ARMAD: ADAPTIVE ROUTING FOR HETEROGENEOUS MULTI-AGENT DEBATE IN LARGE LANGUAGE MODEL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-agent debate (MAD) can improve large language model reasoning, but fixed-depth protocols allocate the same interaction budget to every input, potentially wasting calls when independent answers already agree and risking harmful over-deliberation. We introduce ARMAD, a training-free, black-box-compatible controller that adapts both debate initiation and duration within a fixed heterogeneous model pool. After collecting independent responses, Pre-debate Agreement Routing (PAR) bypasses debate when agreement among rule-normalized final answers reaches a predefined routing threshold. For routed inputs, the Early Agreement Stopping Evaluator (EASE) recomputes the same agreement score after each synchronized revision round and terminates interaction when a separate stopping condition is satisfied or the maximum depth is reached. Responses from the selected round are then aggregated using a deterministic voting rule. We evaluate ARMAD on five reasoning benchmarks against same-pool fixed-depth controls and call-matched routing controls. ARMAD improves over Hetero-Vote (Round-0 voting) on every benchmark and achieves 82.7% macro-average accuracy with 3.96 model calls per item, 67.0% fewer than same-pool fixed three-round debate. At the same call budget, agreement-guided routing achieves higher accuracy than random and response-length allocation on all five benchmarks. Shallow fixed-round policies achieve modestly higher macro-average accuracy at larger call budgets, revealing an accuracy–call trade-off. Together, these results establish answer agreement as an effective control signal for adaptive, call-efficient heterogeneous debate.

1 INTRODUCTION

Large language models (LLMs) have demonstrated strong reasoning capabilities across mathematical problem solving, knowledge-intensive question answering, and complex decision-making tasks. Nevertheless, single-model reasoning remains vulnerable to incomplete knowledge, early reasoning errors, and ineffective self-correction. Multi-agent debate (MAD) provides a complementary test-time reasoning paradigm in which multiple agents independently construct candidate solutions, exchange rationales, critique one another, and revise their predictions through structured interaction (Du et al., 2024; Liang et al., 2024). By exposing models to alternative perspectives, MAD can recover errors that are difficult for a single agent to identify. Recent heterogeneous collaboration frameworks show that agents drawn from different model families can contribute complementary reasoning patterns and knowledge, although model diversity alone does not guarantee a correct consensus (Chen et al., 2024; Wang et al., 2025).

Despite its promise, conventional MAD protocols typically allocate the same number of debate rounds to every input. However, the need for further interaction varies across inputs and over the course of a debate: some questions already produce aligned independent answers, others remain contested, and some discussions converge before reaching the prescribed depth. A fixed schedule cannot distinguish among these cases and therefore allocates the same number of interaction rounds or model calls despite their different interaction needs. Moreover, additional interaction is not uniformly beneficial: it may overturn a correct initial decision or encourage sycophantic convergence rather than genuine error correction (Choi et al., 2025; Pitre et al., 2025; Wang et al., 2024a). The

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

central problem is therefore not whether debate should always be used or avoided, but how debate computation should be allocated: whether interaction should be initiated for a given input and, once initiated, when it should terminate.

To address this problem, we propose **ARMAD**, a training-free framework that uses agreement among the agents' final predictions to allocate debate computation within a fixed heterogeneous model pool. After collecting independent Round 0 responses, Pre-debate Agreement Routing (PAR) bypasses debate when answer agreement reaches a predefined threshold and initiates debate otherwise. For inputs routed to debate, the Early Agreement Stopping Evaluator (EASE) recomputes the same agreement measure after each synchronized revision round and terminates the discussion once a predefined agreement condition is met or the maximum depth is reached. ARMAD then applies deterministic voting to the responses from the selected round. Because the controller uses only rule-normalized final predictions, it requires neither additional training nor access to model logits and is compatible with black-box LLM APIs. Importantly, agreement is used to allocate interaction rather than to certify correctness.

We evaluate ARMAD on five reasoning benchmarks spanning competition-level mathematics, grade-school arithmetic, broad academic knowledge, and graduate-level science. ARMAD achieves higher observed accuracy than Hetero-Vote on all five benchmarks while averaging 3.96 online chat calls per item, 67.0% fewer than fixed three-round Hetero-MAD. We further compare ARMAD with fixed Hetero-MAD at one, two, and three rounds using the same heterogeneous model pool, thereby evaluating adaptive allocation across the fixed-depth accuracy–cost trade-off rather than only against the maximum-depth configuration. Call-matched random and response-length allocation controls achieve lower accuracy than ARMAD on all five datasets, indicating that its gains are not explained by the amount of computation alone. At the same time, analyses of wrong-consensus cases reveal an important boundary: agents can reach high agreement on a shared error. Together, these findings support agreement as a practical allocation signal rather than a correctness certificate.

Our main contributions are summarized as follows:

- We propose ARMAD, a training-free, black-box-compatible method that uses answer agreement to determine whether debate should be initiated and, once initiated, when it should terminate within a fixed heterogeneous model pool.
- We evaluate ARMAD across five reasoning benchmarks against same-pool heterogeneous voting, fixed debate depths from one to three rounds, and call-matched non-agreement routers using the same examples and stored response trajectories.
- We characterize the accuracy–computation trade-off of agreement-guided allocation and analyze unanimous-but-incorrect predictions and minority-expert cases as key limitations of agreement-based control.

2 Related Work

Multi-Agent Debate. MAD allows LLMs to exchange intermediate rationales and revise their answers over multiple rounds (Du et al., 2024; Liang et al., 2024). The paradigm has also been extended to LLM-based evaluation, where multiple evaluator agents deliberate over generated outputs (Chan et al., 2024). However, the benefit of interaction depends on the debate protocol and available budget (Smit et al., 2024). Strong single-agent strategies can match some discussion protocols (Wang et al., 2024a), and gains attributed to debate may partly arise from aggregating independent answers rather than from deliberation itself (Choi et al., 2025). These findings make it necessary to separate pool strength, initial voting, and later interaction. We therefore compare every constituent, Hetero-Vote, and fixed-depth Hetero-MAD using the same agents, examples, prompts, and answer-cluster decision rule.

Agent Heterogeneity and Aggregation. Early MAD systems commonly obtain diversity from multiple samples or prompted roles of a shared backbone (Du et al., 2024; Liang et al., 2024; Zhou & Chen, 2025). Other systems combine evidence across model families. ReConcile uses round-table consensus among different LLMs (Chen et al., 2024), while Mixture-of-Agents separates heterogeneous proposers from a downstream aggregator (Wang et al., 2025). Self-consistency similarly shows that answer-frequency aggregation can improve reasoning without inter-agent in-

teraction (Wang et al., 2023). Heterogeneous performance, however, can conflate the strength of individual models with the effects of voting and debate, and agreement can amplify shared errors or socially induced convergence (Pitre et al., 2025; Estornell & Liu, 2024). We consequently report all standalone agents and analyze wrong-consensus and minority-expert cases rather than treating cross-model agreement as independent evidence of correctness.

