LEARNING WHAT TO DEBATE: ADAPTIVE ARGUMENT SCHEDULING FOR MULTI-AGENT REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-agent debate (MAD) lets large language models (LLMs) read one another's responses and iteratively revise their reasoning. Yet MAD is judged by collective decision quality, while computation is allocated through coarse, full-response rounds. This can repeatedly revisit low-value reasoning, waste computation, and propagate erroneous arguments. We instead formulate MAD as selective computation over a **public argument state**, organizing independently generated rationales into a shared representation of claims, candidate decisions, evidence links, and support and conflict relations. Our method predicts the rescue and damage of candidate revisions, ranks focused revision actions by predicted rescue and call cost, and stops through a damage-aware gate. Opening-relative **promotion**, **acquisition**, and **damage** distinguish revision gains from the starting ensemble. Across eight reasoning benchmarks, AAS-MAD achieves 70.76% pooled accuracy, exceeding the strongest evaluated debate baseline by 1.03 percentage points with 28.84–44.34% fewer tokens. Measured from its own opening rather than from the best baseline, its 1.45-percentage-point gain is the largest among the five MAD methods; at the evaluated policy endpoints, all successful corrections promote existing candidates. On our curated FIFA 2026 dataset, AAS-MAD reaches 68.59% accuracy, improving on the strongest evaluated debate baseline by 1.92 percentage points. These results support allocating deliberation to arguments that can still improve the shared decision, limiting expenditure on unproductive revisions.

1 INTRODUCTION

Step-by-step reasoning makes intermediate inferences explicit and can improve large language models' performance on complex tasks (Wei et al., 2022). An explicit reasoning chain, however, does not certify the validity of its premises or deductions, and intrinsic self-correction without external feedback can fail to repair reasoning errors (Huang et al., 2024). REFINER learns a critic to provide structured feedback on intermediate reasoning, supporting feedback-guided revision (Paul et al., 2024). Chain-of-thought self-consistency (CoT-SC) reduces reliance on a single reasoning trajectory by sampling multiple solutions and aggregating their answers (Wang et al., 2023). Its effectiveness still depends on the alternatives generated. In multi-agent debate, homogeneous reasoning strategies can limit the diversity of candidate solutions and reinforce a shared mental set (Liu et al., 2025).

Multi-agent debate (MAD) extends independent sampling into collective reasoning. Agents inspect one another's rationales, challenge conclusions, and revise their own answers (Du et al., 2024). Interaction can expose a missing premise or recover a correct minority view, but it also creates a channel through which errors spread. MAD-M^2 (Multi-Agent Debate with Memory Masking) is motivated by the observation that erroneous memories from earlier debate rounds can mislead agents that previously answered correctly (Tian et al., 2026). M2CL (multi-LLM context learning) addresses context misalignment that can hinder agents from coherently incorporating useful intermediate results (Hua et al., 2026). Repeated exchanges can therefore reproduce faulty reasoning, amplify shared mistakes, and spend computation on material that no longer affects the decision. Comparisons with self-consistency and ensembling further motivate measuring what interaction adds beyond the decision available before debate (Smit et al., 2024; Choi et al., 2025).

These observations motivate selective argument revision: concentrating computation on updates that can improve the collective decision, while limiting low-value exchanges and the spread of errors.

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.

Many protocols allocate work through complete messages, synchronous rounds, or response-level communication links (Du et al., 2024; Li et al., 2024b; Yun et al., 2026), with memory filtering, context learning, and anti-conformity mechanisms regulating those exchanges (Tian et al., 2026; Hua et al., 2026; Cui et al., 2026). Their unit of exchange nevertheless remains a coarse global round. We investigate whether an explicit claim or reasoning dependency can guide the selection of fine-grained revisions and assess their expected utility.

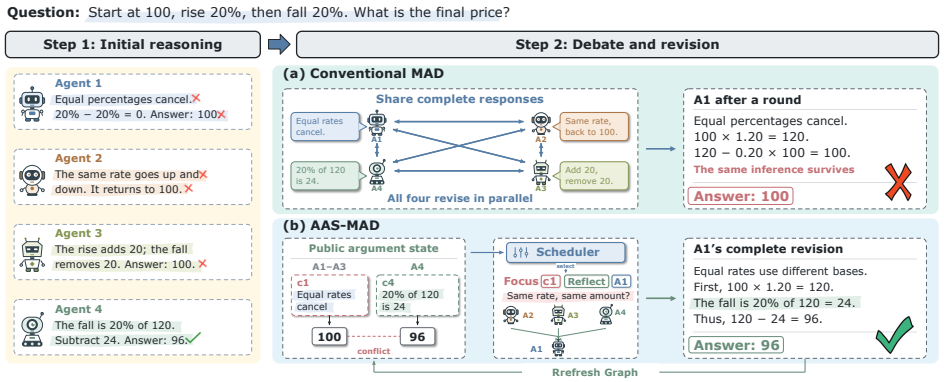

Figure 1: Same reasoning, different collaboration. AAS-MAD makes the cancellation claim addressable and focuses reflection on it. The continuations are illustrative; scoring and stopping are omitted.

