

Multi-Agent Debate with Disagreement-Aware Arbitration for Mathematical Reasoning

Ming Wang^{1*}, Sarah Collins², Joshua Bell³

¹Department of Computer Science, University of Illinois Urbana-Champaign, Urbana, IL, USA

²School of Computer Science, Georgia Institute of Technology, Atlanta, GA, USA

³Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA, USA

ming.wang.cs@outlook.com

Abstract

Multi-agent reasoning promises to improve mathematical problem solving by exposing disagreement before a final answer is selected. This study isolates that coordination problem in a controlled, fully executable setting and evaluates four inference policies on the complete 1,319-problem GSM8K test split: a greedy single chain of thought, lexical self-consistency, three-agent debate, and debate followed by a disagreement-aware arbiter. Executable arithmetic programs were induced from the annotated GSM8K training rationales; 6,999 of 7,473 training problems yielded programs that exactly reconstructed their gold answers. Three retrieval-program agents used raw lexical features, number-normalized structural features, and operation-cue features. The arbiter was a held-out logistic model trained on agreement, support, similarity, trace complexity, and numeric-plausibility features. Initial agents disagreed on 97.12% of test problems. Single-chain and self-consistent inference each solved 27 problems (2.05%). One debate round solved 35 (2.65%), while disagreement-aware arbitration solved 91 (6.90%). The arbiter improved accuracy over the single-chain baseline by 4.85 percentage points; the paired bootstrap 95% interval was 3.56 to 6.29 points and the exact McNemar p-value was 7.37×10^{-13} . Mean token-equivalent costs were 77.6, 725.8, 265.6, and 1,015.0, respectively. Candidate-generation failure accounted for 88.93% of test cases, establishing that arbitration recovers useful minority and extended candidates but does not replace a strong proposer.

Keywords: *Multi-Agent Reasoning; Debate; Mathematical Reasoning; Disagreement; Arbitration; GSM8K; Self-Consistency; Program Induction*

1. Introduction

Mathematical word-problem solving combines language interpretation, quantity grounding, operation selection, and exact calculation. Earlier systems used verb categories, equation templates, semantic parsing, and curated arithmetic and algebra repositories [1]-[5]. They established two lessons: quantities must be distinguished by semantic role rather than surface order, and answers are easier to audit when produced by executable expressions. The same emphasis on traceable evidence appears in retrieval-grounded HDFS failure narratives [6] and source-grounded financial disclosure QA [7]. Accordingly, every agent here emits an arithmetic trace, and candidate construction explores alternative mappings between question numbers and program roles.

Large language models shifted the interface from task-specific parsing to prompted generation. Few-shot learning [8], chain-of-thought prompting [9], zero-shot reasoning [10], and least-to-most decomposition [11] showed that intermediate text can unlock difficult computations. Yet a fluent chain may contain a wrong operation, invalid quantity binding, or plausible calculation attached to the wrong story. Calibration with human uncertainty distributions likewise

separates prediction from confidence [12]. GSM8K was introduced with learned verifiers because generation and selection are distinct problems [13].

Self-consistency turns one chain into a distributional decision by sampling several traces and selecting the modal answer [14]. It helps only when samples explore genuinely different paths and their errors are not perfectly correlated. Spectral rank aggregation likewise shows that reliable ordering depends on the structure and quality of noisy comparisons, not vote count alone [15]. Here, self-consistency tests repeated evidence from one representation; debate tests independent representations; arbitration tests whether a learned selector recognizes useful alternatives.

Subsequent work expanded reasoning controls. STaR bootstraps reasoning from generated rationales [16]; program-aided and program-of-thought methods delegate arithmetic to executable code [17], [18]; Tree of Thoughts searches partial solutions [19]; and Self-Refine and Reflexion add feedback loops [20], [21]. Symbolic solvers provide exact computation after formalization [22]. Retrieval-summary integration for code intelligence further illustrates that finding evidence and explaining it are separate stages

[23]. Across these approaches, multiple reasoning objects create evidence that a selector can exploit.

Selection becomes critical when agents disagree. Process and outcome supervision assign evidence at different levels [24], [25], and Minerva demonstrated the value of domain-focused quantitative training [26]. A verifier trained only on final outcomes may miss structurally wrong traces that land on plausible values, while a process selector may overvalue familiar or lengthy traces. Comparable stakes arise in bank credit-risk classification [27] and medical image diagnosis [28], where confidence must be distinguished from the prediction itself. The arbiter therefore combines agreement, retrieval support, trace complexity, and numeric plausibility. Debate exposes competing answers. AI-safety debate framed adversarial argument as a way to simplify difficult judgments [29], and later work developed more efficient protocols [30]. Multiagent language-model debate allowed several model instances to exchange solutions before voting [31]. A related principle appears in autonomous-driving perception fusion, where complementary views are combined rather than repeatedly sampling one representation [32]. Because intrinsic self-correction remains unreliable without external evidence [33], our agents share candidates backed by independently retrieved executable programs.

A debate vote and an arbiter solve different statistical problems. The vote asks which candidate receives immediate support; the arbiter asks whether the support pattern resembles held-out correct cases. This distinction matters in operational systems, where AIOps fault analysis combines heterogeneous signals before intervention [34] and long-document RAG must reconcile evidence across dispersed clauses [35]. Here, disagreement-aware arbitration uses the structure of disagreement and retains deeper candidates that never became an agent's first choice.

The arbitration problem also connects to ensemble learning. Accuracy depends on member quality and error diversity [36], while raw disagreement is insufficient because diverse systems can still be jointly wrong [37]. Dawid-Skene estimates source reliability from agreement patterns [38]. Our arbiter follows that statistical intuition without assuming fixed agent competence: it learns candidate-level features from held-out GSM8K problems and selects one answer after evidence exchange.

