

# Auditing Multi-Agent LLM Reasoning Trees Outperforms Majority Vote and LLM-as-Judge

Wei Yang\*, Shixuan Li\*, Gengshuo Liu, Heng Ping  
Peiyu Zhang, Paul Bogdan†, Jesse Thomason†

University of Southern California  
{wyang930, sli97750, pbogdan, jessetho}@usc.edu

## Abstract

Multi-agent systems (MAS) can substantially extend the reasoning capacity of large language models (LLMs). Most MAS frameworks aggregate agent outputs via simple majority voting, discarding the evidential structure of reasoning traces. Majority voting is brittle under confabulation consensus, where agents share correlated biases and converge on the same incorrect rationale. We introduce AGENTAUDITOR, which moves beyond frequency-based aggregation by organizing agent traces into a Reasoning Tree that explicitly represents agreements and divergences in their reasoning. AGENTAUDITOR resolves conflicts by comparing branch-level evidence at critical divergence points, turning global adjudication into efficient, localized verification. We further propose Anti-Consensus Preference Optimization (ACPO), which trains the adjudicator with evidence-verified preference supervision to reduce conformity to misleading majority cues. Across four MAS frameworks and multiple reasoning benchmarks, AGENTAUDITOR consistently improves aggregation performance over majority voting, with gains of up to 5% absolute accuracy while remaining token-efficient.

## 1 Introduction

Multi-agent systems (MAS) built on large language models (LLMs) are becoming an increasingly common paradigm for complex problem-solving tasks (Zhang et al., 2024a; Qiao et al., 2024; Han et al., 2025). By orchestrating multiple LLM agents through debate and critique (Du et al., 2023; Liu et al., 2024), dynamic computation graphs (Zhang et al., 2024c; Han et al., 2025), and structured communication topologies (Gabriel et al., 2024; Li et al., 2025a), modern MAS can expand reasoning beyond single-pass generation (Park et al.,

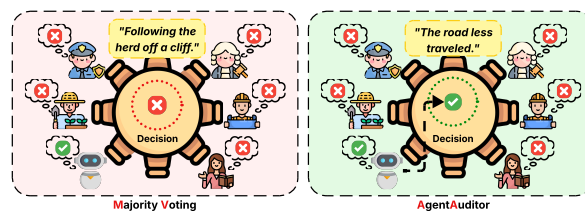


Figure 1: **Majority voting vs. AGENTAUDITOR.** Left: Majority voting leads to an incorrect consensus. Right: AGENTAUDITOR audits localized branch evidence on a reasoning tree to select the correct minority answer.

2023; Zhu et al., 2025).

Despite rich upstream interaction, the final decision is often reduced to majority voting (Hong et al., 2023; Ning et al., 2023). Diverse reasoning traces, intermediate evidence, and agent-level disagreements are compressed into context-free answer counts (Xie et al., 2023; Zhang et al., 2024d).

This compression is problematic because LLM agents are not independent voters: they often share pretraining distributions, alignment biases, and prompt-induced anchors. As a result, multiple agents may reinforce the same erroneous rationale, producing a *confabulation consensus* in which answer frequency no longer reliably tracks validity. We focus on *majority-failure* cases, where the majority answer is wrong but at least one minority trace contains the correct reasoning path.

We propose AGENTAUDITOR, a structure-aware auditing framework for evidence-based aggregation in MAS. AGENTAUDITOR targets cases where the correct reasoning path is present but outnumbered by a correlated erroneous majority (Figure 1), and organizes collective reasoning traces into a *Reasoning Tree* where agreements form shared prefixes and disagreements become explicit branch points. A *Structure-Adaptive Auditor* then performs localized adjudication at *Critical Divergence Points* (CDPs), converting global trace evaluation into targeted branch-level comparison. The core

\* Equal contribution.

† Corresponding authors.

1 Code: <https://github.com/weiyang930/AgentAuditor.git>

framework can be used with a frozen auditor, and we further introduce Anti-Consensus Preference Optimization (ACPO), which fine-tunes the auditor with evidence-verified preference supervision to reduce conformity to popular-but-wrong branches. The main contributions of this work are:

- We identify *confabulation consensus* as a critical failure mode of multi-agent aggregation, where correlated reasoning errors make majority voting suppress correct minority evidence.
- We propose AGENTAUDITOR, a Reasoning Tree-based auditing framework for localized evidence comparison, together with ACPO, a training strategy that reduces conformity to misleading majority cues in the Auditor.
- Extensive experiments across multiple MAS frameworks and reasoning benchmarks show that AGENTAUDITOR consistently improves over majority voting, with particularly strong gains in majority-failure cases where correct answers appear only in minority branches.

## 2 Related Work

### 2.1 LLM-based Multi-Agent Reasoning

LLM-based multi-agent systems (MAS) have become a common strategy for improving complex reasoning by enabling distributed exploration, critique, and coordination (Hong et al., 2023; Chen et al., 2023a; Wu et al., 2024; Zhang et al., 2024c; Ning et al., 2023; Qiao et al., 2024; Pan et al., 2024). Prior work spans (i) pre-structured interaction protocols such as debate-style exchanges or fixed topologies that regulate information flow and verification (Du et al., 2023; Liu et al., 2024), and (ii) self-organizing paradigms that dynamically route, prune, or evolve collaboration graphs for efficiency and diversity (Hu et al., 2024; Zhuge et al., 2024; Zhang et al., 2024c). While these methods primarily optimize how agents interact, they often leave the final adjudication rule comparatively under-specified, which can make the collective outcome vulnerable when agent errors are correlated (Wei et al., 2025; Jiang et al., 2025; Alsadat and Xu, 2024; Lin et al., 2025). A more comprehensive discussion is provided in Appendix C.1.

### 2.2 Evaluation and Adjudication in MAS

A central challenge in multi-agent reasoning is converting diverse reasoning traces into a reliable final decision. Many systems still rely on majority voting or simple ensembling heuristics (Chan et al.,

2023; Chen et al., 2024; Wei et al., 2025), whose reliability is closely tied to the assumption that individual errors are sufficiently independent (Austen-Smith and Banks, 1996). This assumption is fragile in LLM-based MAS: agents often share pretraining distributions, alignment preferences, and prompt-induced anchors, making their errors correlated rather than independent (Jiang et al., 2023; Qiao et al., 2024). Such dependencies can produce *confabulation consensus*, where multiple agents repeat the same unsupported rationale and converge on a plausible but incorrect answer (Mahaut et al., 2024; Coeckelbergh, 2025).

To improve reliability, recent work explores referee agents, debate adjudication, and peer-review style evaluation of reasoning trajectories (Abdelnabi et al., 2023; Du et al., 2023; Wang et al., 2025). These approaches move beyond raw answer counting, but they typically evaluate whole traces or single-agent outputs rather than localizing the specific disagreement that separates competing multi-agent reasoning paths (Lambert et al., 2024; Setlur et al., 2024). This motivates structure-aware adjudication mechanisms that compare branch-level evidence at decision-critical divergences instead of relying solely on answer frequency or full-trace judging. A fuller related-work taxonomy and comparisons appear in Appendix C.2.

## 3 Problem setup.

Given a query  $q$ , a multi-agent system produces a set of agent outputs  $\mathcal{O} = \{o_1, \dots, o_N\}$ , where each output contains a reasoning trace and a final answer  $\hat{y}(o_i)$ . An aggregator maps these outputs to a single prediction,  $\hat{y} = f(\mathcal{O}, q)$ . Majority voting instantiates  $f$  by selecting the most frequent final answer, but this can fail when correlated agents support the same erroneous hypothesis while a correct solution appears only in a minority trace. We call this regime *confabulation consensus*. Our goal is to design an aggregator that is robust to answer multiplicity while remaining sensitive to branch-level evidence. Appendix F.1 provides the full formal setup and objective.

## 4 Methodology

In this section, we describe the main components and overall workflow of AGENTAUDITOR, whose architecture is illustrated in Figure 2. We introduce the Reasoning Tree construction, localized evidence auditing, and the subsequent aggregation

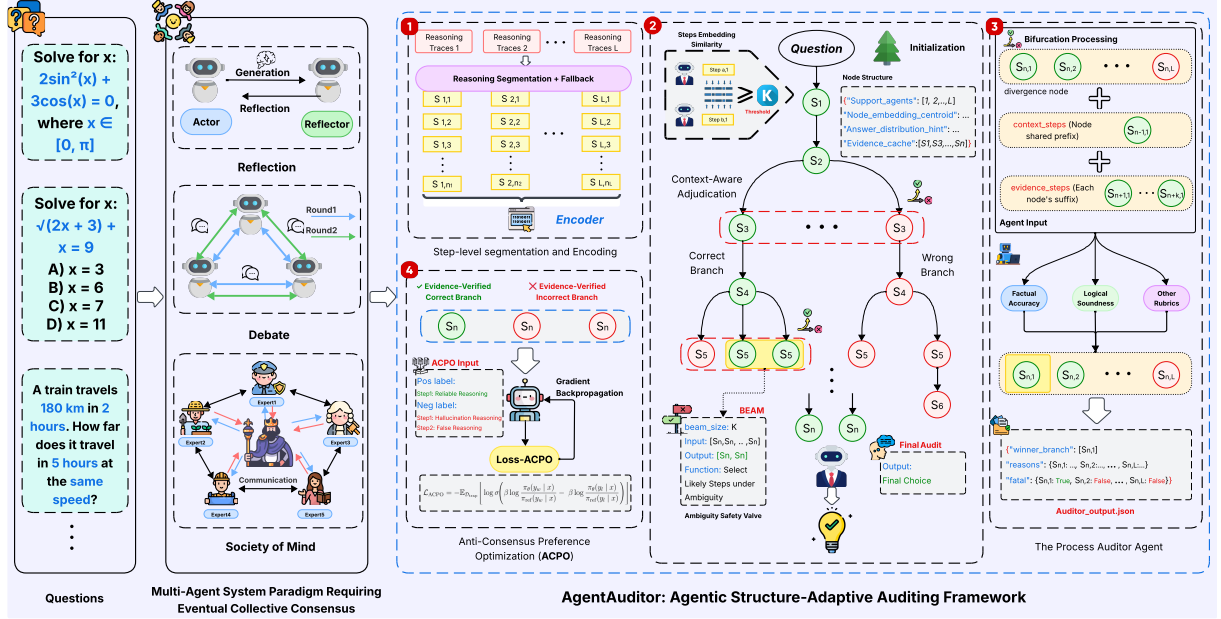


Figure 2: **Overall architecture of AGENTAUDITOR framework.** Given a multi-agent slate of reasoning traces, AGENTAUDITOR performs structural semantic deduplication to construct a compact **Reasoning Tree** of distinct hypotheses. It then audits only decision-critical **Divergence Points** by comparing localized branch evidence and selecting plausible reasoning paths for final aggregation. For learnable auditing, we train the Auditor with Anti-Consensus Preference Optimization on consensus-trap instances.

procedure.

