When Wrong Answers Look Right: AEGIS-D, a Multi-Agent Debate System for High-Precision Verification of Hard Competition Mathematics

Jiayang Sun [1 2] **Jiawei Xu** [3] **Maxm Pan** [4] **Zerui Cheng** [2 †] **Pramod Viswanath** [2 †]

Abstract

High-quality mathematical reasoning data are useful only when their answers and reasoning traces are reliable. We present AEGIS-D (Assessment- and Evidence-Gated Inspection System via Debate), a precision-first multi-agent pipeline for blind verification of competition-level mathematics. Five role-prompted agents exchange structured arguments for up to five rounds, and an assessment gate accepts a candidate only when at least three agents report 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; AEGIS-D achieves **92.5%** precision and 63.8% recall, reducing false positives from 24.7 to 3.0 per run. Because one accepted wrong variant contaminates an entire benchmark problem, we also report problem-level accuracy: the pipeline accepts the correct variant while rejecting every wrong variant for 59.8% of problems on average across three runs. A repeated 2×2 factorial associates role heterogeneity with the largest observed mean false-positive reduction, while post-hoc round and gate-threshold analyses expose the corresponding cost–precision trade-offs. Additional transfer tests provide preliminary evidence that the same precision-first operating point remains useful beyond the constructed benchmark.

[†]Advising authors. [1]The University of Hong Kong, Hong Kong SAR [2]Princeton University, Princeton, New Jersey, USA [3]Tsinghua University, Beijing, China [4]Tencent Hy Team, Tencent, China. Correspondence to: Jiayang Sun <aurorajy@connect.hku.hk>.

The 3rd *AI for Math Workshop* at the 43rd International Conference on Machine Learning (ICML), Seoul, South Korea, 2026.
Copyright 2026 by the author(s).

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 items may come from web-scraped explanations, human-written derivations, adapted competition problems, or LLM-generated attempts; a fluent but subtly wrong solution can silently contaminate downstream use. A secondary source of contamination, studied here only as an extension, is that generated or adapted statements may themselves be malformed.

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 operationalize this objective with AEGIS-D—an Assessment- and Evidence-Gated Inspection System via Debate.

Our primary task is blind answer verification: given a competition-level problem, a candidate answer, and a solution trace, decide whether the answer is CORRECT or WRONG, without access to a reference answer. As a secondary extension for generated items, the verifier may output INVALID when the statement itself has a material defect. 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 primary design questions, each corresponding to a failure mode in high-precision mathematical data filtering.

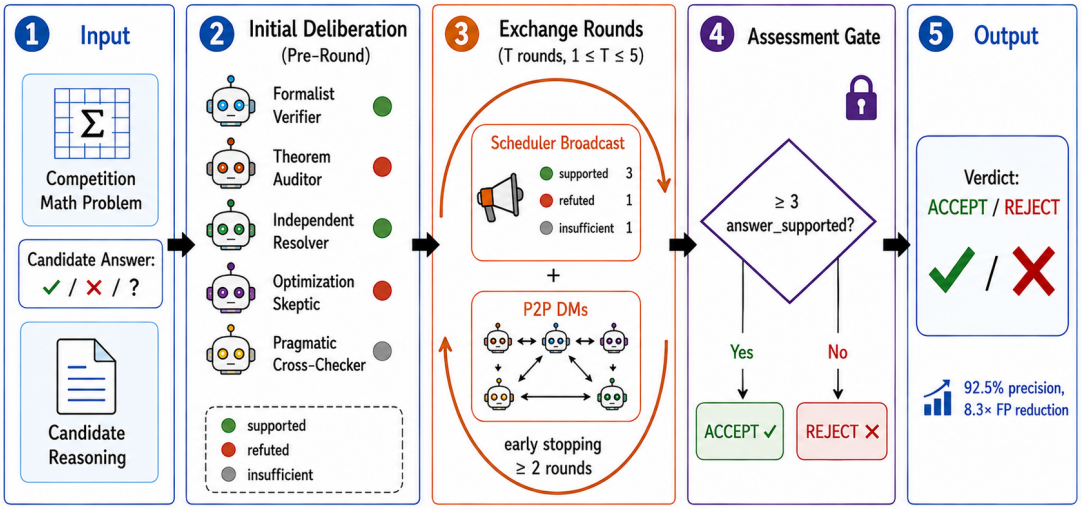

Figure 1. **AEGIS-D: precision-first multi-agent debate.** Five role-prompted agents **(2)** independently check answer correctness and, as a secondary extension, statement validity, then exchange arguments for up to five rounds via scheduler broadcasts and peer-to-peer direct messages **(3)**. A candidate is accepted only when the **assessment gate (4)** is satisfied: ≥ 3 agents hold `answer_supported`—positive confirming evidence, not mere non-refutation; ≥ 3 `problem_invalid` assessments instead reject the item as invalid.

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 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 2×2 factorial ablation and report Q1 as an empirical task-structure comparison rather than a fully compute-matched scaling study. As an additional extension, we test whether the same verifier can distinguish a wrong answer from a malformed, contradictory, or materially ambiguous problem statement. We treat this 69-item statement-validity component as preliminary because it was not the primary task of the original controlled benchmark and may contain lexical or template cues.

Our AEGIS-D pipeline (Figure 1, Section 4) achieves **92.5%** precision, reducing false positives by $8.2\times$ over a single-judge baseline. At the call-count level, an 11-call majority-vote verifier does not fully explain the gain, although our study is not token-matched. A three-run factorial study shows that the largest observed mean false-positive reduction is associated with heterogeneous roles, while assessment gating provides a smaller exactly paired conditional benefit under heterogeneous debate. Post-hoc scoring of the same trajectories shows that two rounds provide a cheaper operating point, while later rounds and stricter gate thresholds trade small amounts of recall for additional false-positive control. These results support precision-first multi-agent debate as a filtering primitive for both mathematical problems and solutions, including benchmark curation, reward labeling, and LLM-generated data pipelines.

2. Related Work

Judges and mathematical verification. LLM-as-a-Judge systems use single-pass or rubric-guided judgments for scalable evaluation (Zheng et al., 2023; Liu et al., 2023; Kim

et al., 2024; Zhu et al., 2025). Process supervision and verifier systems instead inspect intermediate reasoning or invoke external tools (Lightman et al., 2024; Wang et al., 2024; Rahman et al., 2025a; Han et al., 2025; Ospanov et al., 2025). Our setting is reference-free and tool-free: the evaluator must safely accept or reject a competition-level candidate trace using natural-language analysis.

Multi-agent debate and aggregation. Debate can expose errors through disagreement and rebuttal (Du et al., 2024; Liang et al., 2024; Khan et al., 2024; Rahman et al., 2025b). Related evaluation systems use multi-agent discussion, while controlled work highlights correlated reasoning paths, majority pressure, and the importance of group diversity (Chan et al., 2024; Li et al., 2024; Choi et al., 2025; Wu et al., 2025; Li et al., 2026). Our distinction is a precision-first verdict rule: acceptance requires a majority of positive `answer_supported` assessments, and our repeated factorial separately varies role prompting and applies the gate post hoc to identical trajectories.