This paper asks whether lexical self-consistency improves a fixed proposer, whether one debate round corrects more answers than it damages, and whether explicit arbitration recovers minority or extended candidates. The design uses lightweight local agents to keep every trace executable and isolate coordination. That separation mirrors IoT anomaly detection, where feature extraction and anomaly decisions are distinct components [39]. The study measures the value and limits of coordination under weak but transparent proposal generation; it does not claim competitive GSM8K accuracy. The design yields three expectations. Repeated lexical sampling should add little when nearest-neighbor traces are correlated. Cross-view debate should provide a limited net

gain because all agents share one program corpus. A held-out arbiter should outperform voting when correct answers occur as minorities or deeper candidates, but its gain is bounded by candidate coverage. Noisy-neighbor VM degradation modeling offers an analogous warning: multiple signals remain useful only when residual interference is modeled explicitly [40]. All comparisons use paired predictions on the same 1,319 test problems.

Contributions are threefold: a reproducible pipeline that converts GSM8K rationales into executable traces; a three-view debate architecture with a held-out logistic arbiter; and a full analysis of accuracy, disagreement, cost, significance, complexity, calibration, and failure transitions. The final distinction between proposal failure and response policy also appears in predictive DDoS mitigation, where detection and resource allocation are evaluated separately [41]. Debate yields a small net gain, arbitration a larger statistically decisive gain, and proposal coverage remains the primary bottleneck.

2. Method

2.1 Dataset, splits, and evaluation protocol

The experiments use the GSM8K train and test JSONL files. Each record contains a question and a worked answer whose final numeric value follows the delimiter "####". The splits contain 7,473 and 1,319 problems. Every test item was evaluated under each policy, using exact match after rational-number normalization; 5, 5.0, and 10/2 were therefore equivalent.

Faithful programs served three roles. The full set formed the final retrieval index; a fixed 70% formed the development index; and the remaining 30% was split into 1,050 arbiter-training and 1,050 calibration problems. Seed 20260822 fixed the split. Calibration selected k and measured held-out arbiter performance; test answers were never used for training or selection. Table 1 summarizes the corpus.

Split boundaries were applied after fidelity filtering and remained fixed. Arbiter labels came only from its training partition, while calibration labels selected k and assessed the frozen arbiter. The final retrieval index used all faithful training programs because retrieval never observes test outcomes. The program records source-file SHA-256 digests, software versions, configuration, and one JSON record per test problem, making leakage or post-hoc metric substitution detectable.

Table 1. Dataset profile and executable-program coverage.

Split	Examples	Mean question tokens	Mean rationale steps	Mean numeric mentions	Faithful programs
Train	7473	52.28	3.17	3.24	6999 (93.66%)
Test	1319	53.64	3.25	3.34	1208 (91.58%)

2.2 Executable program induction

GSM8K rationales contain arithmetic annotations of the form `<<expression=value>>`. The parser converted each

annotation into an abstract syntax tree supporting addition, subtraction, multiplication, division, parentheses, and unary signs. Intermediate numeric results were replaced by their preceding expression trees, which yielded one expanded expression for the final annotated step. Leaves that matched a number in the question became role variables N_0 , N_1 , and so forth; unmatched leaves remained constants.

A program entered the retrieval corpus only when re-instantiating its role variables with the source question numbers reproduced the gold answer exactly. This fidelity test retained 6,999 of 7,473 training problems (93.66%). The same parser reconstructed 1,208 of 1,319 test rationales (91.58%); those test programs were used only to stratify analysis by operation family, never to generate predictions. Failures came from rationales without arithmetic annotations, algebraic derivations expressed only in prose, or ambiguous reuse of an intermediate value.

Program expansion preserved the order and nesting of the annotated arithmetic. When an earlier annotation computed an intermediate value, a later occurrence of that value was replaced by the earlier expression tree before role normalization. A rationale that first computed $48/2=24$ and then $48+24=72$ therefore became the reusable program $N_0+(N_0/2)$, not a flat expression containing an unexplained constant 24. Exact rational arithmetic was used throughout induction and execution. Division by zero, missing role indices, and values whose magnitude exceeded the execution guard were rejected before entering an agent's support list.

Retrieved programs can use fewer roles than a target question, and target quantities may appear in another order. The solver evaluated the identity mapping and deterministic alternatives, retaining at most 48. Moving m roles by displacement d received $0.78^m \times 0.96^d$. The fixed cap makes search cost explicit, consistent with budget-constrained decision policies [42]. Role-flexible instantiation exposes minority candidates, while the greedy baseline always uses the executable identity mapping.

2.3 Three reasoning agents

The three agents shared one program corpus and arithmetic executor but used independent retrieval representations. Lexical-CoT indexed raw lower-case word unigrams and bigrams. Structural-CoT replaced every numeric mention with NUM and indexed one- to three-grams. Cue-CoT retained NUM, arithmetic and question cues, a generic intervening token, and numeric-mention count. This separation of semantic and neighborhood evidence resembles hybrid e-commerce recommendation designs [43], but every retrieved item here remains an executable arithmetic program. Table 2 lists the exact settings.

For question q and retrieved example e , cosine similarity was multiplied by a numeric-count penalty, a small penalty for programs longer than five operations, and the role-mapping factor. Support for candidate a summed these weights after square-root discounting by executable retrieval rank. Each agent also stored maximum similarity, distinct-template count, and mean operation count.

Each representation was fitted only on its retrieval partition with sublinear term frequency, L2 normalization, and batched sparse cosine search. A deterministic partial sort retained top neighbors and broke ties by source index. Diversity came solely from visible evidence: lexical matching preserved entities and phrasing, structural matching emphasized normalized sentence form, and cue matching retained arithmetic operators, comparison words, units, and numeric count.

The single-chain baseline was the first executable identity-mapped trace returned by Lexical-CoT. This definition yields one explicit calculation per problem and preserves a true greedy baseline. Self-consistency used only Lexical-CoT and aggregated the first k executable traces by a similarity-weighted vote. The calibration sweep tested k in $\{3, 5, 7, 9, 11\}$; $k=9$ achieved the highest calibration accuracy at the lower cost among tied settings.