## 4.1 Epistemic Structure Construction

AGENTAUDITOR operates on a structured abstraction of multi-agent reasoning. Given free-form traces, we construct a *Reasoning Tree* that clusters semantically equivalent reasoning steps and exposes disagreements as branch points. Nodes correspond to recurring atomic reasoning steps, edges encode step-to-step progression, and branches expose substantive divergences, providing a compact substrate for localized auditing under shared context.

### 4.1.1 Trace Atomization

Let the raw output of agent  $a_i \in \mathcal{A}$  be  $o_i = (r_i, \hat{y}_i)$ , where  $r_i$  is a trace text and  $\hat{y}_i$  is the final answer. Because direct comparison across agents is brittle under lexical variation, we segment  $r_i$  into shorter text spans representing local reasoning steps. We use a decomposition function  $\Phi : r_i \rightarrow T_i$  to obtain

$$T_i = \Phi(r_i) = \langle s_{i,1}, s_{i,2}, \dots, s_{i,L_i} \rangle, \quad (1)$$

where each step  $s_{i,j}$  is intended to capture a single logical operation, factual assertion, or calculation. This abstraction provides a common unit for cross-agent alignment.

### 4.1.2 Reasoning Tree Generation

We construct a Reasoning Tree  $\mathcal{F} = (V, E)$  by incrementally inserting the atomized traces  $\mathcal{T} = \{T_1, \dots, T_N\}$  into a shared representation space. Each node  $v \in V$  represents a cluster of aligned reasoning steps and maintains a support set  $\mathcal{S}_v$  recording which agents traverse it. All traces are inserted from a virtual root node, which provides a common starting point for tree construction but does not correspond to any agent-generated reasoning step.

**Semantic alignment and branching.** Let  $u$  be the current node and  $s_{i,j}$  the step to be inserted. We compute a representation  $\mathbf{h}(s_{i,j})$  and compare it with the existing children  $\mathcal{C}(u)$  of  $u$  using a similarity function. Let

$$v^* = \underset{v \in \mathcal{C}(u)}{\operatorname{argmax}} \operatorname{sim}(s_{i,j}, v) \quad (2)$$

denote the best matching child. Given a similarity threshold  $\tau$ , if  $\operatorname{sim}(s_{i,j}, v^*) \geq \tau$ , we integrate  $s_{i,j}$  into the corresponding node. Otherwise, we instantiate a new child for  $s_{i,j}$ , creating a new reasoning branch. When multiple steps are merged into the same node, its representation is updated as the running mean of their step representations. In our implementation, similarity is computed over the step representations using cosine similarity.

**Node attribution.** When inserting trace  $T_i$ , if step  $s_{i,j}$  is assigned to node  $v$ , we add the source agent  $a_i$  to  $v$ 's support set:

$$\mathcal{S}_v \leftarrow \mathcal{S}_v \cup \{a_i\}. \quad (3)$$

Thus,  $\mathcal{S}_v$  records the agent support associated with each reasoning branch. Support frequency is not treated by the local Auditor as direct evidence of correctness, although majority support may later be used as auxiliary information during inference.

## 4.2 Structure-Adaptive Evidence Auditing

Given the Reasoning Tree, AGENTAUDITOR introduces an external Auditor to adjudicate among competing branches. The Auditor receives localized evidence extracted from the tree and selects the branch that is best supported by the shared context. Thus, AGENTAUDITOR resolves disagreements through structure-adaptive auditing rather than evaluating entire traces in isolation. The key idea is to localize adjudication to the points where reasoning paths diverge, so the Auditor compares branch-specific evidence under a shared context.

### 4.2.1 Divergence Packet Construction

We traverse the Reasoning Tree  $\mathcal{F}$  from the root and mark a node  $u$  as a *Critical Divergence Point* (CDP) when it has at least two children, i.e.,  $|\mathcal{C}(u)| \geq 2$ . For each CDP  $u$ , we construct a *Divergence Packet* that contains the shared context before the split and compact branch-specific evidence after it. Let  $H_u = \langle h_1, \dots, h_m \rangle$  be the sequence of representative step texts on the path from the root to  $u$ . For each child  $v \in \mathcal{C}(u)$ , let  $E_v = \langle e_{v,1}, \dots, e_{v,\ell_v} \rangle$  be the next  $\ell_v \leq k$  representative step texts along that branch, truncated earlier if another CDP is reached. The packet is

$$\Psi_u = \langle H_u, \{(E_v, \mathcal{S}_v) \mid v \in \mathcal{C}(u)\} \rangle. \quad (4)$$

Here,  $H_u$  and  $E_v$  are text sequences, and  $\mathcal{S}_v \subseteq \mathcal{A}$  records the agents whose traces pass through branch  $v$  as auxiliary support information. This packet focuses the Auditor on the immediate logical consequence of the divergence, rather than downstream verbosity or full-trace noise.

### 4.2.2 Context-Aware Adjudication

Given  $\Psi_u$ , the Auditor acts as a discriminative function  $f_\theta$  that selects the most reliable outgoing branch based on contextual evidence, rather than judging isolated final answers.

**Auditing rubric.** The Auditor evaluates candidate branches under a structured rubric  $\mathcal{R}$  covering Factual Accuracy ( $\mathcal{R}_{\text{FACT}}$ ), which verifies arithmetic, stated facts, and checkable claims; Logical Soundness ( $\mathcal{R}_{\text{LOG}}$ ), which checks deductive validity and coherence with the shared prefix  $H_u$ ; Constraint Adherence ( $\mathcal{R}_{\text{CON}}$ ), which enforces problem-specific constraints; Assumption Validity ( $\mathcal{R}_{\text{ASSUMP}}$ ), which checks whether introduced assumptions are explicit and well supported; and Trace Hygiene ( $\mathcal{R}_{\text{HYG}}$ ), which favors concise and evidence-dense reasoning without unnecessary or unsupported steps.

**Discriminative output.** The Auditor returns a selected branch  $v^*$ :

$$v^* = f_\theta(\Psi_u; \mathcal{R}). \quad (5)$$

This design decouples generation from verification: instead of re-solving the task from scratch, the Auditor compares the local evidence that separates competing branches.

Starting from the root, AGENTAUDITOR iteratively traverses the tree and invokes  $f_\theta(\Psi_u)$  at each CDP. The selected branch determines the primary traversal path, while an optional beam-preservation mechanism can retain additional plausible branches to reduce the risk of irreversible early pruning.

## 4.3 Adaptive Inference via Conditional Beam Search

Because a single early mis-adjudication can irreversibly prune the correct lineage, we introduce a lightweight conditional beam mechanism that can preserve alternative reasoning paths when additional protection against early pruning is desired. At a CDP  $u$ , AGENTAUDITOR can optionally activate a beam-preservation mechanism instead of immediately committing to a single branch. When enabled, multiple plausible outgoing branches can be retained as active candidates and propagated through subsequent divergences, providing additional protection against irreversible early pruning. When disabled, AGENTAUDITOR simply follows the branch selected by the Auditor. If the surviving lineages eventually yield the same final answer, that answer is returned directly. When their final answers differ, the Auditor performs a final adjudication over the surviving candidates, either by comparing their branch-level evidence or by independently solving the original problem and matching its derived answer. Otherwise, AGENTAUDITOR

falls back to the majority-supported candidate as a conservative default.

## 5 Anti-Consensus Preference Optimization

Structure-adaptive auditing (Sec. 4) localizes where disagreements should be resolved, but the Auditor must also remain evidence-driven when support counts are misleading. We therefore propose Anti-Consensus Preference Optimization (ACPO), which constructs preference supervision from evidence-verified cases and trains the Auditor to decouple branch validity from majority support.

### 5.1 The Sycophancy Challenge in Adjudication

When a divergence packet  $\Psi_u$  contains branches with highly imbalanced support, e.g.,  $|\mathcal{S}_{\text{maj}}| \gg |\mathcal{S}_{\text{min}}|$ , an instruction-tuned Auditor may prefer the majority branch even when its evidence is weaker. We characterize this *sycophancy bias* as:

$$P_{\pi_{\text{ref}}}(y_{\text{maj}} | \Psi_u) > P_{\pi_{\text{ref}}}(y_{\text{min}} | \Psi_u), \quad (6)$$

even when  $y_{\text{maj}} \neq y^*$ ,

where  $y^*$  denotes the ground-truth answer when available. Generic preference tuning may not explicitly correct this bias when support frequency remains correlated with correctness in the training data, allowing the model to use support count as a shortcut for validity.

