# When Wrong Answers Look Right: Multi-Agent Debate for High-Precision Verification of Hard Competition Mathematics

**Anonymous Authors**[1]

## Abstract

High-quality mathematical reasoning data are central to both training and evaluation, but such data are only useful when their answers and reasoning traces are reliable. Candidate solutions may be web-scraped, human-written, adapted from existing competition problems, or generated by LLMs; in all cases, a fluent but subtly wrong solution can silently contaminate downstream use. We present a high precision multi-agent debate pipeline for blind verification of competition-level mathematics, in which five heterogeneous agents independently analyze a candidate answer, exchange structured arguments for up to five rounds, and reach a verdict governed by an assessment-gating criterion that requires positive confirming evidence rather than mere non-refutation. On 195 hard-to-verify variants from 58 IMO/USAMO/Putnam-level problems, a single-judge baseline achieves 55.1% precision; our pipeline achieves **92.5%** precision—an $8.3\times$ false-positive reduction—with 59.8% problem-level accuracy. A $2\times2$ factorial ablation shows that, after accounting for repeated verification, most of the remaining false-positive reduction comes from two architectural mechanisms: *assessment gating* and *heterogeneous agent roles*. These results establish precision-first debate as a practical filtering primitive for mathematical reasoning data, with applications to benchmark curation, reward labeling, RLAIF, and self-improving synthetic-data pipelines. All code and data are publicly available at: https://anonymous.4open.science/r/When-Wrong-Answers-Look-Right-5A57.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

## 1. Introduction

The bottleneck in frontier mathematical AI is no longer only problem solving or data generation—it is *verification*. High-quality mathematical reasoning data are central to both training and evaluation, but such data are only useful when their answers and reasoning traces are reliable. Candidate solutions may come from web-scraped explanations, human-written derivations, adapted competition problems, or LLM-generated attempts; in all cases, a fluent but subtly wrong solution can silently contaminate downstream use.

This makes verification a data filtering problem. For benchmark construction, training-data curation, reward labeling, and RLAIF, false positives are more damaging than false negatives: accepting an incorrect trace introduces a mislabeled example, whereas rejecting a correct trace merely reduces yield. This asymmetry motivates a *precision-first* objective: the verifier should accept only when it has positive evidence that the candidate answer is correct.

We study blind answer verification for competition-level mathematics: given a problem, a candidate answer, and a solution trace, decide whether the candidate is safe to accept without access to a reference answer. A single LLM judge fails on this task for a structural reason: a model trained to recognize plausible mathematical arguments is susceptible to the same surface-coherence cues that make wrong solutions look convincing. On our hard-to-verify variants, a single judge achieves only 55.1% precision, showing that fluent mathematical reasoning is not sufficient evidence of correctness.

**Design questions.** Our study is organized around three design questions, each corresponding to a failure mode in high-precision mathematical data filtering.

**Q1** (*Why verification?*) Can checking a candidate derivation provide a useful signal beyond solving from scratch, especially in the hard competition-math regime where direct solving is unreliable? If verification only replicated the difficulty of solving, it would be a weak primitive for data filtering. However, a verifier can exploit the submitted reasoning trace: it can inspect local algebraic steps, theorem applications, boundary cases, and arithmetic consistency without

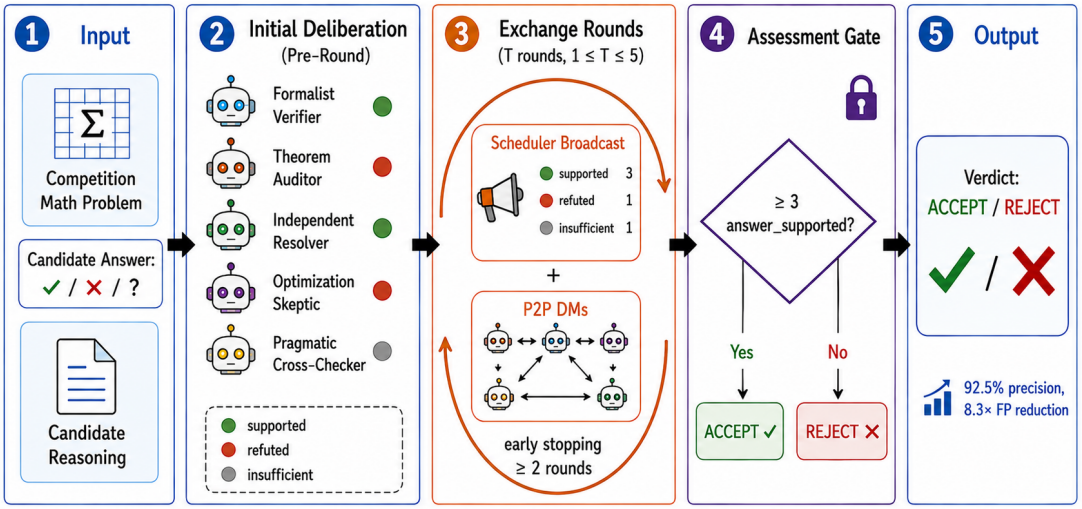


*Figure 1.* **Precision-first multi-agent debate pipeline.** Five heterogeneous agents **(2)** independently deliberate on a candidate answer, then exchange arguments for up to five rounds via scheduler broadcasts and peer-to-peer direct messages **(3)**. A verdict is accepted only when the **assessment gate (4)** is satisfied: $\geq 3$ agents hold `answer_supported`—positive confirming evidence, not mere non-refutation—reducing false positives $8.3\times$ over a single-judge baseline.

reconstructing the entire solution. We therefore compare solving and verification as a diagnostic task structure comparison, while avoiding a claim of a fully compute-scaling law.

**Q2** (*Why debate?*) Does structured exchange in the debate process reduce false positives beyond single-pass judging or repeated independent verification? Single judges are vulnerable to fluent but wrong reasoning, and repeated calls can preserve correlated blind spots when each call evaluates the same trace in isolation. Debate is intended to expose these hidden errors by forcing agents to respond to specific counterarguments, perform new checks, and revise their stance only when a concrete failure or confirmation is found.

**Q3** (*Why heterogeneity and gating?*) Do heterogeneous roles and an assessment gate reduce correlated errors beyond homogeneous debate and flat support/oppose voting? Homogeneous agents may inspect the same parts of a derivation and miss the same flaw, while flat voting can confuse "no error found" with positive evidence of correctness. Our design addresses both issues: heterogeneous agents examine complementary failure modes, and the assessment gate accepts a candidate only when a majority of agents produce `answer_supported`, i.e., positive confirming evidence.

We answer Q2–Q3 through controlled baselines and a $2\times2$ factorial ablation, and report Q1 as an empirical task-structure comparison rather than a fully compute-matched scaling study.

Our pipeline (Figure 1, Section 4) achieves **92.5%** precision, reducing false positives by $8.3\times$ over a single-judge baseline. The gain is not merely a consequence of repeated calls: 59% of the FP reduction is attributable to debate architecture, with assessment gating and role heterogeneity providing independent contributions. These results support precision-first multi-agent debate as a filtering primitive for mathematical reasoning data, including benchmark curation, reward labeling, and LLM-generated data pipelines.

## 2. Related Work

**LLM-as-judge and Agent-as-a-Judge.** LLM-as-a-Judge has become a standard paradigm for scalable evaluation of open-ended model outputs. Zheng et al. (2023) showed that strong LLM judges can approximate human preferences on open-ended chat evaluation, while also documenting important biases and limitations. Follow-up work such as G-Eval (Liu et al., 2023), Prometheus (Kim et al., 2024), and JudgeLM (Zhu et al., 2025) improve judge alignment through chain-of-thought evaluation, fine-grained rubrics, or judge-model tuning. However, these systems still primarily rely on single-pass or monolithic judgment, whereas our setting stresses a different failure mode: competition mathematics requires detecting subtle logical or arithmetic errors rather than judging surface quality.

A recent survey argues that evaluation may be shifting from LLM-as-a-Judge toward Agent-as-a-Judge (You et al., 2026), where evaluators use planning, tool-augmented verification, multi-agent collaboration, and memory to support more robust assessments. Our pipeline is an instance of this broader agentic-evaluation direction in a precision-first mathematical setting: rather than asking one judge whether a solution looks correct, it requires multiple heterogeneous agents to produce positive confirming evidence through

structured disagreement. Beyond generic judge bias, Li et al. (2026a) identify *preference leakage*: LLM judges can be biased when the data-generating model and the judging model are related. This motivates adversarially plausible wrong variants, where fluent reasoning can fool single-pass judges. A recent technical report on LLM-as-a-Verifier (Kwok et al., 2026) scales verifier quality through repeated verification, scoring granularity, and criteria decomposition. Related process-supervision work also suggests that verifying intermediate reasoning can provide a more precise signal than judging only the final outcome (Lightman et al., 2024). Our approach is complementary: instead of deepening one evaluator, we broaden the set of analytical perspectives through heterogeneous agents and structured disagreement.