This perspective leads to three questions. **Representation:** how can agents' evolving reasoning become a public state that exposes support, conflict, and dependence on evidence and argument? **Scheduling:** which inference, revision operation, and participants should receive the next computation? **Stopping:** when is another exchange worth its expected correction benefit, damage risk, and cost? A useful controller must answer these questions without seeing the task's gold answer online.

We propose **AAS-MAD (Adaptive Argument Scheduling for Multi-Agent Debate)**. AAS-MAD turns agents' current reasoning into a public argument state and allocates computation to focused revisions. Two extremely randomized tree (ET) heads, trained on offline action branches, estimate one-step rescue and damage for a bounded menu of program-generated actions. Ranking and stopping share these predictors with separately calibrated cost-sensitive scores. Revised reasoning replaces the previous chain before the public state is rebuilt.

We evaluate AAS-MAD on eight reasoning benchmarks and our FIFA 2026 forecasting dataset. AAS-MAD applies the public argument protocol from initialization onward, then selects which revisions are worth executing. As a complete system, it achieves 70.76% pooled accuracy at 21.09k tokens per item, compared with 69.73% and 37.89k for MAD-M^2. Separating the opening from subsequent interaction, its 1.45-point gain over its own opening ranks first among the five MAD methods, while total token use falls by 28.84–44.34% against the four debate baselines. Component ablations roughly halve the cost of synchronous revision at comparable observed accuracy. FIFA extends this evaluation to a fixed forecasting readout, where adaptive stopping saves 68.01% of the tokens used by a matched six-action control with nearly identical accuracy.

Our contributions are threefold:

1. A **public argument protocol** that exposes claims and reasoning dependencies as targets for complete-chain revisions and refreshes their shared representation after each exchange.

2. **AAS-MAD**, a controller that ranks candidate revisions by predicted rescue and call cost and decides when to stop through a damage-aware gate.

3. An **opening-relative evaluation** across reasoning and forecasting tasks, showing modest net correction and substantial computation savings against the evaluated debate baselines, with stepwise analysis of promotion, acquisition, and damage.

2 Related work

Reasoning with Large Language Models. LLM reasoning can be strengthened through explicit intermediate steps, feedback-guided refinement, and aggregation of sampled solutions (Wei et al., 2022; Paul et al., 2024; Wang et al., 2023). Yet additional computation need not improve correctness, and observed debate gains can partly arise from ensembling rather than interaction (Huang et al., 2024; Smit et al., 2024; Choi et al., 2025). Further analyses examine initialization diversity, confidence, and task-dependent communication benefits (Zhu et al., 2026; Rizvi-Martel et al., 2026). A complementary line of work allocates computation and determines when to stop based on problem difficulty, sampling agreement, or expected decision value (Snell et al., 2025; Li et al., 2024a; Hay et al., 2012). We study this allocation problem within collective reasoning, evaluating additional revisions against an identical opening decision and readout.

Multi-Agent Debate and Collaboration. Multi-agent reasoning develops complementary perspectives through debate (Du et al., 2024; Liu et al., 2025) and regulates information exchange to address conformity, erroneous memories, and context misalignment (Cui et al., 2026; Tian et al., 2026; Hua et al., 2026). Efficiency-oriented approaches reduce redundant exchanges (Li et al., 2024b; Zeng et al., 2025; Zhang et al., 2025; Yang et al., 2026a), selectively invoke debate (Eo et al., 2025; Fan et al., 2026), or adapt communication and collaborative behavior (Wang et al., 2026; Wu et al., 2026; Estornell et al., 2025; Babu et al., 2026). Alongside these coordination mechanisms, graph-based approaches structure thoughts, computations, and agent interactions (Besta et al., 2024; Zhuge et al., 2024; Yun et al., 2026). Argumentation frameworks formalize relations among claims (Dung, 1995; Cayrol & Lagasquie-Schiex, 2005), while structured debate examines reasoning paths and evidence-linked subclaims (Ma et al., 2025; Xu et al., 2026). More directly, LMAD localizes reasoning conflicts, while AgentAuditor audits evidence at reasoning-tree divergences (Gao et al., 2026; Yang et al., 2026b). Our focus is not structure or localization alone, but the predicted decision value of an executable revision. AAS-MAD scores a bounded menu of program-generated revisions using rescue predictions and call costs, sharing its outcome predictors between action ranking and stopping. After each executed revision, it refreshes the public argument state before rescheduling.

