RESOLVE: REUSING CANDIDATE REASONING THROUGH SELECTIVE GENERATIVE MODERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Sampling multiple solutions spends computation on intermediate deductions and unfinished arguments as well as final answers. We introduce ReSolve, a training-free inference procedure that reuses this candidate reasoning through selective generative moderation. An answer-distribution controller invokes a model to examine existing derivations when candidates disagree or lack a parseable answer, then incorporates the generated solution into a bounded loop. Under Hybrid scoring on 130 competition-mathematics problems evaluated with two independently sampled candidate pools, ReSolve obtains 100 and 99 correct answers, compared with 91 and 92 for voting over the same four candidates, with no correct-to-incorrect changes relative to that vote in either pool. Eight-sample self-consistency obtains 94 and 96 correct answers while consuming substantially more tokens; ReSolve uses 46.3% and 47.2% fewer tokens in the two evaluations. A controlled ablation removes visible derivations while retaining answer keys, vote counts, and the per-state output-cap rule, reducing accuracy from 100 to 93 correct despite increasing computation. Selective and always-on Uniform moderation both solve 97 problems, while selectivity reduces moderation tokens by approximately 54% and total pipeline tokens by 6.2%. These results support candidate reasoning as reusable inference computation. They do not establish an accuracy advantage over additional sampling or a distinct benefit from specialized route instructions.

1 INTRODUCTION

Sampling several solutions is a common way to allocate additional inference computation to difficult reasoning problems. By the time four candidates have been generated, the model has already spent computation on their decompositions, intermediate identities, and unfinished arguments. Self-consistency aggregates extracted final answers by frequency (Wang et al., 2023), without directly using those arguments in its decision. A partial derivation contributes no vote if it never reaches an answer, and several candidates can repeat the same omission. Before spending more computation on independent attempts, can we make better use of the reasoning already available?

We study selective reuse of these existing derivations. A moderator receives the problem and candidate evidence, checks disputed deductions, and produces a solution that can extend or revise the candidates. A selector must return an existing response. A generative moderator can complete an unfinished argument or construct a corrected answer. Reuse still has a cost: the model must read the evidence and generate another response, and that response can replace a correct answer with an incorrect one. The allocation decision is therefore both what evidence to reuse and when another call is worth making.

ReSolve implements this idea as selective generative moderation. It samples four initial solutions, groups their parseable answers, and bypasses moderation when the current pool has a single answer group. Otherwise, the answer distribution selects an instruction to inspect a dominant answer, compare two groups, reconcile fragmented evidence, or complete unfinished attempts. The resulting response enters the pool, and an explicit acceptance rule either returns a solution or continues within a bounded number of calls. The same existing model proposes and moderates; no reference answers are available during inference.

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

The key distinction is between *reusing evidence* and *requesting another independent attempt*. We use shared candidates and count all input and output tokens. Under the primary Hybrid score, two independently sampled candidate pools on the same 130 competition-mathematics problems yield 100 and 99 correct answers for ReSolve, versus 91 and 92 for voting over the same four candidates. Moderation corrects nine and seven errors, respectively, with no correct-to-incorrect changes relative to that vote in either pool. Relative to eight-sample self-consistency, the observed counts are 100 versus 94 and 99 versus 96, with 46–47% fewer total tokens. A secondary paired analysis that groups the two evaluations by problem supports improvement over four-candidate voting, while the contrast with eight-candidate voting remains statistically inconclusive.

We then isolate what information and control decisions produce these observations. Removing visible candidate derivations lowers the first evaluation from 100 to 93 correct answers, although the answer-only moderator makes more calls and generates more tokens. Keeping a generic verification instruction and changing only whether agreement states are checked leaves accuracy at 97 correct while more than doubling moderation tokens. By contrast, route-specific instructions do not establish an advantage over a generic Uniform moderator. These findings place the empirical emphasis on *what evidence to reuse* and *when to invoke another model call*.

Our contributions are threefold.

1. **Candidate reasoning as reusable computation.** We formulate existing derivations as evidence for further inference and distinguish generative recovery from fixed-pool answer selection.
2. **Selective generative moderation.** We instantiate this formulation with an explicit, bounded procedure that reads candidate evidence when answers disagree or are unavailable. Its token accounting includes both evidence access and generation.
3. **Controlled evidence for reuse and selectivity.** We evaluate two candidate pools, remove visible derivations, and compare selective with always-on invocation. Recovery and preservation analyses expose both what moderation repairs and what it can lose.

The empirical scope is mathematical reasoning with a 4B backbone. Fresh candidate draws test sensitivity to the sampled pool; they do not by themselves establish generalization to new tasks or rule out benchmark familiarity.

2 ReSolve: selective generative moderation

2.1 From candidate answers to candidate evidence

For a problem x, a proposer p_ϕ generates K candidate solutions $C_0 = \{y_1, \ldots, y_K\}$. Let $a(y)$ extract and normalize a final answer, with $a(y) = \perp$ when extraction fails. The answer histogram is

$$h_C(v) = \sum_{y \in C} \mathbb{1}[a(y) = v], \qquad n_C = \sum_{v \neq \perp} h_C(v). \tag{1}$$

Self-consistency returns a representative of the most frequent valid answer. ReSolve additionally retains the reasoning behind those answers. Its evidence constructor $e(x, C)$ packages the problem, representative arguments, and the available answer-distribution information for a moderator q_θ.

The distinction between selecting and generating is precise. With an evaluation-only correctness function g_x, the fixed-pool selection oracle is

$$O_K(x) = \max_{y \in C_0} g_x(y). \tag{2}$$

Any procedure restricted to returning an element of C_0 is bounded by O_K. A moderator generates $z \sim q_\theta(\cdot \mid e(x, C))$ and can succeed when $O_K = 0$. Over a dataset, the difference between moderation and oracle accuracy decomposes as

$$\sum_x \big(g_x(z_x) - O_K(x)\big) = N_{\text{recover}} - N_{\text{lose}}, \tag{3}$$

where z_x denotes the system's final returned response, either an existing candidate or a new generation. A recovery solves an oracle-unsolved problem and a loss fails on an oracle-solvable one. This accounting identity makes both effects visible. The oracle is an evaluation control with reference-answer access; it is not a deployable system or an upper bound on additional generation.