Test-time scaling and data curation. Self-consistency and test-time scaling improve reasoning by repeated sampling or richer aggregation (Wang et al., 2023; Snell et al., 2024; Zhang et al., 2025). Hard benchmarks and verified-data pipelines increase the need for reliable filtering (Glazer et al., 2024; Center for AI Safety et al., 2026; An et al., 2025; Cheng et al., 2026). Unlike solving benchmarks such as Omni-MATH and OlymMATH (Gao et al., 2024; Sun et al., 2025), our main benchmark attaches plausible candidate traces and measures blind verification. Supplementary related work and a system comparison appear in Appendix B and Table 6.

3. Task and Dataset

3.1. Adversarial Answer Verification

Our primary task is: given a valid competition-level mathematics problem, a candidate 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. As a secondary extension for generated or adapted data, the verifier may output INVALID when the statement contains a material defect such as an undefined variable, unresolved placeholder, contradiction, or ambiguity that prevents a determinate answer.

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. The controlled 195-variant component isolates the primary CORRECT/WRONG task; the invalid-statement component is reported as a preliminary extension.

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. For invalid problems, we additionally report **invalid recall** and the number dangerously predicted CORRECT.

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]

Invalid problem statements. To evaluate statement validation, we collect 69 malformed or under-specified competition-style statements surfaced during problem generation and adaptation, manually confirm their statement-level defects, and assign one `invalid` variant to each. The set includes undefined symbols, unresolved template

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

fields, contradictory conditions, and formula parses that admit materially different tasks. Together with the original answer-verification variants, the full benchmark contains **264 variants across 127 problems**: 58 correct variants, 137 wrong variants, and 69 invalid problem statements. Because many invalid statements arose from generation failures such as unresolved placeholders, this component may contain lexical or template cues; we therefore treat it as a preliminary statement-validity benchmark rather than a comprehensive test of semantic invalidity.

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. This filter is used only to construct hard negatives: the reported Single-Judge numbers should therefore be interpreted as performance on a conditioned hard-negative stress test, not as precision on a natural mixture of mathematical solutions. 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. Correct and wrong traces therefore often differ in generation condition: correct traces are usually reference-equipped, whereas wrong traces are usually natural unguided failures. Plausibility filtering further enriches the benchmark for hard negatives. The controlled set should consequently be read as an adversarial verification stress test, not as an estimate of performance on the natural prevalence or style distribution of mathematical solutions. The plausibility filter intentionally suppresses easy negatives that a simple judge already rejects; performance on a natural mixture may therefore be higher in absolute terms, while the relative gap between systems may differ.

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 5.5. 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. AEGIS-D Architecture

Role-prompted agents. AEGIS-D employs five agents with complementary verification personas:

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.
4. **Optimization Skeptic** — feasibility-vs.-optimality gaps, one-sided bounds, and constraint or geometry-convention violations.
5. **Pragmatic Cross-Checker** — final arithmetic, whether an exact value is uniquely pinned down, and whether an objection is answer-critical or merely exposition-critical.

All use the same base model (`gpt-5.2`) with different system instructions. Role prompting is intended to reduce shared false positives by encouraging complementary checks; it does not make same-model agents statistically independent.

Deliberation and exchange. Before exchange, each agent restates the claim, checks likely failure modes, performs an independent mathematical check, steelmans the strongest opposing case, and returns structured JSON. The `assessment_type` field separates positive support, positive refutation, insufficient evidence, and—for the secondary extension— statement invalidity. AEGIS-D runs for up to five rounds, with early stopping after round 2. Each round broadcasts the vote and assessment breakdown plus unresolved claims; scheduler challenges and targeted peer messages require agents to address specific counterarguments and perform a new check before recommitting. Full schemas and scheduler details appear in Appendix D.

Verdict rule. 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 report *positive confirming evidence*, not merely failed to find a flaw. Independently, if $\geq$ 3 agents carry `assessment_type = problem_invalid`, the final label is INVALID; otherwise, any item that fails the acceptance gate is labeled WRONG.

5. Experiments

5.1. Experimental Setup

Unless otherwise stated, controlled experiments use `gpt-5.2` (`effort="low"`). The controlled answer-verification ladder uses the 195 valid-statement variants from 58 problems; three independent runs are used for main systems, and the ablation rows in the end-to-end ladder are single-run. The separate factorial mechanism study uses three independent runs per role condition and scores gating post hoc on identical trajectories; only the gate contrast within each role condition is therefore exactly paired. Seven systems form 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); **AEGIS-D (Ours)** (full pipeline: heterogeneous personas, exchange, assessment gate).

[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 5.5).

Table 1. Results on 195 competition-mathematics variants from 58 problems. Positive class = `correct`; all controlled 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%
AEGIS-D (Ours)†	**92.5%**	63.8%	**3.0**	59.8%

5.2. Main Results

Table 1 reports all seven systems. AEGIS-D reduces FP from 24.7 to 3.0—an 8.2× reduction—while raising precision from 55.1% to 92.5%. Across three runs, precision ranges from 90.2% to 95.0%, indicating limited observed variation, although three runs are insufficient for a strong stability claim. 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, leaving nearly half of accepted variants incorrect and making it unsuitable for precision-first filtering. **Finding 2** (Heterogeneity Controls FP): In the factorial, the largest observed mean FP reduction is associated with heterogeneous roles (4.33/run with gating enabled); the exactly paired gate contrast adds a further 1.00 FP/run reduction only under heterogeneous debate. **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. AEGIS-D 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. AEGIS-D accepts the true answer for ≈37 problems (recall 63.8%) with only 3.0 FP/run, contaminating ≈3 problems. For benchmark

Table 2. Post-hoc sensitivity on the 195 answer variants (three-run means). Round truncation uses the $\geq 3/5$ gate; threshold sensitivity uses the final early-stopped trajectories. Calls are model calls per variant.

Analysis	Setting	Prec.	Recall	FP	Prob. Acc.	Calls
Rounds	$R = 1$	90.9%	63.2%	3.67	58.0%	5.0
	$R = 2$	90.8%	62.1%	3.67	57.5%	10.0
	$R \leq 5$ (early stop)	**92.5%**	63.8%	3.00	**59.8%**	10.9
Gate	$\geq 2/5$	91.1%	**64.4%**	3.67	**60.3%**	10.9
	$\geq 3/5$	92.5%	63.8%	3.00	59.8%	10.9
	$\geq 4/5$	94.0%	62.6%	2.33	58.6%	10.9
	$5/5$	**94.6%**	60.3%	**2.00**	56.9%	10.9

construction, where any FP on a problem makes the entire entry unusable, this difference is decisive.