Table 2. Retrieval-agent configurations.

Agent	Input representation	N-grams	Max features	Output
Lexical-CoT	Raw lower-case question	Word 1-2	16,000	Greedy trace + support
Structural-CoT	Numbers replaced by NUM	Word 1-3	16,000	Greedy trace + support
Cue-CoT	Arithmetic cues + skeleton	Word 1-3	8,000	Greedy trace + support

2.4 Candidate-sharing debate

The debate pool contained the three initial answers, the self-consistent answer, and the three strongest alternatives from each agent. During one exchange, each agent rescored every shared candidate using its support plus 0.30 times maximum similarity; its original answer received a 0.05 persistence term. Each agent then selected its strongest candidate, and a confidence-weighted vote produced the debate answer. Figure 1 shows the data flow.

The exchange is evidence-bearing rather than conversational. Agents transmit candidate values and support from executable programs, not free-form critiques. This keeps stylistic persuasiveness from becoming an uncontrolled variable and separates candidate expansion from agent rescoring.

One exchange was used for every problem to keep compute comparable. The extended set included supported role remappings even when no agent promoted them. Minority recovery means the gold answer was an initial proposal; extended recovery means it appeared only below a first choice. If absent from the pool, neither debate nor arbitration could return it. A fixed round and saved evidence also create a stable control surface for monitoring and rollback, paralleling drift-aware production safeguards [44].

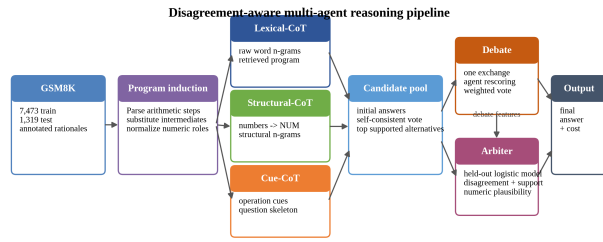


Figure 1. Program induction, three-agent candidate sharing, debate, and disagreement-aware arbitration.

2.5 Disagreement-aware arbitration

The arbiter treated each distinct candidate as a binary instance. Features covered initial and post-exchange votes, self-consistency and debate indicators, distinct-answer count, normalized disagreement entropy, unanimity, aggregate support and margin, integer and sign flags, equality to an input quantity, log magnitude, answer-to-input scale, question length, and numeric-mention count. Agent-specific features recorded choices, support, maximum similarity, template count, and mean program length. Combining heterogeneous behavioral signals before risk scoring is also common in financial-service analytics [45]; here the features describe evidence for one arithmetic candidate.

A standardized logistic regression with balanced class weights was fitted on candidate rows from 1,050 arbiter-training problems. The separate calibration partition produced 7.05% final-answer accuracy. At test time the highest probability won; aggregate support, post-exchange votes, and smaller absolute value broke ties. The model extends reliability aggregation [38] by conditioning on problem evidence. Uncertainty analysis for spectral estimators reinforces the distinction between a ranking score and a calibrated distribution [46].

The specification and transformations were frozen before test evaluation. Final-answer accuracy, not candidate-row accuracy, was reported on calibration because each problem contributes a different candidate count but only one returned answer. Balanced weights addressed incorrect-row prevalence; standardization was fitted on arbiter-training rows and applied unchanged. Probabilities ranked candidates within a problem and were never compared across questions.

2.6 Metrics, cost accounting, and statistical tests

Accuracy is exact-answer fraction. Disagreement is the fraction with at least two distinct initial answers. Three-agent normalized entropy is $-\sum p_i \log(p_i)/\log(3)$, with zero for unanimity. Transitions count wrong-to-right, right-to-wrong, unchanged-right, and unchanged-wrong cases. Failure categories record whether gold appeared initially, only in the extended pool, or nowhere.

Token-equivalent cost used one deterministic Unicode word-and-symbol tokenizer. Single-chain cost included the question and one trace; self-consistency counted k traces; debate counted three question-trace pairs plus the exchange; arbitration added a serialized feature summary. These values compare inference volume under one rule and are not vendor billing estimates.

Paired uncertainty used 5,000 bootstrap resamples of test accuracy differences, and exact McNemar tests compared each method with Single CoT. Analyses also stratified by rationale length, question length, operation family, and disagreement. Per-problem predictions, traces, probabilities, costs, split metadata, checksums, and environment versions were saved.

3. Results and Discussion

3.1 Main accuracy and inference cost

Table 3 and Figure 2 report the primary results. Single CoT and $k=9$ self-consistency each solved 27 of 1,319 problems (2.05%). Debate solved 35 (2.65%), a net gain of eight. Arbitration solved 91 (6.90%), 64 more than Single CoT and 56 more than debate.

Mean token-equivalent cost rose from 77.6 to 725.8, 265.6, and 1,015.0. Figure 3 shows self-consistency is dominated by Single CoT: identical accuracy at 9.36 times the cost. Debate improved accuracy at 3.42 times the cost. Arbitration gained most accuracy but delivered 0.068 correct-fraction points per 1,000 tokens under this normalization.

Absolute accuracy is low because a retrieved program transfers only when story structure, role binding, and operation order align. The paired comparison remains valid: every policy receives candidates from the same corpus, executor, and representations. Arbitration alone changes selection.

Operationally, Single CoT is cheapest, self-consistency spends 648.2 extra token-equivalents without a correction, debate adds 188.0 and eight answers, and arbitration adds more features but 64 answers. Unconditional arbitration maximizes measured accuracy; routing only disputed or high-value cases reduces average cost.

Table 3. Full-test accuracy and token-equivalent cost.