Adaptive Control. Recent work treats multi-agent collaboration as an input-dependent decision. Dynamic agent networks adapt interaction structures (Liu et al., 2024), and MasRouter learns to select collaboration modes, roles, and model backbones (Yue et al., 2025). iMAD trains a debate-decision classifier from structured self-critique features (Fan et al., 2026); adaptive stability detection models consensus dynamics for stopping (Hu et al., 2025); and SELENE combines confidence–likelihood misalignment and semantic disagreement for debate initiation with evidence-weighted judging (Verma et al., 2026). ARMAD introduces a unified, training-free answer-only controller that uses the same agreement score to decide both whether debate begins and when it ends within a fixed heterogeneous model pool, requiring neither learned routers, internal confidence scores, nor auxiliary judges. Its matched-trajectory evaluation controls model composition and, for routing baselines, call budget, isolating the value of agreement-guided allocation.

3 METHOD

Problem Formulation. Given a stable item identifier u and a question q, ARMAD operates on a fixed pool of $K \geq 2$ indexed language-model agents $\mathcal{A} = \{A_1, \ldots, A_K\}$, where $[K] = \{1, \ldots, K\}$. The agents may come from different model families but remain unchanged throughout the inference process. At round t, agent A_i produces a response record $z_i^{(t)}$ containing its generated text and completion status. A task-specific, reference-free parser maps each response to a normalized answer $y_i^{(t)} \in \mathcal{Y}_q \cup \{\perp\}$, where $\mathcal{Y}_q$ is the answer space for q and $\perp \notin \mathcal{Y}_q$ represents an invalid output. The objective is to return a prediction $\hat{y} \in \mathcal{Y}_q \cup \{\perp\}$ while deciding whether debate should be initiated and, once initiated, when it should terminate. These decisions are made without reference answers, model logits, confidence scores, or an auxiliary judge.

Overview of ARMAD. ARMAD allocates debate computation through a two-stage, answer-only controller within a fixed heterogeneous model pool. The overall framework is illustrated in Figure 1. Each agent first produces an independent Round 0 response. Pre-debate Agreement Routing (PAR) uses normalized answer agreement to determine whether Round 0 should be selected directly or the question should enter debate. For routed questions, all agents revise their responses synchronously, and the Early Agreement Stopping Evaluator (EASE) reapplies the same agreement score after each revision round to determine when debate should terminate. ARMAD then applies deterministic voting to the normalized answers from the selected round. Therefore, the adaptive component controls only debate initiation and duration, while the model pool, prompt templates, answer normalization procedure, and aggregation rule remain fixed. Detailed pseudocode for the complete online procedure is provided in Appendix B.2.

Heterogeneous Agent Pool. The agents are instantiated from different model families to provide potentially complementary reasoning patterns and knowledge. Nevertheless, ARMAD does not assume that their errors are statistically independent. All agents receive the same question and Round 0 prompt template $\mathcal{P}_0$, without access to peer responses or model-specific role descriptions. Their independent initial responses are

$$z_i^{(0)} = A_i(\mathcal{P}_0(q)), \quad i \in [K], \qquad \mathbf{Z}^{(0)} = \left(z_1^{(0)}, \ldots, z_K^{(0)}\right). \tag{1}$$

The same K agents participate in every revision round for a routed question. ARMAD therefore adapts the depth of interaction rather than selecting or replacing agents on a per-question basis.

Agreement-Guided Debate Control. ARMAD uses the same answer-agreement statistic for both pre-debate routing and in-debate stopping. The controller operates only on normalized final predictions and does not inspect the correctness of an answer or the internal confidence of an agent.

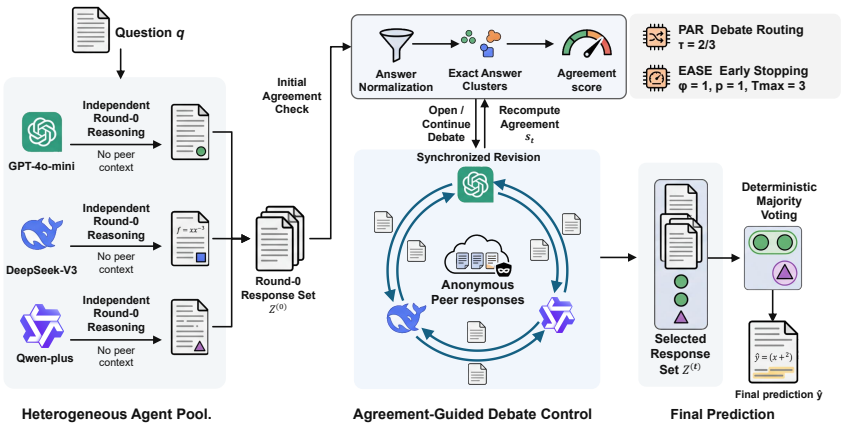

Figure 1: Overview of ARMAD.

Answer normalization and agreement. Let $a_i^{(t)}$ denote the answer extracted from the last complete designated final-answer field in $z_i^{(t)}$. The task-specific normalization function ν_q produces

$$y_i^{(t)} = \nu_q\left(z_i^{(t)}\right). \tag{2}$$

The normalizer returns $\mathrm{canon}_q(a_i^{(t)})$ when the response is complete and the extracted answer is valid; otherwise, it returns $\perp$.

For multiple-choice tasks, canon_q maps extracted answers to canonical option labels. For numerical or symbolic tasks, it performs only conservative, task-compatible surface normalization. Algebraic or numerical equivalence is handled by the offline evaluator and is not used by the online routing controller. Empty, malformed, or length-truncated responses are mapped to $\perp$; in particular, a length-truncated response remains invalid even if an earlier complete final-answer field appears in its text.

Let $\mathbf{Y}^{(t)} = (y_1^{(t)}, \ldots, y_K^{(t)})$ be the normalized answers at round t, and let $\mathcal{C}_t = \{y_i^{(t)} : i \in [K], y_i^{(t)} \neq \perp\}$ contain the valid candidate answers. When $\mathcal{C}_t$ is nonempty, the largest answer cluster has size

$$m_t = \max_{c \in \mathcal{C}_t} \sum_{i=1}^{K} \mathbf{1}\left[y_i^{(t)} = c\right]. \tag{3}$$

We set $m_t = 0$ when $\mathcal{C}_t$ is empty. Normalizing by the fixed pool size gives the agreement score

$$s_t = \frac{m_t}{K}. \tag{4}$$

An invalid response contributes no support to any answer cluster but remains in the denominator K, preventing missing outputs from artificially increasing agreement. Consequently, $s_t \in \{0, 1/K, \ldots, 1\}$, with $s_t = 0$ when all responses are invalid. Importantly, s_t is an allocation signal rather than an estimate or certificate of answer correctness.

Pre-debate agreement routing. After Round 0, PAR compares the initial agreement score s_0 with a threshold $\tau \in (0, 1]$. The routing decision is

$$r_{\mathrm{PAR}}\left(\mathbf{Y}^{(0)}\right) = \mathbf{1}[s_0 < \tau]. \tag{5}$$

When $r_{\mathrm{PAR}} = 0$, ARMAD selects Round 0 and immediately applies the fixed voting rule. Otherwise, the question is routed to debate. PAR neither changes the agent pool nor modifies the generated responses; it only determines whether additional interaction is invoked. High initial agreement may still correspond to a shared error, so skipping debate should not be interpreted as correctness verification.