5.3. Round and Gate-Threshold Sensitivity

Two protocol choices warrant direct sensitivity analysis: the five-round maximum and the $\geq 3/5$ assessment threshold. We therefore rescore the same three completed trajectories after truncating them at each round and after varying the final `answer_supported` threshold. These are exactly paired post-hoc comparisons: no prompts, samples, or model calls change.

Table 2 shows that most items settle quickly: 88.7%, 88.7%, and 86.7% of variants stop after round 2 in the three runs. Stopping all items at round 2 is a cheaper operating point, but the remaining rounds improve mean precision by 1.7 pp, recover 1.7 pp recall, and reduce mean FP from 3.67 to 3.00 while adding fewer than one model call per variant on average. The round-by-round trajectory is not monotone for every run, which cautions against interpreting additional discussion as uniformly beneficial. Because the marginal gain after round 2 is small, we view $R = 2$ as the low-cost operating point and $R \leq 5$ with early stopping as the conservative precision-first setting.

The threshold sweep exposes the intended precision–yield trade-off. The pre-specified $\geq 3/5$ threshold is not uniquely optimal under every metric: $\geq 2/5$ gives slightly higher recall and problem accuracy, whereas $\geq 4/5$ and unanimity further reduce FP at lower yield. We retain $\geq 3/5$ because it is the natural majority-positive-evidence rule used throughout the study, not a threshold selected post hoc to maximize F1.

5.4. Ablation Study: Factorial Attribution

Factorial design with paired gate contrasts. The original system ladder mixes changes to prompts, exchange structure, roles, and aggregation; it is useful as an end-to-end comparison but does not by itself identify causal mechanisms. We therefore run a separate 2×2 factorial over role assignment (homogeneous vs. heterogeneous) and assessment gating (off vs. on). For each role condition, the same completed debate trajectory is scored both with and without the gate, making the gate comparison exactly paired. Each role condition is independently run three times. These results are not identical to the end-to-end ladder in Table 1 because the factorial is a separately sampled mechanism study; we use it for directional mechanism attribution rather than as the headline performance estimate.

Table 3. Three-run 2×2 factorial on the original 195 answer variants (mean ± sample standard deviation). Gate contrasts are paired within role condition; role contrasts are between independent runs.

Roles	Gate	Prec.	Recall	FP	FN
Homogeneous	No	86.5±1.9%	66.1±2.6%	6.00±1.00	19.67±1.53
Homogeneous	Yes	86.4±1.8%	65.5±1.7%	6.00±1.00	20.00±1.00
Heterogeneous	No	93.2±6.4%	62.6±3.6%	2.67±2.52	21.67±2.08
Heterogeneous	Yes	**95.5±4.1%**	60.3±1.7%	**1.67±1.53**	23.00±1.00

Heterogeneity has the largest observed FP reduction. With gating enabled, heterogeneous roles reduce mean FP from 6.00 to 1.67 per run; without gating, they reduce FP from 6.00 to 2.67. The three heterogeneous gated runs produce 3, 2, and 0 FP, compared with 7, 6, and 5 under homogeneous gated debate. This pattern is consistent with the intended mechanism: complementary verification lenses may reduce correlated acceptance of plausible wrong reasoning. Because role conditions use independent runs, this contrast is descriptive rather than an exactly paired causal estimate.

Gating provides a smaller conditional benefit. Under heterogeneous debate, applying the assessment gate to the same trajectories reduces mean FP from 2.67 to 1.67, while lowering recall from 62.6% to 60.3%. Under homogeneous debate, gating does not reduce FP (6.00 in both cells). Thus the exactly paired gate contrast does not support a context-independent gating effect; instead, positive-evidence gating is most useful after diverse roles have surfaced sufficiently discriminative checks. This precision–recall trade-off is aligned with our precision-first objective.

What "heterogeneity" changes. All agents share the same base model, so role prompting should not be interpreted as statistical independence. We directly examine assessment agreement and correlated support errors on the factorial trajectories. Initial pairwise assessment agreement is nearly unchanged (71.3% homogeneous vs. 71.7% heterogeneous), so the benefit is not a generic increase in disagreement. Instead, among truly wrong variants, the number receiving `answer_supported` from at least three agents before exchange falls from 11.0 to 5.0 per run; after debate it falls from 6.0 to 1.67. Thus the observed role benefit is more precisely described as reduced *correlated positive*

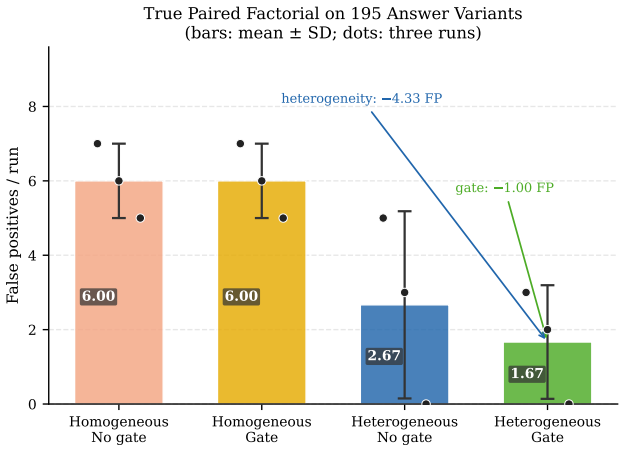

Figure 2. Factorial false positives on the 195 answer variants. Bars show three-run mean ± standard deviation; dots show individual runs. Heterogeneous roles provide the largest FP reduction, while gating adds a smaller benefit only under heterogeneous debate.

support on wrong answers, not as evidence that same-model agents become independent.

Coverage and truncation handling. All heterogeneous runs complete 264/264 variants. The homogeneous runs complete 259/264, 261/264, and 264/264 variants; the remaining eight run-level failures exhaust the fixed 4,096-token output budget on hidden reasoning before producing visible text. We record these as `reasoning_truncated`, assign no inferred verdict, and exclude them from classification metrics. The table reports the answer-verification subset extracted from these full-benchmark trajectories, so the corresponding failed answer variants are also excluded from the homogeneous metrics. Because two homogeneous runs are incomplete and role conditions are independently sampled, the role contrasts are descriptive and we do not report problem-clustered bootstrap confidence intervals.

5.5. 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. Appendix D provides protocol details, and Figure 3 visualizes the diagnostic comparison.

Table 4. Preliminary statement-validity extension. Binary precision treats only CORRECT as accepted.

Setting	N	Binary Prec.	Invalid Recall	Invalid→Correct
Answer + invalid	264	94.7%	85.5%	1

Table 5. Single-run external transfer. OlymMATH audited sensitivity is not an official score on the unmodified benchmark.