Method	Correct	Accuracy (%)	Mean	Median	p95	Accuracy/1k
Single CoT	27	2.05	77.6	74.0	123.0	0.264
Self-consistency	27	2.05	725.8	733.0	1191.0	0.028
3-agent debate	35	2.65	265.6	254.0	396.1	0.100
Debate + arbiter	91	6.90	1015.0	1041.0	1299.2	0.068

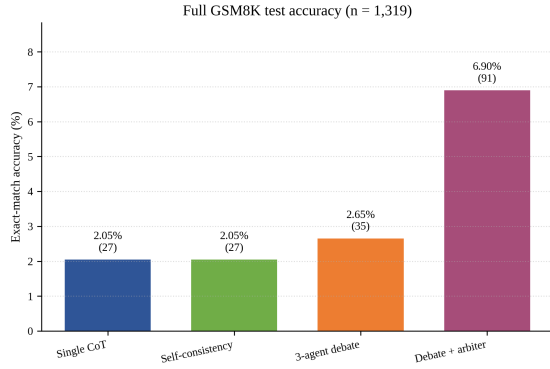


Figure 2. Exact-match accuracy for the four inference policies on all 1,319 test problems.

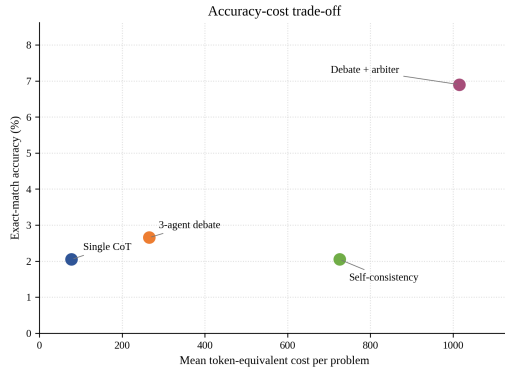


Figure 3. Accuracy-cost positions under one fixed token-equivalent accounting rule.

3.2 Agent diversity and disagreement

Table 4 separates the three initial agents. Cue-CoT was strongest at 3.41%, Structural-CoT reached 2.20%, and Lexical-CoT reached 2.05%. Their mean confidence values did not track accuracy: Cue-CoT had the lowest mean confidence but the highest exact-match rate. This mismatch supports feature-based arbitration rather than direct confidence voting.

Figure 4 shows that only 38 problems (2.88%) produced unanimous initial answers. Two distinct answers occurred on 727 problems (55.12%), and three distinct answers occurred on 554 (42.00%). The overall disagreement rate was therefore 97.12%. High disagreement alone did not guarantee a recoverable answer. As Table 5 shows, arbitration reached 7.02% accuracy in the two-answer stratum and 5.42% in the three-answer stratum, but most disagreements still lacked the gold candidate.

Unanimity was a strong correctness signal: the single-chain answer was correct on 23.68% of unanimous cases, compared with 1.41% across all disagreements. The arbiter improved unanimous accuracy to 26.32%, yet the stratum contained only 38 problems. These results match classical ensemble findings that disagreement must be interpreted jointly with member competence [36], [37]. In this experiment, diversity was abundant; reliable candidate coverage was scarce.

The disagreement strata behaved differently. Partial disagreement usually meant two views retrieved the same program family and the third supplied another binding; full

disagreement more often reflected unrelated templates. Debate reached 1.65% in the two-answer group and 2.53% in the three-answer group, so a third distinct proposal sometimes helped. Attention-based detection work likewise uses complementary feature views to improve selectivity [47]. The arbiter reached 7.02% and 5.42%, because duplicated support in the two-answer group created stronger candidate-level signals. Answer count measured diversity; support structure determined selectability.

Table 4. Initial-agent accuracy and mean internal confidence.

Agent	Correct	Accuracy (%)	Mean confidence
Lexical-CoT	27	2.05	0.393
Structural-CoT	29	2.20	0.389
Cue-CoT	45	3.41	0.367

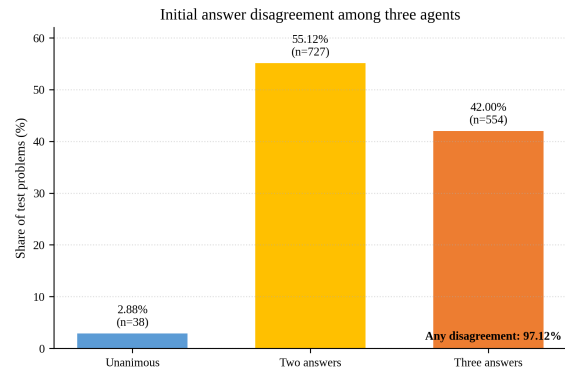


Figure 4. Distribution of one, two, and three distinct initial answers.

Table 5. Accuracy conditioned on initial disagreement.

Stratum	n	Share (%)	Single (%)	Debate (%)	Arbiter (%)
unanimous	38	2.88	23.68	23.68	26.32
partial disagreement	727	55.12	1.38	1.65	7.02
full disagreement	554	42.00	1.44	2.53	5.42

3.3 Self-consistency, debate, and arbitration transitions

The calibration sweep in Table 6 selected $k=9$. Accuracy rose from 1.52% at $k=3, 5$, and 7 to 1.62% at $k=9$ and $k=11$; the lower-cost tied value was retained. On the test set, the weighted vote remained identical to the first lexical answer. The retrieved lexical traces were correlated and often instantiated the same program family, so sampling more neighbors increased cost without adding a decisive alternative.

Table 7 presents the ablation sequence. Initial majority voting solved 2.20%, one exchange lifted accuracy to 2.65%, and the arbiter lifted it to 6.90%. Table 8 explains the changes. Debate corrected 13 single-chain errors and damaged five correct single-chain answers. Arbitration corrected 68 debate errors and damaged 12 debate successes. Relative to Single

CoT, arbitration produced 75 wrong-to-right transitions and 11 right-to-wrong transitions.

The paired tests in Table 9 distinguish descriptive gains from reliable gains. Debate improved accuracy by 0.61 percentage points; its bootstrap interval touched zero and the exact McNemar p-value was 0.096. The arbiter improved accuracy by 4.85 points, with a 95% interval from 3.56 to 6.29 points and $p=7.37 \times 10^{-13}$. Self-consistency had zero paired difference. The evidence therefore supports the arbitration gain, while the one-round debate gain remains modest at this sample size.