Synchronous debate and early stopping. For a routed question, each agent receives its own previous response and the anonymized responses produced by the other agents in the preceding round. Agent A_i revises its response according to

$$z_i^{(t)} = A_i\Big(\mathcal{P}_D\big(q,\, z_i^{(t-1)},\, \pi((z_j^{(t-1)})_{j\in[K]\setminus\{i\}})\big)\Big), \quad t \geq 1, \tag{6}$$

where $\mathcal{P}_D$ is the shared debate prompt template and π removes model identities and presents peer responses in a predetermined configuration order. Every response at round t depends only on outputs from round $t-1$; therefore, all agents revise synchronously and cannot observe another agent's current-round response.

After each completed revision round, EASE recomputes s_t using Eq. 4. Given a stopping threshold $\phi \in (0, 1]$, it maintains a counter h_t of consecutive revision rounds whose within-round agreement reaches ϕ. Starting from $h_0 = 0$, the counter is updated for $t \geq 1$ as

$$h_t = (h_{t-1} + 1)\mathbf{1}[s_t \geq \phi]. \tag{7}$$

Let $p \in \{1, \ldots, T_{\max}\}$ denote the required patience and let $T_{\max} \geq 1$ denote the maximum number of synchronized revision rounds after Round 0. For a routed question, EASE selects the smallest $t \in \{1, \ldots, T_{\max}\}$ satisfying $h_t \geq p$. If no such round exists, it selects Round $T_{\max}$. The patience counter is defined over consecutive high-agreement events; it does not impose an additional requirement that the leading answer label remain unchanged across those rounds. PAR and EASE therefore reuse the same normalized agreement signal but apply it to different decisions: PAR determines whether interaction begins, whereas EASE determines how long an initiated debate continues.

Final Prediction. ARMAD aggregates exactly one selected round. The selected round is Round 0 when PAR skips debate, the first revision round satisfying the EASE condition, or Round $T_{\max}$ when the stopping condition is not met earlier. Invalid answers are excluded from voting, and each valid normalized answer contributes one vote to its corresponding answer cluster. ARMAD returns the cluster with the largest number of votes: this is a strict majority when its support exceeds half of the full pool and a plurality otherwise. We refer to this fixed procedure as deterministic voting throughout the paper. Ties are resolved using a fixed deterministic rule based on the item identifier and normalized candidate answers. If all responses in the selected round are invalid, ARMAD returns $\perp$.

4 Experimental Setup

4.1 Datasets

We evaluate ARMAD on five closed-answer reasoning benchmarks. **MATH Level 5** (Hendrycks et al., 2021) tests competition-level mathematical reasoning, and **GSM8K** (Cobbe et al., 2021) contains grade-school arithmetic word problems. **MMLU** (Hendrycks et al., 2020) measures broad academic knowledge across 57 subjects, whereas **MMLU-Pro** (Wang et al., 2024b) uses more answer options and generally requires deeper reasoning. **GPQA Diamond** (Rein et al., 2023) contains expert-written, graduate-level science questions. Detailed dataset descriptions, the seed-123 sampling procedure, sample sizes, and sampled-ID checks are provided in Appendix A.

4.2 Baselines

We compare ARMAD against representative single-model reasoning, multi-agent debate, aggregation, and adaptive-routing baselines. **Single-CoT** (Wei et al., 2022) applies chain-of-thought prompting to a single model. **Self-Consistency** (Wang et al., 2023) samples multiple reasoning paths from

the same model and selects the most frequent final answer. **Homo-MAD** (Du et al., 2024) uses multiple agents instantiated from the same model and performs fixed-round debate before aggregation. **D-MAD** (Zhou & Chen, 2025) is a prompt-diversified baseline inspired by role-diverse debate, in which agents from the same model family receive different reasoning-style instructions.

MoA (Wang et al., 2025) combines multiple proposer responses through a downstream aggregator without iterative debate. **Hetero-MAD** (Chen et al., 2024) is a ReConcile-inspired fixed-depth debate baseline using agents from different model families; we evaluate its one-, two-, and three-round configurations. **Hetero-Vote** (Choi et al., 2025) aggregates the independent Round 0 predictions from the same heterogeneous pool without debate. **iMAD** (Fan et al., 2026) elicits structured self-critique features and uses a trained lightweight classifier to decide whether multi-agent debate should be triggered. It provides a learned adaptive-debate counterpart to ARMAD's training-free answer-agreement controller. Additional model-pool and shared generation settings are provided in Appendix B.1.

4.3 Evaluation Metrics

We report accuracy (%) and its unweighted mean across the five benchmarks, counting invalid or length-truncated outputs as incorrect. Efficiency is measured by mean LLM calls per item and the relative reduction against fixed three-round Hetero-MAD. Calls capture the allocated interaction budget, with comparable average tokens per call in this comparison. Grading and statistical details are provided in Appendix B; token and latency accounting appear in Appendix G.

4.4 Implementation Details

Our pool comprises gpt-4o-mini, DeepSeek-V3, and qwen-plus. Hetero-Vote, fixed-depth Hetero-MAD, and ARMAD share this pool, prompt templates, answer normalization, and deterministic voting. Default ARMAD uses $K = 3$, $\tau = 2/3$, $\phi = 1$, patience $p = 1$, and $T_{\max} = 3$. All same-pool comparisons use the same examples and stored response trajectories, changing only the allocation or stopping rule. Generation settings, parsing, and trajectory reconstruction are detailed in Appendix B.1.

5 Experimental Results

5.1 Main Results

Table 1 compares ARMAD with single-agent reasoning, sampling-based aggregation, homogeneous and heterogeneous debate, Mixture-of-Agents, and the learned adaptive-debate baseline iMAD across five benchmarks. ARMAD ranks first on every dataset and reaches 82.7% mean accuracy, exceeding MoA by 7.6 points, iMAD by 8.4 points, and fixed Hetero-MAD by 7.4 points. Relative to the strongest competing result on each dataset, its margins are 7.5 points on MATH Level 5, 1.0 on GSM8K, 2.3 on MMLU, 1.4 on MMLU-Pro, and 9.1 on GPQA.

Table 1: Main accuracy results (%) across five reasoning benchmarks. **Bold** and underlined values denote the best and second-best reported results, respectively. Avg. is the unweighted arithmetic mean across the five benchmarks.