Setting	N	Prec.	Recall	FP
Omni-MATH (d>8)	187	88.2%	51.7%	8
OlymMATH unmodified	100	82.0%	80.4%	9
OlymMATH audited sensitivity	100	94.0%	82.5%	3

5.6. Additional Robustness Studies

Statement validity. As a preliminary extension, we evaluate 69 malformed or under-specified statements together with the 195 answer variants. Table 4 shows that only one invalid statement is dangerously accepted as correct. Because many invalid items contain template cues, this experiment is not a primary claim; full confusion matrices and role comparisons appear in Appendix G.

External transfer. We also run a single generate-then-verify pass on Omni-MATH and OlymMATH (Table 5). The generator and verifier both use `gpt-5.2`, so this tests benchmark transfer rather than generator independence. These results are not used for the main claim, but provide preliminary evidence that the precision-first operating point is not unique to the constructed benchmark. Protocol and audit details appear in Appendix F.

Cross-model check. A single-run fixed-prompt Claude Opus replication is directionally consistent with the GPT-5.2 ablation, but because it is not repeated and its exact model snapshot was not retained, we report it only as supplementary evidence (Appendix H).

6. Error Analysis

False positives (3.0/run). Observed residual FPs often 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. In these cases, agent spot-checks fail to cover the flawed sub-step and may produce positive evidence for nearby correct steps. Further reduction may benefit from tool-augmented verification (Han et al., 2025) or cross-provider ensemble diversity.

False negatives (21.0/run, recall 63.8%). Inspection suggests three recurring, non-exhaustive failure modes: **capability saturation**, where agents reach `reasoning_insufficient`; **active false refutation**, where an agent mis-commits to `answer_refuted`; and **mixed evidence**, where a support/uncertainty split fails to clear the gate. The latter two may be addressable via stronger steelman enforcement or tool augmentation (Han et al., 2025).

Illustrative evidence-gating case. In one wrong variant, all five agents initially voted to support the candidate, yet none produced positive confirming evidence: all five recorded the "reasoning insufficient" assessment because the solution invoked an unproved block-extremal formula. A flat vote would have accepted the answer immediately. The structured assessment representation instead identified five weak supporters, allowing the scheduler to target them in the next exchange round; the final assessment gate would also have prevented immediate acceptance because fewer than three agents held `answer_supported`. In that exchange, small-case checks falsified the formula and all five agents switched to oppose. This case illustrates the difference between stance voting and positive-evidence aggregation: apparent consensus is not acceptance evidence when the answer-critical step remains unchecked. The complete trajectory appears in Appendix J.2.

Convergence statistics and per-problem accuracy breakdowns appear in Appendices D.4 and I.

7. Discussion

Verification as selective prediction. AEGIS-D is best understood as a selective predictor rather than a general-purpose mathematical grader. Its objective is to identify a high-quality subset, making precision, false-positive count, and problem-level accuracy more informative than answer-level accuracy alone. For example, MajVote-Verify recovers more correct variants than AEGIS-D, but also accepts substantially more wrong variants. Those false positives create ambiguous benchmark entries; AEGIS-D instead sacrifices yield to make accepted items more trustworthy. This operating point is appropriate for generate–verify–filter workflows, where candidate generation can be scaled more cheaply than expert auditing: rejected candidates can be regenerated, whereas an accepted error may silently enter training data, reward labels, or an evaluation set. The verifier should complement periodic human audits by reducing their volume while preserving a conservative acceptance boundary.

What structured debate contributes. Our results do not imply that communication is intrinsically beneficial or that adding more agents automatically improves verification. The homogeneous no-gate and gated cells in Table 3 have the same mean false-positive count, and round-level trajectories are not monotone for every run. The observed gain depends on information representation and routing: role prompts diversify checks, the scheduler exposes weak support, mandatory new checks turn disagreement into evidence, and the gate prevents uncertainty from counting as confirmation. The illustrative case above makes this distinction concrete: five agents initially agree at the stance level, but their structured assessments reveal that none has verified the decisive theorem. Exchange is useful because it converts that hidden epistemic weakness into a specific falsification task. The unit of aggregation is therefore a typed claim about what has been checked, not merely a vote. The factorial further suggests an interaction: gating helps on heterogeneous trajectories, where agents more often surface discriminative evidence, but provides no observed false-positive reduction on homogeneous trajectories. We therefore interpret role diversity and evidence-aware aggregation as complementary design choices rather than separable guarantees.

Choosing a deployment operating point. The round and threshold sweeps expose several reasonable configurations rather than one universally optimal setting. A two-round run captures most convergence at lower cost; the full early-stopped protocol adds fewer than one call per variant on average and offers a modest additional precision gain. Likewise, a $2/5$ gate retains more correct candidates, whereas stricter gates reduce false positives at lower yield. The appropriate choice depends on the downstream cost ratio. Benchmark release and reward-model labeling may justify the conservative $3/5$ or $4/5$ settings because a false acceptance can affect many later experiments. Interactive tutoring or exploratory data generation may prefer a lower threshold followed by human review. The assessment counts also provide a natural triage signal beyond the final binary decision:

unanimous positive evidence can be routed directly to a retained pool, mixed support to targeted re-verification, and widespread insufficiency to a stronger model or external tool. Although we do not evaluate this cascaded policy, it follows directly from the pipeline's structured outputs and could reduce cost without weakening the acceptance criterion.

Implications for verifier design. Residual errors indicate two different bottlenecks. False positives often require broader *coverage*: agents validate nearby steps but miss the one subtle arithmetic or global-condition failure that changes the answer. False negatives instead often require greater *capability*: agents correctly recognize that a proof is difficult but cannot construct enough positive evidence. These failure modes motivate different extensions. Coverage may improve through executable spot checks, symbolic algebra, theorem retrieval, or agents from different model families. Capability-limited cases may require higher inference effort, longer derivations, or formal proof assistance. Importantly, adding tools should preserve the same epistemic distinction used here: tool failure or absence of a counterexample is not positive support, whereas a reproduced computation, verified certificate, or checked theorem application can raise the evidence grade. This suggests a natural way to carry the assessment-gate idea into hybrid verifiers, although whether it remains effective with tools must be tested.

Auditability and reproducibility. The structured trajectory records each assessment, evidence grade, check, decisive issue, and stance change. Unlike a scalar score, it reveals whether acceptance rested on an independent derivation, a boundary-case test, theorem applicability, or repeated claims. Although agents exchange compressed summaries, complete outputs remain available for auditing and targeted dataset maintenance.

Safe use and escalation. High acceptance precision does not constitute a correctness guarantee, and the remaining errors preclude use as an autonomous mathematical authority. Reproducibly checked items can enter a provisional pool, while borderline items should be escalated to stronger models, tools, or experts and accepted items periodically audited. Distribution shifts may require source-specific calibration. The contribution is an auditable filtering primitive, not a replacement for trusted mathematical validation.