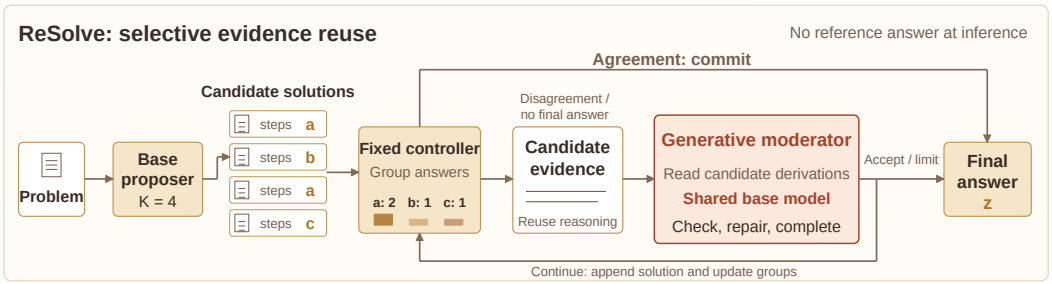

Figure 1: **Selective reuse of candidate reasoning.** Four sampled solutions provide both an answer distribution and visible derivations. Agreement permits a direct decision; otherwise, a generative moderator checks, repairs, or completes candidate reasoning. New solutions update the pool until acceptance or the call limit. Proposer and moderator share existing model weights, and reference answers are absent at inference.

State	Histogram condition	Intervention
No final answer	$n_C = 0$	Complete or repair unfinished attempts
Agreement	One valid answer group	Commit without a new call
Dominant answer	Top share $> \tau$	Audit the majority and competing evidence
Two groups	Two groups, no dominant answer	Adjudicate the competing arguments
Fragmented	More than two, no dominant answer	Reconcile useful deductions

Table 1: Controller states, evaluated in the displayed order. Shares use the number of *parseable* answers. In particular, agreement can occur with fewer than K parseable candidates. We use $\tau = 0.5$.

2.2 Selective intervention and bounded feedback

The controller partitions the current histogram into five states (Table 1). Agreement permits a direct decision. For disagreement, the moderator checks a majority argument, compares two answer groups, or reconciles more fragmented evidence. If no final answer can be extracted, unfinished attempts are supplied as weak evidence for repair. None of these inference operations receives the reference answer or correctness feedback.

The evidence view retains the first representative of each answer group and its multiplicity, ordered by frequency with candidate-order tie breaking. A later derivation supporting an existing key increases its count without replacing that representative. A new key can introduce a new representative. Rendering prefers the post-`</think>` derivation when nonempty and otherwise uses the original response, within the same evidence limits across compared methods. No-answer repair instead reads a bounded prefix of the current attempts (Appendix A).

Feedback updates answer support and selected evidence. Generated responses join the pool, but later calls do not necessarily see every new derivation. For example, consider a pool with answer counts $\{a : 3, b : 1\}$. If moderation returns b, the pool becomes $\{a : 3, b : 2\}$ and still calls for a majority audit. A second parseable moderator response of b meets the stability criterion and permits an overturn; a response of a instead confirms the leading answer. In two-group or fragmented states, selecting an existing answer permits immediate acceptance, while a new answer requires continued processing. These are deterministic acceptance conditions on answer keys, not certificates of mathematical correctness. An unparseable response does not reset that sequence; its text is eligible for no-answer repair but is not an answer-group representative.

After each moderation call, its answer is appended to the pool and the histogram is recomputed. State-specific acceptance and a stability check determine whether to return a solution. The procedure allows at most $R = 3$ moderation calls and uses a stability patience of two. At the round limit it applies the deterministic answer-selection fallback. The full acceptance rules and extraction conventions are given in Appendix A. This separation between *whether to invoke*, *what to ask*, and *when to stop* allows each component to be examined experimentally.

2.3 A FIXED MODEL AND AN EXPLICIT INFERENCE POLICY

The proposer and moderator share the same Qwen3-4B-Thinking-2507 weights (Yang et al., 2025). No parameter update is required. The moderator receives the problem and the rendered candidate evidence; its generated response enters the same answer-extraction and grouping pipeline as the original proposals. Reference answers and evaluator feedback are absent from this process.

2.4 ACCOUNTING FOR INFERENCE COMPUTATION

For each problem we count all proposal and moderation input and output tokens:

$$T(x) = \sum_{i=1}^{K} \left(T_i^{\text{in}} + T_i^{\text{out}}\right) + \sum_{t=1}^{R_x} \left(T_{m,t}^{\text{in}} + T_{m,t}^{\text{out}}\right). \tag{4}$$

The first sum is shared by four-candidate voting and moderation: both have already generated the same solutions. The second sum measures the additional cost of reading and extending that evidence, and vanishes when the controller bypasses moderation. Eight-sample self-consistency instead generates four more independent solutions. We report the full token consumption of each choice, including reading candidate evidence. Token counts characterize inference workload; they do not assert equivalent latency or FLOPs for arbitrary model and serving configurations.

3 EXPERIMENTS

3.1 EVALUATION DESIGN

Two candidate pools on the same problems. The principal evaluation contains AIME24, AIME25, and HMMT25 (30 problems each), plus CMIMC25 (40), for 130 distinct problems. F2 and F3 denote two independently sampled candidate pools for these same problems. Each evaluation generates eight Qwen3-4B solutions per problem. SC@4, Oracle@4, ReSolve, and Uniform use the first four; SC@8 and Oracle@8 use all eight. The additional four candidates are hidden from the moderators. F3 keeps the questions, frozen inference rules, and grading code unchanged. Its batched serving implementation was checked for equivalent per-problem controller behavior. These are repetitions of candidate sampling on previously studied questions, not additional independent test problems.

Controls and generation. *Uniform* uses a general verification instruction with the same visible evidence, invocation gate, acceptance conditions, and stopping controller as ReSolve. *Answer-only* removes the visible derivations while retaining answer keys and counts. *Direct32* solves independently without candidate evidence or voting fallback. Proposal temperature/top-p are 0.7/0.9; moderation and Direct32 use 0.1/1.0. Calls share a 32,768-token context window, with output capacity equal to the remaining context after the prompt and a 32-token margin. Answer-only uses the capacity computed from its state's Full rendering to isolate evidence removal. We retain historical same-pool studies H1/D1 for supporting mechanism analyses, with their separate 16K output budget documented in the appendix.

Scoring and paired analysis. We use the prespecified Hybrid score, which accepts an answer when either the frozen rule grader or CompassVerifier-3B (CV) accepts it (OpenCompass, 2025). CV is a sensitivity analysis. A scoring audit identified an equivalent-form false negative in the CV-only SC@4 and SC@8 scores, reinforcing the need to distinguish the two views (Section B.1). Within each candidate pool, we report exact two-sided McNemar tests. Holm correction covers the two prespecified contrasts, ReSolve versus SC@8 and versus Uniform. The same-pool SC@4 comparison is a secondary analysis. Across F2 and F3, the problem is the statistical unit: the reported sign-flip analysis groups its two paired outcomes rather than treating 260 outcomes as independent problems. Appendix B separates these tests and their scope.

