480$$ and $\quad 480 \div 20 = 24,$ any tiling must use exactly 24 tiles.	First, let us explain. A 4×5 rectangle has an area of 20. Given that $20 \times 24 = 480$ and 480 divided by 20 equals 24, any tiling requires exactly 24 tiles.
Moreover, since all tiles are congruent (up to rotation) and their sides must be parallel to the sides of the large rectangle, one may show (by a standard "grid-forcing" argument) that if a rectangle is tiled by congruent (non-square) rectangles then the tiling "extends" to a complete set of horizontal and vertical lines. In other words, every tiling by congruent non-square rectangles is necessarily a grid tiling.	Moreover, since all tiles are congruent (up to rotation) and their sides must align with the sides of the large rectangle, it can be shown (via a standard grid-forcing argument) that tiling a rectangle with congruent non-square rectangles necessitates a grid-like structure. In other words, every tiling by congruent non-square rectangles is necessarily a grid tiling.
Now, note that a 4×5 rectangle has two possible orientations. In one orientation the longer side (of length 5) is vertical and the shorter side (of length 4) is horizontal; in the other the roles are reversed. But look at the dimensions of the big rectangle: • The vertical side is 20 and the horizontal side is 24.	Now, note that a 4×5 rectangle can be oriented in two ways: one with the longer side (length 5) vertical and the shorter side (length 4) horizontal, and the other with these dimensions reversed. But consider the dimensions of the large rectangle: • The vertical side is 20 and the horizontal side is 24.
If we try to fit a 4×5 rectangle with the 5-side vertical, then its horizontal side is 4. Since 24 is divisible by 4 ($24 \div 4 = 6$) but 20 is not divisible by 4 ($20 \div 4 = 5$, and then $5 \times 4 = 20$ would force the other dimension to be 5, not 20), the only possibility is to have every tile placed in the orientation in which the 5-side is vertical and the 4-side is horizontal. In that case the vertical cell has height 5 and the horizontal cell has width 4. Then the tiling is given by • dividing the 20-side into $20 \div 5 = 4$ equal segments, and • dividing the 24-side into $24 \div 4 = 6$ equal segments. This produces a 4×6 grid of 5×4 rectangles.	If we attempt to fit a 4×5 rectangle with the 5-side vertical, its horizontal side would be 4. Since 24 is divisible by 4 ($24 \div 4 = 6$) but 20 is not divisible by 4 ($20 \div 4 = 5$, and $5 \times 4 = 20$ would force the other dimension to be 5, not 20), the only viable orientation is one in which the 5-side is vertical and the 4-side is horizontal. In this case, the vertical cell has height 5 and the horizontal cell has width 4. Then the tiling is achieved by • dividing the 20-side into $20 \div 5 = 4$ equal segments, and • dividing the 24-side into $24 \div 4 = 6$ equal segments. This produces a 4×6 grid of 5×4 rectangles.
One may check that no "staggered" or "non-grid" tiling is possible (in fact, one may prove that any tiling of a rectangle by congruent non-square rectangles must be a grid tiling). Also, if one were to try to use the other orientation–with the 4-side vertical–then the horizontal side would be 5, but 24 is not divisible by 5.	One may verify that no staggered or non-grid tiling is feasible (in fact, one may prove that any tiling of a rectangle by congruent non-square rectangles must be a grid tiling). Additionally, if one were to try the other orientation–with the 4-side vertical–the horizontal side would be 5, but 24 is not divisible by 5.

Continued on next page

Table 12 (continued)

Original trace	**Style-normalized trace**
Thus the tiling is unique. Since rotations and reflections are considered distinct in the statement, even if one were to flip the entire tiling, one would still have the same grid; there is no alternative. Therefore, the number of ways to divide a 20×24 rectangle into 4×5 rectangles is $\boxed{1}$.	Thus the tiling is unique. Since rotations and reflections are considered distinct in the statement, even if one were to flip the entire tiling, the grid remains unchanged; there is no alternative. Therefore, the number of ways to divide a 20×24 rectangle into 4×5 rectangles is $\boxed{1}$.