3 Problem Formulation

Task and public state. Given a task x, admissible evidence E_x, and answer space $\mathcal{Y}_x$, four agents independently produce reasoning chains $\mathbf{r}_0$ and answer commitments $\mathbf{d}_0$. At a completed-exchange boundary, the public computational state and collective decision are

$$s_t = (x, E_x, G_t, \mathbf{d}_t, \mathbf{r}_t, h_t, t), \qquad \hat{y}_t = \rho(s_t), \tag{1}$$

where G_t is the current argument graph, h_t is the interaction history, and ρ is a fixed task readout. Let ν denote native answer aggregation: $\rho = \nu$ for general reasoning, whereas FIFA applies a fixed forecasting readout to the same opening and endpoint. Local controller labels use ν in both settings, so on FIFA they guide native revisions while endpoint evaluation uses ρ. Gold labels y appear only in offline supervision and evaluation. The task and admissible evidence remain fixed as revisions change the available reasoning.

Revision control objective. The controller selects a legal revision $a_t \in \mathcal{A}^+(s_t)$ or STOP and returns $\hat{y}_\tau$ at $\tau \le H$, where H is the action horizon. Model sampling and programmatic graph updates induce $P(s' \mid s, a)$, while STOP returns $\rho(s)$. These elements define a finite-horizon metalevel MDP with explicit stopping (Hay et al., 2012). Its objective balances opening-relative correctness gain against nominal call cost. AAS-MAD approximates the sequential problem with one-step rescue and damage predictions rather than a learned full-trajectory value.

For a policy π choosing revisions or stopping, the task-level objective is

$$J(\pi) = \mathbb{E}_\pi\left[u(\rho(s_\tau), y) - u(\rho(s_0), y) - \eta \sum_{t<\tau} c(s_t, a_t)\right], \qquad \tau \le H, \tag{2}$$

where u is binary task correctness, c counts nominal generation calls, and $\eta \ge 0$ specifies the preference for computation. Immediate stopping has zero incremental value. The opening provides

the reference decision; its cost is common to all continuation policies. For forecasting, the local predictions under ν serve as proxies for this objective under ρ. Section 4 specifies the resulting one-step controller.

Opening-relative outcomes. We measure what deliberation adds relative to the same opening and readout. Let C_0 be the set of opening answer commitments. An initially incorrect decision can be corrected by **promotion**, when the correct answer is already in C_0, or by **acquisition**, when it is absent. **Damage** changes an initially correct decision into an incorrect one. With their indicators denoted by P_t, A_t, D_t, respectively, the per-case change in correctness is

$$u(\hat{y}_t, y) - u(\hat{y}_0, y) = P_t + A_t - D_t. \tag{3}$$

Here u is binary correctness under the registered scorer. Acquisition captures gains unavailable to a selector confined to C_0; Appendix A gives the coverage bound and accounting details. These endpoint outcomes differ from the one-step labels used by the controller.

4 AAS-MAD: Adaptive Argument Scheduling

Figure 2 summarizes AAS-MAD. It constructs a bounded revision menu from the public graph, predicts each action's rescue and damage, ranks the menu by predicted rescue and call cost, and gates its top action. Execution rewrites one complete reasoning chain, rebuilds the graph, and repeats.

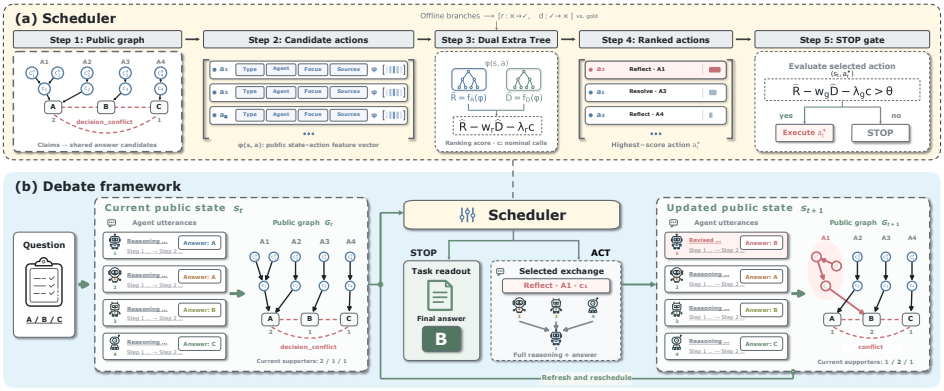

Figure 2: AAS-MAD overview. (a) The scheduler constructs public state–action features, predicts rescue and damage with two ET heads, ranks legal actions by predicted rescue and call cost, and applies a STOP gate to the selected action. The two decisions share the heads with separately calibrated coefficients: the ranker reads predicted rescue and call cost, while the gate adds predicted damage. (b) An executed exchange replaces the revised reasoning and refreshes the public graph before rescheduling; STOP returns the task readout. The illustrated answer change is schematic. Gold labels supervise offline branches only.