3.2 REUSING FOUR CANDIDATES VERSUS SAMPLING EIGHT

Improvement over the same candidate vote. Table 2 reports Hybrid results for both independently sampled pools. ReSolve improves SC@4 from 91 to 100 correct in F2 and from 92 to 99 in

Method	F2 correct	F2 (%)	F3 correct	F3 (%)
SC@4	91	70.00	92	70.77
SC@8	94	72.31	96	73.85
Direct32	82	63.08	80	61.54
Uniform	97	74.62	99	76.15
ReSolve	100	76.92	99	76.15
Oracle@4	99	76.15	100	76.92
Oracle@8	100	76.92	108	83.08

Table 2: Hybrid results on two independent candidate pools for the same 130 problems. ReSolve and Uniform access only the first four candidates. Oracle rows use evaluation reference answers. The evaluations are repeated observations of the same questions.

F3: gains of 6.92 and 5.38 percentage points. The paired changes are nine fixes with no breaks and seven fixes with no breaks, respectively. In F2, 74 of the 91 SC@4-correct problems bypass moderation; all 17 that are actually moderated remain correct. Across both pools, 16 fixes and no breaks give a mean gain of 6.15 percentage points. These are repeated observations of 130 problems, not 260 distinct questions.

The secondary paired analysis gives within-pool $p = 0.0039$ and $p = 0.0156$, and combined problem-level $p = 0.0003$ (Appendix B). F2 moderation adds only 6.4% to the token cost of the shared four proposals. This comparison asks what reading and extending already-generated reasoning adds beyond its final-answer vote.

A different allocation of the inference budget. SC@8 solves 94 problems in F2 and 96 in F3, compared with 100 and 99 for ReSolve. The F2 totals remain 100 versus 94 under the rule grader alone (Table 8). F2 has seven wins and one loss ($p = 0.0703$, Holm $p = 0.1406$); F3 has seven wins and four losses ($p = 0.5488$, Holm $p = 1$). The combined gain is 3.46 percentage points with problem-level $p = 0.093$. The observed accuracy ordering repeats, but its margin narrows and does not establish superiority over additional sampling. Nor does a nonsignificant difference establish equivalence.

The token comparison is more consistent. ReSolve uses 12.46M versus 23.19M tokens in F2 and 12.35M versus 23.38M in F3: reductions of 46.3% and 47.2% (Figure 2). For F2, the shared four proposals consume 11.711M tokens; moderation adds 0.753M, including 0.195M input tokens for reading evidence and 0.558M output tokens. The next four independent proposals consume 11.483M. The cost advantage therefore includes the price of accessing previous reasoning rather than treating evidence as free input. F3 makes 47 moderation calls, compared with 51 in F2, with no call reaching its output cap. SC@8 adds samples on every problem, whereas moderation is selective; this comparison changes both which problems receive computation and what computation they receive.

What the instruction controls establish. Uniform solves 97 problems in F2 and 99 in F3. The ReSolve–Uniform paired outcomes are 4/1 and 2/2 wins/losses; combined problem-level $p = 0.51$. Specialized route instructions therefore do not establish an improvement over generic candidate-conditioned verification. Direct32 solves 82 and 80 problems, but uses a different budget; its lower F2 cost of 2.88M tokens prevents interpreting its accuracy gap as a matched-cost comparison. The central evidence concerns reuse versus voting and the observed tradeoff against further sampling.

3.3 Does the visible candidate reasoning matter?

A controlled removal of derivations. Answer-only retains the problem, group headers, extracted answers, vote counts, route instructions, decoding settings, and controller. It deletes the rendered derivations from both original and subsequently generated candidates. Each call's cap is computed from the Full rendering of that same state. Shorter input consequently does not grant extra output capacity. All 44 initially triggered problems match the original Full prompt and cap before removal; the 86 agreement outputs are reused unchanged. The intervention concerns visible written solutions, not hidden thinking traces.

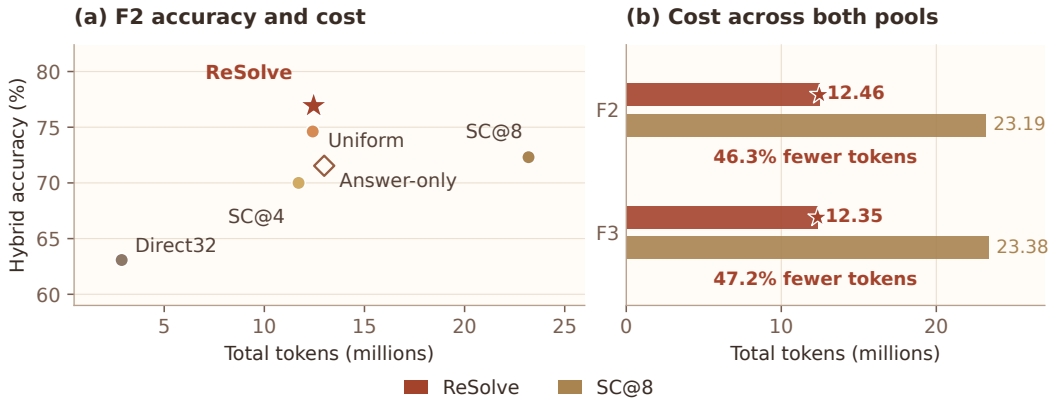

Figure 2: **Accuracy and token cost with Hybrid scoring.** (a) F2 controls, including the answer-only intervention; the star marks ReSolve. (b) Whole-pipeline token cost for ReSolve and SC@8 on both independent candidate pools. Each method's total includes the proposals it uses and all moderation input/output tokens.

F2 configuration	Correct / 130	Calls	Moderation (M)
ReSolve (Full)	100	51	0.753
Answer-only	93	65	1.276
Uniform loop, selective	97	50	0.706
Uniform loop, always-on	97	136	1.522
Uniform one-call, selective	97	44	0.619
Uniform one-call, always-on	97	130	1.435

Table 3: F2 controls under Hybrid. Moderation tokens include inputs and outputs; all rows share an additional 11.711M proposal tokens. Full versus Answer-only removes derivations under the same per-state output-cap rule. Selective versus always-on Uniform adds agreement checks, shared across the two Uniform variants.