### 5.2 Constructing the ‘‘Consensus Trap’’ Dataset

To counter majority-induced priors, we construct a targeted preference dataset  $\mathcal{D}_{\text{trap}}$  from evidence-verifiable cases with competing correct and incorrect branches. Rather than restricting supervision to majority-failure cases, we include different support configurations so that branch correctness cannot be inferred from support frequency alone. Given multi-agent traces with ground truth  $y^*$ , we retain instances in which correct and incorrect reasoning branches coexist and can be reliably distinguished using the ground-truth outcome. For each retained instance, we build its Reasoning Tree, locate the earliest relevant divergence separating competing outcomes, and identify a branch  $b_{\text{gt}}$  leading to  $y^*$  and a branch  $b_{\text{err}}$  leading to an incorrect answer. We then extract a divergence packet  $\Psi_{\text{div}}$  containing the shared context, short branch evidence, and support statistics, and form preference

triplets  $(x, y_w, y_l)$  with  $x = \Psi_{\text{div}}$ , where  $y_w$  selects  $b_{\text{gt}}$  and  $y_l$  selects  $b_{\text{err}}$ . We balance branch presentation across preference examples to reduce positional and support-related shortcuts, encouraging the Auditor to discriminate branches based on their evidence rather than support frequency.

### 5.3 Optimization Objective

We fine-tune the Auditor policy  $\pi_{\theta}$  from a fixed reference model  $\pi_{\text{ref}}$  using DPO on  $\mathcal{D}_{\text{trap}}$ . For each triplet  $(x, y_w, y_l)$ , where  $y_w$  is the correct choice and  $y_l$  is the incorrect choice under the same audit input  $x$ , we optimize:

$$\mathcal{L}_{\text{ACPO}} = -\mathbb{E}_{\mathcal{D}_{\text{trap}}} \left[ \log \sigma \left( \beta \log \frac{\pi_{\theta}(y_w | x)}{\pi_{\text{ref}}(y_w | x)} - \beta \log \frac{\pi_{\theta}(y_l | x)}{\pi_{\text{ref}}(y_l | x)} \right) \right]. \quad (7)$$

Here  $\beta$  controls the strength of the implicit KL regularization to  $\pi_{\text{ref}}$ . Training on evidence-verified preferences reduces support-driven shortcuts and encourages evidence-grounded auditing.

## 6 Experiments

We evaluate AGENTAUDITOR on four math-intensive and general reasoning benchmarks GSM8K, MATH, AMC, and MMLU. We compare against representative single-agent baselines (*Vanilla*, *CoT*, *Self-Consistency*) and widely used multi-agent collaboration frameworks (*LLM-Debate*, *DyLan*, *GPTSwarm* and *AgentPrune*). AGENTAUDITOR is integrated as a drop-in adjudication module that replaces only the final aggregation stage for fair comparison. All models are instantiated using publicly available instruction-tuned checkpoints Llama3-3B/8B, Qwen2.5-3B/7B (Dubey et al., 2024; Bai et al., 2023). The full experimental setup is provided in Appendix D.1.

### 6.1 RQ1: Does AGENTAUDITOR consistently improve MAS aggregation?

Results in Table 1 demonstrate that AGENTAUDITOR consistently outperforms Majority Voting (MV) and LLM-as-Judge across four MAS architectures and four reasoning benchmarks. Across these settings, AGENTAUDITOR achieves an average absolute improvement of approximately 3.2% over MV, with particularly clear gains on challenging benchmarks such as AMC and GSM8K. These

Table 1: **Main Results (%) on Reasoning Benchmarks.** We compare AGENTAUDITOR against standard Majority Voting (MV) across four different Multi-Agent System (MAS) architectures. Baseline results for single-agent methods are provided for reference. AGENTAUDITOR consistently achieves the best performance across all datasets. Values in parentheses indicate absolute improvement over the MV baseline.

| Category     | Method                | GSM8K               | AMC                 | MATH                | MMLU                | Average             |
|--------------|-----------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| Single-Agent | Vanilla               | 72.76               | 8.03                | 42.85               | 57.99               | 45.41               |
|              | CoT                   | 74.22               | 11.65               | 46.93               | 61.57               | 48.59               |
|              | SC (Self-Consistency) | 80.79               | 12.45               | 51.28               | 68.30               | 53.21               |
| LLM-Debate   | w/ MV                 | 83.52               | 19.28               | 54.85               | 67.59               | 56.31               |
|              | w/ LLM-as-Judge       | 83.17               | 18.07               | 55.20               | 67.02               | 55.87               |
|              | w/ AGENTAUDITOR       | <b>87.43</b> (+3.9) | <b>24.65</b> (+5.3) | <b>57.93</b> (+3.1) | <b>69.65</b> (+2.1) | <b>59.92</b> (+3.6) |
| DyLan        | w/ MV                 | 82.03               | 19.68               | 55.32               | 66.85               | 55.97               |
|              | w/ LLM-as-Judge       | 83.85               | 16.87               | 54.27               | 67.13               | 55.53               |
|              | w/ AGENTAUDITOR       | <b>87.58</b> (+5.5) | <b>23.53</b> (+3.9) | <b>58.12</b> (+2.8) | <b>68.37</b> (+1.5) | <b>59.40</b> (+3.4) |
| GPTSwarm     | w/ MV                 | 84.89               | 15.66               | 56.69               | 69.67               | 56.73               |
|              | w/ LLM-as-Judge       | 82.64               | 18.07               | 53.53               | 68.34               | 55.65               |
|              | w/ AGENTAUDITOR       | <b>88.15</b> (+3.3) | <b>21.38</b> (+5.7) | <b>58.81</b> (+2.1) | <b>70.75</b> (+1.1) | <b>59.77</b> (+3.0) |
| AgentPrune   | w/ MV                 | 84.38               | 16.47               | 54.37               | 69.09               | 56.08               |
|              | w/ LLM-as-Judge       | 83.39               | 19.28               | 54.23               | 67.84               | 56.19               |
|              | w/ AGENTAUDITOR       | <b>87.34</b> (+2.9) | <b>20.56</b> (+4.1) | <b>57.05</b> (+2.7) | <b>70.12</b> (+1.0) | <b>58.77</b> (+2.7) |

results highlight a fundamental limitation of MV: by collapsing reasoning traces into context-free answer counts, it allows a frequent but flawed rationale to override a correct minority solution. In contrast, AGENTAUDITOR constructs a *Reasoning Tree* to expose substantive divergences and performs localized evidence audits at *Critical Divergence Points* (CDPs), allowing aggregation to depend on the quality of branch-level reasoning rather than answer frequency alone. The consistent improvements across diverse MAS architectures further suggest that structure-aware auditing and ACPO provide a general aggregation mechanism that is robust to different forms of consensus and reasoning errors.

## 6.2 RQ2: Performance under Majority Failure (Minority Correct)

Table 2 focuses on the regime that most clearly exposes confabulation consensus: **MinC** instances where the majority answer is wrong even though a correct minority hypothesis exists. This setting directly breaks frequency-based aggregation, so MV deterministically achieves 0% on MinC by construction, reflecting that increased multiplicity of an erroneous hypothesis cannot be voted away. In contrast, AGENTAUDITOR recovers a large fraction of these majority-wrong cases, reaching **65.35%** on GSM8K and **81.82%** on AMC, the best results among all methods and substantially outperforming LLM-as-Judge. This margin suggests that generic judging is insufficient. Resolving majority-wrong cases requires exploiting the struc-

Table 2: **Percent correctness under Majority-Correct (MajC) and Minority-Correct (MinC) regimes.** We evaluate aggregation and adjudication methods on two complementary subsets: **MajC** (majority answer is correct) and **MinC** (majority answer is wrong but a correct minority exists; the “hard” majority-failure regime targeted by confabulation consensus). AGENTAUDITOR substantially improves robustness on MinC while largely preserving performance on MajC.

| Method        | GSM8K MajC   | GSM8K MinC   | AMC MajC     | AMC MinC     |
|---------------|--------------|--------------|--------------|--------------|
| MV            | 100.00       | 0.00         | 100.00       | 0.00         |
| LLM-as-Judge  | 89.60        | 6.70         | 83.33        | 72.73        |
| LLM-as-Solver | 76.87        | 34.65        | 75.00        | 3.64         |
| AGENTAUDITOR  | <b>97.28</b> | <b>65.35</b> | <b>91.67</b> | <b>81.82</b> |

tured disagreement in the slate and adjudicating using branch-level evidence rather than support cues. This also explains why LLM-as-Solver is substantially weaker on MinC: re-solving from scratch is less reliable than auditing the localized evidence that separates competing hypotheses. Importantly, AGENTAUDITOR remains strong on **MajC** cases (97.28% on GSM8K; 91.67% on AMC), indicating that improved minority recovery does not come from broadly sacrificing standard consensus instances, but from selectively correcting the hard majority-wrong failures that voting fundamentally cannot address.