8. Limitations

Yield and operating point. The verifier deliberately trades recall for precision: recall is 63.8%, and difficult correct traces often end in `reasoning_insufficient`. This operating point is most useful when candidate generation is cheap: a large pool can still yield many accepted items, whereas false positives directly corrupt the retained set. The threshold sweep makes this trade-off explicit, but does not eliminate the underlying capability ceiling.

Dataset and statistical scope. The controlled benchmark contains only 195 variants from 58 problems and deliberately selects plausible hard negatives. Correct traces are usually reference-equipped, wrong traces are usually unguided failures, and plausibility filtering suppresses easy negatives; the reported numbers should therefore be read as an adversarial stress test rather than natural-distribution performance. Our solve–verify comparison is call-level rather than token-matched, and two homogeneous factorial runs contain eight output truncations in total.

Model diversity and extensions. Role-prompted agents share one base model and are not independent. The external and cross-model studies are single-run, while the statement-validity extension may contain template cues and lacks a simple lexical baseline. Larger cross-provider, source-matched, repeated, and tool-augmented evaluations remain future work.

9. Conclusion

We presented AEGIS-D, an assessment- and evidence-gated multi-agent debate system for blind verification of competition-level mathematical answers. On 195 hard variants from 58 IMO/USAMO/Putnam-level problems, it achieves 92.5% precision and reduces false positives by $8.2\times$ over a single judge. Acceptance is based on positive confirming evidence rather than unrestricted majority voting. Repeated factorial and sensitivity analyses suggest that role prompts reduce correlated positive support for wrong answers, while stricter gates trade recall for additional false-positive control. These results support precision-first debate as a practical filtering primitive, while leaving larger cross-provider and tool-augmented verification to future work.

Code and Data Availability

Code, prompts, benchmark variants, and evaluation artifacts are available at https://github.com/aurorajy1108/When-Wrong-Answers-Look-Right.

Impact Statement

Automated mathematical-data validation may reduce annotation costs but relies on fallible evaluators; precision-first rejection mitigates this risk.

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.

Cui, Y., Fu, H., Zhang, H., Wang, L., and Zuo, C. Free-MAD: Consensus-free multi-agent debate. *CoRR*, abs/2509.11035, 2025. doi: 10.48550/ARXIV.2509.11035. URL https://doi.org/10.48550/arXiv.2509.11035.

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.

Gao, B., Song, F., Yang, Z., Cai, Z., Miao, Y., Dong, Q., Li, L., Ma, C., Chen, L., Xu, R., Tang, Z., Wang, B., Zan, D., Quan, S., Zhang, G., Sha, L., Zhang, Y., Ren, X., Liu, T., and Chang, B. Omni-math: A universal olympiad level mathematic benchmark for large language models. *CoRR*, abs/2410.07985, 2024. doi: 10.48550/ARXIV.2410.07985. URL https://doi.org/10.48550/arXiv.2410.07985.

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/.

Kaesberg, L. B., Becker, J., Wahle, J. P., Ruas, T., and Gipp, B. Voting or consensus? decision-making in multi-agent debate. *CoRR*, abs/2502.19130, 2025. doi: 10.48550/ARXIV.2502.19130. URL https://doi.org/10.48550/arXiv.2502.19130.

Khan, A., Hughes, J., Valentine, D., Ruis, L., Sachan, K., Radhakrishnan, A., Grefenstette, E., Bowman, S. R., Rocktäschel, T., and Perez, E. Debating with more persuasive LLMs leads to more truthful answers. *arXiv preprint arXiv:2402.06782*, 2024.

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, 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, 2026. 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.

Liu, T., Wang, X., Huang, W., Xu, W., Zeng, Y., Jiang, L., Yang, H., and Li, J. Groupdebate: Enhancing the efficiency of multi-agent debate using group discussion. *CoRR*, abs/2409.14051, 2024. doi: 10.48550/ARXIV.2409.14051. URL https://doi.org/10.48550/arXiv.2409.14051.

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.

Rahman, S., Gorantla, S., Gupta, A., Roy, S., Peng, N., and Liu, Y. SPARK: Stepwise process-aware rewards for reference-free reinforcement learning. *arXiv preprint arXiv:2512.03244*, 2025a.

Rahman, S., Issaka, S., Suvarna, A., Liu, G., Shiffer, J., Lee, J., Parvez, M. R., Palangi, H., Feng, S., Peng, N., Choi, Y., Michael, J., Jiang, L., and Gabriel, S. AI debate aids assessment of controversial claims. *arXiv preprint arXiv:2506.02175*, 2025b.

Shen, C., Wong, Z. H., He, R., Liang, H., Qiang, M., Meng, Z., Zhao, Z., Zeng, B., Zhu, Z., Cui, B., and Zhang, W. Let's verify math questions step by step. *CoRR*, abs/2505.13903, 2025. doi: 10.48550/ARXIV.2505.13903. URL https://doi.org/10.48550/arXiv.2505.13903.

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.

Sun, H., Min, Y., Chen, Z., Zhao, W. X., Liu, Z., Wang, Z., Fang, L., and Wen, J. Challenging the boundaries of reasoning: An olympiad-level math benchmark for large language models. *CoRR*, abs/2503.21380, 2025. doi: 10.48550/ARXIV.2503.21380. URL https://doi.org/10.48550/arXiv.2503.21380.

Wang, P., Li, L., Shao, Z., Xu, R., Dai, D., Li, Y., Chen, D., Wu, Y., and Sui, Z. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439, 2024. doi: 10.18653/v1/2024.acl-long.510. URL https://aclanthology.org/2024.acl-long.510/.

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.

Wu, H., Li, Z., and Li, L. Can LLM agents really debate? a controlled study of multi-agent debate in logical reasoning. *CoRR*, abs/2511.07784, 2025. doi: 10.48550/ARXIV.2511.07784. URL https://doi.org/10.48550/arXiv.2511.07784.

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.

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.

Zhao, Y., Li, Y., Bo, Z., Takezoe, R., Hui, H., Guang, M., Ren, L., Qin, X., and Long, K. SATQuest: A verifier for logical reasoning evaluation and reinforcement fine-tuning of LLMs. *CoRR*, abs/2509.00930, 2025. doi: 10.48550/ARXIV.2509.00930. URL https://doi.org/10.48550/arXiv.2509.00930.

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.

Zhou, Y., Xu, A., Wang, P., Xiong, C., and Joty, S. Evaluating judges as evaluators: The JETTS benchmark of LLM-as-judges as test-time scaling evaluators. *CoRR*, abs/2504.15253, 2025. doi: 10.48550/ARXIV.2504.15253. URL https://doi.org/10.48550/arXiv.2504.15253.