Full solves 100 problems versus 93 for Answer-only, with seven fixes and zero breaks ($p = 0.0156$). Four fixes have parseable candidate answers: two dominant-answer states, one two-group state, and one fragmented state. The remaining three have no extracted initial answer. The benefit therefore extends beyond completing unfinished attempts. In the four disagreement cases, the answer-only moderator ends on an existing incorrect answer group. Full evidence can support either preserving the correct group or overturning an incorrect majority.

Answer-only makes 65 calls versus 51 for Full and generates 1.257M versus 0.558M moderator tokens. Its total pipeline cost is approximately 12.99M rather than 12.46M, despite reducing moderation input from 0.195M to 0.019M tokens. Twenty answer-only calls reach the output cap, compared with one Full call. These observations are consistent with existing derivations reducing the work needed to reach a solution within the tested budget. They do not prove the same gain under a different cap policy. In particular, allowing the shorter answer-only prompt to spend its saved context on longer generation remains a separate control.

3.4 Does selective invocation save computation?

The final-protocol F2 control adds one Uniform verification call to each of the 86 agreement problems and reuses the existing outputs for the triggered problems. A parseable verification answer replaces the previous response; otherwise the original response is retained. This isolates the effect of checking the states normally skipped by the gate.

Selective and always-on Uniform both solve 97 problems, with no correctness changes across the 86 added checks. Eighty-five checks return parseable answers and one changes the extracted answer without changing correctness. The closed-loop comparison uses 50 versus 136 total calls and

F2 initial state	N	vs. SC@4	vs. SC@8	vs. Uniform
Agreement	86	0/0	0/0	0/0
Dominant answer	18	2/0	2/0	3/0
Two groups	8	1/0	0/0	0/0
Fragmented	12	3/0	2/1	1/1
No answer	6	3/0	3/0	0/0

Table 4: F2 fixes/breaks by initial state under Hybrid. Comparators isolate same-pool voting, additional proposals, and the moderation instruction. Small strata are descriptive.

0.706M versus 1.522M moderation tokens. Selectivity saves approximately 54% of moderation tokens; the total pipeline reduction is approximately 6.2% because the shared proposals dominate cost. A one-call version reaches the same 97 correct with 44 versus 130 calls and 0.619M versus 1.435M moderation tokens (Table 3).

This control supports the agreement bypass on this pool, not the proposition that agreement guarantees correctness. Twelve of the 86 agreement states are wrong, and additional Uniform checking does not repair them. Selectivity avoids work that did not improve outcomes in the observed control. A different verifier or a targeted disagreement-inducing intervention could behave differently; such interventions are not evaluated here.

3.5 Where gains and preservation failures occur

Under Hybrid scoring, F2's nine fixes over SC@4 arise from dominant-answer (two), two-group (one), fragmented (three), and no-answer states (three), with no breaks. Against SC@8, the net improvements are two dominant-answer, one fragmented, and three no-answer problems. Against Uniform, all of the net gain comes from dominant-answer states. The route decomposition identifies where the observed improvements occur; it does not establish a separate advantage for each instruction.

Fixed-pool oracle accounting exposes a complementary failure mode. In F2, ReSolve recovers three no-answer problems and loses two oracle-solvable problems. These recoveries show completion of unfinished attempts, not repair of four completed but incorrect answers. In F3, it recovers two and loses three. The corresponding totals are 100 versus Oracle@4's 99, and 99 versus Oracle@4's 100. Generative recovery is possible in both pools, but exceeding the oracle in aggregate is not a stable result. In F3, three of the four losses to SC@8 occur in no-answer states, where the additional proposals can supply a valid answer unavailable to the moderator. Generating more candidates and completing existing reasoning address different failure modes.

Historical controls are consistent with this interpretation. On H1, ReSolve solves 127/160 compared with 119 for SC@4, with four recoveries and three preservation losses relative to the Hybrid oracle. In D1, one Uniform call with evidence solves 126 versus 119 for a fresh solve on the same triggered problems under the same 16K output cap. The earlier comparison also changes prompt wording; the F2 answer-only intervention more directly isolates visible derivations. H1's Uniform and operation-permutation controls remain within two correct answers of ReSolve (Appendix E).

3.6 A case of completing candidate evidence

Table 5 illustrates a quartic-root problem from the historical pool. None of the four candidates has an extractable final answer, but one contains a useful sum-of-squares identity. The moderator verifies that identity and completes the complex quadratic, obtaining 322. Uniform also succeeds, so the example explains evidence reuse rather than instruction superiority. The identity and root were checked independently. Absence of a selectable answer does not imply absence of useful intermediate computation; Appendix F gives the provenance and a separate majority-error example.

3.7 Token savings and sequential latency

The F2 timing audit records no moderation time on agreement problems. Among 44 triggered problems, the added moderation time has median 108 seconds and 90th percentile 304 seconds;

Table 5: **Completing candidate reasoning on CMIMC 2025.** The initial pool has no extractable final answer, but contains an identity that supports a correct completion. All descriptions are edited summaries, not verbatim model quotations.

Example: a quartic root (CMIMC25, cache index 7). Problem: Let $P(x) = x^4 + 20x^3 + 29x^2 - 666x + 2025 > 0$ for real x. A first-quadrant root has the form $r = (a + bi + \sqrt{c + di})/2$, with integer a, b, c, d. Find $a + b + c + d$. **Reference answer: 322.**

Initial candidate pool / SC@4	**ReSolve**
All four attempts lack an extractable final answer, so answer voting has no valid answer to select. However, the fourth attempt already contains $P(x) = (x^2 + 10x - 36)^2 + (x + 27)^2$. *This useful intermediate identity remains visible in the moderator's actual input.*	Checks the decomposition and completes the complex quadratic: $x^2 + (10 - i)x - (36 + 27i) = 0$. The discriminant is $243 + 88i$, giving $r = \dfrac{-10 + i + \sqrt{243 + 88i}}{2}$. Thus $-10 + 1 + 243 + 88 = 322$.
Extracted answers: $[\perp, \perp, \perp, \perp]$	**One moderation call.** Answer: $\boxed{322}$ ✓

Control: Uniform also returns 322 in one call. This case illustrates evidence reuse and completion, rather than superiority over Uniform or a claim that the initial reasoning was entirely wrong.

averaged across the full panel, the increment is 52 seconds per problem. Thirty-seven triggered problems use one round and seven use two. Sequential checking can therefore remain costly even when it produces fewer tokens than another batch of independent candidates.