## 6.3 RQ3: Is AGENTAUDITOR Token-Efficient?

Table 3 shows that AGENTAUDITOR improves the accuracy-cost trade-off by auditing only localized

Table 3: **Per-sample token cost averaged across tasks.** AGENTAUDITOR uses substantially fewer total tokens by auditing only decision-critical divergences, while LLM-as-Solver incurs high output cost from re-solving.

| Method        | In Tkns    | Out Tkns   | Total      |
|---------------|------------|------------|------------|
| LLM-as-Judge  | 1698       | 64         | 1762       |
| LLM-as-Solver | 1559       | 487        | 2046       |
| AGENTAUDITOR  | <b>868</b> | <b>105</b> | <b>973</b> |

evidence rather than regenerating full solutions. Concretely, AGENTAUDITOR uses only 973 total tokens per sample, which is a 44.8% reduction relative to LLM-as-Judge (1762) and a 52.4% reduction relative to LLM-as-Solver (2046). The savings primarily come from inputs. AGENTAUDITOR consumes 868 input tokens, roughly half of LLM-as-Judge and LLM-as-Solver, because it audits only decision-critical divergence packets instead of re-feeding entire multi-agent traces. Output cost further differentiates the approaches. LLM-as-Solver spends 487 output tokens to reconstruct solutions from scratch, which is expensive and often redundant because the multi-agent slate has already surfaced the competing hypotheses. By contrast, AGENTAUDITOR remains discriminative and generates only 105 output tokens to resolve local disagreements, so computation is concentrated on the minimal evidence needed for selection. As the number or length of agent traces grows, avoiding repeated full-trace inspection also provides a natural advantage for scaling aggregation to larger reasoning slates.

#### 6.4 RQ4: Does AGENTAUDITOR Generalize Across LLM Backbones?

Table 4 evaluates AGENTAUDITOR under different backbone LLMs. Across all tested backbones and MAS protocols, replacing majority voting with AGENTAUDITOR consistently improves the corresponding MV baseline, indicating that the gains come from the adjudication mechanism rather than being tied to a particular generator family. The improvements are most pronounced on the weaker backbone (LLaMA-3B), where AGENTAUDITOR yields roughly +3.3% to +4.1% points over MV across protocols. As the backbone strengthens to Qwen-3B and Qwen-7B, the MV baselines approach saturation and the remaining errors concentrate on harder cases, so the absolute gains shrink while remaining uniformly positive. The same trend across LLM-Debate, DyLan, GPTSwarm,

Table 4: **Performance (%) of AGENTAUDITOR across different LLM backbones** on GSM8K. AGENTAUDITOR yields consistent improvements over all baselines, demonstrating strong generalization across model scales and architectures.

| Model            | LLaMA-3B            | Qwen-3B             | Qwen-7B             |
|------------------|---------------------|---------------------|---------------------|
| Vanilla          | 46.85               | 83.37               | 90.88               |
| CoT              | 50.14               | 84.56               | 90.98               |
| SC               | 54.21               | 88.60               | 92.95               |
| LLM-Debate w/ MV | 75.84               | 87.14               | 93.63               |
| w/ AGENTAUDITOR  | <b>79.53 (+3.7)</b> | <b>88.73 (+1.6)</b> | <b>94.08 (+0.4)</b> |
| DyLan w/ MV      | 76.47               | 88.10               | 93.15               |
| w/ AGENTAUDITOR  | <b>79.81 (+3.3)</b> | <b>88.93 (+0.8)</b> | <b>93.63 (+0.5)</b> |
| GPTSwarm w/ MV   | 69.19               | 86.78               | 92.27               |
| w/ AGENTAUDITOR  | <b>73.28 (+4.1)</b> | <b>88.09 (+1.3)</b> | <b>92.84 (+0.6)</b> |
| AgentPrune w/ MV | 65.02               | 86.43               | 92.44               |
| w/ AGENTAUDITOR  | <b>68.65 (+3.6)</b> | <b>87.92 (+1.5)</b> | <b>93.15 (+0.7)</b> |

and AgentPrune further suggests that the method is robust to differences in how multi-agent reasoning trajectories are generated and coordinated. Moreover, the persistent gains under stronger backbones indicate that structure-aware adjudication remains useful even when the underlying agents already achieve high standalone reasoning accuracy. These results indicate that AGENTAUDITOR behaves as a backbone-agnostic plug-in for multi-agent aggregation, replacing frequency-driven selection with evidence-based branch auditing.

#### 6.5 RQ5: Do AGENTAUDITOR Modules All Contribute?

Figure 3 isolates three design choices in AGENTAUDITOR: conditional beam search, the step-splitting module, and the embedding encoder. Removing the beam search consistently degrades performance, indicating that maintaining alternative lineages helps prevent irreversible commitment errors from brittle early decisions. Conversely, replacing the step splitter with an LLM-based alternative yields only marginal changes, suggesting that structure-aware auditing is robust to segmentation heuristics as long as coarse semantic boundaries are preserved. Finally, using LLM-based embeddings provides a small but consistent gain in step alignment and divergence detection. These results indicate that while higher-fidelity representations help, the primary performance gains stem from the core structure-aware auditing and anti-consensus training rather than specific upstream components.

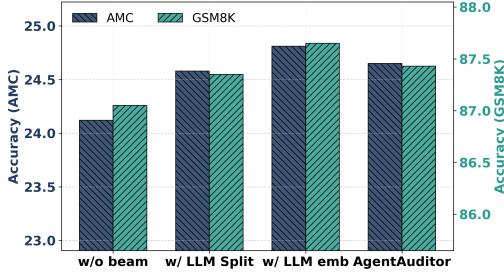


Figure 3: **Key module ablations.** Removing beam search hurts performance; splitting and embedding variants have smaller effects.

## 6.6 RQ6: Case Study

Figure 4 shows an example where Majority Voting fails under confabulation consensus. Although the agents share a plausible high-level plan, their traces diverge at decision-critical steps. Rather than counting final answers or judging full traces, AGENTAUDITOR audits local branch evidence under the shared prefix. The Auditor flags one branch for mixing flavor-specific quantities across different pizza sizes and rejects another for introducing an unsupported entity (“Kate”) into the headcount. These errors are locally plausible and can therefore survive answer-level aggregation, even though they violate basic unit and scope constraints once the relevant reasoning steps are compared directly. By exposing these semantic incompatibilities, AGENTAUDITOR converts global aggregation into localized verification and recovers the correct minority path. The example also illustrates why the tree representation is useful beyond simple answer disagreement: it reveals *where* otherwise fluent reasoning paths become incompatible, allowing the Auditor to focus on the evidence responsible for the final disagreement. The local audits therefore concentrate verification on the precise failure points rather than repeatedly re-evaluating complete reasoning trajectories.

## 7 Conclusion

This paper introduces AGENTAUDITOR, a structure-aware framework for multi-agent LLM aggregation. By organizing reasoning traces into a Reasoning Tree and auditing Critical Divergence Points, the framework can recover correct minority paths that majority voting may suppress. Experiments across MAS frameworks, reasoning benchmarks, and LLM backbones show consistent gains with lower token cost,

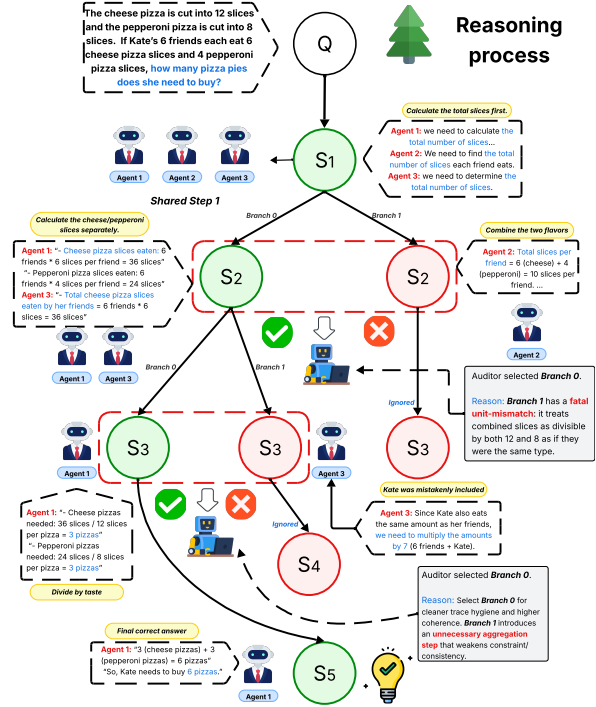


Figure 4: **Case Study.** Majority Voting selects an incorrect consensus, while AGENTAUDITOR detects a unit mismatch and an unsupported population assumption, identifies the invalid reasoning branches, and recovers the correct minority path.

highlighting the value of localized evidence-based aggregation. These results suggest that modeling where reasoning paths diverge is more informative than relying only on agreement at the final-answer level. Future work will explore tool-augmented verification and broader agentic tasks.

## Limitations

AGENTAUDITOR is most effective in majority-failure settings where multiple agents produce meaningfully different reasoning paths and a correct minority hypothesis exists among them. If the traces are nearly identical, all agents follow the same incorrect reasoning, or the correct solution is absent from the candidate set, structural auditing has little evidence with which to recover the answer. Likewise, when the underlying LLM or MAS framework is already very strong and majority voting is close to saturation, the remaining headroom for improvement can be limited. Therefore, the method should not be expected to improve every instance or every multi-agent system; its main advantage lies in cases of confabulation consensus, where an incorrect majority coexists with a distinguishable correct minority path.

## Acknowledgments

The authors acknowledge the support by the National Science Foundation (NSF) under the NSF Award 2243104 under the Center for Complex Particle Systems (COMPASS), the CAREER Award CPS-1453860, NSF Mid-Career Advancement Award BCS-2527046, the U.S. Army Research Office (ARO) under Grant No. W911NF-23-1-0111, the Defense Advanced Research Projects Agency (DARPA) Young Faculty Award and DARPA Director Fellowship Award under Grant Number N66001-17-1-4044, Intel Faculty Awards, and a Northrop Grumman grant. P.B. is also grateful to the National Institutes of Health (NIH) for Grants R01 AG079957, “Interpretable machine learning to synergize brain age estimation and neuroimaging genetics,” and RF1 AG082201, “Neurovascular calcification and AD/DRD in two nonindustrial Native American populations,” and is grateful for the wonderful experience of developing and receiving these two grants. The views, opinions, and/or findings in this article are those of the authors and should not be interpreted as official views or policies of the Department of Defense, the National Institutes of Health, or the National Science Foundation.