Method	MATH L5	GSM8K	MMLU	MMLU-Pro	GPQA	Avg.
Single-CoT	43.0	92.0	82.0	61.5	47.5	65.2
Self-Consistency	50.0	94.0	84.0	66.0	52.0	69.2
Homo-MAD	53.5	64.5	66.5	54.0	49.5	57.6
D-MAD	59.0	<u>95.5</u>	66.5	53.5	54.5	65.8
Hetero-Vote	50.5	93.0	87.5	71.5	56.6	71.8
Hetero-MAD	<u>63.5</u>	89.0	84.5	76.0	<u>63.6</u>	<u>75.3</u>
MoA	60.5	90.9	84.0	79.0	61.0	75.1
iMAD	62.3	84.8	<u>89.2</u>	<u>80.6</u>	54.7	74.3
ARMAD	**71.0**	**96.5**	**91.5**	**82.0**	**72.7**	**82.7**

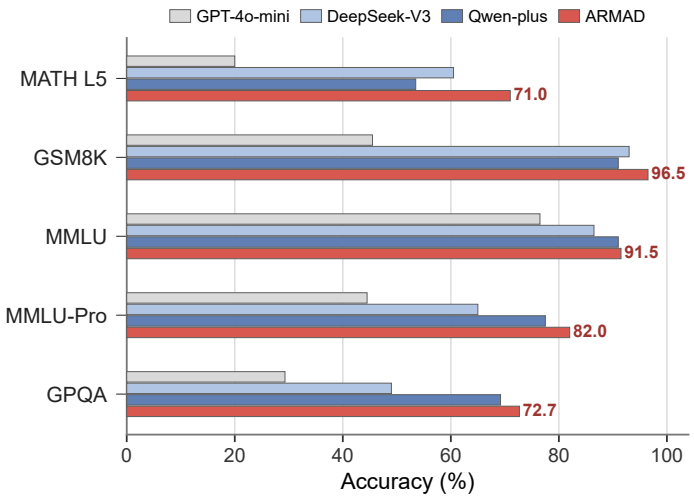

Figure 2: Accuracy of the three independently evaluated constituent models and ARMAD across five benchmarks. Values next to the red bars denote ARMAD accuracy.

The two same-pool controls clarify where these gains arise. Relative to Hetero-Vote, ARMAD improves all five datasets, with gains ranging from 3.5 points on GSM8K to 20.5 points on MATH Level 5. This rules out heterogeneous Round-0 voting as a complete explanation of the result. Relative to fixed Hetero-MAD, ARMAD gains 6.0–9.1 points across the five datasets, showing that the operating point is not reproduced by applying the same debate depth uniformly.

Figure 2 separately compares ARMAD with its three independently evaluated constituent models. ARMAD outperforms the strongest constituent by 10.5 points on MATH Level 5, 3.5 on GSM8K, 0.5 on MMLU, 4.5 on MMLU-Pro, and 3.5 on GPQA. The small MMLU margin reflects an already strong Qwen-plus result, whereas the larger MATH margin shows where heterogeneous coordination contributes most. The improvement across all five datasets cannot be explained by merely including one stronger model in the pool.

Accuracy–call trade-off. Table 2 summarizes ARMAD's call use relative to same-pool fixed Round 3 Hetero-MAD (12 calls per item). Across the five benchmarks, ARMAD uses 3.13–5.57 calls per item and averages 3.96 calls over the 998 evaluated items, a 67.0% reduction. Savings are largest on MMLU (73.9%) and GSM8K (71.9%), where initial agreement frequently avoids debate, and smallest on MATH Level 5 (53.6%), where more items enter discussion. Using the same stored response trajectories, uniformly selecting Round 1 or Round 2 yields modestly higher macro-average accuracy than ARMAD, but increases the call budget from 3.96 to 6 or 9 calls per item, respectively. ARMAD thus retains competitive reasoning accuracy with substantially fewer model invocations; the shallow fixed-depth policies offer additional accuracy at larger call budgets. Per-dataset fixed-round and matched-budget results are reported in Appendix C; full token and reconstructed-latency measurements are provided in Appendix G.

Table 2: Chat calls per item and reduction relative to fixed three-round Hetero-MAD, which uses 12 calls per item. The aggregate is weighted by the 998 evaluated items.

Dataset	MATH L5	GSM8K	MMLU	MMLU-Pro	GPQA Diamond	Size-weighted
ARMAD calls	5.57	3.38	3.13	3.81	3.94	3.96
Reduction	53.6%	71.9%	73.9%	68.3%	67.2%	67.0%

5.2 Ablation Studies

We isolate the routing signal used by PAR and the stopping decision made by EASE using the same examples and stored response trajectories as Default ARMAD. For PAR, we preserve the default selected-depth histogram within each dataset—and therefore the same chat-call budget—while reassigning debate depth either randomly or according to Round 0 response length. For EASE, PAR and deterministic voting remain fixed, but routed examples continue to the maximum depth instead of stopping when the agreement condition is satisfied. Table 3 reports changes relative to Default ARMAD: the upper panel gives net accuracy changes in percentage points, and the lower panel gives changes in chat calls per item.

Table 3: Paired changes relative to Default ARMAD. The routing controls are call matched, and Random Routing is averaged over 5,000 permutations. Avg. is the unweighted mean.

Controller Variant	MATH L5	GSM8K	MMLU	MMLU-Pro	GPQA	Avg.
Accuracy change (percentage points)						
Default ARMAD	0.0	0.0	0.0	0.0	0.0	0.0
PAR → Random Routing	−8.7	−2.4	−2.4	−2.6	−4.2	−4.1
PAR → Length Routing	−5.0	−2.0	−2.0	−1.0	−4.0	−2.8
w/o EASE	0.0	0.0	0.0	0.0	0.0	0.0
Chat-call change per item						
Default ARMAD	0.00	0.00	0.00	0.00	0.00	0.00
PAR → Random Routing	0.00	0.00	0.00	0.00	0.00	0.00
PAR → Length Routing	0.00	0.00	0.00	0.00	0.00	0.00
w/o EASE	+1.12	+0.30	+0.19	+0.68	+0.74	+0.61

Pre-debate Agreement Routing. To separate the routing criterion from the amount of computation, we replace PAR with two call-matched policies. Random Routing permutes ARMAD's selected debate depths across examples, whereas Length Routing assigns deeper trajectories to examples with longer Round 0 responses while preserving the same number of examples at each depth. Both replacements reduce accuracy on every benchmark without changing the call budget. Relative to Random Routing, PAR improves accuracy by 2.4 points on GSM8K and MMLU and by 8.7 points on MATH Level 5. Relative to Length Routing, its gains range from 1.0 point on MMLU-Pro to 5.0 points on MATH Level 5. These paired comparisons support normalized cross-model answer agreement as a more informative allocation signal than either random assignment or response length. Full permutation intervals, shallow fixed-round controls, and the non-call-matched all-debate control are reported in Appendix C.

Early Agreement Stopping. The *w/o EASE* variant retains PAR but forces every routed example to continue through $T_{\max}$. It increases calls on all five benchmarks, from 0.19 additional calls per item on MMLU to 1.12 on MATH Level 5, with a mean increase of 0.61. Its net dataset-level accuracy change is 0.0 points on each benchmark. On the same stored response trajectories, EASE therefore removes post-agreement rounds that do not improve aggregate accuracy, supporting its role as an efficiency mechanism. This empirical equality is specific to the evaluated trajectories and should not be interpreted as a general non-inferiority guarantee.

5.3 Sensitivity Analysis

We examine how PAR, EASE, and the agent-pool configuration change the accuracy–computation operating point. Figure 3 summarizes the trends, while Appendix D retains the exact per-dataset values. Panels (a) and (b) use the same examples and stored response trajectories as the default configuration and report paired changes relative to it rather than repeating the corresponding absolute scores. They use unweighted means across the five main benchmarks. Panel (c) reports absolute results on the same 50 MMLU-Pro items for all four pool configurations.

PAR threshold. PAR produces the clearest accuracy–computation trade-off. With $K = 3$, $\tau = 1/3$ effectively accepts any valid initial answer and therefore provides little agreement filtering. Relative to the $\tau = 2/3$ default, it saves 0.93 calls per item but loses 6.4 accuracy points on average.