The arbiter was not majority voting. Its largest standardized coefficients favored integer and nonnegative answers and penalized implausible scale and magnitude; lexical choice, numeric count, cue-program length, template, and similarity terms also contributed. These are conditional associations. Vote features alone could not explain 46 successes from candidates no agent initially selected. The flexibility also caused 12 debate-to-arbiter regressions, motivating a safeguard that preserves a strong incumbent unless an alternative clears a calibrated margin.

Table 6. Calibration sweep for lexical self-consistency.

k	Calibration accuracy (%)	Mean token cost	Decision
3	1.52	203.4	
5	1.52	333.3	
7	1.52	454.4	
9	1.62	562.5	Selected
11	1.62	652.8	

Table 7. Incremental ablation from one agent to arbitration.

Configuration	Accuracy (%)	Mean token cost
Lexical agent only	2.05	77.6
Lexical self-consistency (k=9)	2.05	725.8
Three-agent initial majority	2.20	232.2
Three agents plus debate	2.65	265.6
Debate plus disagreement-aware arbiter	6.90	1015.0

Table 8. Paired correctness transitions.

Comparison	Wrong->right	Right->wrong	Right->right	Wrong->wrong	Net
Single CoT -> 3-agent debate	13	5	22	1279	+8
3-agent debate -> Debate + arbiter	68	12	23	1216	+56
Single CoT -> Debate + arbiter	75	11	16	1217	+64

Table 9. Paired significance tests against Single CoT.

Comparison	Delta (pp)	95% CI (pp)	Single only	Method only	McNemar p
Self-consistency vs Single CoT	0.00	[0.00, 0.00]	0	0	1
3-agent debate vs Single CoT	0.61	[0.00, 1.21]	5	13	0.0963
Debate + arbiter vs Single CoT	4.85	[3.56, 6.29]	11	75	7.37e-13

3.4 Complexity and operation families

Figure 5 and Table 10 show a sharp decline with rationale length. Arbitration solved 13.18% of one- or two-step problems, 4.43% of three- or four-step problems, and 1.79% of problems with at least five steps; Single CoT reached 3.41%, 1.68%, and 0.45%. Debate solved no five-step case, while arbitration recovered four. Because GSM8K is English-only, this complexity analysis excludes linguistic variation; bilingual CS education results show that language context can affect learning and engagement [48].

Operation-family results in Table 11 show a less monotonic pattern. Additive problems were the most transferable: among 118 cases, arbitration reached 17.80% and debate reached 7.63%. The 872 mixed-operation problems reached 4.59% under arbitration. Multiplicative-only problems numbered 204; their initial and debate accuracies were 1.96%, while arbitration reached 9.31%. The 14 no-operation cases reached 7.14% under the three non-arbiter policies and 0% under arbitration. Among 111 unparsed test rationales, debate reached 4.50% and arbitration reached 9.91%.

These strata expose two properties of the retrieval programs. Additive templates transferred most directly because operand roles were often local and monotonic. Arbitration recovered several multiplicative and unparsed cases from deeper candidates. Mixed-operation stories remained hardest because they combined quantity binding with operation order; ratios, percentages, rates, hidden constants, unit conversions, and division direction created plausible distractors.

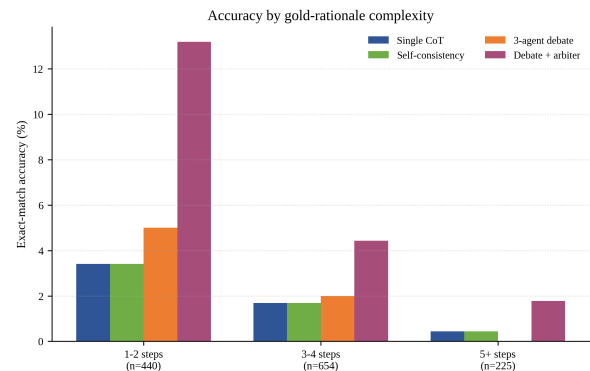


Figure 5. Accuracy by the number of annotated arithmetic steps in the gold rationale.

Table 10. Accuracy by gold-rationale complexity.

Steps	n	Single (%)	Self-consistency (%)	Debate (%)	Arbiter (%)
1-2 steps	440	3.41	3.41	5.00	13.18
3-4 steps	654	1.68	1.68	1.99	4.43
5+ steps	225	0.44	0.44	0.00	1.78

Table 11. Accuracy by operation family reconstructed from the gold rationale.

Family	n	Single (%)	Self-consistency (%)	Debate (%)	Arbiter (%)
Additive	118	1.69	1.69	7.63	17.80
Mixed	872	1.95	1.95	1.83	4.59
Multiplicative	204	1.96	1.96	1.96	9.31
No Operation	14	7.14	7.14	7.14	0.00
Unparsed	111	2.70	2.70	4.50	9.91

3.5 Failure taxonomy and calibration

Table 12 and Figure 6 divide all 1,319 final outcomes. Candidate-generation failure dominated: on 1,173 problems (88.93%), the final answer was wrong and the gold value was absent from the initial and extended candidate set. The arbiter correctly recovered 35 minority candidates already proposed by an initial agent and 46 extended candidates that appeared only in support lists. It also made 27 selection errors when a correct initial candidate was available.

The taxonomy separates two priorities. Better arbitration addresses 27 selection errors and protects correct majority answers; better proposal generation addresses 1,173 missing-candidate failures and offers much larger headroom. Stronger semantic parsers, program-aided language models [17], [18], symbolic formulations [22], and process verifiers [24], [25] directly target coverage. Related enterprise marketing systems likewise distinguish predictive evidence from the decision that acts on it [49].