E PROMPTS

We provide the prompts used for style normalization and final-answer decision. Figure 11 shows the style-normalization prompt, and Figure 12 illustrates the decision prompt before and after adverse-frequency bias injection. The bias injection stage uses the same decision instruction as the bias-controlled stage; only the candidate presentation is changed.

```
Target style: a single uniform plain-prose voice.
  - No markdown headers (no '#', no '##').
  - No bullet lists; write in continuous sentences.
  - LaTeX only for inline equations and any \boxed{...}; do NOT decorate prose.
  - Short paragraphs; transitions like "First,", "Next,", "Therefore,".
  - Preserve any \boxed{...} exactly.

LENGTH (hard): the passage is {n} chars; keep your rewrite within {0.85n}-{1.15n} chars.

=== Rewrite the following passage in the target style ===
Rewrite ONLY the surface wording, phrasing, and formatting.
Keep ALL content exactly as it is:
every statement, number, equation, and \boxed{...} stays unchanged in meaning and value.
Do NOT add anything that is not already there -- no commentary, no confirmations or checks
("this is correct", "this checks out", "consistent with", "indeed", ...), no corrections, no flags.
Do NOT remove or reorder content. If something looks off, leave it exactly as written.
You are only changing how it reads, not what it says.

Preserve any final \boxed{...} exactly (character-for-character). Output ONLY the rewritten
     passage, nothing else.

## Passage ({n} chars)
{trace}

## Your rewrite (target: {0.85n}-{1.15n} chars)
```

Figure 11: **Style-normalization prompt.** The prompt instructs the rewriter to change only wording and formatting while preserving reasoning content and final answers; a length constraint limits deviation from the original trace.

```
You are given several candidate solutions (anonymized) to the same problem. Decide which one's
     final answer is correct. If you believe none of them is correct, answer none.

Problem:
Eight circles of radius $34$ are sequentially tangent, and two of the circles are tangent to $AB$
     and $BC$ of triangle $ABC$, respectively. $2024$ circles of radius $1$ can be arranged in the
     same manner. The inradius of triangle $ABC$ can be expressed as $\frac{m}{n}$, where $m$ and
     $n$ are relatively prime positive integers. Find $m+n$.

--- Solution 0 (answer: 2297) ---
[FULL NORMALIZED REASONING TRACE FOR 2297]

--- Solution 1 (answer: 254) ---
[FULL NORMALIZED REASONING TRACE FOR 254]

--- Solution 2 (answer: 197) ---
[FULL NORMALIZED REASONING TRACE #1 FOR 197]

--- Solution 3 (answer: 8503) ---
[FULL NORMALIZED REASONING TRACE FOR 8503]

Reason step by step about which final answer is correct, then put the chosen final answer within
     \boxed{}, or \boxed{none} if none of the candidates' answers is correct.
```

(a) Prompt for the initial bias-controlled decision.

```
You are given several candidate solutions (anonymized) to the same problem. Decide which one's
     final answer is correct. If you believe none of them is correct, answer none.

Problem:
Eight circles of radius $34$ are sequentially tangent, and two of the circles are tangent to $AB$
     and $BC$ of triangle $ABC$, respectively. $2024$ circles of radius $1$ can be arranged in the
     same manner. The inradius of triangle $ABC$ can be expressed as $\frac{m}{n}$, where $m$ and
     $n$ are relatively prime positive integers. Find $m+n$.

--- Solution 0 (answer: 2297) ---
[FULL NORMALIZED REASONING TRACE FOR 2297]

--- Solution 1 (answer: 254) ---
[FULL NORMALIZED REASONING TRACE FOR 254]

--- Solution 2 (answer:  197) ---
[FULL NORMALIZED REASONING TRACE #1 FOR 197]

--- Solution 3 (answer: 8503) ---
[FULL NORMALIZED REASONING TRACE FOR 8503]

--- Solution 4 (answer:  197) ---
[FULL NORMALIZED REASONING TRACE #2 FOR 197]

Reason step by step about which final answer is correct, then put the chosen final answer within
     \boxed{}, or \boxed{none} if none of the candidates' answers is correct.
```

(b) Prompt for the second-pass decision under adverse-frequency bias injection.

Figure 12: **Decision prompts before and after adverse-frequency bias injection.** (a) Bias-controlled presentation with one trace per answer hypothesis. (b) After the controlled decision selects an answer other than 197, an additional existing trace makes 197 the higher-frequency competitor without changing the problem, answer hypotheses, or decision instruction.