**Multi-agent debate for reasoning and evaluation.** Multi-agent collaboration has been widely explored as a way to reduce single-judge instability and expose hidden errors. Du et al. (2024) showed that multi-agent debate improves factuality in open-domain question answering, while Liang et al. (2024) demonstrated that encouraging divergent thinking in debate reduces sycophancy. In evaluation settings, ChatEval (Chan et al., 2024) and MATEval (Li et al., 2024) use multi-agent discussion to improve open-ended text assessment. However, Choi et al. (2025) proved analytically that for *homogeneous* agents under *flat majority voting*, debate exchange yields no expected improvement. Our ablation identifies two mechanisms that move the setting beyond homogeneous flat voting: *assessment gating*, which separates positive evidence from uncertainty, and *role heterogeneity*, which encourages agents to examine different failure modes (Section 5.4).

**Tool-augmented verification and reward signals.** VerifiAgent (Han et al., 2025) proposes a unified, training-free verification agent that combines meta-verification with adaptive tool-based verification across mathematical, logical, commonsense, and hybrid reasoning tasks. HERMES (Ospanov et al., 2025) takes a more formal route for mathematical reasoning by interleaving informal LLM reasoning with Lean-based proof checking. Agentic Reward Modeling (Peng et al., 2025) further shows that reward models can be strengthened by incorporating verifiable correctness signals rather than relying only on preference-based supervision. Our setting is complementary: we study blind, precision-first verification of competition-level mathematical answers without external tools. Tool augmentation is a natural future direction for reducing the remaining false positives and false negatives.

**Test-time compute scaling.** Self-consistency (Wang et al., 2023) improves answer accuracy by majority-voting over independent completions. Snell et al. (2024) showed optimal test-time compute allocation can outperform model scaling. Zhang et al. (2025) showed reasoning paths carry richer logical structure than final-answer consistency captures, and that graph-based aggregation outperforms flat majority voting. Our pipeline addresses the homogeneous re-sampling bottleneck differently: heterogeneous agents explore structurally distinct verification paths, producing approximately independent error signals.

**Mathematical data curation and benchmarks.** FrontierMath (Glazer et al., 2024) and Humanity's Last Exam (Center for AI Safety et al., 2026) illustrate the growing demand for difficult, high-quality evaluation data. AMO-Bench (An et al., 2025) provides recent competition-style mathematical problems, which we use as one source for variant construction. VeRA (Cheng et al., 2026) studies verified reasoning data augmentation through executable specifications, representing one route to high-quality synthetic reasoning data. Our setting is complementary: we study source-agnostic blind verification of candidate answers and solution traces, without assuming an executable specification, a trusted generation process, or access to ground-truth answers.

The combination of **Olympiad-level** competition mathematics (IMO/USAMO/Putnam), blind verification, a precision-first objective, tool-free heterogeneous debate, and systematic factorial attribution is not matched by any prior system (Appendix A, Table 2).

## 3. Task and Dataset

### 3.1. Adversarial Verification Task

The task is: given a competition-level mathematics problem, a candidate numerical answer, and a full solution trace, output CORRECT or WRONG. Unlike simple answer matching—where verification reduces to checking whether the candidate equals a stored ground truth—the pipeline has no access to the reference answer and must determine correctness through mathematical analysis alone.

The task is source-agnostic: the candidate answer and reasoning trace may be human-written, web-scraped, adapted from existing problems, or LLM-generated. Our benchmark uses plausible LLM-style wrong reasoning as a stress test because such errors are fluent, surface-coherent, and difficult for simple judges to reject. The verifier's deployment goal is not to identify the data source, but to decide whether a candidate is safe to accept into a training, evaluation, or reward labeling pool.

**Metrics.** Our primary metric for data filtering is **precision**: the fraction of accepted variants that are truly correct. A false positive silently admits an incorrect example into a benchmark, training set, or reward-labeling pool; a false negative merely reduces yield. *Problem-level accuracy* is

reported alongside precision as a secondary metric: a problem is correctly handled only when the pipeline (a) *accepts* the correct variant *and* (b) *rejects all* wrong variants in a run. This avoids inflating credit for high-recall systems that also accept wrong variants, exposing contamination that variant-level precision can hide. Recall and false-positive count complete the error profile.

### 3.2. Dataset

**Dataset design principles.** Our goal is not to benchmark problem generation. Instead, we construct a stress test for verification: each item contains a candidate answer and a plausible reasoning trace, and the verifier must decide whether it is safe to accept into a training or evaluation pool. The dataset is designed around three principles: (1) hard mathematical content, (2) reliable labels, and (3) plausible wrong reasoning. We evaluate on **195 hard-to-verify variants** derived from **58 IMO/USAMO/Putnam-level problems** spanning combinatorics, number theory, geometry, and algebra. Problems are drawn from three sources: (i) parameter-perturbed instances from AMO-Bench (An et al., 2025), accepted only after two guided solvers agreed and one unguided solver failed—validating correctness and non-trivial difficulty; (ii) parameter-perturbed instances from IMO and USAMO competition archives; (iii) expert-contributed problems with hand-verified answers. Each problem yields 3–4 variants, exactly one labeled `correct` and the rest `wrong`.[1]

**Variant construction.** Regardless of problem source, each variant comprises four independently constructed components; their construction procedures are as follows:

**Correct answers** are established via *three-model triangulation*: `gpt-5.2` with reference solution, `claude-opus-4-5` with reference solution, and `gpt-5.2` without any reference. Here "reference solution" means the original competition problem's official answer and derivation, provided so that the model can validate the *parameter-perturbed* variant (rather than the original problem). A candidate is accepted only when the two reference-equipped models agree *and* the zero-shot model fails—ensuring correctness via cross-provider consensus and non-trivial difficulty via zero-shot failure. All 58 correct answers were manually reviewed.

**Wrong answers** come primarily from *natural LLM failures*: the model solves each problem from scratch; genuinely wrong outputs (misapplied theorem, flawed counting argument, incorrect boundary case) are retained together with the model's own reasoning, preserving the fluency and surface coherence that makes them hard to detect. Supplementary strategies broaden coverage: (i) *parameter reuse* (numerically plausible alternatives from neighboring problem instances) and (ii) *targeted error injection* (a small minority covering error categories underrepresented in natural failures). All wrong answers pass *plausibility filtering*: a simple judge must fail to immediately identify the variant as incorrect. Using LLM failures does not restrict the deployment setting to LLM-generated data; rather, it provides a challenging proxy for plausible-but-wrong reasoning that can arise in any large-scale data curation pipeline. **Correct reasoning** comes from the reference-equipped model; for 7 hard problems, human-guided iterative solving (intermediate hints, certificate searches) produced verified traces. **Wrong reasoning** is the model's own failed-attempt output, with an error embedded mid-step while the conclusion remains end-to-end plausible.

**Quality guarantee.** Three independent gates act in sequence: (1) *correctness* via multi-provider consensus on the accepted answer, (2) *difficulty* via zero-shot solve failure, and (3) *plausibility* via simple-judge filtering. Gate (3) is typically satisfied without special effort: a model that cannot solve the problem tends to produce wrong answers that superficially resemble correct ones, precisely because it follows the same solution template. Wrong variants are plausible rather than random distractors, forcing verifiers to locate mathematical failures rather than rely on stylistic cues.

**Task difficulty.** `gpt-5.2 (effort="low")` solves problems from scratch at 33.3%, 44.9%, and 39.2% across three independent runs (mean ≈39%), confirming that the dataset is difficult for direct solving. We also compare majority-vote solving with majority-vote verification in Section 5.3. Because solving produces one answer per problem whereas verification evaluates candidate variants individually, this comparison should be interpreted as a task-structure comparison rather than a complete compute-scaling study. All systems use the same base model and effort setting.

## 4. Pipeline Architecture

### 4.1. Agent Composition

The pipeline employs **five heterogeneous agents**, each with a distinct verification persona:

1. **Formalist Verifier** — algebraic consistency, arithmetic, bound confusion, equality-case handling.
2. **Theorem Auditor** — applicability of cited theorems, correct use of lemmas.
3. **Independent Resolver** — attempts a short independent re-derivation to cross-check the stated answer.