## References

- Sahar Abdelnabi, Amr Goma, Sarath Sivaprasad, Lea Schönherr, and Mario Fritz. 2023. Llm-deliberation: Evaluating llms with interactive multi-agent negotiation games. *arXiv preprint arXiv:2310.01444*.
- Shayan Meshkat Alsadat and Zhe Xu. 2024. Multi-agent reinforcement learning in non-cooperative stochastic games using large language models. *IEEE Control Systems Letters*.
- David Austen-Smith and Jeffrey S Banks. 1996. Information aggregation, rationality, and the condorcet jury theorem. *American political science review*, 90(1):34–45.
- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.
- Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201*.
- Ching Chang, Yidan Shi, Defu Cao, Wei Yang, Jeehyun Hwang, Haixin Wang, Jiacheng Pang, Wei Wang, Yan Liu, Wen-Chih Peng, and 1 others. 2025. A survey of reasoning and agentic systems in time series with large language models. *arXiv preprint arXiv:2509.11575*.
- Junzhe Chen, Xuming Hu, Shuodi Liu, Shiyu Huang, Wei-Wei Tu, Zhaofeng He, and Lijie Wen. 2024. Llmarena: Assessing capabilities of large language models in dynamic multi-agent environments. *arXiv preprint arXiv:2402.16499*.
- Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chen Qian, Chi-Min Chan, Yujia Qin, Yaxi Lu, Ruobing Xie, and 1 others. 2023a. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. *arXiv preprint arXiv:2308.10848*, 2(4):6.
- Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2023b. Universal self-consistency for large language model generation. *arXiv preprint arXiv:2311.17311*.
- Yiqun Chen, Jinyuan Feng, Wei Yang, Meizhi Zhong, Zhengliang Shi, Rui Li, Xiaochi Wei, Yan Gao, Yi Wu, Yao Hu, and 1 others. 2026a. Self-compression of chain-of-thought via multi-agent reinforcement learning. *arXiv preprint arXiv:2601.21919*.
- Yiqun Chen, Qi Liu, Yi Zhang, Weiwei Sun, Xinyu Ma, Wei Yang, Daiting Shi, Jiaxin Mao, and Dawei Yin. 2025. Tourrank: Utilizing large language models for documents ranking with a tournament-inspired strategy. In *Proceedings of the ACM on Web Conference 2025*, pages 1638–1652.
- Yiqun Chen, Wei Yang, Erhan Zhang, Shijie Wang, Qi Liu, Zechun Niu, Bin Zhang, Haitao Li, Rui Li, Lingyong Yan, and 1 others. 2026b. Unitymaso: A general rl optimization framework for llm-based multi-agent systems. *arXiv preprint arXiv:2605.26646*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Mark Coeckelbergh. 2025. Llms, truth, and democracy: an overview of risks. *Science and Engineering Ethics*, 31(1):4.
- Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multi-agent debate. In *Forty-first International Conference on Machine Learning*.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

- Xueyang Feng, Zhi-Yuan Chen, Yujia Qin, Yankai Lin, Xu Chen, Zhiyuan Liu, and Ji-Rong Wen. 2024. Large language model-based human-agent collaboration for complex task solving. *arXiv preprint arXiv:2402.12914*.
- Adrian Garret Gabriel, Alaa Alameer Ahmad, and Shankar Kumar Jeyakumar. 2024. Advancing agentic systems: Dynamic task decomposition, tool integration and evaluation using novel metrics and dataset. *arXiv preprint arXiv:2410.22457*.
- Ai Han, Junxing Hu, Pu Wei, Zhiqian Zhang, Yuhang Guo, Jiawei Lu, and Zicheng Zhang. 2025. Joyagents-r1: Joint evolution dynamics for versatile multi-llm agents with reinforcement learning. *arXiv preprint arXiv:2506.19846*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.
- Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiwu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, and 1 others. 2023. Metagpt: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*.
- Shengran Hu, Cong Lu, and Jeff Clune. 2024. Automated design of agentic systems. *arXiv preprint arXiv:2408.08435*.
- Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. *arXiv preprint arXiv:2306.02561*.
- Zhouyang Jiang, Bin Zhang, Airong Wei, and Zhiwei Xu. 2025. Qllm: Do we really need a mixing network for credit assignment in multi-agent reinforcement learning? *arXiv preprint arXiv:2504.12961*.
- Minae Kwon, Sang Michael Xie, Kalesha Bullard, and Dorsa Sadigh. 2023. Reward design with language models. *arXiv preprint arXiv:2303.00001*.
- Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, and 1 others. 2024. Rewardbench: Evaluating reward models for language modeling. *arXiv preprint arXiv:2403.13787*.
- Boyi Li, Zhonghan Zhao, Der-Hong Lee, and Gaoang Wang. 2025a. Adaptive graph pruning for multi-agent communication. *arXiv preprint arXiv:2506.02951*.
- Shawn Li, Chenxiao Yu, Han Wang, Wei Yang, Ryan Rossi, Franck Dernoncourt, Xiyang Hu, Philip Yu, Chaowei Xiao, Huan Zhang, and 1 others. 2026. Fortis: Benchmarking over-privilege in agent skills. *arXiv preprint arXiv:2605.09163*.
- Shixuan Li, Wei Yang, Peiyu Zhang, Xiongye Xiao, Defu Cao, Yuehan Qin, Xiaole Zhang, Yue Zhao, and Paul Bogdan. 2025b. Climatellm: Efficient weather forecasting via frequency-aware large language models. *arXiv preprint arXiv:2502.11059*.
- Muhan Lin, Shuyang Shi, Yue Guo, Vaishnav Tadiparthi, Behdad Chalaki, Ehsan Moradi Pari, Simon Stepputtis, Woojun Kim, Joseph Campbell, and Katia Sycara. 2025. Speaking the language of teamwork: Llm-guided credit assignment in multi-agent reinforcement learning. *arXiv preprint arXiv:2502.03723*.
- Tongxuan Liu, Xingyu Wang, Weizhe Huang, Wenjiang Xu, Yuting Zeng, Lei Jiang, Hailong Yang, and Jing Li. 2024. Groupdebate: Enhancing the efficiency of multi-agent debate using group discussion. *arXiv preprint arXiv:2409.14051*.
- Matéo Mahaut, Laura Aina, Paula Czarnowska, Momchil Hardalov, Thomas Müller, and Lluís Màrquez. 2024. Factual confidence of llms: on reliability and robustness of current estimators. *arXiv preprint arXiv:2406.13415*.
- Xuefei Ning, Zinan Lin, Zixuan Zhou, Zifu Wang, Huazhong Yang, and Yu Wang. 2023. Skeleton-of-thought: Prompting llms for efficient parallel generation. *arXiv preprint arXiv:2307.15337*.
- Bo Pan, Jiaying Lu, Ke Wang, Li Zheng, Zhen Wen, Yingchaojie Feng, Minfeng Zhu, and Wei Chen. 2024. Agentcoord: Visually exploring coordination strategy for llm-based multi-agent collaboration. *arXiv preprint arXiv:2404.11943*.
- Joon Sung Park, Joseph O’Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.
- Feng Peiyuan, Yichen He, Guanhua Huang, Yuan Lin, Hanchong Zhang, Yuchen Zhang, and Hang Li. 2024. Agile: A novel reinforcement learning framework of llm agents. *Advances in Neural Information Processing Systems*, 37:5244–5284.
- Heng Ping, Arijit Bhattacharjee, Peiyu Zhang, Shixuan Li, Wei Yang, Anzhe Cheng, Xiaole Zhang, Jesse Thomason, Ali Jannesari, Nesreen Ahmed, and 1 others. 2025a. Verimoo: A mixture-of-agents framework for spec-to-hdl generation. *arXiv preprint arXiv:2510.27617*.
- Heng Ping, Shixuan Li, Peiyu Zhang, Anzhe Cheng, Shukai Duan, Nikos Kanakaris, Xiongye Xiao, Wei Yang, Shahin Nazarian, Andrei Irimescu, and 1 others. 2025b. Hdllcore: A training-free framework for mitigating hallucinations in llm-generated hdl. In *2025 IEEE International Conference on LLM-Aided Design (ICLAD)*, pages 108–116. IEEE.
- Chen Qian, Zihao Xie, Yifei Wang, Wei Liu, Kunlun Zhu, Hanchen Xia, Yufan Dang, Zhuoyun Du, Weize