4.1 Public Argument State

The program parses each natural-language chain into atomic claims, addressable dependencies, and a task decision, then projects them into

$$G_t = (V_t, R_t^+, R_t^-, L_t), \tag{4}$$

where V_t contains claims, decisions, and dependencies; R_t^+ and R_t^- encode declared support and conflict; and L_t records authorship and cited evidence from the shared pool. Authorship, endorsement, and provenance remain distinct, so repeated endorsement does not create a new source.

Opening prompts assign local identifiers to reasoning steps and ask agents to name the premises of each derived claim. The parser creates claim nodes from these steps and support relations from their references. For example, a step L_3 derived from L_1, L_2 yields two support links sharing the same inference identifier, preserving the joint premise set. Explicit attacks target the referenced claim; the final reasoning step supports the parsed answer commitment. Cited evidence identifiers are bound to

entries in the shared pool. The scheduler can address both claims and inference relations, allowing a revision to question either a proposition or the step connecting premises to a conclusion.

After an exchange, each complete revision replaces its predecessor and the program rebuilds the active graph from the current chains; earlier utterances remain in provenance. Withdrawn reasoning therefore leaves active decision support, while new claims and dependencies shape the next action menu.

4.2 Candidate revision actions

Each candidate revision is represented as

$$a = (v, o, i, S) \in \mathcal{A}^+(s), \qquad \mathcal{A}(s) = \mathcal{A}^+(s) \cup \{\textsc{Stop}\}, \tag{5}$$

with focus v, operation o, executor i, and source participants S. **Source-conditioned reflection** asks the executor to reconsider its reasoning using the other agents' current chains. **Dispute resolution** directs a challenge and response toward one shared focus and elicits a complete revised chain. The focus is a program-selected claim or reasoning dependency supported by the action interface.

Programmatic rules instantiate all four fields before learned scoring; the scheduler selects among complete candidates rather than optimizing the fields independently. With four agents, the menu contains one reflection and at most one eligible dispute per executor. Reflection uses the three peers as sources, while dispute eligibility fixes its focus and participants. A focused action conditions a complete-chain rewrite on an explicit claim or dependency; it does not edit an isolated graph node. Each candidate has an 88-dimensional $\phi(s, a)$ describing public argument structure, decision support and disagreement, evidence links, participants, operation, cost, and action history.

Reflection anchors on the executor's own reasoning, preferentially its relation to the current answer, and provides complete peer chains as comparison material. In Resolve, the initiating agent challenges a peer's argument at the shared focus; the responding agent produces the revised chain and decision. Eligibility follows public disagreement and graph relations. Thus participants have operation-specific roles, and the updated answer can belong to the responder. After either operation, candidate generation runs again on the refreshed state.

4.3 Learning rescue and damage predictors

Training branches execute alternative legal actions from the same saved state. Using the native decision $\nu(s)$ and task equivalence $\equiv$, offline scoring of a branch (s, a, s') produces two binary labels:

$$\begin{aligned} \mathrm{res}(s, a, s') &= \mathbf{1}\{\nu(s) \not\equiv y,\ \nu(s') \equiv y\}, \\ \mathrm{dam}(s, a, s') &= \mathbf{1}\{\nu(s) \equiv y,\ \nu(s') \not\equiv y\}. \end{aligned} \tag{6}$$

We fit two extremely randomized tree regressors (Geurts et al., 2006) to the same public state–action feature interface, one for rescue and one for damage:

$$\hat{p}_{\mathrm{res}}(s, a) = f_R(\phi(s, a)), \qquad \hat{p}_{\mathrm{dam}}(s, a) = f_D(\phi(s, a)). \tag{7}$$

The targets estimate $\overline{\mathrm{res}}(s, a) = \mathbb{E}[\mathrm{res} \mid s, a]$ and $\overline{\mathrm{dam}}(s, a) = \mathbb{E}[\mathrm{dam} \mid s, a]$, so native one-step correctness changes by

$$\mathbb{E}[\Delta u \mid s, a] = \overline{\mathrm{res}}(s, a) - \overline{\mathrm{dam}}(s, a). \tag{8}$$

Both labels concern the collective decision: correcting one agent can leave the aggregate answer unchanged, in which case the branch has $\mathrm{res} = \mathrm{dam} = 0$. The same is true of an incorrect-to-incorrect or correct-to-correct transition. Equation 8 follows because the only nonzero correctness changes are rescue $(+1)$ and damage (-1).