[1] 37 problems contribute 3 variants; 21 contribute 4: $37\times3 + 21\times4 = 195$.

4. **Optimization Skeptic** — boundary conditions, constraint violations, optimality-vs-feasibility gaps.

5. **Pragmatic Cross-Checker** — numerical spot checks, small-case sanity tests, symbolic verification.

All five agents run the same base model (`gpt-5.2`) with different system instructions. Heterogeneity is critical: debate improves factuality when agents bring diverse perspectives (Du et al., 2024), but homogeneous agents are liable to reinforce shared errors through correlated sampling. This design operationalizes the DynaDebate principle (Li et al., 2026b): a diverse population of initial stances yields richer information exchange than a homogeneous one.

### 4.2. Initial Deliberation (Pre-Round)

Before any exchange, each agent independently executes five steps: **(A)** restate the claim precisely; **(B)** enumerate 2–4 domain-specific failure modes and check each against the submitted reasoning—*priming adversarial scrutiny* before any positive evidence is weighed; **(C)** perform one independent check (spot check, arithmetic, bound argument, theorem applicability); failure forces `evidence_grade=weak`; **(D)** steelman the opposition, classifying the strongest counterargument as **FATAL / UNCERTAIN / REFUTED**—only FATAL may flip a verdict, preventing cascading stance changes from vague concern; and **(E)** commit to structured JSON output.

The `assessment_type` field distinguishes three epistemic states: `answer_supported` (positive evidence confirms correctness), `answer_refuted` (positive evidence confirms incorrectness), and `reasoning_insufficient` (uncertainty without a specific flaw; the full field value includes the suffix `_but_answer_not_refuted`). This three-way split is the basis for the assessment-gating verdict rule (Section 4.4).

### 4.3. Exchange Rounds

After the initial deliberation, the pipeline runs up to $R{=}5$ exchange rounds with early stopping permitted only after at least two full rounds, preventing premature consensus. In each round, every agent receives:

**(1) Scheduler broadcast.** The current vote tally, `assessment_type` breakdown, strongest support and oppose claims from the previous round (ranked by confidence $\times$ evidence grade), and a *disagreement digest*—a structured enumeration of the top unresolved opposing claims that every agent is explicitly required to address. The disagreement digest is the most impactful component: without it, agents tend to restate prior positions rather than engaging specific counter-arguments.

**(2) Direct messages (DMs).** Each agent's inbox is assembled by two independent mechanisms: (a) *Scheduler-generated challenges*: the highest-priority support and oppose agents each receive a message quoting the other's central claim; weak supporters are pressured to produce a concrete check or switch stance; undecided agents are instructed to compare competing claims and state which they can verify. All challenge text is derived programmatically from the previous round's `swing_issue` and `summary` fields—no LLM call is invoked by the scheduler. (b) *Peer-to-peer DMs*: each agent may include up to two targeted messages to named recipients in its `outbound_dm` field, enabling focused bilateral exchanges on specific disputed claims.

Each agent then follows a four-step in-round protocol: **Step 1** (address open disagreements), **Step 2** (perform one new independent check), **Step 3** (steelman the other side, same FATAL/UNCERTAIN/REFUTED classification), **Step 4** (commit; if changing stance, record the triggering agent and reason—silent flips are prohibited).

**Summary propagation.** Agents carry forward only a $\leq$400-character summary of their prior rationale rather than the full chain. Propagating full reasoning chains causes context length to grow linearly with rounds, increases attention dilution, and causes agents to anchor on verbose history rather than engaging new evidence (Yang et al., 2026). The compressed summary retains the key claim and single most important check.

### 4.4. Verdict Rules

After the final round, the verdict is accepted if and only if the **assessment gate** is satisfied: $\geq$3 of 5 agents carry `assessment_type = answer_supported`—meaning they independently obtained *positive confirming evidence*, not merely failed to find a flaw. Since `answer_supported` implies `support`, the assessment gate subsumes the supermajority criterion on the acceptance path; a separate supermajority pre-check is applied only as a fast-fail guard against malformed JSON output where the `assessment_type` field is missing or invalid. This precision-first filter ensures that acceptance requires a clear majority of agents to have found definitive positive evidence through independent checking—not mere absence of refutation.

## 5. Experiments

### 5.1. Experimental Setup

All systems use `gpt-5.2` (`effort="low"`) on the 195-variant, 58-problem dataset; three independent runs per main system; ablations are single-run. Seven systems form

*Table 1.* Results on 195 competition-mathematics variants from 58 problems. Positive class = `correct`; all systems use `gpt-5.2` (`effort="low"`). †3-run average; ‡single run.

| System | Prec. | Recall | FP | Prob. Acc. |
|---|---|---|---|---|
| Single-Judge† | 55.1% | 52.3% | 24.7 | 33.3% |
| MajVote-Solve† | 61.8% | 42.5% | 15.3 | 42.5% |
| MajVote-Verify† | 75.1% | 81.6% | 15.7 | 60.9% |
| MajVote-Precise‡ | 75.9% | 70.7% | 13.0 | 56.9% |
| Debate-NoGate‡ | 84.8% | 67.2% | 7.0 | 60.3% |
| Debate-Uniform‡ | 88.1% | 63.8% | 5.0 | 58.6% |
| **Debate (Ours)†** | **92.5%** | 63.8% | **3.0** | 59.8% |

a controlled ladder (Table 1): **Single-Judge** (1 call/variant); **MajVote-Solve** ($N$=11 solve attempts/problem, majority vote answers each variant); **MajVote-Verify** ($N$=11 calls/variant, threshold $\lceil 0.6N \rceil$=7);[2] **MajVote-Precise** (same as MajVote-Verify, precision-first prompt, no exchange—isolates prompt quality); **Debate-NoGate** (full debate, stance-only verdict, no assessment gate); **Debate-Uniform** (full debate + gate, five identical agents); **Debate (Ours)** (full pipeline: heterogeneous personas, exchange, assessment gate).

### 5.2. Main Results

Table 1 reports all seven systems; Figure 2 shows the precision–recall landscape. The Debate pipeline reduces FP from 24.7 to 3.0—an 8.3× reduction—while raising precision from 55.1% to 92.5%. Run variance is low (precision range: 90.2–95.0%), confirming stable verdicts. MajVote-Precise gains only 0.8 pp in precision and 2.7 fewer FP over MajVote-Verify; the majority of gains require the debate structure.

**Finding 1** (Precision Ceiling): Single-judge evaluation achieves only 55.1% precision—near chance—and repeated calls alone cannot close this gap. **Finding 2** (Architecture > Compute): 59% of FP reduction is attributable to debate architecture, not additional inference budget. **Finding 3** (Verification as useful filtering signal): Majority-vote verification accepts correct variants much more often than majority-vote solving in our diagnostic comparison, suggesting that checking a given reasoning trace can provide useful signal when solving from scratch is unreliable.

**Problem-level accuracy and benchmark quality.** Debate and MajVote-Verify achieve similar problem-level accuracy (59.8% vs. 60.9%), but their error profiles differ critically. MajVote-Verify accepts the true answer for ≈47 of 58 problems (recall 81.6%), yet its 15.7 FP/run contaminate many additional problems where wrong variants are *also* accepted—making them unusable despite the correct label being present. Debate accepts the true answer for ≈37 problems (recall 63.8%) with only 3.0 FP/run, contaminating ≈3 problems. For benchmark construction, where any FP on a problem makes the entire entry unusable, this difference is decisive.

[2]MajVote-Solve and MajVote-Verify allocate calls differently: solving produces one answer per problem, whereas verification evaluates candidate variants individually. We therefore treat the solve-vs-verify comparison as diagnostic rather than a fully compute-matched scaling study (Section 5.3).