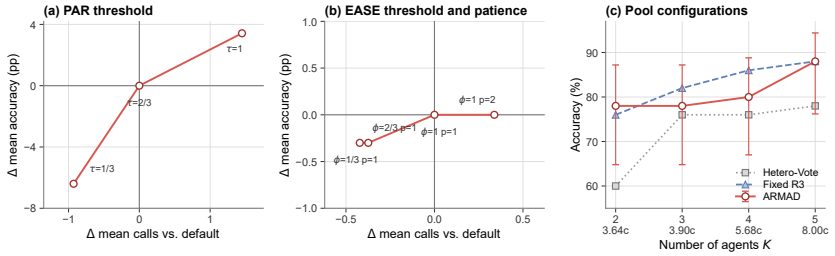

Figure 3: Sensitivity analysis of PAR, EASE, and agent-pool configurations. Panels (a)–(b) report mean changes relative to the default setting; error bars in panel (c) denote 95% Wilson intervals.

Unanimity routing ($\tau = 1$) adds 1.45 calls and gains 3.4 points on average. The stricter threshold recovers 1.5–9.1 points on four difficult benchmarks relative to the default, but decreases GSM8K by 0.5 points. Thus, increasing τ is not uniformly accuracy-monotone even though it increases calls on every dataset.

EASE threshold and patience. EASE moves along a much flatter frontier. Relative to the $\phi = 1, p = 1$ default, $\phi = 1/3$ saves 0.42 calls with a 0.3-point mean decrease, and $\phi = 2/3$ saves 0.37 calls with the same 0.3-point decrease. Increasing patience from one to two adds 0.34 calls on average without changing mean accuracy. These results support EASE primarily as a resource-control mechanism: PAR determines whether corrective interaction occurs, while EASE determines how much of an opened trajectory is consumed.

Number of agents. At $K = 5$, it matches fixed Round 3 debate at 88.0% while using 60% fewer calls than the corresponding 20-call fixed budget. However, the four points jointly change model composition, thresholds, tie structure, and visible peer context; the non-monotone PAR skip rates and wide intervals at $n = 50$ preclude interpreting the curve as a causal scaling law. It should instead be read as evidence that the controller remains operational across four tested pool configurations.

5.4 Further Analysis

On the same evaluation examples and stored response trajectories, ARMAD improves substantially over Hetero-Vote, but wrong-consensus and minority-expert cases remain. We identify 42 unanimous-but-wrong Round-0 cases, showing that high agreement can conceal a shared misconception rather than certify correctness. Among 63 selected-round cases in which exactly one model is correct, deterministic voting and a sentence-embedding-based alternative each recover the correct answer only once.

Appendix F provides per-dataset transition statistics, wrong-consensus breakdowns, aggregation results, and a detailed minority-expert case study. All counts use the same evaluation examples and stored response trajectories as the same-pool comparisons.

6 Conclusion

We presented ARMAD, a training-free, black-box-compatible controller that uses answer agreement to adapt both debate initiation and duration within a fixed heterogeneous model pool. Across five reasoning benchmarks, ARMAD improved over Hetero-Vote on every benchmark and achieved 82.7% macro-average accuracy with 3.96 model calls per item, representing a 67.0% reduction in calls relative to same-pool fixed three-round debate. At matched call budgets, it consistently outperformed random and response-length routing. Shallow fixed-round policies achieved modestly higher accuracy at larger call budgets, positioning ARMAD as a lower-call operating point rather than an accuracy-maximizing policy. Analyses of wrong-consensus and minority-expert cases further showed that agreement is useful for allocating debate calls but cannot certify correctness.

Ethics and Safety Statement

This work studies call-efficient multi-agent reasoning using existing benchmark datasets. ARMAD reduces unnecessary model interactions, but its outputs remain subject to errors and biases in the underlying LLMs. In particular, agreement among agents can reflect shared misconceptions and should not be treated as a guarantee of correctness. Applications involving consequential decisions require independent validation and appropriate human oversight. The use of benchmark data and model APIs should respect applicable licenses, privacy requirements, and terms of service.

AI Use Statement

Generative AI tools were used to assist with manuscript revision, including language editing, terminology consistency, equation presentation, and LaTeX formatting. The LLMs used as experimental agents and their settings are described in Section 4.4. The authors take full responsibility for the accuracy, originality, and integrity of the final manuscript.

References

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. Chateval: Towards better LLM-based evaluators through multi-agent debate. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=FQepisCUWu.

Justin Chen, Swarnadeep Saha, and Mohit Bansal. Reconcile: Round-table conference improves reasoning via consensus among diverse llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7066–7085, 2024.

Hyeong Kyu Choi, Jerry Zhu, and Sharon Li. Debate or vote: Which yields better decisions in multi-agent large language models? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=iUjGNJzrF1.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first international conference on machine learning*, 2024.

Andrew Estornell and Yang Liu. Multi-llm debate: Framework, principals, and interventions. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 28938–28964. Curran Associates, Inc., 2024. doi: 10.52202/079017-0911. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/32e07a110c6c6acf1afbf2bf82b614ad-Paper-Conference.pdf.

Wei Fan, JinYi Yoon, and Bo Ji. iMAD: Intelligent multi-agent debate for efficient and accurate LLM inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 29403–29411, 2026. doi: 10.1609/aaai.v40i35.40181. URL https://ojs.aaai.org/index.php/AAAI/article/view/40181.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=7Bywt2mQsCe.

Tianyu Hu, Zhen Tan, Song Wang, Huaizhi Qu, and Tianlong Chen. Multi-agent debate for LLM judges with adaptive stability detection. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://papers.neurips.cc/paper_files/paper/2025/hash/42475c537936b2394b5015e871765056-Abstract-Conference.html.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 17889–17904, 2024.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. Dynamic LLM-agent network: An LLM-agent collaboration framework with agent team optimization, 2024. URL https://openreview.net/forum?id=i43XCU54Br.

Priya Pitre, Naren Ramakrishnan, and Xuan Wang. Consensagent: Towards efficient and effective consensus in multi-agent llm interactions through sycophancy mitigation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 22112–22133, 2025.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof Q&A benchmark. *arXiv preprint arXiv:2311.12022*, 2023. doi: 10.48550/arXiv.2311.12022.

Andries Petrus Smit, Nathan Grinsztajn, Paul Duckworth, Thomas D. Barrett, and Arnu Pretorius. Should we be going MAD? a look at multi-agent debate strategies for LLMs. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 45883–45905. PMLR, 2024. URL https://proceedings.mlr.press/v235/smit24a.html.

Akshay Verma, Swapnil Gupta, Deepak Gupta, Prateek Sircar, and Siddharth Pillai. SELENE: Selective and evidence-weighted LLM debating for efficient and reliable reasoning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 5: Industry Track)*, pp. 95–104. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.eacl-industry.7. URL https://aclanthology.org/2026.eacl-industry.7/.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Y Zou. Mixture-of-agents enhances large language model capabilities. In *International Conference on Learning Representations*, volume 2025, pp. 33944–33963, 2025.

Qineng Wang, Zihao Wang, Ying Su, Hanghang Tong, and Yangqiu Song. Rethinking the bounds of LLM reasoning: Are multi-agent discussions the key? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6106–6131, 2024a.