The audit's recorded end-to-end ratio to SC@8 is 1.18 at the median and 1.41 at the 90th percentile on triggered problems. These are measurements of the logged serving configuration, which uses eager execution and single-sequence moderation. GPU sharing was not recorded, and generating only four proposals was not timed separately. The ratios are not a controlled estimate of a production four-proposal pipeline. We report token consumption as workload and retain latency as a deployment constraint, without translating the 46–47% token reduction into a corresponding speedup.

What the controls establish. Under Hybrid scoring across two independently sampled pools, moderation improves on voting over the same four candidates with no observed regressions. The F2 answer-only intervention supports a contribution from visible derivations beyond answer counts alone. The F2 always-on control shows that checking agreement states adds computation without observed benefit. Together, these controls support reusing already-generated reasoning selectively. They do not establish that further independent sampling is less accurate, that the current route-specific instructions are necessary, or that fewer tokens imply lower latency.

4 Related Work

Sampling and adaptive computation. Chain-of-thought prompting elicits intermediate reasoning (Wei et al., 2022); self-consistency aggregates sampled answers (Wang et al., 2023). Adaptive-Consistency adjusts sample counts using agreement (Aggarwal et al., 2023), while test-time compute allocation studies search and refinement as a function of difficulty (Snell et al., 2024). Adaptive sampling and selective moderation both allocate further computation according to intermediate evidence. They differ in whether that computation generates independent attempts or inspects existing derivations.

Candidate selection and generative fusion. Universal Self-Consistency uses an LLM to select the most consistent candidate (Chen et al., 2023). LLM-Blender combines pairwise ranking with generative fusion of selected outputs (Jiang et al., 2023). Thus, both model-based selection and generation from candidates are established mechanisms. ReSolve studies what visible derivations

and selective invocation add with shared candidates and fixed model weights. It also measures recovery and preservation failures relative to a shared fixed-pool selection oracle.

Verification and feedback. Process verification scores intermediate steps (Lightman et al., 2024); FOBAR reconstructs a masked problem value from a candidate answer (Jiang et al., 2024). Self-Refine revises an output through feedback (Madaan et al., 2023). Multiagent debate and reconciliation exchange arguments (Du et al., 2024; Liang et al., 2024; Chen et al., 2024), and Mixture-of-Agents conditions later generations on preceding outputs (Wang et al., 2024). Our controls separate derivation access, invocation, and instruction choice within a shared candidate pool; Uniform's competitive results make these distinctions essential.

Search over reasoning states. Tree of Thoughts evaluates and backtracks over intermediate thoughts (Yao et al., 2023); Graph of Thoughts permits richer dependencies (Besta et al., 2024). ReSolve begins with independent responses and uses their answer histogram to control bounded interventions. It neither maintains an explicit proof-state search structure nor assumes a step-wise reward model. No direct performance comparison with these search methods is claimed.

5 Discussion: practical interpretation

The experiments suggest a simple allocation principle: once several expensive reasoning trajectories already exist, additional compute need not be spent only on independent resampling. When candidates disagree or lack a final answer, reading and extending existing derivations can recover useful work that voting discards. The answer-only control isolates the contribution of visible derivations under the shared output-cap rule, while the always-on control shows where additional verification was unnecessary on the evaluated pool. The comparison with SC@8 also reveals complementary failure modes: further samples can supply a missing answer, while moderation can complete or correct reasoning already available.

For inference design, this motivates treating candidate generation and subsequent processing as separate decisions. A system can first inspect agreement, then read selected derivations when another call is warranted, while charging both reading and generation to its budget. ReSolve instantiates this pattern with the same existing model as proposer and moderator. The competitive Uniform control makes the practical lesson useful beyond one prompt family: candidate evidence and selective invocation are design choices that can be examined independently of specialized instructions. In this setting, already-generated reasoning is a resource for further computation, including when it has not yet produced a selectable answer.

6 Conclusion

Generating four solutions has already incurred the cost of their reasoning. ReSolve reuses that evidence and continues only when candidates disagree or lack an answer. Across two candidate pools, this improves four-candidate voting with no observed regressions and uses 46–47% fewer tokens than eight-sample self-consistency, without establishing an accuracy advantage over the latter. Evidence removal and always-on checking support both reuse and selectivity under the tested protocol. The principal evaluation uses one 4B backbone and two candidate pools on the same 130 problems; Appendix H details the remaining scope and limitations.

AI use statement

Generative AI tools assisted with experiment planning, evaluation-script implementation and checking, organization of result summaries, and drafting and editing. Generated prose is not treated as experimental evidence. Numerical claims are linked to the archived evaluations and reports described in the appendix. The authors are responsible for reviewing and verifying the submitted content.

Reproducibility statement

The appendix specifies extraction, grouping, evidence rendering, acceptance, fallback, sampling parameters, and token accounting. F2 and F3 use independent candidate pools on the same 130 problems, and joint statistical analyses group outcomes by problem. The evidence-removal and invocation controls reuse their specified F2 outputs. Historical controls have a separate budget and are not pooled into a cross-protocol mean. Both Hybrid and CV scoring views are reported, with paired counts distinguished from marginal accuracy. The evidence-provenance section identifies which results were recomputed locally and which rely on completed experiment reports.

References

Pranjal Aggarwal, Aman Madaan, Yiming Yang, and Mausam. Let's sample step by step: Adaptive-Consistency for efficient reasoning and coding with LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12375–12396, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.761. URL https://aclanthology.org/2023.emnlp-main.761/.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17682–17690, 2024. doi: 10.1609/aaai.v38i16.29720. URL https://ojs.aaai.org/index.php/AAAI/article/view/29720.

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

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. Universal self-consistency for large language model generation. *arXiv preprint arXiv:2311.17311*, 2023. URL https://arxiv.org/abs/2311.17311.

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

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. LLM-Blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14165–14178, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.792. URL https://aclanthology.org/2023.acl-long.792/.

Weisen Jiang, Han Shi, Longhui Yu, Zhengying Liu, Yu Zhang, Zhenguo Li, and James Kwok. Forward-backward reasoning in large language models for mathematical verification. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 6647–6661, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.397. URL https://aclanthology.org/2024.findings-acl.397/.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 17889–17904, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.992. URL https://aclanthology.org/2024.emnlp-main.992/.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/aca97732e30bcf1303bc22ac3924fd16-Abstract-Conference.html.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-Refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46534–46594, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/91edff07232fb1b55a505a9e9f6c0ff3-Abstract-Conference.html.

OpenCompass. CompassVerifier-3B. https://huggingface.co/opencompass/CompassVerifier-3B, 2025.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024. URL https://arxiv.org/abs/2408.03314.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. Mixture-of-agents enhances large language model capabilities. *arXiv preprint arXiv:2406.04692*, 2024. URL https://arxiv.org/abs/2406.04692.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022. URL https://papers.neurips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 11809–11822, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/271db9922b8d1f4dd7aaef84ed5ac703-Abstract-Conference.html.