4.4 Cost-sensitive scheduling and stopping

The same fitted heads enter both decisions through a cost-sensitive score:

$$q_{w,\lambda}(s, a) = \hat{p}_{\mathrm{res}}(s, a) - w\hat{p}_{\mathrm{dam}}(s, a) - \lambda c(s, a), \qquad w, \lambda \geq 0, \tag{9}$$

where $c(s, a)$ is the nominal number of calls required by the exchange. Ranking and stopping use separately configured coefficients, while sharing the predictors and feature interface:

$$a_t^* = \arg\max_{a \in \mathcal{A}^+(s_t)} q_{w_r,\lambda_r}(s_t, a), \qquad \begin{cases} \text{execute } a_t^*, & q_{w_g,\lambda_g}(s_t, a_t^*) > \theta, \\ \text{STOP}, & \text{otherwise.} \end{cases} \tag{10}$$

At $w = 1$, the score estimates one-step correctness gain minus nominal call cost; varying w changes damage aversion. The gate evaluates only the ranker's first choice, and a veto ends deliberation. Execution also stops at an empty menu, the six-action limit, or when the preceding exchange changes neither the semantic graph nor its revised public chain. Otherwise the exchange refreshes the graph and action menu. The cost term directly compares a two-call dispute with a one-call reflection.

Ranking compares revisions within one state, while the gate compares the selected revision with stopping. Separate coefficients allow these decisions to express different cost and damage preferences; in the reported configuration the ranker's damage weight is zero, so ranking reads predicted rescue and call cost, and only the gate reads predicted damage (Appendix B.2). Equal ranking scores use a deterministic action-identifier tie-break. The threshold θ sets the continuation boundary on the learned score scale.

4.5 Offline fitting and controller configuration

The two heads are fitted to observed one-step branches and run locally. The reported controller uses one ranking rule and one STOP rule, permitting Reflect and Resolve for at most six exchanges. Appendix B.2 gives the fitted models and coefficients.

5 Experiments

We evaluate AAS-MAD on eight reasoning benchmarks (3,598 cases) and 104 FIFA 2026 fixtures with three seeds. The reasoning benchmarks are ProofWriter (Tafjord et al., 2021), LogiQA2 (Liu et al., 2023), FLDx2 (Morishita et al., 2024), ForecastQA (Jin et al., 2021), TriviaQA (Joshi et al., 2017), GPQA (Rein et al., 2024), FEVEROUS (Aly et al., 2021), and HOVER (Jiang et al., 2020). The two heads are trained on 5,101 labeled one-step branches from 1,139 examples across four reasoning domains, with 341 separate validation examples; Appendix Table A1 assigns every dataset to its fitting, tuning, or evaluation pool, and no two pools share a task identity or a normalized evidence group. All language-model comparisons use Qwen3-Flash with thinking disabled; debate uses four agents. Costs sum known returned input and output tokens for the complete pipeline, and failures remain in each registered denominator. Appendix B gives the protocols.

5.1 Reasoning accuracy and computation

Tables 1 and 2 show that AAS-MAD solves 2,546/3,598 cases (70.76%) at 21.09k tokens per item. The strongest debate baseline, MAD-M^2, reaches 69.73% at 37.89k; the other debate baselines reach 68.95–69.68% at 29.63–35.73k. Selective revision therefore gives the best pooled accuracy with the lowest pooled token cost among these methods.

Table 1: Cross-domain evaluation. A. ProofWriter, LogiQA2, FLDx2, ForecastQA

	ProofWriter (400)		LogiQA2 (500)		FLDx2 (500)		ForecastQA (500)	
Method	Acc. ↑	kTok ↓	Acc. ↑	kTok ↓	Acc. ↑	kTok ↓	Acc. ↑	kTok ↓
Vanilla MAD	80.25	23.82	**85.80**	27.15	87.80	34.74	59.40	15.32
Sparse MAD	80.75	21.16	84.40	23.72	86.00	30.02	58.20	13.73
FREE-MAD	81.75	25.71	84.20	29.92	**89.00**	36.34	56.80	18.43
MAD-M^2	81.75	27.71	85.60	29.92	88.00	42.19	60.40	13.84
AAS-MAD	**87.25**	24.27	85.60	6.19	84.60	17.97	**62.00**	4.37

Accuracy (%) and known returned input+output kTok per item; n in parentheses. AAS-MAD uses the configured dual-head controller with Resolve and its stepwise STOP gate ($K \leq 6$). Its matched opening and endpoint solve 2,494 and 2,546 of these 3,598 cases (+1.45 pp); the best competing endpoint solves 2,509.