Figure 7 shows expected calibration error 0.738 and Brier score 0.615. Selected scores exceeded observed accuracy in every populated high-confidence bin. Balanced logistic training improved within-problem ranking but not calibration because candidate rows were imbalanced and dependent. The scores are therefore ranking values, not correctness probabilities.

Table 12. Final outcome taxonomy.

Category	Count	Share (%)
candidate-generation failure	1173	88.93
correct: extended-candidate recovery	46	3.49
correct: minority recovery	35	2.65
shared reasoning error	28	2.12
arbiter selection error	27	2.05
correct: unanimous	10	0.76

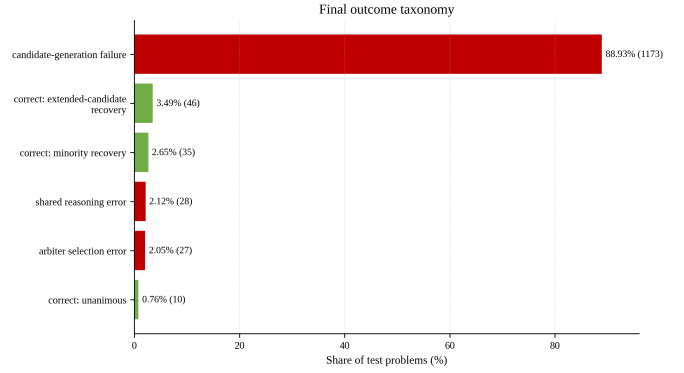


Figure 6. Correct recoveries and unresolved failure categories for the final arbiter.

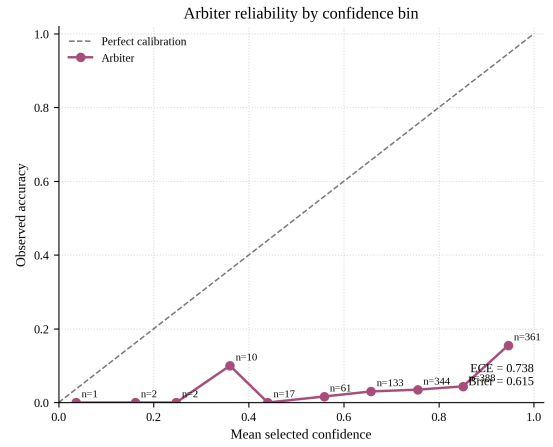


Figure 7. Observed accuracy versus selected logistic score by confidence bin.

3.6 Measured case behavior

Table 13 shows three mechanisms. In the bicycle-stop case, Cue-CoT proposed 25 while the others proposed -65 and 560; exchange moved two agents to 25, and debate and arbitration were correct. In the spoon-package case, no initial agent proposed 10, but role-flexible support generated it and the arbiter selected it over 264. In the raspberry case, debate selected the correct 187, but the arbiter preferred 93.

Debate succeeds when one agent holds a correct trace and the others find support for it. Arbitration succeeds when gold exists deeper in the support lists and resembles held-out correct candidates. It fails when a compact distractor receives a stronger score than the correct minority, and no coordination policy succeeds when the candidate generator never constructs gold.

Table 13. Representative test transitions recorded by the experiment.

Problem cue	Initial answers	Debate	Arbiter	Gold	Observed mechanism
Bike stops: 60, 20, 15	-65 / 560 / 25	25	25	25	Debate and arbiter recover minority
Spoon package: +5, -3, total 12	264 / 264 / -0.58	264	10	10	Arbiter recovers extended candidate

Problem cue	Initial answers	Debate	Arbiter	Gold	Observed mechanism
Raspberries: 6 x 20 + 67	93 / 18 / 187	187	93	187	Arbiter overturns correct debate

4. Limitations

First, proposal quality is limited. The agents retrieve and instantiate arithmetic programs rather than create new semantic parses. Traces are exact and reproducible, but transfer fails on new quantity roles, implicit constants, unit conversions, or prose-only algebra. The 6.90% best accuracy quantifies this restriction.

Second, the study isolates coordination without a neural language-model proposer. Its conclusions apply to these retrieval-program agents: arbitration improves selection under a fixed generator, self-consistency fails when traces are correlated, and missing candidates dominate errors. The effect size need not transfer to frontier models with different candidate and argument distributions.

Third, the parser reconstructed 93.66% of training rationales. Excluded items include valid reasoning without machine-readable arithmetic annotations. Equation, verbal-fraction, and unit-semantic handling would expand the corpus and alter candidate coverage.

Fourth, token-equivalent cost uses a deterministic tokenizer rather than a vendor subword vocabulary. The cross-method ranking is consistent, but the values are not API billing estimates; runtime and memory also reflect the reported CPU implementation.

Fifth, evaluation uses one English-language benchmark. Tests on SVAMP, ASDiv, MAWPS, and MATH [1]-[4], [50] are needed to separate GSM8K-specific retrieval effects from general arbitration behavior. Multilingual and culturally varied word problems are also important, given evidence that instructional language changes engagement [48].

Sixth, the arbiter is overconfident, and subgroup analyses depend on benchmark annotations. Calibration, abstention, and protection for strong incumbents are required before scores guide high-stakes decisions, a concern also evident in clinical MRD prediction [51]. Several problems admit shorter equivalent programs, and 111 test rationales were not reconstructed. Main exact-match results include every question, but subgroup estimates inherit annotation dependence.

5. Conclusion

This study evaluated Single CoT, self-consistency, three-agent debate, and disagreement-aware arbitration on all 1,319 GSM8K test problems using one fixed corpus of executable training-rationale programs. The agents disagreed on 97.12% of problems. Self-consistency raised mean cost from 77.6 to 725.8 without changing accuracy; debate raised accuracy from 2.05% to 2.65%; and the held-out arbiter reached 6.90%, a decisive 4.85-point gain over Single CoT.

The gains combined 35 minority recoveries and 46 extended-candidate recoveries. Yet 1,173 problems failed because no candidate matched gold. Debate and arbitration improve

selection when diverse evidence contains the answer; they do not compensate for a weak proposer.