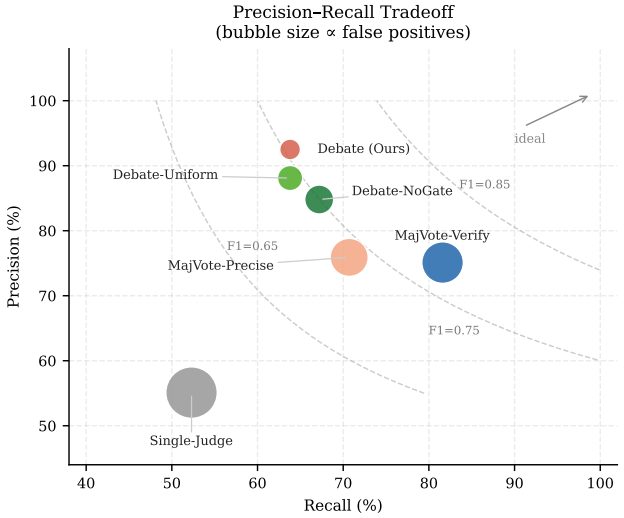


*Figure 2.* Precision–recall tradeoff across six plotted pipeline variants. Bubble size proportional to FP/run; dashed grey lines are iso-F1 curves. Debate (red, top-right) dominates all other plotted systems in precision.

### 5.3. Verify vs. Solve: A Diagnostic Task-Structure Comparison

We include a solve-vs.-verify comparison to test whether a candidate reasoning trace provides useful signal beyond solving from scratch. MajVote-Solve uses $N$=11 independent solve attempts per *problem*, majority-votes for an answer, and labels each variant as correct if and only if the variant's candidate answer matches the majority answer. MajVote-Verify uses $N$=11 calls per *variant*, directly examining each candidate answer against its reasoning trace.

This comparison is intentionally diagnostic rather than a full compute-scaling study. The two procedures allocate computation differently: solving produces one answer for the whole problem, while verification evaluates each candidate variant separately. Thus the problem-level comparison mixes task structure and call allocation. Nevertheless, the variant-level behavior is informative. MajVote-Verify achieves higher precision (75.1% vs. 61.8%) and much higher recall (81.6% vs. 42.5%) than MajVote-Solve, while producing nearly the same number of false positives (15.7 vs.

15.3). The main gap is therefore recall: verification is more willing to accept the true variant when it is accompanied by a reasoning trace.

We interpret this as evidence that checking a proposed derivation can be easier than reconstructing an answer from scratch in this setting. A verifier can inspect local steps, theorem applications, boundary cases, and arithmetic consistency without reproducing the entire solution. However, a fully compute-matched solve-vs.-verify scaling study—for example, equalizing total calls across problem sizes and variant counts—remains future work. (Figure 4, Appendix C).

### 5.4. Ablation Study: Decomposing the FP Reduction

The 21.7-unit total FP reduction has two sources: additional compute and architectural design. MajVote-Verify, a repeated-verification baseline without debate structure, accounts for $-9.0$ FP (41%). The remaining **59% is architectural** ($-12.7$ FP), split between prompt quality ($-2.7$ FP) and the structural core of the debate design: assessment gating and role heterogeneity.

**Factorial design for the structural core.** To isolate the contributions of assessment gating and role heterogeneity without path-order dependence, we use all four cells of a $2\times2$ factorial design (Table 1, Figure 3):

| | No Heterogeneity | Heterogeneity |
|---|---|---|
| No Gating | MajVote-Precise (13.0) | Debate-NoGate (7.0) |
| Gating | Debate-Uniform (5.0) | **Debate** (3.0) |

A sequential ablation is path-dependent: going through Debate-Uniform first attributes $-8.0$ FP to gating and $-2.0$ FP to heterogeneity, whereas going through Debate-NoGate first attributes $-6.0$ FP to heterogeneity and $-4.0$ FP to gating. We therefore report **Shapley-fair values** (average over both orderings), which uniquely satisfy additivity and symmetry: **assessment gating** $-6.0$ **FP (60%)** and **role heterogeneity** $-4.0$ **FP (40%)**. Each mechanism exhibits diminishing marginal returns in the presence of the other: gating contributes $-8.0$ FP without heterogeneity but only $-4.0$ FP with it already present, and symmetrically for heterogeneity ($-6.0$ vs. $-2.0$ FP), reflecting that each one reduces the marginal room left for the other.

**Assessment gating moves beyond flat voting even without heterogeneity** (MajVote-Precise→Debate-Uniform, $-8.0$ FP without heterogeneity; $-4.0$ FP with heterogeneity already present; Shapley $-6.0$ FP). These systems share the same homogeneous agents, the same prompt, and the same call count; only the verdict rule differs. Choi et al.'s analysis focuses on homogeneous agents and flat voting; our setting changes the aggregation rule by distinguishing `answer_supported` from `reasoning_insufficient`. (Appendix D).

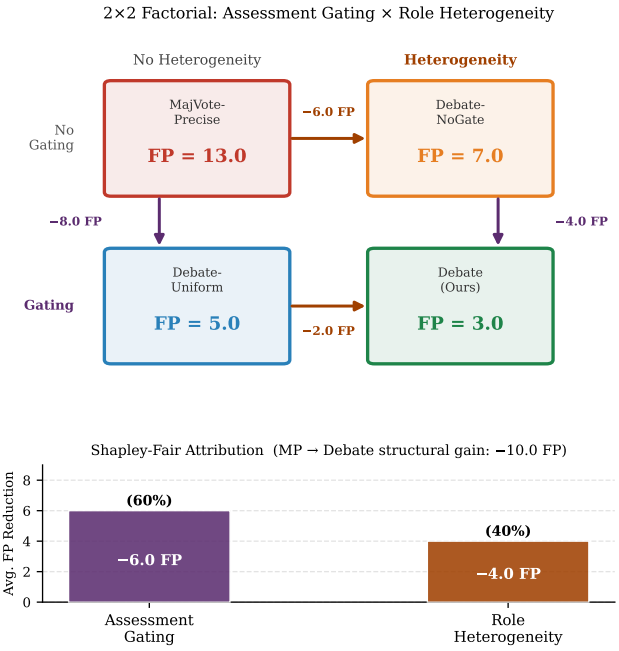


*Figure 3. Top*: $2\times2$ factorial design; each cell shows FP/run. Arrows: dark violet = gating ($-8.0$ or $-4.0$ FP), burnt sienna = heterogeneity ($-6.0$ or $-2.0$ FP). *Bottom*: Shapley-fair attribution—gating 60%, heterogeneity 40% of the $-10.0$ FP structural gain.

**Role heterogeneity further reduces correlated errors** ($-6.0$ FP without gating; $-2.0$ FP with gating present; Shapley $-4.0$ FP). Five role-differentiated personas bring genuinely distinct information: an algebraic-consistency lens, a theorem-applicability check, an independent re-derivation, a boundary-condition probe, and a numerical spot-check. Two agents with different lenses who independently agree are less likely to share a correlated error than two agents with the same lens, making heterogeneous consensus strictly more informative than homogeneous consensus and introducing a second route by which debate exchange generates signal beyond flat voting.

**Interpretation.** The factorial pattern suggests that gating and heterogeneity address different failure modes. Gating primarily prevents weak support from becoming acceptance: agents may fail to find an error, yet still lack positive evidence that the submitted answer is correct. Heterogeneity instead reduces correlated misses by forcing the same derivation through different verification lenses. The diminishing returns in the factorial design are therefore expected: once one mechanism removes many easy false positives, fewer cases remain for the other mechanism to fix.

## 6. Error Analysis

**False positives (3.0/run).** The residual FPs share a consistent structure: the error is numerically subtle (an off-by-one, a modular slip), embedded within an otherwise valid template, with a final answer that passes all sanity checks. No single agent's spot-check covers the flawed sub-step; all five produce positive evidence for the steps they can verify. The $8.3\times$ FP reduction stalls at answers genuinely indistinguishable from correct ones without symbolic computation; resolution requires tool-augmented verification (Han et al., 2025) or cross-provider ensemble diversity.

**False negatives (21.0/run, recall 63.8%).** Three failure modes: **capability saturation** (≈70%): all agents reach `reasoning_insufficient`, unresolvable without model improvement; **active false refutation** (≈25%): at least one agent mis-commits to `answer_refuted`; **mixed evidence** (≈5%): a support/uncertainty split fails to clear the gate. The latter two are addressable via stronger steelman enforcement or tool augmentation (Han et al., 2025).

Convergence statistics appear in Appendix C.4; per-problem accuracy breakdown in Appendix E; and a case study in Appendix F.2.

## 7. Discussion

**Broader use as a data-quality primitive.** Precision-first verification is useful beyond synthetic benchmark construction. This is consistent with recent Agent-as-a-Judge discussions, which argue that evaluation may move from single-pass scoring toward agentic workflows combining decomposition, collaboration, and verification (You et al., 2026). Any pipeline that consumes mathematical reasoning data—web-scraped solutions, human-written explanations, adapted competition problems, LLM-generated variants, benchmark items, or reward-labeling traces—needs to reject plausible but incorrect reasoning before the data are used for training or evaluation. Our verifier is therefore best viewed as a high-precision filtering primitive: it trades yield for label quality, which is the appropriate tradeoff when false positives silently contaminate downstream systems.