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

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024b.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. doi: 10.52202/068431-1800. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

Yanwei Yue, Guibin Zhang, Boyang Liu, Guancheng Wan, Kun Wang, Dawei Cheng, and Yiyan Qi. MasRouter: Learning to route LLMs for multi-agent systems. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15549–15572, 2025. doi: 10.18653/v1/2025.acl-long.757. URL https://aclanthology.org/2025.acl-long.757/.

Yan Zhou and Yanguang Chen. Adaptive heterogeneous multi-agent debate for enhanced educational and factual reasoning in large language models. *Journal of King Saud University Computer and Information Sciences*, 37(10):330, 2025.

A Dataset Details and Sampling

Sampling protocol. The frozen evaluation set contains 998 items across five benchmarks. Dataset preparation and subset selection use seed 123. For datasets requiring subsampling, examples are deterministically ranked using a SHA-256 digest derived from the seed and the stable item identifier. For stratified datasets, examples are ranked within each metadata group and selected in round-robin order across groups. This procedure makes the selected examples independent of their original file order. The complete prepared files, ordered sampled identifiers, and their SHA-256 fingerprints are included in the anonymous artifact.

Dataset	Prepared Pool	Evaluated	Selection Procedure
MATH Level 5	1,324	200	Stratified across seven mathematical subjects; each subject contributes 28 or 29 examples.
GSM8K	1,319	200	Deterministic sample from the test split without category stratification.
MMLU	14,042	200	Stratified across 57 subjects; each subject contributes three or four examples.
MMLU-Pro	12,032	200	Stratified across 14 categories; each category contributes 14 or 15 examples.
GPQA Diamond	198	198	All examples in the Diamond split are retained.
Total	—	**998**	

Table 4: Prepared dataset pools and frozen evaluation samples. All sampling uses seed 123.

MATH Level 5. We construct the MATH Level 5 pool from the test split of `EleutherAI/hendrycks_math` (Hendrycks et al., 2021). Only problems labeled `Level 5` are retained. The resulting pool covers algebra, counting and probability, geometry, intermediate algebra, number theory, prealgebra, and precalculus. Reference answers are extracted from the boxed answers in the official solutions. We select 200 examples using deterministic subject-stratified sampling, preventing subjects with larger source partitions from dominating the evaluation sample.

GSM8K. We use the `main` test split of `openai/gsm8k` (Cobbe et al., 2021), which contains 1,319 grade-school arithmetic word problems. Each reference answer is extracted from the numerical answer following the official `####` delimiter. We select 200 examples using deterministic sampling without stratification.

MMLU. We use the `all` configuration and test split of `cais/mmlu` (Hendrycks et al., 2020). The prepared pool contains 14,042 four-option questions across 57 subjects in the humanities, social sciences, natural sciences, and professional domains. Questions are presented with canonical option labels, and predictions are evaluated against the corresponding reference label. We select 200 examples using subject-stratified sampling, retaining three or four questions from every subject.

MMLU-Pro. We use the test split of `TIGER-Lab/MMLU-Pro` (Wang et al., 2024b), containing 12,032 questions across 14 categories: biology, business, chemistry, computer science, economics, engineering, health, history, law, mathematics, other, philosophy, physics, and psychology. Compared with MMLU, MMLU-Pro contains more answer options and places greater emphasis on reasoning-intensive questions. We select 200 examples using category-stratified sampling, retaining 14 or 15 examples from each category.

GPQA Diamond. We use all 198 examples from the Diamond split of GPQA (Rein et al., 2023). These multiple-choice questions were written and validated by domain experts and cover graduate-level biology, physics, and chemistry. Each item contains one correct answer and three incorrect alternatives. The four options are deterministically shuffled for each item using a local seed derived from seed 123 and the stable source information, preventing the correct answer from occupying a fixed option position.

Dataset integrity. For each dataset, the run manifest records the hash of the complete prepared file and the hash of the ordered sampled-item identifiers. The runner verifies these values before

resuming an experiment, preventing silent changes to either the source data or the frozen evaluation sample. Exact identifiers and hashes are provided in the anonymous artifact.

B Additional Evaluation and Implementation Details

Multiple-choice answers are converted to canonical option labels before comparison with the reference. GSM8K predictions undergo deterministic numerical normalization, whereas MATH predictions are evaluated using the task-specific mathematical equivalence grader. Algebraic equivalence is used only for offline correctness evaluation and does not affect ARMAD's online agreement calculation. Missing, empty, malformed, or length-truncated responses are retained and scored as incorrect.

Accuracy is computed separately over the frozen items of each dataset. The overall average is the unweighted mean of the five dataset-level accuracies. We report Wilson intervals for individual headline accuracies and percentage-point differences for headline method comparisons. Paired replay controls report within-snapshot changes under identical stored trajectories. We do not attach paired significance tests to independently evaluated headline rows because those tests require their item-level joint outcomes. Efficiency is measured using the number of chat-model calls that would be consumed by the selected response prefix.

B.1 Additional Implementation Details

The main heterogeneous pool uses `gpt-4o-mini-2024-07-18`, `DeepSeek-V3`, and `qwen-plus`. The retained single-model and homogeneous baselines use GPT-4o-mini, while Hetero-Vote-only, fixed-depth Hetero-MAD, and ARMAD use all three constituent models. All new model calls use temperature zero and permit at most 8,192 output tokens. A complete trajectory consists of the independent Round 0 responses and three synchronized revision rounds.

Responses are instructed to place their final prediction in a designated `<FINAL>...</FINAL>` field. When a response contains multiple complete fields because of self-correction, the last complete field is used. Empty, malformed, or length-truncated responses are marked invalid. During debate, model identities are removed, and each agent receives its own preceding response together with the anonymized peer responses from the preceding round. Peer responses are presented once in a predetermined configuration order.

For controlled comparison, the collector records the complete Round 0–3 trajectory for every agent and item. Each adaptive or fixed-depth policy is then reconstructed from the prefix it would have consumed. A policy cannot access any response produced after its selected stopping round. This procedure allows same-item and same-response comparisons while preserving the online-equivalent number of calls associated with each policy. Default ARMAD uses no embedding request; the sentence-embedding SOD variant makes one embedding request for the selected response set. Full prompt templates, model configurations, sampled identifiers, parsing rules, and validation scripts are included in the anonymous artifact.

Result provenance. The appendix keeps two result types separate. Headline accuracy tables reproduce the independently evaluated systems reported in the main paper. Threshold, allocation, stopping, and diagnostic analyses use a validated stored-trajectory production snapshot and are reported as within-snapshot changes relative to its default controller. We do not translate those replay changes into synthetic absolute headline scores.

B.2 Complete Inference Procedure

Algorithm 1 provides the complete online inference procedure of ARMAD. It includes Round 0 routing, synchronized revision, agreement-based early stopping, and deterministic majority voting.