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 6. 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., 2026)	Reasoning	AMC/AIME	N (H)	Debate + vote	Code, Search	✓	×
VerifiAgent (Han et al., 2025)	Math / Code	Easy–Med.	1 (loop)	Agentic chain	Python, Z3	×	×
AEGIS-D (Ours)	**Comp. Math**	**Olympiad**	**5 (H)**	**Debate + gate**	**—**	✓	✓

Within this representative comparison, our system is distinguished by jointly targeting **Olympiad-level** competition mathematics (IMO/USAMO/Putnam), operating without ground-truth access, pursuing a precision-first objective, and achieving 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 is insufficient for precision-first filtering (55.1%). Our main methodological distinction is the combination of an assessment gate with a 2×2 factorial attribution study.

B. Supplementary Related Work

Debate efficiency and decision protocols. Beyond improving raw reasoning accuracy, recent multi-agent debate work studies how communication topology and aggregation rules affect cost and reliability. GroupDebate partitions agents into discussion groups and exchanges intermediate group results, reducing redundant communication (Liu et al., 2024). Free-MAD instead removes the requirement that agents converge, using anti-conformity and a score over the full debate trajectory rather than a final majority vote (Cui et al., 2025). Controlled comparisons of voting and consensus protocols likewise show that the final decision rule can materially change debate outcomes, independently of the underlying discussion (Kaesberg et al., 2025). These systems optimize general reasoning performance or efficiency. Our assessment gate serves a narrower data-filtering objective: it rejects mathematical items unless a majority supplies positive confirming evidence, even when the final stance vote would accept them.

Executable and formal verification. Another line of work obtains reliable labels from executable specifications or formal proof systems. SATQuest generates satisfiability-based reasoning tasks and objectively checks answers with a SAT solver, supporting controlled evaluation and reinforcement fine-tuning (Zhao et al., 2025). Similarly, VeRA uses executable specifications for verified data augmentation, while HERMES and VerifiAgent introduce formal or adaptive tool-based checks into mathematical reasoning (Cheng et al., 2026; Ospanov et al., 2025; Han et al., 2025). Such approaches provide stronger guarantees when a trusted checker can be constructed. Our setting addresses the complementary case in which candidate problems and solutions arrive from heterogeneous sources, no executable specification or reference answer is available, and the verifier must decide whether an item is safe to retain using blind natural-language analysis.

Relation to process supervision and statement validation. Process-supervision methods learn or apply step-level correctness signals to mathematical solutions (Lightman et al., 2024; Wang et al., 2024), whereas judge-evaluation benchmarks examine whether evaluators remain useful inside test-time scaling procedures (Zhou et al., 2025). MathQ-Verify extends verification upstream to ill-posed mathematical questions (Shen et al., 2025). Our task unifies these concerns at inference time: agents first decide whether the statement is valid, then seek local evidence about the submitted derivation, and finally route the item to CORRECT, WRONG, or INVALID without trained verifier supervision.

Table 7. Per-run results for systems with retained variant-level run records. 195 variants, 58 problems. All controlled 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
AEGIS-D (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. Full Results Table

D. Pipeline Protocol Details

D.1. Initial Deliberation JSON Schema

Each agent's initial output includes the following fields: `verdict` (`correct`/`wrong`/`invalid`), `assessment_type` (`answer_supported` / `answer_refuted` / `reasoning_insufficient_but_answer_not_refuted` / `problem_invalid`), `evidence_grade` (`strong`/`medium`/`weak`), `rationale` (structured verdict-oriented analysis), `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).

D.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. The broadcast separately identifies `problem_invalid` claims and asks all agents to determine whether each claim concerns a genuine statement defect rather than an answer or reasoning error.

D.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.

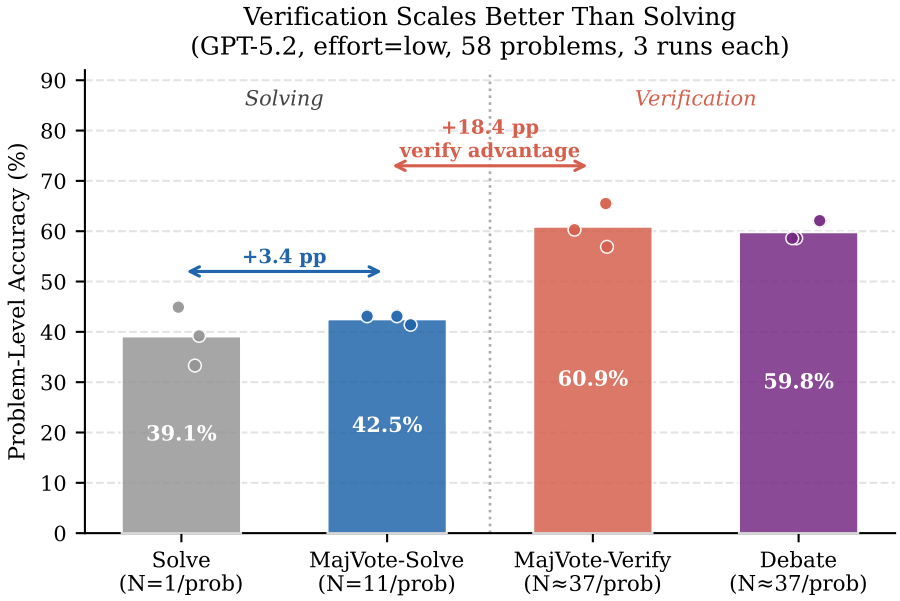

Figure 3. Diagnostic comparison of solving vs. verification. The two procedures allocate calls differently: solving produces one answer per problem, whereas verification evaluates candidate variants individually. At the variant level (Table 1), both systems produce similar false-positive counts (15.3 vs. 15.7), while verification achieves higher recall (81.6% vs. 42.5%).

D.4. Debate Convergence and FP Ladder

E. Why Debate Breaks the Martingale

Choi et al. (2025) show under a stylized homogeneous-agent model that the expected vote tally can behave as a martingale: debate exchange need not add information beyond a single majority vote over the same number of calls. Our factorial clarifies which departures from that setting matter.

(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. Applied to identical heterogeneous trajectories, it reduces mean FP from 2.67 to 1.67, although it does not reduce FP for homogeneous trajectories.

(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. Different lenses are intended to reduce correlated error relative to agents sharing a lens. The factorial associates heterogeneity with the largest observed mean FP reduction: heterogeneity reduces mean FP from 6.00 to 1.67 with gating, and from 6.00 to 2.67 without gating. This is consistent with the prediction of Li et al. (2026) and the graph-theoretic view of self-consistency in Zhang et al. (2025).