Table 2: Cross-domain evaluation. B. TriviaQA, GPQA, FEVEROUS, HOVER

	TriviaQA (500)		GPQA (198)		FEVEROUS (500)		HOVER (500)	
Method	Acc. ↑	kTok ↓	Acc. ↑	kTok ↓	Acc. ↑	kTok ↓	Acc. ↑	kTok ↓
Vanilla MAD	60.60	12.17	67.68	107.90	47.20	39.82	**69.60**	48.22
Sparse MAD	60.80	11.61	**68.18**	86.12	46.00	38.15	69.20	44.98
FREE-MAD	**62.40**	16.62	65.15	104.96	45.40	42.41	68.40	51.28
MAD-M^2	61.60	9.61	67.68	107.04	46.00	52.59	68.00	59.95
AAS-MAD	62.00	6.25	**68.18**	72.11	**57.80**	42.65	60.40	26.35

Same cases and accounting as Panel A. Bold marks the best accuracy, underlining the second-best accuracy, and a green tint the lowest cost within each dataset and metric.

Where selective computation pays off. AAS-MAD leads on ProofWriter, ForecastQA, and FEVEROUS, ties the best GPQA accuracy, and has the lowest total token cost per item on six of eight tasks. On ProofWriter and FEVEROUS, it raises the strongest baseline accuracy by 5.50 and 10.60 points, respectively. On LogiQA2, it matches MAD-M^2 at 85.60% with 6.19k rather than 29.92k tokens; on GPQA, it matches Sparse MAD at 68.18% with 72.11k rather than 86.12k. Both comparisons reach similar collective decisions with less computation than full-response protocols.

The opening controls identify how these advantages arise. On ProofWriter, a strong 84.25% opening is further improved to 87.25%; on FEVEROUS, the protocol establishes a 57.40% opening and retains that advantage through revision to 57.80%. Initialization thus supplies candidate reasoning while exchanges improve or preserve it. FLDx2 shows the complementary pattern, with the largest revision gain (+4.20 pp) from a weaker opening, although its endpoint remains below FREE-MAD. HOVER also favors the baseline endpoints. Thus the pooled advantage combines strong initial decisions on some tasks with effective correction on others, rather than a uniform per-task ranking.

5.2 What additional revisions contribute

The complete AAS-MAD protocol begins with a competitive opening decision (69.32% accuracy). Subsequent selective revisions improve it from 2,494 to 2,546 correct decisions (+1.45 pp), the **largest pooled opening-relative gain** among the five MAD methods in Figure 3. MAD-M^2 improves from 2,465 to 2,509 (+1.22 pp), using 28.15k additional tokens per case versus AAS-MAD's 10.23k, so AAS-MAD obtains an 18.18% larger net gain with 36.34% of the additional tokens, a 3.25× net correction per additional token. AAS-MAD also has the largest opening-relative gains on FLDx2 (+4.20 pp) and TriviaQA (+1.40 pp), while using the fewest additional tokens per case on all eight tasks: on FLDx2, its 21 net corrections cost 10.76k additional tokens per case, and on TriviaQA it converts the same 303 correct opening decisions as Sparse MAD into 310 rather than 304, at 2.94k rather than 7.60k additional tokens.

At the endpoints of the final adaptive policy on the 3,598 reasoning cases, we observe 136 promotions, no acquisitions, and 84 opening-relative damages, yielding 52 net corrections. This is specific to the evaluated policy endpoints rather than an invariant of AAS-MAD. The separate six-action diagnostic continuations contain seven acquisitions at depth six (Table A4). Promotion and acquisition classify corrections by whether the correct answer was present among the opening commitments; they do not determine whether subsequent revisions introduced useful new reasoning.

Preserving correction gains through selective stopping. Appendix C.1 reports the depth-wise decomposition and its dataset-level patterns: correction saturates on some tasks, damage overtakes correction early on others, and further revision remains useful on the rest. These differences motivate state-dependent stopping rather than a common early cutoff. The diagnostic paths share the final policy's openings but can differ in their continuations, so they do not isolate the effect of STOP. Appendix C reports operation-cost sensitivity.

5.3 Component ablations

Table 3 compares eight ablations on one registered 300-case slice. The synchronous controls revise all four agents once; other conditions alter ranking, operations, features, or answer repair (Appendix B.6).