- Chen, Cheng Yang, and 1 others. 2024. Scaling large language model-based multi-agent collaboration. *arXiv preprint arXiv:2406.07155*.
- Shuofei Qiao, Ningyu Zhang, Runnan Fang, Yujie Luo, Wangchunshu Zhou, Yuchen Eleanor Jiang, Chengfei Lv, and Huajun Chen. 2024. Autoact: Automatic agent learning from scratch for qa via self-planning. *arXiv preprint arXiv:2401.05268*.
- Yuehan Qin, Li Li, Linxin Song, Wei Yang, Jiate Li, Yuqing Yang, and Yue Zhao. 2026. Memory retrieval for changing preferences. *arXiv preprint arXiv:2606.02976*.
- Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. 2024. Rewarding progress: Scaling automated process verifiers for llm reasoning. *arXiv preprint arXiv:2410.08146*.
- Yu Shang, Yu Li, Keyu Zhao, Likai Ma, Jiahe Liu, Fengli Xu, and Yong Li. 2024. Agentsquare: Automatic llm agent search in modular design space. *arXiv preprint arXiv:2410.06153*.
- Ziyu Wan, Yunxiang Li, Xiaoyu Wen, Yan Song, Hanjing Wang, Linyi Yang, Mark Schmidt, Jun Wang, Weinan Zhang, Shuyue Hu, and 1 others. 2025. Rema: Learning to meta-think for llms with multi-agent reinforcement learning. *arXiv preprint arXiv:2503.09501*.
- Xiao Wang, Jia Wang, Yijie Wang, Pengtao Dang, Sha Cao, and Chi Zhang. 2025. Mars: toward more efficient multi-agent collaboration for llm reasoning. *arXiv preprint arXiv:2509.20502*.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Yuan Wei, Xiaohan Shan, and Jianmin Li. 2025. Lero: Llm-driven evolutionary framework with hybrid rewards and enhanced observation for multi-agent reinforcement learning. *arXiv preprint arXiv:2503.21807*.
- Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, and 1 others. 2024. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First Conference on Language Modeling*.
- Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, James Xu Zhao, Min-Yen Kan, Junxian He, and Michael Xie. 2023. Self-evaluation guided beam search for reasoning. *Advances in Neural Information Processing Systems*, 36:41618–41650.
- Zhenran Xu, Senbao Shi, Baotian Hu, Jindi Yu, Dongfang Li, Min Zhang, and Yuxiang Wu. 2023. Towards reasoning in large language models via multi-agent peer review collaboration. *arXiv preprint arXiv:2311.08152*.
- Bingyu Yan, Zhibo Zhou, Litian Zhang, Lian Zhang, Ziyi Zhou, Dezhuang Miao, Zhoujun Li, Chaozhao Li, and Xiaoming Zhang. 2025. Beyond self-talk: A communication-centric survey of llm-based multi-agent systems. *arXiv preprint arXiv:2502.14321*.
- Wei Yang, Defu Cao, Jiacheng Pang, Muyan Weng, and Yan Liu. 2026. Adaptive collaboration with humans: Metacognitive policy optimization for multi-agent llms with continual learning. *arXiv preprint arXiv:2603.07972*.
- Wei Yang, Jiacheng Pang, Shixuan Li, Paul Bogdan, Stephen Tu, and Jesse Thomason. 2025a. Maestro: Learning to collaborate via conditional listwise policy optimization for multi-agent llms. *arXiv preprint arXiv:2511.06134*.
- Wei Yang and Jesse Thomason. 2026. Learning to deliberate: Meta-policy collaboration for agentic llms with multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 29820–29828.
- Wei Yang, Muyan Weng, Jiacheng Pang, Defu Cao, Heng Ping, Peiyu Zhang, Shixuan Li, Yue Zhao, Qiang Yang, Mengdi Wang, and 1 others. 2025b. Toward evolutionary intelligence: Llm-based agentic systems with multi-agent reinforcement learning. *Available at SSRN 5819182*.
- Wen Ye, Wei Yang, Defu Cao, Yizhou Zhang, Luminyuan Tang, Jie Cai, and Yan Liu. 2026. Ts-reasoner: Domain-oriented time series inference agents for reasoning and automated analysis. *Transactions on Machine Learning Research*.
- Ceyao Zhang, Kaijie Yang, Siyi Hu, Zihao Wang, Guanghe Li, Yihang Sun, Cheng Zhang, Zhaowei Zhang, Anji Liu, Song-Chun Zhu, and 1 others. 2024a. Proagent: building proactive cooperative agents with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17591–17599.
- Erhan Zhang, Yiqun Chen, Zechun Niu, Wei Yang, Xiaochi Wei, Yan Gao, Yi Wu, Yao Hu, and Jiaxin Mao. 2026. Oases: Outcome-aligned search-evaluation co-training for agentic search. *arXiv preprint arXiv:2604.03675*.
- Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. 2024b. Cut

the crap: An economical communication pipeline for llm-based multi-agent systems. *arXiv preprint arXiv:2410.02506*.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, and 1 others. 2024c. Aflow: Automating agentic workflow generation. *arXiv preprint arXiv:2410.10762*.

Wenqi Zhang, Yongliang Shen, Linjuan Wu, Qiuying Peng, Jun Wang, Yueting Zhuang, and Weiming Lu. 2024d. Self-contrast: Better reflection through inconsistent solving perspectives. *arXiv preprint arXiv:2401.02009*.

Guobin Zhu, Rui Zhou, Wenkang Ji, and Shiyu Zhao. 2025. Lamarl: Llm-aided multi-agent reinforcement learning for cooperative policy generation. *IEEE Robotics and Automation Letters*.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. 2024. Gptswarm: Language agents as optimizable graphs. In *Forty-first International Conference on Machine Learning*.

## A Potential Risks

Although AGENTAUDITOR improves robustness to misleading majority consensus, it does not guarantee correct aggregation. The Auditor may still make incorrect branch-level judgments, especially when the available reasoning traces are highly similar, contain subtle shared errors, or do not include a correct hypothesis. Moreover, the structured auditing process may create a stronger appearance of reliability than simple voting, which could lead users to over-trust the final decision. Its effectiveness may also depend on the quality and diversity of the candidate reasoning generated by the underlying multi-agent system. In open-ended or safety-critical domains, locally plausible but incorrect evidence may remain difficult to detect without external grounding. Therefore, the framework should be used with appropriate caution in high-stakes applications, where additional verification, external tools, or human oversight may be necessary.

## B Use of AI Assistants

We used ChatGPT for limited writing assistance, primarily to improve grammar and presentation of the manuscript. We also used Claude for code-quality improvement and debugging support during development. All technical content, experimental results, analyses, and scientific conclusions were produced, reviewed, and verified by the authors.

## C Related Work

### C.1 LLM-based Multi-Agent Collaboration for Reasoning

Recent advances in large language models (LLMs) (Ping et al., 2025b; Chang et al., 2025; Ye et al., 2026; Li et al., 2025b; Chen et al., 2025; Yang et al., 2025b; Qin et al., 2026) have inspired a surge of interest in multi-agent systems (MAS) as a means to extend single-model reasoning capacity (Wu et al., 2024; Chen et al., 2026b; Li et al., 2026; Yan et al., 2025; Zhang et al., 2026; Yang et al., 2026; Ping et al., 2025a). Early frameworks demonstrate that collective interaction among LLMs can mitigate individual cognitive limitations, enabling richer reasoning through distributed exploration and mutual critique (Hong et al., 2023; Chen et al., 2023a; Jiang et al., 2023; Ning et al., 2023; Qiao et al., 2024; Pan et al., 2024). Existing methods largely fall into two paradigms. *Prestructured collaboration* imposes

fixed interaction topologies, such as chains, trees, or debate graphs, to orchestrate discussion and verification (Du et al., 2023; Liu et al., 2024; Qian et al., 2024). These designs enhance coherence but remain bounded by static, human-specified structures that cannot adapt to dynamic task complexity. In contrast, *self-organizing paradigms* dynamically adjust collaboration graphs through routing, pruning, or evolutionary mechanisms, exemplified by frameworks like DyLan, GPTSwarm, and AFLOW (Yang et al., 2025a; Shang et al., 2024; Zhang et al., 2024b; Zhuge et al., 2024; Zhang et al., 2024c; Yang and Thomason, 2026; Chen et al., 2026a). While these approaches improve communication efficiency and diversity, most focus on *coordination dynamics* rather than epistemic reliability (Wan et al., 2025; Peiyuan et al., 2024; Feng et al., 2024). Their optimization is procedural, determining who speaks or when, rather than epistemological, leaving the collective still vulnerable to correlated reasoning errors and unverified consensus (Wei et al., 2025; Jiang et al., 2025; Alsatat and Xu, 2024; Lin et al., 2025).

### C.2 Evaluation and Adjudication in Multi-Agent Reasoning

While multi-agent coordination improves the diversity of reasoning, ensuring the *correctness* of collective outcomes remains a core challenge. Early systems aggregate results through *majority voting* or rule-based ensembling (Chan et al., 2023; Chen et al., 2024; Wei et al., 2025), assuming that consensus approximates truth. In practice, however, agents trained on similar distributions often exhibit *correlated errors*, causing the majority to converge on plausible yet incorrect answers, a phenomenon described as the "tyranny of the majority" (Du et al., 2023; Liu et al., 2024). To improve reliability, recent frameworks introduce *referee* or *reviewer* agents that evaluate peers' reasoning trajectories (Abdelnabi et al., 2023; Du et al., 2023; Zhang et al., 2024b). Debate-style protocols exchange arguments before a third-party judge determines the winner, while *self-consistency* and *peer review* methods aggregate explanations instead of final answers (Chen et al., 2023b; Xu et al., 2023; Wang et al., 2025). Although these approaches enhance reasoning control, they often depend on heuristic templates or task-specific rules rather than a principled notion of evidence and logical soundness. Meanwhile, research in *LLM evaluation and verification* explores automatic rea-

soning validation through reward modeling (Kwon et al., 2023; Setlur et al., 2024) and reflective self-evaluation (Xie et al., 2023; Zhang et al., 2024d), yet these methods focus on single-agent reflection rather than distributed adjudication across agents.

## D Experiments

### D.1 Experimental Setup

**Tasks and Datasets.** We evaluate our framework on four representative reasoning benchmarks that collectively assess both domain-specific and general reasoning abilities of large language model (LLM) agents. **GSM8K** focuses on grade-school mathematical reasoning through structured word problems. **MATH** and **AMC** provide competition-level problems requiring multi-step symbolic reasoning and algebraic manipulation. To measure general reasoning competence, we further include the **MMLU** benchmark, which spans multiple academic and professional domains. For all datasets (Cobbe et al., 2021; Hendrycks et al., 2020), we follow the official evaluation splits and report the standard metrics: *Solve Rate* for GSM8K, MATH, and AMC, and *Accuracy* for MMLU.