C Matched-Budget and Full-Debate Controls

Table 5 adds controls between the three-call heterogeneous vote and the twelve-call fixed Round 3 endpoint. To avoid mixing absolute scores from independently evaluated headline systems with the

Algorithm 1 Complete ARMAD Inference Procedure

Require: Question q; agents $\mathcal{A}$; thresholds τ and ϕ; patience p; maximum round $T_{\max}$

Obtain independent Round 0 responses $Z^{(0)}$
Normalize their final answers to obtain $Y^{(0)}$
Compute agreement score s_0
if $s_0 \geq \tau$ **then**

  return $\mathrm{Majority}(Y^{(0)})$
end if
$h \leftarrow 0$
for $t = 1$ to $T_{\max}$ **do**
  Obtain synchronized revisions $Z^{(t)}$
  Normalize their final answers to obtain $Y^{(t)}$
  Compute agreement score s_t
  if $s_t \geq \phi$ **then**
    $h \leftarrow h + 1$
  else
    $h \leftarrow 0$
  end if
  if $h \geq p$ **then**

    return $\mathrm{Majority}(Y^{(t)})$
  end if
end for

return $\mathrm{Majority}(Y^{(T_{\max})})$

replay snapshot, every accuracy entry is expressed as a paired change from the snapshot's Default ARMAD. Fixed Round 1 is already strong on MATH and GPQA, showing that a shallow uniform policy can be preferable when extra calls are affordable. The sharper allocation test holds ARMAD's selected-round histogram fixed within each dataset and permutes those depths over items. ARMAD exceeds the mean random allocator by 2.4–8.7 points on all five datasets; 5,000-permutation upper-tail p-values range from .0002 to .0010. It also exceeds the equal-call response-length allocator by 1.0–5.0 points.

Dataset	Fixed R1 Δ	Fixed R2 Δ	Random Δ [95%]	Length Δ	Matched calls
MATH L5	+4.0	+3.5	−8.7 [−11.5,−6.0]	−5.0	5.57
GSM8K	−1.0	−1.5	−2.4 [−3.0,−1.5]	−2.0	3.38
MMLU	−1.0	+2.0	−2.4 [−3.0,−2.0]	−2.0	3.13
MMLU-Pro	+1.5	+2.5	−2.6 [−4.0,−1.0]	−1.0	3.81
GPQA Diamond	+7.6	+8.1	−4.2 [−6.0,−2.0]	−4.0	3.94

Table 5: Within-snapshot accuracy changes (percentage points) relative to Default ARMAD for shallow fixed debate and allocation controls. Fixed R1 and R2 consume 6 and 9 chat calls per item, respectively. Random and response-length allocation receive exactly the default selected-round histogram; the bracketed range is the 2.5–97.5 percentile interval over 5,000 permutations, expressed relative to Default. Matched calls are shared by Random, Length, and Default.

All-debate control. Table 6 reports paired changes when PAR is disabled and every example is routed to debate. This variant uses 2.01–3.14 more chat calls per example than the default controller and is therefore not a budget-matched component ablation. Its higher accuracy on four of five datasets quantifies the corrective interaction forgone by routing, while the matched-budget controls isolate whether agreement allocates the available calls better than the two alternative routers.

Dataset	Accuracy Δ	Chat-call Δ
MATH L5	+4.0	+2.01
GSM8K	−0.5	+2.89
MMLU	+1.5	+3.14
MMLU-Pro	+3.0	+2.72
GPQA Diamond	+9.1	+2.73

Table 6: Within-snapshot changes produced by disabling PAR and routing every item to debate, relative to Default ARMAD. Accuracy changes are in percentage points and call changes are per item. This resource upper bound is not a budget-matched component ablation.

D Detailed Sensitivity Results

Tables 7–9 provide the exact values summarized by the sensitivity figure in the main paper. The PAR and EASE tables report accuracy and chat calls for every main benchmark. The pool table additionally reports vote and fixed-debate accuracy, Wilson intervals, and realized routing and stopping rates.

PAR Setting	**MATH L5**	**GSM8K**	**MMLU**	**MMLU-Pro**	**GPQA Diamond**	**Avg.**
Accuracy change (percentage points)						
$\tau = 1/3$	−17.5	−2.5	−2.5	−3.0	−6.5	−6.4
Default ($\tau = 2/3$)	0.0	0.0	0.0	0.0	0.0	0.0
$\tau = 1$	+4.0	−0.5	+1.5	+3.0	+9.1	+3.4
Chat-call change per item						
$\tau = 1/3$	−2.57	−0.38	−0.13	−0.76	−0.80	−0.93
Default ($\tau = 2/3$)	0.00	0.00	0.00	0.00	0.00	0.00
$\tau = 1$	+1.57	+1.60	+0.94	+1.31	+1.83	+1.45

Table 7: PAR sensitivity under majority aggregation. Both panels report paired changes relative to the $\tau = 2/3$ default on the stored-trajectory snapshot.

EASE Setting	**MATH L5**	**GSM8K**	**MMLU**	**MMLU-Pro**	**GPQA Diamond**	**Avg.**
Accuracy change (percentage points)						
$\phi = 1/3, p = 1$	+0.5	−0.5	−1.0	−0.5	0.0	−0.3
$\phi = 2/3, p = 1$	0.0	−0.5	−0.5	−0.5	0.0	−0.3
Default ($\phi = 1, p = 1$)	0.0	0.0	0.0	0.0	0.0	0.0
$\phi = 1, p = 2$	0.0	0.0	0.0	0.0	0.0	0.0
Chat-call change per item						
$\phi = 1/3, p = 1$	−1.34	−0.15	−0.02	−0.31	−0.29	−0.42
$\phi = 2/3, p = 1$	−1.20	−0.15	−0.01	−0.30	−0.21	−0.37
Default ($\phi = 1, p = 1$)	0.00	0.00	0.00	0.00	0.00	0.00
$\phi = 1, p = 2$	+0.60	+0.15	+0.13	+0.39	+0.42	+0.34

Table 8: EASE threshold and patience sensitivity under majority aggregation with PAR fixed at $\tau = 2/3$. Both panels report paired changes relative to $\phi = 1, p = 1$ on the stored-trajectory snapshot.

K	Vote acc.	Fixed R3 acc.	ARMAD acc. [95% CI]	ARMAD calls	PAR skip	EASE stop
2	60.0	76.0	78.0 [64.8,87.2]	3.64	54.0%	40.0%
3	76.0	82.0	78.0 [64.8,87.2]	3.90	80.0%	18.0%
4	76.0	86.0	80.0 [67.0,88.8]	5.68	78.0%	16.0%
5	78.0	88.0	88.0 [76.2,94.4]	8.00	66.0%	32.0%

Table 9: Four joint pool-size/composition configurations on the same 50-item MMLU-Pro sensitivity subset. Accuracy is in percent; calls are chat calls under majority aggregation. These subset results are separate from the 200-item MMLU-Pro headline evaluation.

E Confidence Intervals and Headline Comparisons

E.1 Wilson Intervals

Table 10 reports constituent-model intervals for the standalone results visualized in the main paper. Table 11 reproduces the headline Hetero-Vote, fixed Hetero-MAD, and ARMAD accuracies and gives their Wilson intervals. The descriptive best-single row is omitted because its entries duplicate one of the three constituents and its identity changes by dataset.

Dataset	GPT-4o-mini	DeepSeek-V3	Qwen-plus
MATH L5	20.0 [15.0,26.1]	60.5 [53.6,67.0]	53.5 [46.6,60.3]
GSM8K	45.5 [38.7,52.4]	93.0 [88.6,95.8]	91.0 [86.2,94.2]
MMLU	76.5 [70.2,81.8]	86.5 [81.1,90.6]	91.0 [86.2,94.2]
MMLU-Pro	44.5 [37.8,51.4]	65.0 [58.2,71.3]	77.5 [71.2,82.7]
GPQA Diamond	29.3 [23.4,36.0]	49.0 [42.1,55.9]	69.2 [62.4,75.2]

Table 10: Constituent accuracy (%) with 95% Wilson intervals.