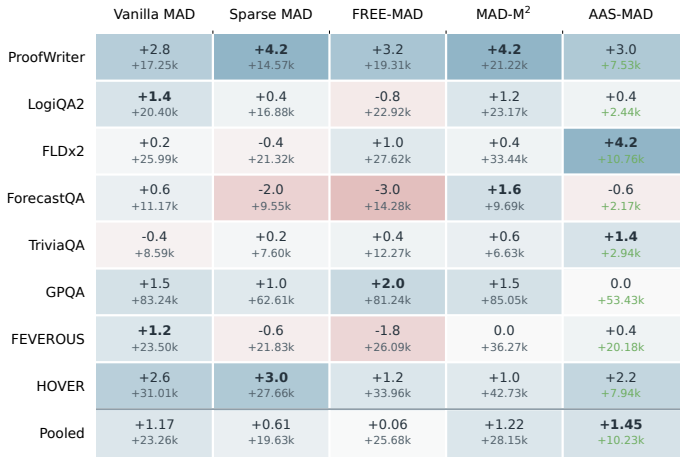

Figure 3: Revision gain and cost relative to each method's own opening. Cells show accuracy change (pp, top) and additional kTok per case (bottom). Bold marks the largest gain; green marks the lowest additional cost. Pooled values weight 3,598 cases. Numerical values appear in Table A3.

Table 3: Mechanism ablations. All rows share the registered 300-case reasoning slice.

Condition	Acc. (%)	Δ (pp)	kTok	Calls
AAS-MAD (full; STOP, max 6)	**74.67**	–	18.65	4.94
Random action + STOP	73.67	-1.00	14.45	4.38
Reflect only + STOP	74.33	-0.33	18.40	4.91
Resolve only + STOP	73.33	-1.33	13.22	4.39
At most one Reflect	73.00	-1.67	11.99	4.24
No graph inputs: refit both heads + STOP	73.67	-1.00	17.19	4.84
No scheduler: graph + Vanilla MAD one round	74.00	-0.67	37.35	8.04
No graph: DAG + Vanilla MAD one round	74.33	-0.33	35.97	8.05
No answer-format retries	74.00	-0.67	24.35	4.94

AAS-MAD is the full method with stepwise STOP, and $K \leq 6$; the random-action, Reflect-only, and one-Reflect arms keep the same gate, and the one-Reflect arm halts after its first accepted revision. The two one-round arms synchronously revise four agents once. Resolve-only stops at $K = 0$ in 263/300 cases. Removing graph inputs refits both heads but keeps the graph-derived menu and context. All conditions score the same 300 cases; kTok and Calls cover the complete inference pipeline.

Allocating revisions matters more than revising every agent. The full controller solves 224/300 cases at 18.65k tokens, compared with 222–223 at 35.97–37.35k for synchronous revision. The graph-context control is especially informative: retaining the shared representation but revising all four agents once roughly doubles token use without improving observed accuracy. Structured context alone therefore does not deliver the full efficiency benefit; selecting which revision to execute contributes to the observed accuracy–cost trade-off.

The remaining controls test flexibility and information. One-Reflect costs 11.99k but loses five correct decisions, showing the value of allowing further exchanges when selected. Reflect-only retains 223/300 correct at 18.40k, indicating that inexpensive reflection accounts for most of the full system's performance. Resolve-only stops immediately on 263/300 cases and reaches 220/300: a menu containing only the more expensive operation offers fewer opportunities that pass the gate. Random ranking and removing graph inputs each reduce the result to 221/300. The latter refits both heads while retaining the graph-derived menu and context, so its comparison supports the contribution of graph features to prediction. These small-slice results favor ranking informative revisions and continuing beyond one exchange when useful.

5.4 FIFA 2026 FORECASTING AND STOPPING

FIFA tests reasoning with uncertain rankings, form, lineups, and market evidence. We use regulation-time win/draw/loss labels and fixed evidence snapshots. The same configured controller is used on FIFA; all local methods share a terminal probability reader, prior fusion, and draw adjustment. Table 4 covers the local reasoning methods; football-specific reference forecasts with different information sets are reported in Appendix D (Rezaei & Samadi, 2026; Dixon & Coles, 1997; Groll et al., 2019; Zeileis, 2026).

Table 4: FIFA 2026: local reasoning methods.