**Baselines.** We compare our framework, AGENTAUDITOR, against a comprehensive suite of single- and multi-agent reasoning paradigms. Single-agent baselines include the *Vanilla* prompting strategy, *Chain-of-Thought* (CoT) (Wei et al., 2022), and *Self-Consistency* (SC) (Wang et al., 2022). Multi-agent frameworks encompass debate-based and collaborative systems such as *LLM-Debate*, and *DyLan*. We also consider dynamic workflow architectures, including *AgentPrune* and *GPTSwarm*. To ensure a fair comparison, AGENTAUDITOR is integrated as a plug-in adjudication module that replaces the final voting stage with our evidence-audited selection process, while keeping the upstream candidate generation identical across all methods.

**Implementation Details.** Following common practice, we employ three agents and three rounds of interaction unless otherwise specified. All models are instantiated from publicly available instruction-tuned checkpoints, including Llama-3.1-8B-Instruct, Llama-3.2-3B-Instruct, Qwen2.5-7B-Instruct, and Qwen2.5-3B-Instruct (Dubey et al., 2024; Bai et al., 2023). Across all settings, we report the mean performance over three random seeds. To ensure a fair and reproducible compar-

ison, we keep the overall experimental pipeline aligned with the baselines, including identical evaluation protocols and consistent prompt formatting. Unless explicitly noted, the AGENTAUDITOR module shares the same backbone family as the generator agents and does not require additional supervised training beyond the preference optimization applied in our method variants. Unless otherwise specified, we report results averaged over three random seeds. For a subset of experiments, we report results from a single run due to computational cost.

For multi-agent trace generation, we use nucleus sampling with  $\text{top-}p = 0.95$ , while varying the decoding temperature within  $[0.7, 1.0]$  to encourage diverse but stable reasoning across agents. The maximum generation length is searched within 512–1024 tokens depending on the task and MAS protocol. The Auditor uses low-temperature decoding with  $T \in [0.0, 0.3]$  to provide stable, single-pass adjudication without additional self-consistency or multi-sample voting. For preference optimization, we apply LoRA with rank  $r = 16$ , scaling factor  $\alpha = 32$ , and dropout 0.05. We search the learning rate over  $\{1 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}, 5 \times 10^{-4}\}$  and train for 1–3 epochs, selecting the configuration according to validation performance. For the structural aggregation module, we search the reasoning-tree merge threshold  $\tau$  over a broad range within  $[0.5, 1.0]$ , together with several candidate beam sizes and decoding budgets. Across these settings, we select hyperparameters using the same validation criterion and keep the overall search budget comparable across methods. All training experiments are run on 32 NVIDIA A100 GPUs in total, and we fix all remaining optimization and infrastructure settings to match the corresponding baseline configurations as closely as possible.

**Prompt Templates.** To support transparency and reproducibility, we document the instruction prompt used by the AGENTAUDITOR module. The prompt casts the model as a process auditor that selects the most reliable reasoning branch at a divergence point. Each candidate branch is provided as a structured packet containing (i) the number of supporting agents, (ii) a shared *Context* prefix, (iii) local *Evidence* steps from that branch, and (iv) the answer distribution among supporting agents as an auxiliary hint.

### AGENTAUDITOR Prompt

You are a Process Auditor for math reasoning.

You will see multiple candidate BRANCHES that diverge from a shared reasoning prefix.

Select the branch that is most reliable based on PROCESS QUALITY, not popularity.

Evaluate branches using:

- 1) Factual integrity: correct arithmetic/algebra, with no obvious calculation errors.
- 2) Logical coherence: steps follow with no unsupported leaps or contradictions.
- 3) Constraint coverage: respects the question, units, bounds, and required format.
- 4) Assumption validity: assumptions are explicit and reasonable.
- 5) Trace hygiene: reasoning is concise and evidence-dense.

Return STRICT JSON only:

<outputformat>

Original Problem:

<question>

Candidates:

<body>

**Candidate Branch Packet Format.** Each candidate in <body> is formatted as follows.

### Candidate Branch Block Format

[Branch b]

Support: <support>

Context (shared prefix, last steps):

- <ctx\_step\_1>

- <ctx\_step\_2>

...

Branch evidence (next steps):

- <ev\_step\_1>

- <ev\_step\_2>

...

Answer distribution among agents on this branch (hint only):

<answer\_distribution>

## E Theoretical Analysis: Why Voting Fails Under Correlated Errors and Why Structure Helps

This section provides a *supporting* theoretical perspective on the failure mode we target. Our goal is not to introduce a new theory or claim general guarantees for arbitrary LLM behaviors. Instead, we use a simple analytical lens to explain (i) why majority voting can fail when agent errors are correlated, leading to confabulation consensus, and (ii) why organizing redundant reasoning into semantic branches enables evidence-based adjudication beyond raw answer frequency. The analysis should be read as an interpretation of the underlying mechanism rather than a complete characterization of all possible deployments.

### E.1 Majority Voting Under Correlated Correctness

Let  $X_i \in \{0, 1\}$  be the correctness indicator from Eq. (12), with  $\mathbb{P}(X_i = 1) = p$  for all  $i$ . Define the sample mean  $\bar{X} = \frac{1}{N} \sum_{i=1}^N X_i$  and let  $\rho$  denote the average pairwise correlation in Eq. (13). Under this homogeneous Bernoulli setting,

$$\text{Var}(\bar{X}) = \frac{p(1-p)}{N} (1 + (N-1)\rho). \quad (8)$$

**Proposition E.1** (Failure of independence assumption). *If  $\rho = 0$  and  $p > 1/2$ , then  $\text{Var}(\bar{X}) = O(1/N)$  and  $\bar{X}$  concentrates around  $p$ , recovering the classical CJT intuition. If  $\rho > 0$  is bounded away from 0, then  $\text{Var}(\bar{X})$  does not vanish as  $N \rightarrow \infty$  and approaches  $p(1-p)\rho$ . Thus, increasing the number of agents alone does not guarantee increasingly reliable majority decisions under persistent positive correlation.*

**Interpretation.** When errors are correlated, the ensemble can behave like an “echo chamber”: a shared bias may simultaneously shift many agents toward the same incorrect answer. Consequently, even a large  $N$  may provide substantially fewer independent signals than the raw number of agents suggests. This motivates aggregation mechanisms that examine the structure and validity of the underlying reasoning rather than relying only on answer multiplicity.

## E.2 Reasoning Trees as Semantic Deduplication and Auditing as Discrimination

Our method organizes the multiset of raw outputs  $\mathcal{O} = \{o_1, \dots, o_N\}$  into a set of semantic reasoning branches. Abstractly, let

$$\mathcal{T} : \{o_1, \dots, o_N\} \mapsto \{B_1, \dots, B_K\}, \quad K \leq N, \quad (9)$$

where each branch  $B_k$  represents a distinct reasoning lineage and may be supported by multiple agents. When many agents produce semantically redundant variants of the same trajectory,  $K$  can be substantially smaller than  $N$ . The Reasoning Tree aligns such redundant reasoning into shared nodes and lineages, reducing the extent to which repeated variants are treated as independent evidence.

At each divergence, the resulting decision can be viewed locally as an evidence-based comparison among competing branches. For intuition, consider a simplified two-branch case with one correct branch  $B_{\text{cor}}$  and one erroneous branch  $B_{\text{err}}$ . Let the Auditor choose between them using the corresponding divergence packet containing shared context and branch-specific evidence. Define its *discrimination accuracy* as

$$q = \mathbb{P}(\text{Auditor selects } B_{\text{cor}} \mid B_{\text{cor}}, B_{\text{err}}). \quad (10)$$

**Proposition E.2** (Branch-level auditing reduces direct multiplicity bias). *Suppose an observed MAS slate contains  $n_{\text{err}}$  outputs supporting an erroneous branch  $B_{\text{err}}$  and  $n_{\text{cor}}$  outputs supporting a correct branch  $B_{\text{cor}}$ , with  $n_{\text{err}} > n_{\text{cor}}$ . Under perfect answer extraction, majority voting deterministically selects  $y_{\text{err}}$ . After the traces are organized into semantic branches, the local decision is no longer determined solely by the number of redundant outputs. In the two-branch toy case, the probability that the Auditor selects the correct branch is  $q$  from Eq. (10). Thus, branch-level auditing can recover consensus-error instances that majority voting necessarily misses in this setting; when  $q > 1/2$ , the Auditor favors the correct branch more often than random selection.*

**Interpretation.** The key shift is from a purely frequency-driven decision (“how many agents said this?”) toward a *validity-sensitive* comparison (“which branch is better supported by evidence and logic?”). The Reasoning Tree groups redundant reasoning while exposing the semantic points at which competing hypotheses diverge, allowing

the Auditor to concentrate verification on these localized differences. Support information may still serve as auxiliary context, but it no longer uniquely determines the local branch decision. This perspective helps explain why the approach is particularly useful in majority-failure settings where correlated errors inflate the apparent support of an incorrect hypothesis while a distinguishable correct minority remains available.

## F Method

### F.1 Problem Formulation

We study answer aggregation for a single query (problem)  $q$  with a unique ground-truth answer  $y^*$ . A multi-agent system (MAS) instantiates  $N$  agents  $\mathcal{A} = \{a_1, \dots, a_N\}$  and produces a set of candidate outputs  $\mathcal{O} = \{o_1, \dots, o_N\}$ , where each  $o_i$  contains an agent’s reasoning trace and a final answer. Let  $\hat{y}(o_i)$  denote the final answer extracted from  $o_i$ .