Dataset	Hetero-Vote	Fixed Hetero-MAD	ARMAD
MATH L5	50.5 [43.6,57.4]	63.5 [56.6,69.9]	71.0 [64.4,76.8]
GSM8K	93.0 [88.6,95.8]	89.0 [83.9,92.6]	96.5 [93.0,98.3]
MMLU	87.5 [82.2,91.4]	84.5 [78.8,88.9]	91.5 [86.8,94.6]
MMLU-Pro	71.5 [64.9,77.3]	76.0 [69.6,81.4]	82.0 [76.1,86.7]
GPQA Diamond	56.6 [49.6,63.3]	63.6 [56.7,70.0]	72.7 [66.1,78.5]

Table 11: Headline group-method accuracy (%) with 95% Wilson intervals computed from the reported correct counts ($n = 200$ except GPQA Diamond, $n = 198$).

E.2 Aggregate Headline Differences

Table 12 restates the headline differences between ARMAD and the two same-pool controls. These are aggregate accuracy differences rather than paired inferential statistics; W/L counts, McNemar tests, and paired bootstrap intervals from the separate replay snapshot are not transferred to the headline rows.

Dataset	ARMAD	Hetero-Vote	Δ	Fixed R3	Δ
MATH L5	71.0	50.5	+20.5	63.5	+7.5
GSM8K	96.5	93.0	+3.5	89.0	+7.5
MMLU	91.5	87.5	+4.0	84.5	+7.0
MMLU-Pro	82.0	71.5	+10.5	76.0	+6.0
GPQA Diamond	72.7	56.6	+16.1	63.6	+9.1

Table 12: Headline accuracy comparisons. Each Δ is ARMAD minus the corresponding baseline in percentage points. Item-level joint outcomes are unavailable for these headline rows, so no paired confidence interval or significance test is reported.

F Diagnostic Analyses

The analyses in this section use the paired-replay diagnostic snapshot. They characterize transitions within that snapshot and are not an item-level decomposition of the independently evaluated headline systems.

F.1 Answer Transitions and Wrong Consensus

Table 13 reports conditional accuracy for items skipped and debated by PAR, together with transitions relative to Hetero-Vote. Across the five benchmarks, ARMAD fixes 68 initially incorrect decisions while regressing two initially correct decisions. MATH Level 5 accounts for 36 fixes and one regression. The table also identifies 42 cases in which all three Round-0 answers agree on the same incorrect answer. These unanimous-but-wrong cases are concentrated on GPQA Diamond (18) and MMLU-Pro (15), demonstrating that answer agreement cannot distinguish reliable consensus from a shared misconception.

Dataset	PAR skip	Skip acc.	Debate acc.	Fixes	Regressions	Wrong consensus
MATH L5	59.0%	86.4%	73.2%	36	1	1
GSM8K	92.5%	95.7%	80.0%	6	1	2
MMLU	96.5%	91.2%	100.0%	5	0	6
MMLU-Pro	83.5%	76.6%	87.9%	7	0	15
GPQA Diamond	81.3%	61.5%	73.0%	14	0	18
Overall	**82.6%**	**82.8%**	**77.6%**	**68**	**2**	**42**

Table 13: Routing and answer-transition diagnostics from the paired replay snapshot. Skip and debate accuracy are conditional on the PAR decision. Fixes and regressions compare ARMAD with Hetero-Vote within this diagnostic snapshot; wrong consensus counts items for which all three Round-0 answers agree but are incorrect. These transitions should not be interpreted as differences between the independently reported headline systems.

F.2 Minority-Expert and Aggregation Analysis

We compare the default majority rule with sentence-embedding aggregation at the round selected by ARMAD. Table 14 reproduces the headline majority accuracies and separately reports the paired aggregation change measured in the replay snapshot. That change is zero on all five datasets, while the embedding variant requires one additional API request per item. Among 63 selected-round cases in which exactly one of the three models is correct, each rule selects that correct minority answer only once. The embedding rule changes the selected answer cluster in one retained case, but does not rescue a majority error or regress a majority success.

The changed case is `MATH-L5/precalculus/346`. Qwen-plus is the only agent to return the reference answer, 43. Majority voting selects 26, whereas sentence-embedding aggregation selects 62. The semantic rule therefore changes which incorrect answer is returned without identifying the knowledgeable minority. This case illustrates why similarity to other responses should not be treated as evidence of correctness.

Dataset	Headline majority acc.	Embedding accuracy Δ	Extra requests
MATH L5	71.0	0.0	+1.00
GSM8K	96.5	0.0	+1.00
MMLU	91.5	0.0	+1.00
MMLU-Pro	82.0	0.0	+1.00
GPQA Diamond	72.7	0.0	+1.00

Table 14: Aggregation summary. The first column reproduces Default ARMAD's headline majority accuracy (%). The remaining columns report the paired replay change from replacing majority with sentence-embedding aggregation and its additional embedding request per item.

G Trace Validation and Resource Accounting

All trajectory-based controls and diagnostic results in this appendix come from a validated 998-item production snapshot. The validator checks expected item and round counts, serialized prompts and responses, model and dataset hashes, finish reasons, and embedding dimensions. Missing or length-truncated answers are retained and scored as invalid rather than silently dropped. Validation reports no unresolved errors. The retained snapshot contains 11,976 chat calls and 3,992 batched embedding requests; reported policy consumption uses only the selected chat prefix and, for the sentence-embedding variant, one selected embedding request.

The anonymous artifact contains the exact configurations, sampled IDs, full prompts, parser and grader source, environment lock, raw and normalized responses, usage and finish metadata, embeddings, SHA-256 manifests, validation report, and scripts for the stored-trajectory analyses. The offline-control manifest records seed 20260725, 5,000 randomizations, 5,000 bootstrap replicates, the source-trace hash, and zero API calls. Paired inferential outputs retained with that snapshot are diagnostic only and are not presented as tests of the independently evaluated headline rows.

Table 15 reports three resource dimensions. Default ARMAD uses only chat calls and chat tokens. Latency is the sum of recorded per-round maximum model latencies over the selected prefix; it is a reconstructed synchronized estimate rather than an observed adaptive replay. We omit dollar cost because the proxy endpoint does not provide a stable, model-specific tariff for every alias.

Dataset	Hetero-Vote	Fixed Hetero-MAD	ARMAD
MATH L5	3.00/4,478/52.6	12.00/24,664/80.6	5.57/13,291/63.2
GSM8K	3.00/1,348/8.6	12.00/8,615/23.2	3.38/1,922/9.3
MMLU	3.00/1,222/9.4	12.00/8,756/39.7	3.13/1,464/10.2
MMLU-Pro	3.00/1,977/16.4	12.00/12,275/47.2	3.81/3,338/19.3
GPQA Diamond	3.00/3,959/46.4	12.00/22,743/80.8	3.94/7,014/50.3

Table 15: Selected-prefix resource use per item in the stored-trajectory production snapshot. Each cell is chat calls / chat tokens / reconstructed latency in seconds.