F. External-Benchmark Transfer

For Omni-MATH and OlymMATH, we evaluate a generate-then-verify deployment setting. `gpt-5.2` (`effort="low"`) first solves each problem from scratch and outputs a candidate answer plus reasoning trace. The candidate is labeled against the benchmark reference answer, then passed to the unchanged debate pipeline without reference access. Each external problem contributes one candidate variant. Unlike the controlled main benchmark, these results use a single generation and a single debate run. Because generation and verification both use `gpt-5.2`, this protocol tests transfer across benchmarks

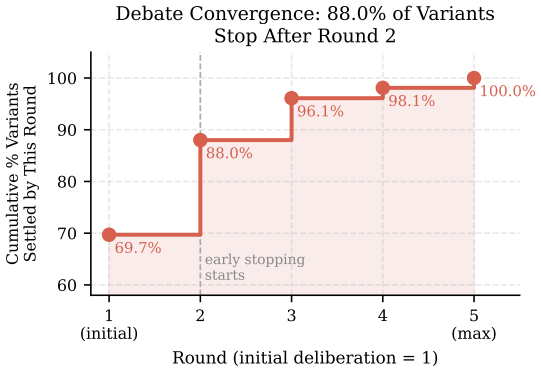

Figure 4. Cumulative debate convergence by exchange round. Across the three final runs, 88.0% of variants stop after round 2, keeping average cost at 10.9 calls/variant despite a 5-round maximum; only 1.9% require all five rounds.

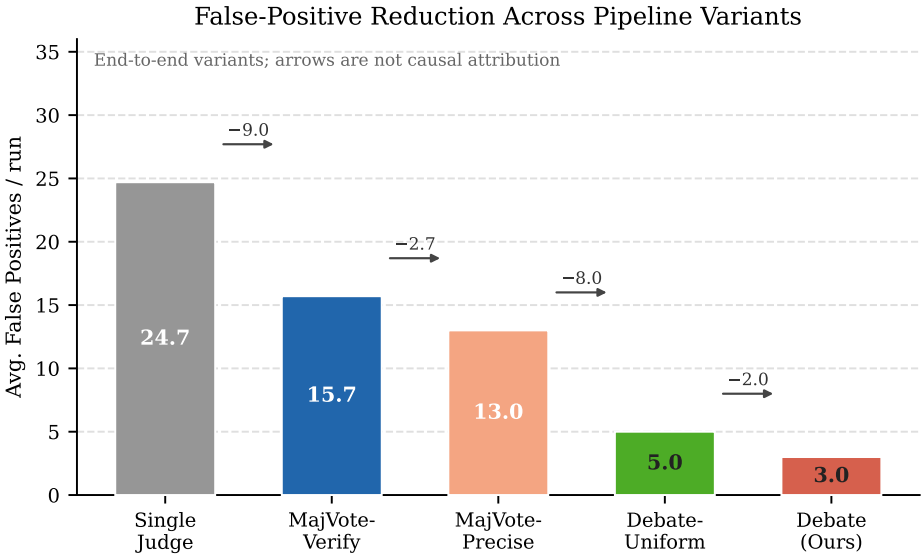

Figure 5. Step-by-step false-positive reduction across pipeline variants, shown as an end-to-end system ladder. Arrows show observed FP differences between adjacent variants, not causal mechanism attribution.

but not across candidate-generating models.

Omni-MATH. We extract all 187 Omni-MATH problems with difficulty greater than 8. The generator produces 116 reference-correct and 71 incorrect candidates. AEGIS-D rejects 63 incorrect candidates (true negatives), while all eight apparent false positives end with unanimous `answer_supported`. At least four of the eight apparent FPs involve evaluation artifacts: $0 < t \leq 4$ vs. $0 < t \leq 4$; an equivalent inequality with an added equality condition; a permutation-valued reference attached to a problem asking for constants; and a numeric reference attached to a construction task. The remaining four are substantive mathematical disagreements. Two are clear verifier failures: a missing special case for $p = 2$ in a finite-field polynomial problem, and an incorrect necklace extremal formula ($2n - 1$ instead of $n^2 - n - 1$). The other two concern the scope of a functional-equation classification and a tournament characterization, where all agents endorse an over-generalized or over-restrictive theorem statement without checking the global necessity conditions. Most false negatives are conservative: 51 of 56 end with a majority of agents at `reasoning_insufficient`, including 46 where all five agents are insufficient. Representative FN cases include long synthetic-geometry proofs, functional equation classifications, constructive sequences, and combinatorial existence proofs where the candidate answer is correct but the text-only verifier cannot obtain enough positive evidence to pass the gate.

The false-negative asymmetry is equally diagnostic. Of 56 correct traces rejected by AEGIS-D, 55 are labeled `wrong` and one is labeled `invalid`; 51 finish with a majority at `reasoning_insufficient`, and 46 finish with all five agents in

Table 8. Confusion matrices for single-run external-benchmark transfer. Omni-MATH uses unadjusted reference/equivalence labels. OlymMATH reports both the unmodified benchmark result and an audited sensitivity analysis.

Setting	TP	TN	FP	FN	Prec.	Recall	Acc.
Omni-MATH (d>8)	60	63	8	56	88.2%	51.7%	65.8%
OlymMATH en-hard (unmodified)	41	40	9	10	82.0%	80.4%	81.0%
OlymMATH en-hard (audited sensitivity)	47	40	3	10	94.0%	82.5%	87.0%

Table 9. Manual audit of all eight apparent Omni-MATH false positives. "Artifact" denotes an answer-equivalence or reference-schema issue rather than a demonstrated verifier error.

Problem	Audit class	Main issue
P99	Artifact	Equivalent $\leq$ versus $\leq$ answer strings were not matched.
P159	Artifact	The task asks for constants, but the attached reference is a permutation.
P167	Artifact	The candidate gives the reference inequality together with its equality condition.
P187	Artifact	A construction task is paired with a numeric reference.
P37	Verifier FP	The accepted classification misses the exceptional case $p = 2$.
P129	Verifier FP	The accepted necklace extremal formula is incorrect.
P141	Scope disagreement	Agents endorse a functional-equation classification without establishing completeness.
P162	Scope disagreement	Agents over-generalize a tournament characterization without checking the necessary global condition.

that state. Only five rejected correct traces contain any active `answer_refuted` assessment. Thus Omni-MATH's high FN count is primarily the intended conservative effect of the positive-evidence gate on globally difficult proofs, rather than widespread false refutation.

OlymMATH. We evaluate all 100 OlymMATH en-hard problems, balanced across algebra, combinatorics, geometry, and number theory. Before scoring, we manually audit apparent label disagreements. Four English prompts leave a mathematical convention under-specified (e.g., whether reflections are allowed or whether tuples are ordered); we rewrite these prompts to state the benchmark's intended convention. Two additional disagreements are resolved by manual mathematical checking. Table 8 reports both the unmodified benchmark result and the audited sensitivity analysis. We treat the audited row as a sensitivity analysis rather than an official score on the unmodified OlymMATH benchmark.