**Self-improving AI as one deployment path.** One concrete deployment is the Generate → Verify → Filter → Train loop. LLMs can generate large pools of candidate problems and solution traces, but only a high-precision verifier can decide which candidates are safe to retain. In this setting, our system supplies the verification/filtering stage rather than the generator itself. This is particularly relevant to AI4Math-style self-evolving mathematical agents, where progress depends not only on generating new problems or solutions, but also on reliably validating them.

**Limitations and future work.** Although the proposed verifier substantially reduces false positives, several limitations remain. First, recall at 63.8% still limits yield. Most false negatives arise from capability saturation: agents often recognize that a derivation is hard but fail to obtain sufficient positive evidence to pass the assessment gate. Mitigations include higher-effort inference, cross-provider ensemble diversity, and tool-augmented verification (Han et al., 2025).

Second, our solve-vs.-verify comparison is diagnostic rather than a complete compute-scaling study. Solving and verification allocate computation differently: solving produces one answer per problem, while verification evaluates candidate variants individually. A fully compute-matched study that equalizes total calls across problem sizes, variant counts, and inference budgets remains future work.

Third, our evaluation is limited to a relatively small dataset of 195 variants from 58 competition-mathematics problems. Future work should scale to larger and more diverse benchmarks to enable stronger significance testing and more fine-grained analysis across problem types, difficulty levels, and error modes.

Finally, the current system is restricted to mathematics and deliberately tool-free. Extending precision-first debate to other verifiable domains, such as physics, chemistry, and engineering, would test whether the approach is a general primitive for scientific verification rather than a domain-specific math pipeline. Future systems could also invoke Python, symbolic algebra systems, theorem provers, web search, or domain-specific tools to ground judgments in external evidence rather than text-only deliberation.

## 8. Conclusion

We present a precision-first multi-agent debate pipeline for blind verification of competition-level mathematics: 92.5% precision and an $8.3\times$ false-positive reduction over a single-judge baseline. The system is designed as a high-precision filtering primitive for mathematical reasoning data, where false positives are more harmful than false negatives. Five heterogeneous verification personas, structured exchange, direct-message challenges, and an assessment gate requiring positive confirming evidence each target a distinct failure mode of single-pass or flat-vote verification. A $2\times2$ factorial ablation shows that assessment gating and role heterogeneity independently reduce false positives, supporting multi-agent debate as a practical mechanism for reliable mathematical data filtering. More broadly, our results suggest that verification can be treated as a source-agnostic data-quality primitive rather than as a task tied to any single data generator.

## Impact Statement

This paper presents work toward automated verification of mathematical reasoning, with applications in training-data filtering, benchmark curation, reward labeling, RLAIF, and self-improving mathematical AI systems. The technology could reduce reliance on human annotation in AI training and evaluation pipelines, but it may also increase dependence on automated evaluation systems whose errors are difficult to audit. The precision-first design is intended to mitigate this risk by favoring rejection over accepting uncertain or weakly supported examples.

## References


An, S., Cai, X., Cao, X., Li, X., Lin, Y., Liu, J., Lv, X., Ma, D., Wang, X., Wang, Z., and Zhou, S. Amo-bench: Large language models still struggle in high school math competitions. *CoRR*, 2025. URL https://doi.org/10.48550/arXiv.2510.26768.

Center for AI Safety, Scale AI, and HLE Contributors Consortium. A benchmark of expert-level academic questions to assess AI capabilities. *Nature*, 649:1139–1146, 2026. doi: 10.1038/s41586-025-09962-4. URL https://arxiv.org/abs/2501.14249.

Chan, C., Chen, W., Su, Y., Yu, J., Xue, W., Zhang, S., Fu, J., and Liu, Z. Chateval: Towards better llm-based evaluators through multi-agent debate. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024. URL https://openreview.net/forum?id=FQepisCUWu.

Cheng, Z., Liu, J., Wu, C., Yao, J., Viswanath, P., Zhang, G., and Huang, W. Vera: Verified reasoning data augmentation at scale. *CoRR*, abs/2602.13217, 2026. URL https://doi.org/10.48550/arXiv.2602.13217.

Choi, H. K., Zhu, X., and Li, Y. Debate or vote: Which yields better decisions in multi-agent large language models? *CoRR*, abs/2508.17536, 2025. URL https://doi.org/10.48550/arXiv.2508.17536.

Du, Y., Li, S., Torralba, A., Tenenbaum, J. B., and Mordatch, I. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, pp. 11733–11763, 2024. URL https://proceedings.mlr.press/v235/du24e.html.

Glazer, E., Erdil, E., Besiroglu, T., Chicharro, D., Chen, E., Gunning, A., Olsson, C. F., Denain, J., Ho, A., de Oliveira Santos, E., Järviniemi, O., Barnett, M., Sandler, R., Vrzala, M., Sevilla, J., Ren, Q., Pratt, E., Levine, L., Barkley, G., Stewart, N., Grechuk, B., Grechuk, T., Enugandla, S. V., and Wildon, M. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in AI. *CoRR*, abs/2411.04872, 2024. URL https://doi.org/10.48550/arXiv.2411.04872.

Han, J., Buntine, W. L., and Shareghi, E. Verifiagent: a unified verification agent in language model reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 16410–16431, 2025. URL https://aclanthology.org/2025.findings-emnlp.891/.

Kim, S., Shin, J., Choi, Y., Jang, J., Longpre, S., Lee, H., Yun, S., Shin, S., Kim, S., Thorne, J., and Seo, M. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024. URL https://openreview.net/forum?id=8euJaTveKw.

Kwok, J., Li, S., Atreya, P., Liu, Y., Pavone, M., Stoica, I., and Mirhoseini, A. LLM-as-a-verifier: A general-purpose verification framework, 2026. Notion Blog. https://llm-as-a-verifier.notion.site/.

Li, D., Sun, R., Huang, Y., Zhong, M., Jiang, B., Han, J., Zhang, X., Wang, W., and huan liu. Preference leakage: A contamination problem in LLM-as-a-judge. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=grIvSXVJ65.

Li, Y., Zhang, S., Wu, R., Huang, X., Chen, Y., Xu, W., Qi, G., and Min, D. Mateval: A multi-agent discussion framework for advancing open-ended text evaluation. In *Database Systems for Advanced Applications - 29th International Conference, DASFAA 2024, Gifu, Japan, July 2-5, 2024, Proceedings, Part VII*, pp. 415–426, 2024. URL https://doi.org/10.1007/978-981-97-5575-2_31.

Li, Z., Zheng, Z., Chen, W., Zhao, J., Chen, Y., Xu, T., and Chen, E. Dynadebate: Breaking homogeneity in multi-agent debate with dynamic path generation. *CoRR*, abs/2601.05746, 2026b. doi: 10.48550/ARXIV.2601.05746. URL https://doi.org/10.48550/arXiv.2601.05746.

Liang, T., He, Z., Jiao, W., Wang, X., Wang, Y., Wang, R., Yang, Y., Shi, S., and Tu, Z. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 17889–17904, 2024. URL https://doi.org/10.18653/v1/2024.emnlp-main.992.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. In Kim, B., Yue, Y., Chaudhuri, S., Fragkiadaki, K., Khan, M., and Sun, Y. (eds.), *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/aca97732e30bcf1303bc22ac3924fd16-Paper-Conference.pdf.

Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R., and Zhu, C. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 2511–2522, 2023. URL https://doi.org/10.18653/v1/2023.emnlp-main.153.

Ospanov, A., Feng, Z., Sun, J., Bai, H., Xin, S., and Farnia, F. HERMES: towards efficient and verifiable mathematical reasoning in llms. *CoRR*, abs/2511.18760, 2025. doi: 10.48550/ARXIV.2511.18760. URL https://doi.org/10.48550/arXiv.2511.18760.

Peng, H., Qi, Y., Wang, X., Yao, Z., Xu, B., Hou, L., and Li, J. Agentic reward modeling: Integrating human preferences with verifiable correctness signals for reliable reward systems. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 15934–15949, 2025. URL https://aclanthology.org/2025.acl-long.775/.

Snell, C., Lee, J., Xu, K., and Kumar, A. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *CoRR*, abs/2408.03314, 2024. doi: 10.48550/ARXIV.2408.03314. URL https://doi.org/10.48550/arXiv.2408.03314.