A stronger system would combine disagreement-aware control with semantic program generation, external calculation, process verification, calibrated confidence, and abstention. Disagreement then becomes a routing signal: easy unanimous cases stop early, recoverable conflicts receive arbitration, and unsupported cases trigger new reasoning rather than a forced vote.

References

- [1] A. Patel, S. Bhattamishra, and N. Goyal, "Are NLP models really able to solve simple math word problems?" in Proc. NAACL-HLT, pp. 2080-2094, 2021.
- [2] W. Ling, D. Yogatama, C. Dyer, and P. Blunsom, "Program induction by rationale generation: Learning to solve and explain algebraic word problems," in Proc. ACL, pp. 158-167, 2017.
- [3] R. Koncel-Kedziorski, S. Roy, A. Amini, N. Kushman, and H. Hajishirzi, "MAWPS: A math word problem repository," in Proc. NAACL-HLT, pp. 1152-1157, 2016.
- [4] S.-Y. Miao, C.-C. Liang, and K.-Y. Su, "A diverse corpus for evaluating and developing English math word problem solvers," in Proc. ACL, pp. 975-984, 2020.
- [5] M. J. Hosseini, H. Hajishirzi, O. Etzioni, and N. Kushman, "Learning to solve arithmetic word problems with verb categorization," in Proc. EMNLP, pp. 523-533, 2014.
- [6] J. Nie and D. Zheng, "Ambiguity-Aware HDFS Log Anomaly Detection with Retrieval-Augmented Failure Narratives and Selective Refusal," J. Adv. Comput. Syst., vol. 3, no. 1, pp. 66-80, Jan. 2023, doi: 10.69987/JACS.2023.30105.
- [7] S. Zhou, Z. Li, and E. Wang, "Evidence-Grounded RAG for Tokenized Trade Receivable Disclosure QA under U.S. Capital Market Standards," J. Adv. Comput. Syst., vol. 3, no. 7, pp. 41-57, Jul. 2023, doi: 10.69987/JACS.2023.30704.
- [8] T. B. Brown et al., "Language models are few-shot learners," in Advances in Neural Information Processing Systems, vol. 33, pp. 1877-1901, 2020.
- [9] J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," in Advances in Neural Information Processing Systems, vol. 35, pp. 24824-24837, 2022.
- [10] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," in Advances in Neural Information Processing Systems, vol. 35, pp. 22199-22213, 2022.
- [11] D. Zhou et al., "Least-to-most prompting enables complex reasoning in language models," in Proc. Int. Conf. Learning Representations, 2023.
- [12] Z. S. Zhong, R. Ma, and H. Zhao, "Human-Uncertainty Distillation for Calibrated Vision Models on CIFAR-10H," J. Adv. Comput. Syst., vol. 3, no. 2, pp. 77-89, Feb. 2023, doi: 10.69987/JACS.2023.30206.
- [13] K. Cobbe et al., "Training verifiers to solve math word problems," arXiv:2110.14168, 2021.