**Majority voting.** A standard aggregator is majority voting (MV),

$$\hat{y}_{\text{MV}} = \underset{y}{\operatorname{argmax}} \sum_{i=1}^N \mathbb{I}[\hat{y}(o_i) = y], \quad (11)$$

which estimates the mode of the empirical answer distribution. Under the classical Condorcet Jury Theorem (CJT), if each agent is correct with probability  $p > 1/2$  and the correctness events are independent, then  $\mathbb{P}(\hat{y}_{\text{MV}} = y^*) \rightarrow 1$  as  $N \rightarrow \infty$ .

**Confabulation consensus (correlated errors).** In LLM-based MAS, agent errors can be highly correlated due to shared pretraining, prompt anchoring, and interaction dynamics, leading to *confabulation consensus*: many agents converge to the same (but incorrect) answer with similar reasoning patterns. To formalize this regime, define the correctness indicator

$$X_i = \mathbb{I}[\hat{y}(o_i) = y^*], \quad i \in \{1, \dots, N\}, \quad (12)$$

and the average pairwise correlation

$$\rho = \frac{2}{N(N-1)} \sum_{1 \leq i < j \leq N} \operatorname{Corr}(X_i, X_j). \quad (13)$$

We refer to *hard* instances as those for which  $\rho$  is non-negligible (often  $\rho > 0$ ), violating the independence assumption in CJT. Equivalently, one can view correlated errors as arising from a latent “bias”

variable  $Z$  that, when activated, increases the probability that many agents produce the same incorrect answer  $y_{\text{err}} \neq y^*$  with  $Z \sim \text{Bernoulli}(\pi)$ :

$$\begin{aligned} \mathbb{P}(\hat{y}(o_i) = y_{\text{err}} \mid Z = 1) &\gg \\ \mathbb{P}(\hat{y}(o_i) = y_{\text{err}} \mid Z = 0). \end{aligned} \quad (14)$$

In this regime, MV can be dominated by the multiplicity of a shared error mode, i.e.,  $\hat{y}_{\text{MV}} = y_{\text{err}}$  even when a minority of agents output  $y^*$ .

**Goal.** We seek an aggregation procedure  $f$  that maps a set of candidate outputs  $\mathcal{O}$  to a single final prediction  $\hat{y}$ ,

$$\hat{y} = f(\mathcal{O}, q), \quad (15)$$

such that it is (i) *robust to quantity*: it does not automatically privilege an answer solely because it appears many times, and (ii) *sensitive to validity*: it can recover  $y^*$  when the correct reasoning exists but is outnumbered by correlated errors. Our method operationalizes this goal by (a) *structural semantic deduplication* of redundant reasoning and (b) *auditing* at critical divergence points, moving beyond purely frequency-based aggregation toward evidence-based discrimination.

## F.2 Case Study

Figure 5 illustrates an example where *Majority Voting* fails due to confabulation consensus, while AGENTAUDITOR succeeds by focusing adjudication on decision-critical semantic divergences. All agents share an apparently consistent macro-plan (compute slice totals, then convert slices to pies). However, this shared prefix hides two qualitatively different error modes that only become salient once we inspect *where* the reasoning topology first forks into incompatible semantic commitments (CDPs). In this instance, two out of three agents reach incorrect answers through different yet locally fluent transformations, so aggregating by final answer amplifies error even when the traces appear coherent at a glance.

**CDP-1: Type/unit mismatch induced by premature aggregation.** The first substantive divergence occurs when one agent collapses flavor-specific quantities into a single scalar “total slices per friend” (6+4) and proceeds as if this combined count could be converted under both denominators (12 slices per cheese pie and 8 slices per pepperoni pie). This is a latent *type error*: “cheese-slice” and “pepperoni-slice” are not interchangeable because they belong to distinct conversion regimes. Once

merged, the quantity can no longer be mapped consistently back to pie counts without reinstating the per-flavor partition. AGENTAUDITOR identifies this as an early inconsistency and favors the branch that preserves flavor-wise accounting (36 cheese slices and 24 pepperoni slices), effectively enforcing a basic invariant for correctness: unit-consistent conversions must be completed before any cross-type summation.

**CDP-2: Constraint violation via an unsupported multiplicative assumption.** A later divergence appears during the slice-to-pie conversion stage, where another agent introduces an extra “ $\times 7$ ” factor by implicitly including Kate as an additional eater (6 friends + Kate), even though the statement specifies only the friends’ consumption. Unlike CDP-1, this is a *scope/quantifier error* that changes the population being counted. AGENTAUDITOR identifies this branch as inconsistent with the problem statement and favors the alternative grounded in the stated entities and quantities. The Auditor also favors *trace discipline*, avoiding unnecessary transformations or unsupported assumptions that can remain locally plausible and survive naive voting.

**Why structural tree auditing helps.** This case highlights how structural tree auditing reshapes the failure surface relative to majority voting. Majority voting operates only on final answers and therefore cannot distinguish between unit-consistent, statement-grounded reasoning and fluent but invalid reasoning that violates an earlier invariant or constraint. In contrast, the Reasoning Tree exposes the earliest semantic fork points, allowing the Auditor to localize verification to a small number of CDPs rather than re-evaluating entire traces. By identifying the unit mismatch at CDP-1 and the scope violation at CDP-2, AGENTAUDITOR prevents locally plausible errors from dominating aggregation simply because they receive greater support. The selected reasoning path preserves the relevant quantities and constraints, performing clean per-flavor conversions (36/12 and 24/8) and yielding the correct purchase count: 3 cheese pies and 3 pepperoni pies, totaling 6. More broadly, the example shows that structural auditing helps by exposing the specific semantic commitments responsible for disagreement rather than relying only on final-answer selection.

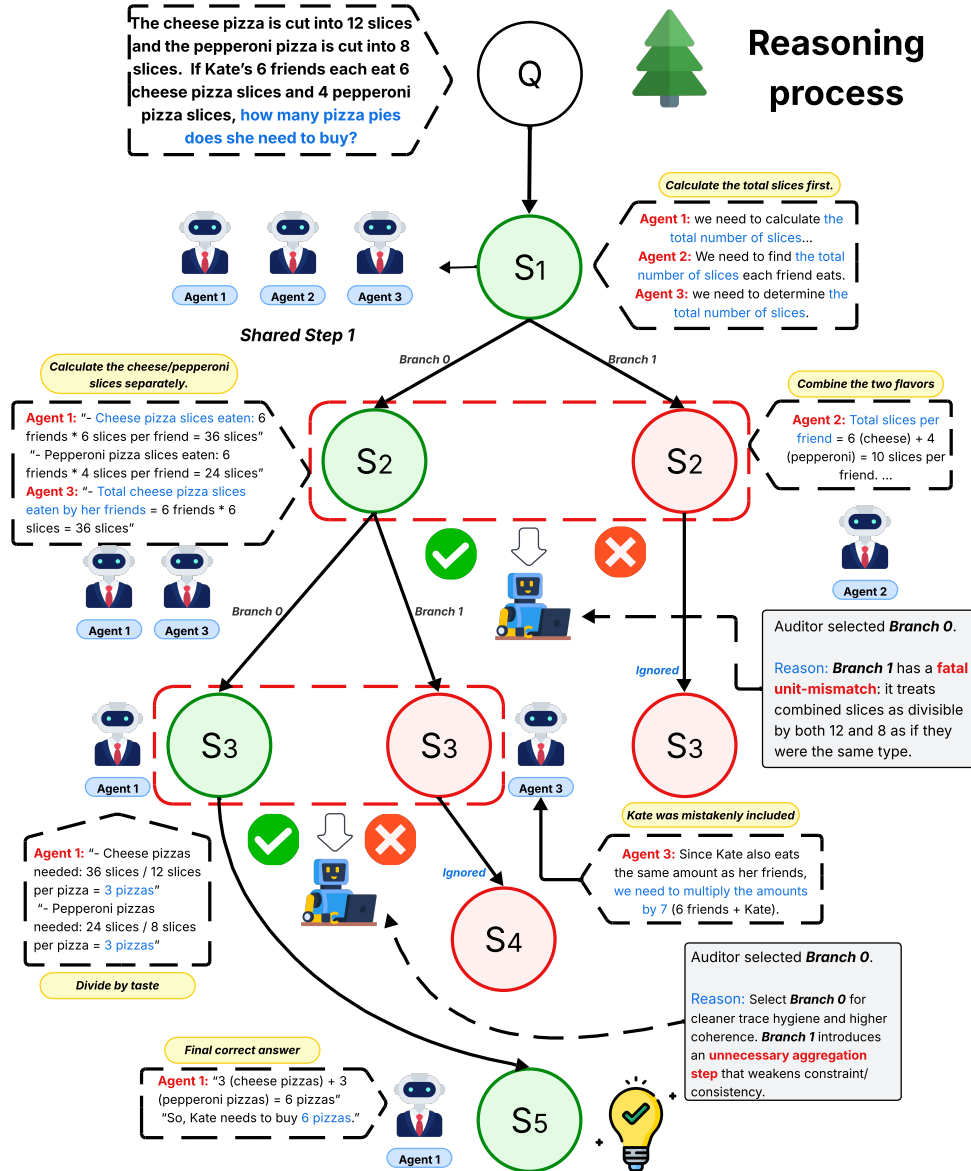


Figure 5: **Case Study.** An example where majority voting converges to an incorrect consensus due to correlated fluent errors. AGENTAUDITOR audits only decision-critical divergence points, detects a flavor-unit mismatch and an unsupported population assumption, prunes the invalid branches early, and selects the correct per-flavor conversion path.