Wang, X., Wei, J., Schuurmans, D., Le, Q. V., Chi, E. H., Narang, S., Chowdhery, A., and Zhou, D. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Yang, Z., Guo, Z., Huang, Y., Wang, Y., Shi, W., Wang, Y., Liang, X., and Tang, J. Accordion-thinking: Self-regulated step summaries for efficient and readable LLM reasoning. *CoRR*, abs/2602.03249, 2026. URL https://doi.org/10.48550/arXiv.2602.03249.

You, R., Cai, H., Zhang, C., Xu, Q., Liu, M., Yu, T., Li, Y., and Li, W. Agent-as-a-judge, 2026. URL https://arxiv.org/abs/2601.05111.

Zhang, C., Shu, C., Shareghi, E., and Collier, N. All roads lead to rome: Graph-based confidence estimation for large language model reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 31814–31824, 2025. URL https://doi.org/10.18653/v1/2025.emnlp-main.1620.

Zheng, L., Chiang, W., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., and Stoica, I. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

Zhu, L., Wang, X., and Wang, X. Judgelm: Fine-tuned large language models are scalable judges. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025. URL https://openreview.net/forum?id=xsELpEPn4A.

# A. System Comparison Table

*Table 2.* Comparison of representative LLM evaluation and verification systems across seven key dimensions. *Domain*: competition-level math (Comp. Math), general NLP, long-horizon agent benchmarks (Agents). *Prob. Level*: difficulty of math problems, where applicable ("—" = non-math domain; Easy–Med. = GSM8K / MATH-dataset level; AMC/AIME = competition level up to and including AIME; **Olympiad** = IMO/USAMO/Putnam). *Agents*: H = heterogeneous roles; K = $K$ i.i.d. re-calls. *Verdict*: flat vote, debate + flat vote, or debate + assessment gate. *Tools*: external Python/solver tools. *Blind*: no ground-truth access. *Prec.-first*: optimized to minimize false positives.

| System | Domain | Prob. Level | Agents | Verdict | Tools | Blind | Prec.-first |
|---|---|---|---|---|---|---|---|
| LLM-as-a-Judge (Zheng et al., 2023) | General NLP | — | 1 | Single-pass | — | ✓ | × |
| LLM-as-a-Verifier (Kwok et al., 2026) | Agents | — | $1\times K$ | MajVote | — | ✓ | × |
| ChatEval (Chan et al., 2024) | General NLP | — | $N$ (H) | Debate + vote | — | ✓ | × |
| DynaDebate (Li et al., 2026b) | Reasoning | AMC/AIME | $N$ (H) | Debate + vote | Code, Search | ✓ | × |
| VerifiAgent (Han et al., 2025) | Math / Code | Easy–Med. | 1 (loop) | Agentic chain | Python, Z3 | × | × |
| **Debate (Ours)** | **Comp. Math** | **Olympiad** | **5 (H)** | **Debate + gate** | **—** | ✓ | ✓ |

Our work is the only system in this comparison that simultaneously targets **Olympiad-level** competition mathematics (IMO/USAMO/Putnam), operates without ground-truth access, pursues a precision-first objective, and achieves high precision (92.5%) without external tools. Prior math-focused systems (DynaDebate, VerifiAgent) evaluate on standard benchmarks up to AMC/AIME difficulty; scaling to the Olympiad regime introduces qualitatively harder proof-based derivations where single-judge precision collapses to near chance (55.1%). The assessment gate and the $2\times2$ factorial attribution study are unique to our design.

# B. Full Results Table

*Table 3.* Per-run results for all systems. 195 variants, 58 problems. All systems: `gpt-5.2`, `effort="low"`. †MajVote-Verify precision/recall: per-run values; avg computed from pooled TP/FP. ‡Single-run result (ablation variant).

| System | Run | Acc | Prec | Recall | F1 | FP | FN | Prob. Acc |
|---|---|---|---|---|---|---|---|---|
| Single-Judge | r1 | 72.8% | 54.5% | 51.7% | 53.1% | 25 | 28 | 18/58 |
| | r2 | 72.8% | 54.9% | 48.3% | 51.4% | 23 | 30 | 18/58 |
| | r3 | 73.9% | 55.9% | 56.9% | 56.4% | 26 | 25 | 22/58 |
| | **avg** | **73.2%** | **55.1%** | **52.3%** | **53.6%** | **24.7** | **27.7** | **19.3/58** |
| MajVote-Verify ($N = 11$)† | r1 | 87.7% | 76.6% | 84.5% | 80.3% | 15 | 9 | 38/58 |
| | r2 | 85.6% | 73.4% | 81.0% | 77.1% | 17 | 11 | 33/58 |
| | r3 | 86.2% | 75.4% | 79.3% | 77.3% | 15 | 12 | 35/58 |
| | **avg** | **86.5%** | **75.1%** | **81.6%** | **78.2%** | **15.7** | **10.7** | **35.3/58** |
| MajVote-Precise‡ | – | 84.6% | 75.9% | 70.7% | 73.2% | 13 | 17 | 33/58 |
| Debate-NoGate‡ | – | 86.7% | 84.8% | 67.2% | 75.0% | 7 | 19 | 35/58 |
| Debate-Uniform‡ | – | 86.7% | 88.1% | 63.8% | 74.0% | 5 | 21 | 34/58 |
| Debate (Ours) | r1 | 87.2% | 90.2% | 63.8% | 74.7% | 4 | 21 | 34/58 |
| | r2 | 87.2% | 92.3% | 62.1% | 74.2% | 3 | 22 | 34/58 |
| | r3 | 88.7% | 95.0% | 65.5% | 77.6% | 2 | 20 | 36/58 |
| | **avg** | **87.7%** | **92.5%** | **63.8%** | **75.5%** | **3.0** | **21.0** | **34.7/58** |

# C. Pipeline Protocol Details

## C.1. Initial Deliberation JSON Schema

Each agent's initial output includes the following fields: `verdict` (binary support/oppose), `assessment_type` (`answer_supported` / `answer_refuted` / `reasoning_insufficient_but_answer_not_refuted`),

`evidence_grade` (`strong`/`medium`/`weak`), `rationale` (full chain-of-thought), `summary` ($\leq$400 chars, propagated to subsequent rounds following the Accordion principle (Yang et al., 2026)), `swing_issue` (single most decisive factor, one sentence), `key_checks` (list of specific checks performed), `confidence` ($[0, 1]$ scalar), `outbound_broadcast` (one claim for all agents to address), `outbound_dm` (list of 0–2 targeted messages to specific agents).

### C.2. Scheduler Logic

The scheduler is deterministic and generates no LLM calls. It assembles the broadcast and challenge messages programmatically from the previous round's `swing_issue` and `summary` fields. Three challenge categories are generated: (i) cross-challenge between the highest-priority support and oppose agents; (ii) weak-supporter pressure demanding a concrete check or stance switch; (iii) undecided-resolution directing `reasoning_insufficient` agents to compare competing claims.

### C.3. Problem Difficulty Distribution

Of the 58 problems, GPT-5.2 (effort=low) solves from scratch: $\geq$2/3 runs: 25 problems; exactly 1/3 runs: 9 problems; 0/3 runs: 24 problems. The 24 unsolvable-from-scratch problems represent the hardest tier, where candidate-trace verification appears most useful relative to direct solving.

### C.4. Debate Convergence and FP Ladder

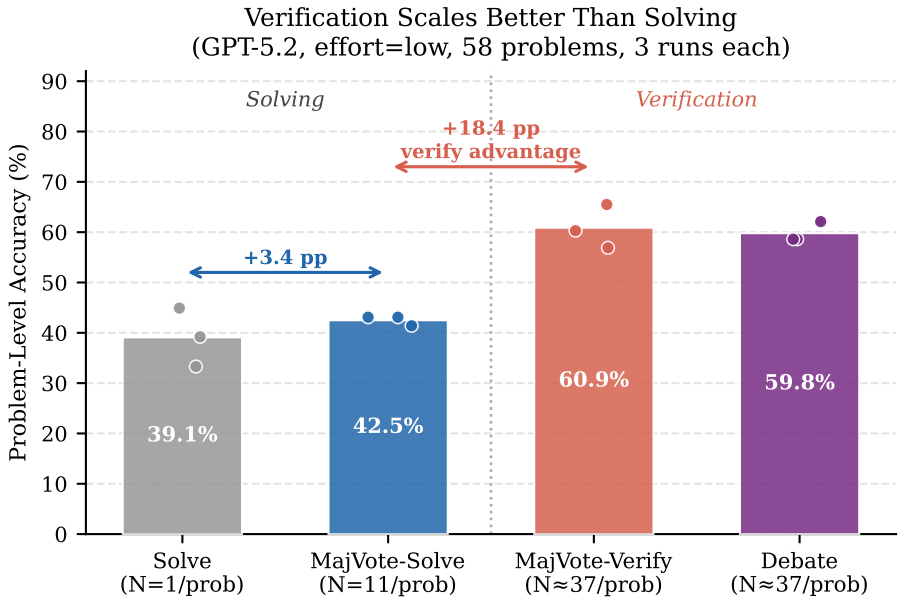


*Figure 4.* Diagnostic comparison of solving vs. verification. The two procedures allocate calls differently: solving produces one answer per problem, whereas verification evaluates candidate variants individually. At the variant level (Table 1), both systems produce similar false-positive counts (15.3 vs. 15.7), while verification achieves higher recall (81.6% vs. 42.5%).