G. Full Benchmark Problem-Validity Evaluation

The fourth `assessment_type`, `problem_invalid`, records that the problem itself is malformed, contradictory, or under-specified. AEGIS-D outputs `invalid` when a majority reaches this assessment. The full benchmark evaluates answer correctness and statement validity jointly: 195 answer variants from valid problems plus 69 invalid problem statements. Because variants are evaluated independently, we construct this composite result by combining the three-label run on the original 195 variants with the complete run on all 69 invalid statements; no item is duplicated or omitted. For the answer variants, this uses the fixed trajectory from the full-benchmark evaluation rather than the three-run average in Table 1; the composite result is therefore a single-run diagnostic.

Table 10. Three-label confusion matrix on the full 264-variant benchmark. Rows are true labels and columns are pipeline predictions.

True label	N	Pred. correct	Pred. wrong	Pred. invalid
Correct	58	36	20	2
Wrong	137	1	134	2
Invalid	69	1	9	59

The full benchmark has 90.9% binary accept/reject accuracy and 86.7% three-label exact accuracy. Binary acceptance precision is 94.7%: among 38 accepted variants, 36 are truly correct, one is a wrong answer, and one is an invalid problem.

Invalid precision is 93.7% (59/63), and invalid recall is 85.5% (59/69). The most common invalid-statement error is conservative rejection as `wrong`, not dangerous acceptance.

The three-run direct validity comparison reveals a complementary role trade-off. The single-agent baseline achieves 94.9% precision and 90.3% recall; homogeneous debate improves recall to 92.2% and overall accuracy to 93.7%; heterogeneous debate instead raises precision to 97.2% while lowering recall to 84.1%. Thus, the role diversity that best protects answer acceptance is not automatically optimal for detecting every malformed statement.

Reasoning truncations in the factorial. All three heterogeneous-role runs complete all 264 variants. The homogeneous-role runs complete 259, 261, and 264 variants. An exact-prompt diagnostic on a representative missing trajectory returns `status=incomplete` with `incomplete_details.reason=max_output_tokens`: all 4,096 output tokens are consumed by hidden reasoning before visible output is produced. We record the eight persistent run-level failures as `reasoning_truncated`, assign no verdict, and exclude them from classification metrics.

H. Cross-Model Fixed-Prompt Replication

We repeat the main verification ladder with Claude Opus while keeping the dataset, prompts, role instructions, scheduler, verdict rules, and call budgets fixed; only the base model changes. The result JSON files do not serialize the model name, and no matching run log for this fixed-prompt ablation was retained. We therefore report the provider family rather than asserting an exact snapshot. Because these are single-run results and absolute calibration differs sharply from `gpt-5.2`, we use them to test directional mechanism transfer rather than to compare model quality.

Table 11. Claude Opus fixed-prompt replication on the original 195 variants. All rows are single runs.

System	Prec.	Recall	FP	FN	Acc.	Prob. Acc.
Single-Judge	30.2%	72.4%	97	16	42.1%	6.9%
MajVote-Precise (gated)	44.2%	79.3%	58	12	64.1%	17.2%
Debate-NoGate	53.9%	84.5%	42	9	73.9%	36.2%
Debate-Uniform	51.5%	91.4%	50	5	71.8%	31.0%
AEGIS-D (Ours)	**58.8%**	86.2%	**35**	8	**77.9%**	**41.4%**

Despite lower absolute precision, both structural comparisons preserve the direction observed with `gpt-5.2`: adding the assessment gate to heterogeneous debate reduces FP from 42 to 35, and adding heterogeneous roles to gated debate reduces FP from 50 to 35. AEGIS-D reduces the Claude Opus single judge's FP count from 97 to 35. This provides directional evidence across a second base model, but the remaining 35 false positives also show that architecture cannot substitute for base-model verification capability or calibration.

I. Per-Problem Analysis

Figure 6 visualizes the per-problem outcome across three runs for MajVote-Verify and AEGIS-D side by side. Table 12 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.

I.1. MajVote-Verify and AEGIS-D Problem-Level Correct Counts (out of 3 runs)

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

AEGIS-D-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). AEGIS-D'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: AEGIS-D achieves 0/3 despite no FP contamination,

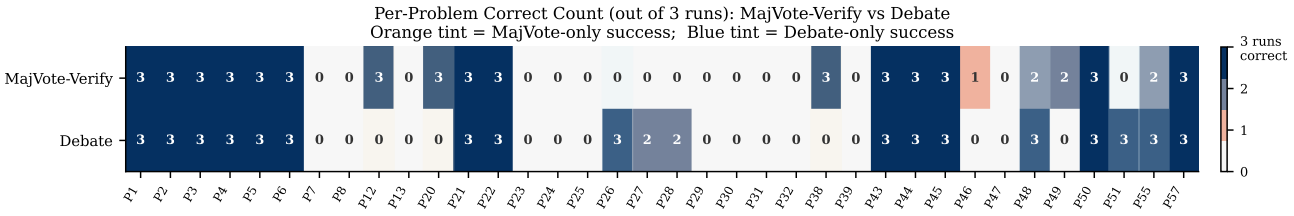

Figure 6. Per-problem correct count (0–3 runs) for MajVote-Verify (left) and AEGIS-D (right) on all 58 problems. **Orange** columns: MajVote-Verify succeeds but AEGIS-D fails (geometry problems where all agents produce `reasoning_insufficient`, triggering overcautious rejection). **Blue** columns: AEGIS-D 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 12. Per-problem correct count (0–3) for MajVote-Verify and AEGIS-D. A problem is "correct" only when the correct variant is accepted *and* all wrong variants are rejected in that run. Selected problems; the preceding heatmap shows the full data. Abbrev.: Dbt = AEGIS-D, 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

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 AEGIS-D. Even at the boundary of capability, AEGIS-D'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.

J. Dataset Example and Debate Case Study

J.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 1849×1849 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. The reference-equipped reasoning applies this formula to $N = 1849 = 43^2$, yielding $1849 + 86 - 3 = 1932$.

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 (43×43 factorization) is mathematically meaningful, and the error—an unverified formula—is not surfaced by surface-level checks.

J.2. Case Study: Exchange Overturns a 5–0 Initial Majority

We trace AEGIS-D on Problem 25, Variant 2 (answer $= 2012$, true label: *wrong*). This case illustrates how evidence-aware scheduling and exchange-round counterexamples overturn an initially unanimous but weakly supported stance.

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 flat vote that stopped after initial deliberation would accept this variant: 5–0 satisfies any supermajority threshold. Our protocol, however, always performs at least one exchange round, even after unanimous initial stances. The assessment types expose that the apparent consensus contains no positive support, allowing the scheduler to target every agent as a weak supporter.

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 4×4 board requires ≥ 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 ≥ 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)* **Evidence-aware diagnosis**: the assessment types distinguish a weak 5–0 stance from five positively supported judgments and enable targeted scheduler pressure. This case does not isolate the final assessment gate, because both gated and ungated debate execute the exchange round. *(ii)* **Exchange-driven falsification**: the scheduler correctly identifies weak-support agents and requires concrete checks, triggering the falsification that was absent from 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.