A IMPLEMENTATION AND INFERENCE PROTOCOLS

Answer extraction and grouping. The implementation first attempts to extract the last boxed answer. If none is available, it uses conservative short-answer patterns within the last 1,200 characters. Normalization removes presentation-level LaTeX delimiters and spacing, outer braces, and trailing periods, and lowercases the result. This grouping rule is not a symbolic-equivalence oracle. It is used consistently for invocation and voting, independently of the evaluation grader.

The controller first handles zero parseable answers, then a single answer group, then a strict majority among parseable answers, then a two-group split, and finally fragmented disagreement. For the fresh training-free protocol, evidence exposes the first representative of each answer group, its key, and its count, with at most six groups (two in the two-group state). Repeated support for an existing key changes its count without replacing the first representative; a new key can introduce its generated response as a representative. Unparseable responses are not shown when rendering answer groups. With no parseable answer, repair instead exposes up to the first six attempts in the current pool, including appended moderator attempts when they fit. Thus, feedback can update answer counts without exposing a revised argument. A representative uses at most 6,000 characters; long text retains a short head and an approximately 75% tail with an explicit omitted-middle marker. The candidate block is capped at 36,000 characters and the rendered prompt at 48,000. A nonempty post-`</think>` derivation is preferred when available. These are evidence-construction limits; the fresh protocol applies no additional token-context clipping to an already-rendered moderator prompt.

Acceptance and fallback. At the beginning of a round, a single answer group commits without a new call. Otherwise the response is appended. A majority audit commits if it returns the current leading key; a two-group or fragmented intervention commits if it returns a key already present before that call. A generated key can also commit after appearing twice consecutively among parseable moderator responses; an unparseable response does not reset the stored streak. At the call limit, the fallback selects the most frequent parseable key and its latest supporting response. Ties follow the first encountered group order. If no answer can be extracted, the fallback is the first original response. The stored convergence label for consensus can occur after a call; invocation counts are therefore taken from call records rather than inferred from that label.

Fresh-run sampling. The proposer and moderator are Qwen3-4B-Thinking-2507, served with vLLM (Kwon et al., 2023). Each problem receives eight separately sampled proposals. The first four proposals form the shared moderation pool; the additional four are used only by the eight-sample controls. Each call has output cap $32768 - T^{\text{in}} - 32$, with at least 1,024 available output tokens required for a moderation call. Proposal temperature/top-p are 0.7/0.9; moderation uses 0.1/1.0. F2 and F3 use these candidate-compaction and output-cap rules throughout. No forced-close rescue or reference-answer feedback is used in either pool.

Uniform moderation. Uniform shares the evidence, including counts, as well as the invocation and stopping rules. It replaces the specialized majority, two-group, and fragmented-disagreement instructions with a common instruction to verify and repair. The no-answer state retains the repair operation. Thus, it isolates an instruction intervention within the controller; it is neither count-blind nor an always-on verifier.

Historical studies. H1/D1 share a recorded 160-problem pool, adding 30 BRUMO25 questions, with a 16,384-token moderator output cap. D1 reuses relevant outputs and adds 102 agreement checks and 58 independent solves. Overlapping results are not independent replications, and these costs are not pooled with the fresh protocol.

F3 execution. F3 resamples the eight candidates for every problem while retaining the F2 problem texts and frozen inference, utility, and grading code. Candidate requests are batched across problems, and independent per-problem controller threads submit calls through a batching proxy. The final batched implementation was validated against sequential execution for equivalent per-problem controller behavior; debugging runs are excluded from the reported results. This execution change should not be interpreted as a controlled latency comparison with F2.

B SCORES AND PAIRED COMPARISONS

Method	AIME24	AIME25	CMIMC25	HMMT25	Total
SC@4	26/26	24/24	25/24	16/16	91/90
Oracle@4	27/27	25/25	27/26	20/20	99/98
SC@8	27/27	25/25	26/25	16/16	94/93
Oracle@8	27/27	25/25	28/27	20/20	100/99
ReSolve	27/27	25/25	27/27	21/21	100/100
Uniform	27/27	26/26	25/25	19/19	97/97
Direct32	23/23	24/24	20/19	15/15	82/81

Table 6: F2 per-task correct counts as Hybrid/CV. Task sizes are 30, 30, 40, and 30. Hybrid and CV score the same outputs; the main text uses Hybrid.

Pool	Score	Comparator	Fix/break	Exact p	Holm p
F2	Hybrid	SC@8	7/1	0.07031	0.14062
F2	Hybrid	Uniform	4/1	0.37500	0.37500
F2	Hybrid	SC@4	9/0	0.00391	—
F3	Hybrid	SC@8	7/4	0.54883	1.00000
F3	Hybrid	Uniform	2/2	1.00000	1.00000
F3	Hybrid	SC@4	7/0	0.01562	—
F2	CV	SC@8	8/1	0.03906	0.07812
F2	CV	Uniform	4/1	0.37500	0.37500
F2	CV	SC@4	10/0	0.00195	—
F3	CV	SC@8	7/4	0.54883	1.00000
F3	CV	Uniform	2/2	1.00000	1.00000
F3	CV	SC@4	7/1	0.07031	—

Table 7: Within-pool tests for ReSolve. Holm correction covers SC@8 and Uniform within each pool and score. SC@4 is a secondary comparison. Combined problem-level tests use a different analysis described in the text.

Hybrid is the main scoring view in this manuscript, consistent with the original evaluation protocols. CV supplies a sensitivity analysis of the same outputs. Under Hybrid, F2/F3 totals are 100/99 for ReSolve, 94/96 for SC@8, and 97/99 for Uniform. Under CV, they are 100/97, 93/94, and 97/97. These are alternative scores of shared responses, not separate generations.

Within each pool, exact two-sided McNemar tests use fixes and breaks. The prespecified two-comparison family comprises SC@8 and Uniform; Holm correction is performed within each pool and scoring policy. The SC@4 comparison is secondary and was added after the initial evaluation. Its unadjusted p-values should not be read as prespecified evidence. Treating it as a third member of the F2 Hybrid Holm family would give $p = 0.0117$, without changing the two original adjusted values.