## D. Why Debate Breaks the Martingale

Choi et al. (2025) prove that for homogeneous agents in iterated argument exchange, the expected vote tally is a martingale: debate adds no information beyond a single majority vote over the same number of calls. Our ablation (Debate-Uniform FP=5.0 vs. MajVote-Verify FP=15.7) shows this prediction fails when three mechanisms are simultaneously present:

**(1) Assessment gating.** The verdict rule rejects answers even when a *majority* of agents voted `support`, if fewer than three produced `answer_supported` (positive evidence) rather than `reasoning_insufficient` (uncertainty). This precision-biased filter has no analogue in majority voting and contributes approximately **31%** of the structural FP reduction over MajVote-Verify: removing it raises FP from 3.0 to 7.0 (Debate-NoGate, Table 1).

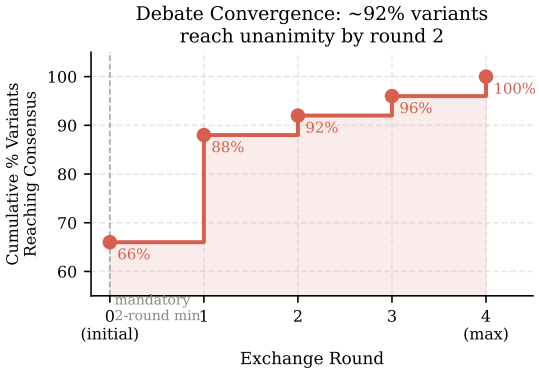


*Figure 5.* Cumulative debate convergence by exchange round. 92% of variants reach unanimous consensus by round 2, keeping average cost at ≈11 calls/variant despite a 5-round maximum. Only 8% require the full five rounds.

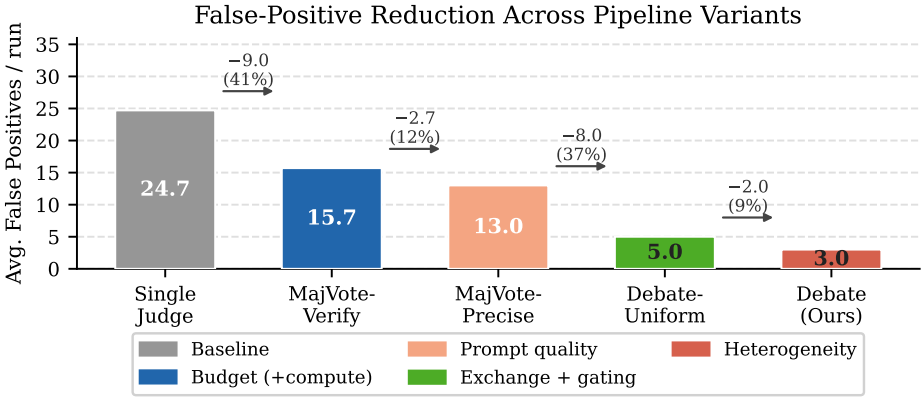


*Figure 6.* Step-by-step false-positive reduction across pipeline variants, coloured by mechanism type. Arrows show the FP delta and its share of the total 21.7-unit reduction.

**(2) Adversarial framing.** Wrong variants carry the model's own fluent reasoning, so a naive majority vote ("does this look correct?") reliably misjudges them. The debate protocol forces agents to steelman the opposition and find a FATAL objection—a materially stronger epistemic standard. Even *homogeneous* agents surface genuine disagreement on contested variants (61 of 195 variants per run undergo at least one stance change), generating signal that a flat vote cannot capture.

**(3) Role heterogeneity.** Each persona analyzes the problem through a distinct lens: algebraic consistency, theorem applicability, independent re-derivation, boundary conditions, numerical spot-checking. Two agents with different lenses who independently agree have a lower probability of correlated error than two agents sharing a lens. The information value of heterogeneous consensus is strictly higher, and accounts for an additional **16%** of structural FP reduction (Debate-Uniform → Debate: FP=5.0→3.0). This is consistent with the prediction of Li et al. (2026b) and the graph-theoretic view of self-consistency in Zhang et al. (2025).

## E. Per-Problem Analysis

Figure 7 visualizes the per-problem outcome across three runs for MajVote-Verify and Debate side by side. Table 4 lists per-problem correct counts for a representative subset. A problem is counted as correct in a given run only when the pipeline accepts the correct variant *and* rejects all wrong variants; 0–3 indicates how many of the three independent runs achieved this.

### E.1. MajVote-Verify and Debate Problem-Level Correct Counts (out of 3 runs)

The table reveals five interpretable patterns related to the four difficulty tiers described in Section 6:

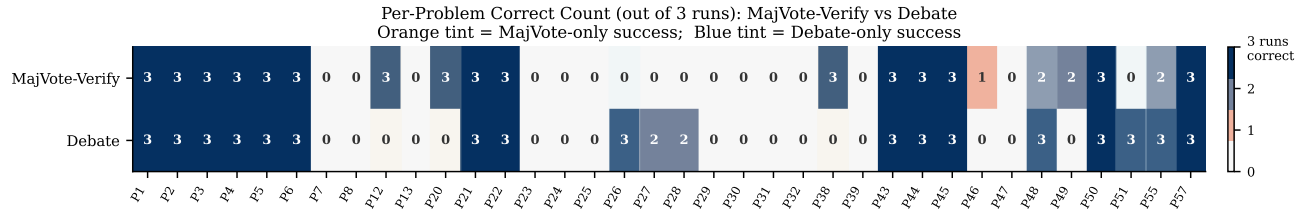


*Figure 7.* Per-problem correct count (0–3 runs) for MajVote-Verify (left) and Debate (right) on all 58 problems. **Orange** columns: MajVote-Verify succeeds but Debate fails (geometry problems where all agents produce `reasoning_insufficient`, triggering overcautious rejection). **Blue** columns: Debate succeeds but MajVote-Verify fails (problems with plausible wrong variants that require multi-round adversarial argument to reject; majority voting accepts the wrong variant alongside the correct one). **White/light** cells: neither system succeeds consistently (large-derivation problems that hit a raw capability ceiling regardless of pipeline structure).

*Table 4.* Per-problem correct count (0–3) for MajVote-Verify and Debate. A problem is "correct" only when the correct variant is accepted *and* all wrong variants are rejected in that run. Selected problems; full data in Figure 7. Abbrev.: Dbt = Debate, MV = MajVote-Verify.

| Prob | MV | Dbt | Prob | MV | Dbt | Prob | MV | Dbt |
|---|---|---|---|---|---|---|---|---|
| P1 | 3 | 3 | P22 | 3 | 3 | P43 | 3 | 3 |
| P2 | 3 | 3 | P23 | 0 | 0 | P44 | 3 | 3 |
| P3 | 3 | 3 | P24 | 0 | 0 | P45 | 3 | 3 |
| P4 | 3 | 3 | P25 | 0 | 0 | P46 | 1 | 0 |
| P5 | 3 | 3 | P26 | 0 | 3 | P47 | 0 | 0 |
| P6 | 3 | 3 | P27 | 0 | 2 | P48 | 2 | 3 |
| P7 | 0 | 0 | P28 | 0 | 2 | P49 | 2 | 0 |
| P8 | 0 | 0 | P29 | 0 | 0 | P50 | 3 | 3 |
| P12 | 3 | 0 | P30 | 0 | 0 | P51 | 0 | 3 |
| P13 | 0 | 0 | P31 | 0 | 0 | P52 | 1 | 1 |
| P20 | 3 | 0 | P32 | 0 | 0 | P55 | 2 | 3 |
| P21 | 3 | 3 | P38 | 3 | 0 | P57 | 3 | 3 |

**Debate-only success (Tier 2).** P26 and P51 are the clearest examples: MajVote-Verify achieves 0/3 despite accepting the correct variant in each run, because it also accepts one or more wrong variants (FP contamination disqualifies the problem). Debate's adversarial exchange correctly rejects these wrong variants through cascading counterexamples that emerge from multi-round bilateral challenges. These are exactly the cases where structured disagreement adds value that repetition cannot: the same simple prompt, applied 11 times, consistently misjudges the same plausible wrong argument.