- [14] X. Wang, J. Wei, D. Schuurmans, Q. V. Le, E. H. Chi, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," in Proc. Int. Conf. Learning Representations, 2023.
- [15] Z. S. Zhong and S. Ling, "Improved theoretical guarantee for rank aggregation via spectral method," *Inf. Inference*, vol. 13, no. 3, Art. no. iaac020, Sep. 2024, doi: 10.1093/imaia/iaac020.
- [16] E. Zelikman, Y. Wu, J. Mu, and N. D. Goodman, "STaR: Bootstrapping reasoning with reasoning," in *Advances in Neural Information Processing Systems*, vol. 35, pp. 15476-15488, 2022.
- [17] L. Gao et al., "PAL: Program-aided language models," arXiv:2211.10435, 2022.
- [18] W. Chen, X. Ma, X. Wang, and W. W. Cohen, "Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks," arXiv:2211.12588, 2022.
- [19] S. Yao et al., "Tree of thoughts: Deliberate problem solving with large language models," in *Advances in Neural Information Processing Systems*, vol. 36, pp. 11809-11822, 2023.
- [20] A. Madaan et al., "Self-Refine: Iterative refinement with self-feedback," in *Advances in Neural Information Processing Systems*, vol. 36, pp. 46534-46594, 2023.
- [21] N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao, "Reflexion: Language agents with verbal reinforcement learning," in *Advances in Neural Information Processing Systems*, vol. 36, pp. 8634-8652, 2023.
- [22] J. He-Yueya, G. Poesia, R. E. Wang, and N. D. Goodman, "Solving math word problems by combining language models with symbolic solvers," arXiv:2304.09102, 2023.
- [23] Y. Li, "Findable then Explainable: Retrieval-Summary Integration for Code Intelligence on a Lightweight CodeSearchNet Subset," *J. Adv. Comput. Syst.*, vol. 4, no. 7, pp. 65-82, Jul. 2024, doi: 10.69987/JACS.2024.40706.
- [24] H. Lightman et al., "Let's verify step by step," arXiv:2305.20050, 2023.
- [25] J. Uesato et al., "Solving math word problems with process- and outcome-based feedback," arXiv:2211.14275, 2022.
- [26] A. Lewkowycz et al., "Solving quantitative reasoning problems with language models," arXiv:2206.14858, 2022.
- [27] Q. Xin, R. Song, Z. Wang, Z. Xu, and F. Zhao, "Enhancing Bank Credit Risk Management Using the C5.0 Decision Tree Algorithm," *J. Comput. Technol. Appl. Math.*, vol. 1, no. 4, pp. 100-107, Nov. 2024, doi: 10.5281/zenodo.14032041.
- [28] Z. Zhong, M. Zheng, H. Mai, J. Zhao, and X. Liu, "Cancer image classification based on DenseNet model," *J. Phys.: Conf. Ser.*, vol. 1651, no. 1, Art. no. 012143, 2020, doi: 10.1088/1742-6596/1651/1/012143.
- [29] G. Irving, P. Christiano, and D. Amodei, "AI safety via debate," arXiv:1805.00899, 2018.
- [30] J. Brown-Cohen, G. Irving, and G. Piliouras, "Scalable AI safety via doubly-efficient debate," arXiv:2311.14125, 2023.
- [31] Y. Du et al., "Improving factuality and reasoning in language models through multiagent debate," arXiv:2305.14325, 2023.
- [32] Y. Wang, S. Du, Q. Xin, Y. He, and W. Qian, "Autonomous Driving System Driven by Artificial Intelligence Perception Fusion," *Acad. J. Sci. Technol.*, vol. 9, no. 2, pp. 193-198, Feb. 2024, doi: 10.54097/e0b9ak47.
- [33] J. Huang et al., "Large language models cannot self-correct reasoning yet," arXiv:2310.01798, 2023.
- [34] Y. He, Y. Pan, Y. Wang, S. Du, and Q. Xin, "Intelligent Fault Analysis with AIOps Technology," *J. Theory Pract. Eng. Sci.*, vol. 4, no. 1, pp. 94-100, Feb. 2024, doi: 10.53469/jtpes.2024.04(01).13.
- [35] S. Zhou, Z. Li, and E. Wang, "Long-Document RAG for Contractual and Insurance Clause Analysis in Receivables RWA Structures," *J. Adv. Comput. Syst.*, vol. 4, no. 8, pp. 88-104, Aug. 2024, doi: 10.69987/JACS.2024.40810.
- [36] T. G. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Systems*, LNCS 1857, pp. 1-15, 2000.
- [37] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Mach. Learn.*, vol. 51, pp. 181-207, 2003.
- [38] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the EM algorithm," *J. Roy. Stat. Soc. C*, vol. 28, no. 1, pp. 20-28, 1979.
- [39] Q. Xin, Z. Xu, L. Guo, F. Zhao, and B. Wu, "IoT traffic classification and anomaly detection method based on deep autoencoders," *Appl. Comput. Eng.*, vol. 69, no. 1, pp. 64-70, Jul. 2024, doi: 10.54254/2755-2721/69/20241511.
- [40] J. Nie and D. Zheng, "Noisy-Neighbor-Aware VM Degradation Risk Modeling with Unsupervised Residual Fusion," *J. Adv. Comput. Syst.*, vol. 4, no. 4, pp. 112-123, Apr. 2024, doi: 10.69987/JACS.2024.40409.
- [41] B. Wang, Y. He, Z. Shui, Q. Xin, and H. Lei, "Predictive optimization of DDoS attack mitigation in distributed systems using machine learning," *Appl. Comput. Eng.*, vol. 64, no. 1, pp. 94-99, May 2024, doi: 10.54254/2755-2721/64/20241350.
- [42] H. Zhang, "Risk-Aware Budget-Constrained Auto-Bidding under First-Price RTB: A Distributional Constrained Deep Reinforcement Learning Framework," *J. Adv. Comput. Syst.*, vol. 4, no. 6, pp. 30-47, Jun. 2024, doi: 10.69987/JACS.2024.40603.
- [43] K. Xu, H. Zhou, H. Zheng, M. Zhu, and Q. Xin, "Intelligent classification and personalized recommendation of e-commerce products based on machine learning," in *Proc. 6th Int. Conf. Computing and Data Science*, 2024, doi: 10.54254/2755-2721/64/20241365.
- [44] H. Zhang, "DriftGuard: Multi-Signal Drift Early Warning and Safe Re-Training/Rollback for CTR/CVR

- Models," *J. Adv. Comput. Syst.*, vol. 3, no. 7, pp. 24-40, Jul. 2023, doi: 10.69987/JACS.2023.30703.
- [45] T. Yang, Q. Xin, X. Zhan, S. Zhuang, and H. Li, "Enhancing Financial Services through Big Data and AI-Driven Customer Insights and Risk Analysis," *J. Knowl. Learn. Sci. Technol.*, vol. 3, no. 3, pp. 53-62, 2024, doi: 10.60087/jklst.vol3.n3.p53-62.
- [46] Z. S. Zhong and S. Ling, "Uncertainty quantification of spectral estimator and MLE for orthogonal group synchronization," arXiv:2408.05944, Aug. 2024.
- [47] Z. Ling, Q. Xin, Y. Lin, G. Su, and Z. Shui, "Optimization of autonomous driving image detection based on RFACnv and triplet attention," *Appl. Comput. Eng.*, vol. 77, no. 1, pp. 210-217, Jul. 2024, doi: 10.54254/2755-2721/77/2024MA0067.
- [48] A. G. S. Raj, H. Zhang, V. Abhyankar, S. Mukherjee, E. Zhang, J. Williams, R. Halverson, and J. M. Patel, "Impact of Bilingual CS Education on Student Learning and Engagement in a Data Structures Course," in *Proc. 19th Koli Calling Int. Conf. Comput. Educ. Res.*, 2019, Art. no. 24, pp. 1-10, doi: 10.1145/3364510.3364518.
- [49] J. Wang, Q. Xin, Y. Liu, J. Wang, and T. Yang, "Predicting Enterprise Marketing Decision Making with Intelligent Data-Driven Approaches," *J. Ind. Eng. Appl. Sci.*, vol. 2, no. 3, pp. 12-19, Jun. 2024, doi: 10.5281/zenodo.11357252.
- [50] D. Hendrycks et al., "Measuring mathematical problem solving with the MATH dataset," in *Proc. NeurIPS Datasets and Benchmarks Track*, 2021.
- [51] J. Chen, J. Xiong, Y. Wang, Q. Xin, and H. Zhou, "Implementation of an AI-based MRD Evaluation and Prediction Model for Multiple Myeloma," *Front. Comput. Intell. Syst.*, vol. 6, no. 3, pp. 127-131, Jan. 2024, doi: 10.54097/zJ4MnbWW.