The combined F2/F3 analysis groups repeated outcomes by problem and reports a sign-flip test of the average paired difference. It has 130 problem units with two candidate pools each. For SC@4 the Hybrid count difference is $16/260 = 6.15$ percentage points ($p = 0.0003$); for SC@8 it is $9/260 = 3.46$ points ($p = 0.093$); for Uniform it is $3/260 = 1.15$ points ($p = 0.51$). The count denominator summarizes observations, whereas dependence is handled at the problem level in the test. These combined tests are reported from the completed experiment analysis; they are not obtained by applying McNemar to pooled outcomes as if every observation were independent. No adjustment across all exploratory analyses is claimed.

B.1 SCORER DISAGREEMENTS AND CLIPPING

In CMIMC25 index 16, SC@4 and SC@8 return $2271049/129600$, equivalent to the reference $1507^2/360^2$. The rule accepts the answer and CV rejects it. Thus F2's main SC@8 comparison is 100 versus 94 with 7/1 paired changes; the CV-only 100 versus 93 with 8/1 includes an equivalent-

Method	F2 rule	F2 CV	F2 Hybrid	F3 CV	F3 Hybrid
ReSolve	100	100	100	97	99
Uniform	97	97	97	97	99
SC@8	94	93	94	94	96
SC@4	91	90	91	91	92
Direct32	81	81	82	79	80

Table 8: Scoring sensitivity on shared outputs. F2 rule values are from the audit; F2 Hybrid/CV values are reconciled to archived outcomes. Provisional manual judgments do not replace recorded scores.

form false negative. Answer-only exhibits the same extra CV discrepancy. We retain each scoring policy's recorded result and do not count this disagreement as an additional mathematical repair.

F2 deduplicates candidate and returned-response texts into 1,258 unique scoring keys. The audit reports 15 rule/CV disagreements, including equivalent-form judgments and incomplete responses. The F2 ReSolve and Uniform outputs have no rule/CV disagreements. The report's provisional manual adjudications are not used as a third primary benchmark. This distinction avoids treating an analyst's judgment as a fully validated grading protocol.

To fit the verifier context, 291 scoring texts are clipped; the rule grader uses the full response. A replay examines every discordant output in the reported comparisons under both scoring views, covering 28 problems and 139 scoring inputs. Replayed flags match the recorded clipping flags. No clipping operation removes an existing final answer, and no ReSolve output in the SC@8 or Uniform comparison is clipped. Five extracted-answer sensitivity verdicts on the affected main-comparison questions agree with the original labels. These checks address answer loss under clipping without certifying the correctness of every retained derivation or verifier decision.

CV contrast	Discordant problems	Clipped inputs	Final answers lost
ReSolve vs. SC@8	9	2	0
ReSolve vs. Uniform	5	0	0
ReSolve vs. Answer-only	8	2	0
ReSolve vs. Direct32	23	17	0
ReSolve vs. Oracle@4	6	17	0
ReSolve vs. Oracle@8	7	39	0

Table 9: Reported replay of CV clipping on discordant F2 comparisons. Oracle comparisons inspect candidate scoring inputs as well as moderator outputs, so input counts can exceed problem counts. Rows overlap and must not be summed. No existing final answer is removed; clipped Answer-only outputs already lacked final answers before clipping.

C CONTROLLED EVIDENCE AND INVOCATION INTERVENTIONS

The answer-only intervention reuses the F2 questions and first four candidates. Each retained proposal block is reduced to its header, final answer, and vote count; no-answer blocks retain their attempt header and unparsed-answer marker. The same restriction applies to later moderator-generated candidates. The Full representation supplies the visible post-`</think>` written derivation where available, under the original renderer limits.

At each state, the output cap is $32768 - T^{\text{in}}_{\text{Full,state}} - 32$. The counterfactual Full prompt is tokenized but not sent to the model. After the two trajectories diverge, this matches the same state's Full representation, not the original run's later states. The controller, sampling settings, maximum of three calls, and patience of two are unchanged. First-round prompt hashes, token counts, and caps match the original Full run on all 44 triggered problems. Regrading those Full outputs reproduces all 44 original labels under both Hybrid and CV. The follow-up uses two single-sequence engines per GPU with cache utilization 0.42 rather than 0.88; it reports no other engine-setting change.

The seven Hybrid improvements over Answer-only are AIME24 index 25, CMIMC25 index 15, HMMT25 indices 15 and 5, AIME25 index 27, HMMT25 index 28, and CMIMC25 index 4. Four have parseable candidate answers; three have none. Full has no Hybrid loss. Under CV there is one additional recorded improvement, the equivalent-answer discrepancy described above. Of the 44 triggered answer-only problems, 29 use one call, nine use two, and six use three, yielding 65 calls. Twenty calls reach the output cap, including 15 on no-answer problems. Cost totals include the same proposal pool.

For always-on Uniform, the final-protocol control adds one generic verification call on each original agreement problem and reuses existing Uniform outputs elsewhere. An unparseable check falls back to the original response. The one-call and closed-loop variants share the same added agreement checks, so they are controls on the same underlying outputs rather than independent replications. Both show zero fixes and zero breaks among those 86 checks. Moderation token totals in the report are rounded, and percentages computed from them are approximate.

D ROUTE AND RECOVERY DETAILS

Initial state	N	ReSolve	SC@4	SC@8	Uniform	Oracle@4
Agreement	86	74	74	74	74	74
Dominant answer	18	14	12	12	11	14
Two groups	8	5	4	5	5	5
Fragmented	12	4	1	3	4	6
No answer	6	3	0	0	3	0

Table 10: F2 route-wise Hybrid correct counts, recomputed from archived per-question records. Agreement is defined by answer groups rather than correctness.

F2's initial route counts are 86 agreement, 18 dominant-answer, eight two-group, 12 fragmented, and six no-answer problems. Under Hybrid, the SC@4 fixes are 2, 1, 3, and 3 in the four triggered categories, with no breaks. Against SC@8 the corresponding outcomes are 2/0, 0/0, 2/1, and 3/0. Against Uniform they are 3/0, 0/0, 1/1, and 0/0. Both moderators solve three of six no-answer problems. These are descriptive strata rather than independent significance tests.

Table 11 separates agreement by parseability. Only 54 of the 86 skipped problems have four parseable candidates; 14 have just one. Agreement is therefore consistency among available answers, not necessarily four-way agreement. The 74 SC@4-correct skipped problems and 17 SC@4-correct triggered problems account for all 91 correct votes. All 17 triggered correct votes are preserved; the nine fixes occur among the other 27 triggered problems. The archived F3 summary does not provide this conditional denominator or parseability breakdown.

Parseable candidates	Agreement problems	Correct under Hybrid
1	14	6
2	7	5
3	11	10
4	54	53

Table 11: F2 agreement states by the number of parseable initial candidates. SC@4 and ReSolve share the skipped outputs; all 86 added Uniform checks leave correctness unchanged. Counts are descriptive.