**MajVote-only success (Tier 3).** P38 and P49 show the opposite: Debate achieves 0/3 despite no FP contamination, because all five agents independently output `reasoning_insufficient` for the *correct* variant, failing the assessment gate. P12 and P20 exhibit the same pattern. These are coordinate-geometry and trigonometric problems where agents correctly identify that they cannot fully re-derive the computation through text-only spot checks, but incorrectly treat this uncertainty as grounds for rejection. The assessment gate, designed to enforce positive evidence, here overfires: "I cannot independently verify this computation" becomes a false refusal.

**Large-derivation failures (Tier 4).** P30–32 (Domino series) and P29 (grid-game) fail both systems in every run; P27–28 are boundary cases discussed below as partial successes. These hit a raw capability ceiling: at `effort="low"`, the model cannot reproduce the derivation independently, so every agent produces `reasoning_insufficient`. Raising `effort` or adding symbolic tools (Han et al., 2025) are the most direct remedies.

**Partial success cases.** P27–28 show 0/3 for MajVote-Verify but 2/3 for Debate. Even at the boundary of capability, Debate's structured exchange allows recovery in some runs: when one agent finds a checkable sub-step in the correct variant's reasoning, the disagreement digest propagates it to the others, occasionally enabling the correct verdict. MajVote-Verify never recovers on these because its majority vote is too sensitive to FP contamination from correlated errors.

**Consistent successes (Tier 1).** Problems P1–6, P21–22, P43–45, P50, P57 achieve 3/3 for both systems. These are algebraically or arithmetically checkable problems where any capable verifier—single-judge or debate—can directly validate intermediate steps without re-deriving the full solution. They constitute the "easy" stratum where pipeline investment above simple majority voting provides minimal marginal return.

# F. Dataset Example and Debate Case Study

## F.1. Dataset Entry: Problem 25 (Olympiad-Level Combinatorics)

Problem 25 is representative of the dataset's difficulty tier: the base model *never* solves it from scratch across three attempts, and its wrong variants exploit common misapplications of advanced combinatorial theory.

**Problem statement.** Consider a $1849 \times 1849$ grid of unit squares. Rectangular tiles with sides parallel to the grid lines may be placed on the grid (non-overlapping, integer side lengths). Determine the *minimum* number of tiles required so that exactly one unit square in every row and exactly one unit square in every column remains uncovered.

**Correct variant (answer $= 1932$).** The key reduction: a set of exactly one uncovered square per row and per column is a permutation matrix; the problem minimizes the number of axis-parallel rectangles tiling the complement. A sharp result gives the minimum as $N + \lceil 2\sqrt{N} \rceil - 3$ for an $N \times N$ board. Since $N = 1849 = 43^2$ and $\lceil 2\sqrt{1849} \rceil = 86$, the answer is $1849 + 86 - 3 = 1932$. The reasoning constructs an explicit 43-block staircase permutation achieving this bound and proves no fewer tiles can suffice.

**Wrong variant 1 (answer $= 3696$).** The argument claims the minimum is $2n - 2$ for any $n \times n$ board and supplies an explicit construction using $2n - 2$ tiles (tiling the upper triangle and lower triangle separately). The construction is correct; the error is that the claimed minimum is *not* a lower bound—the permutation can be chosen to allow far fewer tiles when $n$ is a perfect square. The reasoning is coherent and the upper-bound construction is valid, making this variant superficially convincing.

**Wrong variant 2 (answer $= 2012$).** The candidate leverages $1849 = 43 \times 43$ to invoke an alleged "standard block-extremal formula":

$$R(N) = N + 2a + 2b - 9 \quad \text{when } N = ab,$$

applying it with $a = b = 43$ to obtain $1849 + 86 + 86 - 9 = 2012$. The derivation is presented as authoritative, with no proof of the formula and no explicit tiling construction, yet the argument reads fluently and correctly identifies the block-factorization structure. This is the hardest variant: the answer is plausible, the approach ($43 \times 43$ factorization) is mathematically meaningful, and the error—an unverified formula—is not surfaced by surface-level checks.

## F.2. Case Study: Assessment Gating Catches a 5–0 Initial Majority

We trace the Debate (Ours) pipeline on Problem 25, Variant 2 (answer $= 2012$, true label: *wrong*). This case illustrates how the assessment gate and exchange-round counterexamples together prevent a false positive that a majority-vote pipeline would have produced.

**Initial deliberation.** All five agents independently evaluate the candidate's reasoning. Every agent flags the same structural weakness—the formula $R(N) = N + 2a + 2b - 9$ is asserted without proof—but cannot supply a concrete refutation. Consequently, all five vote `support` (vote tally: 5–0), yet all five assign `assessment_type = reasoning_insufficient_but_answer_not_refuted`.

Representative initial swing issues (paraphrased from agent JSON):

- Agent 1 (Formalist Verifier): "No rigorous lower bound and no explicit construction; the claimed formula $R(N) = N + 2a + 2b - 9$ is unsupported and appears dubious under small-case sanity checks."
- Agent 2 (Independent Resolver): "Whether there is a genuine known theorem giving the exact minimum for the complement of a permutation matrix when $N$ is a perfect square; the candidate's formula reference is unattributed."

A naive majority-vote pipeline (**Debate-NoGate**) would *accept* this variant here: 5–0 satisfies any supermajority threshold. The assessment gate rejects it: fewer than three agents hold `answer_supported`, so the pipeline moves to the exchange round.

**Scheduler broadcast (after initial deliberation).** The scheduler detects all five as "weak-support" agents and issues the following directive (verbatim excerpt):

> *"assessment mix:* `reasoning_insufficient`$= 5$ *— weak-support agents (must provide a concrete check or switch to oppose): 1, 2, 3, 4, 5."*

All five are required to either find a concrete check confirming the formula or switch to oppose.

**Exchange round (Round 1 → Round 2).** Each agent independently performs a small-$N$ sanity check on the formula, arriving at distinct counterexamples:

- **Agent 1** (Formalist Verifier): $N = 2 = 1 \times 2 \Rightarrow R(2) = 2 + 2 + 4 - 9 = -1$, impossible.
- **Agent 2** (Independent Resolver): $N = 4 = 2 \times 2 \Rightarrow R(4) = 4 + 4 + 4 - 9 = 3$, but the diagonal-hole complement of a $4 \times 4$ board requires $\geq 4$ rectangles (explicit construction).
- **Agent 3** (Theorem Auditor): $N = 1 = 1 \times 1 \Rightarrow R(1) = 1 + 2 + 2 - 9 = -4$, impossible.
- **Agent 4** (Optimization Skeptic): $N = 3$ explicit tiling analysis shows formula unreliable; no valid lower bound in candidate solution.
- **Agent 5** (Pragmatic Cross-Checker): $N = 4$ structural argument: in the diagonal-uncovered arrangement, any valid rectangle must lie wholly on one side of the diagonal, forcing $\geq 4$ tiles.

Every agent independently falsifies the formula via a different small case. All five flip to `oppose`. Final vote: **0–5** (`wrong`). **Correct prediction.**

**Why this case matters.** Three distinct mechanisms are active simultaneously: *(i)* **Assessment gating**: the initial 5–0 majority is not accepted because no agent produced positive confirming evidence; Debate-NoGate would have output a false positive here. *(ii)* **Exchange-driven falsification**: the scheduler correctly identifies weak-support agents and requires concrete checks, triggering the falsification round that was never initiated in the initial deliberation. *(iii)* **Role heterogeneity**: the five agents approach the small-$N$ sanity check differently ($N = 1$, $N = 2$, $N = 3$, $N = 4$ algebraic, $N = 4$ structural), producing five independently valid counterexamples—any one of which is sufficient, but their diversity provides mutual reinforcement and reduces the chance that all five make a correlated error on the same check.