F2's three recoveries beyond Oracle@4 all arise in no-answer states; both losses occur in fragmented states. F3 reverses the net balance with two recoveries and three losses. Its ten no-answer states, compared with six in F2, also contain three of the four ReSolve losses to SC@8. The aggregate oracle is bounded only for selection from its specified pool and evaluator. It is neither a deployable baseline nor an upper bound on a generator with additional computation.

E HISTORICAL MECHANISM CONTROLS

Method	AIME24	AIME25	BRUMO25	CMIMC25	HMMT25	Total
Oracle@4	27/26	25/25	24/24	30/30	20/20	126/125
SC@4	23/23	24/24	24/24	29/29	19/19	119/119
ReSolve	26/26	26/26	24/24	30/30	21/21	127/127
Matched operation	25/25	26/26	24/24	31/31	20/20	126/126
Uniform, closed loop	26/26	25/25	24/24	31/31	20/20	126/126
Operation permutation 1	24/24	26/26	24/24	32/32	20/20	126/126
Operation permutation 2	26/26	25/25	24/24	30/30	20/20	125/125

Table 12: H1 training-free results as Hybrid/CV counts ($N = 160$). Task sizes are 30, 30, 30, 40, and 30. All rows share the same four-candidate pool.

Configuration	H/CV	Calls	Input/q (K)	Output/q (K)	Total/q (K)
SC@4	119/119	0	0.00	0.00	87.67
ReSolve	127/127	74	2.01	4.56	94.24
Uniform, closed loop	126/126	68	1.78	4.36	93.80
One specialized call, original gate	125/125	58	1.50	3.56	92.73
One Uniform call, original gate	126/126	58	1.50	3.74	92.90
One Uniform call, validity gate	126/126	86	1.71	5.82	95.20
One Uniform call, always	126/126	160	2.30	8.40	98.37
ReSolve, validity gate	127/127	102	2.23	6.64	96.54
Uniform loop, validity gate	126/126	96	1.99	6.45	96.10
Fresh solve, original gate (16K)	119/118	58	0.06	5.73	93.46

Table 13: D1 on the H1 pool ($N = 160$). Input/output columns count moderation only; total includes proposals. Calls are panel totals and tokens are per-question means. The original gate triggers on 58 questions.

H1 and D1 share a historical 160-problem candidate pool that additionally contains 30 BRUMO25 questions. Their overlapping rows are reused observations. All displayed historical configurations use existing model weights. H1's Hybrid oracle decomposition is four recoveries and three losses; the corresponding CV decomposition is five and three. Both satisfy the recovery/loss identity with their own oracle totals (126 and 125). On the ten no-answer problems, ReSolve solves four while SC@4 solves none. The 102 agreement states include nine errors, unchanged by D1's always-on Uniform.

In D1, one candidate-conditioned Uniform call and a fresh independent solve act on the same 58 triggered questions under a 16K output cap. Hybrid scores are 126 and 119; CV scores are 126 and 118. The original answer is used on skipped questions. This is supporting evidence about candidate-conditioned reasoning, but changing the prompt means it is less isolated than the final-protocol answer-only intervention. Selective and always-on Uniform are both 126 under Hybrid while consuming different moderation budgets. These historical costs are not pooled with the 32K-context F2/F3 protocols.

F CASE-STUDY EVIDENCE

The main-text quartic example and the geometry example below come from the historical panel. Their identifiers are zero-based cache indices, not official contest question numbers. Descriptions are edited mathematical summaries, not verbatim model transcripts.

CMIMC25, index 7. The polynomial is $P(x) = x^4 + 20x^3 + 29x^2 - 666x + 2025$. Expanding $(x^2+10x-36)^2+(x+27)^2$ verifies the identity found in the fourth unfinished attempt and retained in the moderator's input. The first-quadrant root is $(-10 + i + \sqrt{243 + 88i})/2$, giving the required sum 322. The identity, discriminant, and quadrant were checked independently. All four initial final answers are unparseable and marked incorrect under both scoring views. Their finish reasons are not recorded, so absence of an answer is not relabeled as observed length termination. The numeral 322 occurs in unrelated arithmetic in an initial trace; the claim concerns absence of a correct selectable final response. Uniform also succeeds, with 8,558 output tokens versus 7,502 for ReSolve.

AIME25, index 28. Three candidates answer 153 and one answers 104. The majority calculation omits an equilateral triangle. Checking the region decomposition gives $(208-55-49)\sqrt{3} = 104\sqrt{3}$ and answer 104. ReSolve returns the correct answer after two calls and 21,421 output tokens. Uniform reaches 104 on its first call but returns to 153 on its second. This example concerns discovering and retaining a correction; it does not show that a generic verification instruction cannot find the correction.

G EVIDENCE PROVENANCE AND SCOPE

F2/H1/D1 counts and paired tests are recomputed from archived per-question records. F3, invocation, latency, evidence-removal, and clipping results use completed experiment reports; marginal and paired-count arithmetic is checked here, but remote generation and grading artifacts were not replayed locally. Combined problem-level tests are taken from the completed analysis. Historical protocols remain separate. The evidence concerns mathematical inference with the specified backbone and budgets.

H SCOPE AND LIMITATIONS

Observed preservation depends on the reference decision. Neither pool loses a problem answered correctly by SC@4, yet moderation loses two and three problems, respectively, that contain a correct candidate according to Oracle@4. Improving the vote therefore does not guarantee retaining every correct candidate, or avoiding regressions on future pools. The SC@8 and Uniform comparisons likewise leave accuracy superiority and the value of specialized instructions unresolved.

Both principal evaluations reuse the same 130 questions. Independent candidate sampling addresses pool variability while leaving benchmark familiarity and task generalization unresolved. The answer-only and final-protocol always-on interventions use F2; F3 repeats the main comparison rather than those individual interventions. The backbone is a 4B reasoning model, so the observed tradeoff does not establish a universal cost frontier across model sizes or domains. Acceptance and grouping also depend on answer extraction, whose errors can differ for open-ended answers.

Evidence construction and budgeting remain part of the intervention. Bounded representative views can omit a useful step; giving every candidate equal space or reallocating saved prompt context may change results. Our token accounting includes reading and generation, but not a hardware-normalized measure of computation. The latency audit shows why fewer tokens need not imply a faster response. Finally, Hybrid scoring reduces some equivalence errors without making the rule grader or verifier infallible. The mathematical answer audit and clipping replay identify specific failure modes rather than certify all model outputs.