Method	n	Acc. ↑	Brier ↓	NLL ↓	RPS ↓	kTok ↓	Calls ↓
Single	312	63.78	0.513	0.877	0.155	8.336	2.000
Vote4	312	64.42	0.512	0.878	0.153	30.762	5.006
CoT-SC8	312	62.50	0.527	0.900	0.158	44.244	9.000
Vanilla MAD	312	64.10	0.508	0.871	0.152	79.879	9.006
Sparse MAD	312	66.67	0.506	0.865	0.152	74.705	9.003
FREE-MAD	312	64.74	0.512	0.872	0.154	81.934	9.019
MAD-M^2	312	66.03	0.509	0.876	0.152	92.985	13.529
AAS-MAD	312	**68.59**	0.506	0.874	0.151	27.099	5.391

Acc. is regulation-time W/D/L accuracy (%). All rows use Qwen3-Flash, 104 matches with three seeds ($n = 312$), and the same probability reader, prior fusion, and draw adjustment. Best-value marks apply within this table. Reference forecasts use their native probabilities and information sets, without our terminal reader, prior fusion, or draw adjustment, so they are not matched controller comparisons; their token and call costs are not applicable. Sources and protocols are in Appendix D; matched stopping controls are in Table A6.

Forecast quality with substantially less deliberation. AAS-MAD achieves the highest local accuracy, $214/312$ (68.59%), at 27.10k tokens per item. Sparse MAD, the strongest debate baseline by accuracy, solves $208/312$ at 74.71k; MAD-M^2 solves $206/312$ at 92.99k. AAS-MAD also exceeds Vote4 and CoT-SC8 while using fewer tokens. Because every local method uses the same probability reader, prior fusion, and draw adjustment, these differences compare the reasoning supplied to a common forecasting pipeline. The best local RPS (0.151) and tied best rounded Brier score (0.506) further show that the accuracy advantage is compatible with competitive probability forecasts; Sparse MAD retains the best NLL.

Under a fixed evidence snapshot, further exchanges reinterpret existing match information without acquiring new evidence. CoT-SC8 and all four debate baselines spend more tokens without reaching AAS-MAD's accuracy, which favors limiting repeated deliberation once the available evidence already yields a strong forecast.

Matched paths isolate the value of stopping. The matched six-action control solves $213/312$ at 84.72k tokens, compared with AAS-MAD's $214/312$ at 27.10k; all 312 prefixes match through stopping. Terminating these same reasoning paths saves 68.01% of the full-pipeline tokens at nearly identical observed accuracy. Fixing the opening, ranker, and terminal readout locates this saving in the avoided exchanges. The matched opening already solves $213/312$, and revision contributes one rescue without damage under the common probability head (Table A6). FIFA therefore demonstrates the value of preserving predictive quality without routinely spending a six-action budget.

6 CONCLUSION

AAS-MAD uses a public argument state, predicted rescue, and call costs to select focused revisions, and a damage-aware gate to decide when to stop. Our retrospective evaluation shows higher pooled reasoning accuracy with substantially fewer tokens than the evaluated debate baselines. Opening-relative analysis shows that, at the evaluated policy endpoints, successful corrections promote answers already present in the ensemble. These findings support selective deliberation: allocating computation according to the expected value of the next revision rather than extending debate uniformly.

AI USE STATEMENT

In this work, we used generative AI tools for the following tasks with required disclosure: formulating and refining the mathematical statements reported in the paper; providing feedback on the research methodology and the experimental design; assisting with the translation of author-written notes and materials; and interpreting and discussing experimental results.

We have not used generative AI tools to propose the research idea, to build the theoretical or conceptual framing of the method, to provide critical ingredients for proving mathematical claims, to write proofs, to design the method or its components, or to clean and reformat the datasets. Qualitative and thematic data analysis is not applicable to this work.

Additionally, we have used generative AI tools for tasks with recommended disclosure: brainstorming; sourcing and searching for information, including locating relevant literature and checking bibliographic entries; creating and modifying scientific figures; writing the implementation code for the authors' method and experimental pipeline; helping organize results; drafting the initial version of some paragraphs; and polishing the language of the paper.

The division of labor is explicit: the authors designed the method, its components, and the experimental protocol, while generative AI tools assisted in writing the implementation code under author direction, and all AI-assisted code was reviewed and revised by the authors.

We have reviewed all AI-assisted work: the mathematical statements were re-derived and checked by the authors, and the AI-assisted text and code were reviewed and revised by the authors. The research idea, the method, the experimental protocol, and the reported measurements are the authors' own work: the authors proposed the idea and used brainstorming with AI tools to operationalize and iterate it. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

The FIFA collection supports research on collective reasoning under uncertainty. Predictions may be wrong despite traceable evidence and are not betting advice. Dataset documentation records provenance and temporal availability, while external forecasts retain their original information conditions.

REFERENCES

Rami Aly, Zhijiang Guo, Michael Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. FEVEROUS: Fact extraction and VERification over unstructured and structured information. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/68d30a9594728bc39aa24be94b319d21-Abstract-round1.html.

Abhijith Babu, Ramneet Kaur, Vishal Pramanik, Olivera Kotevska, Nathaniel D. Bastian, Susmit Jha, Sunny Raj, Yanzhao Wu, Sumit Kumar Jha, and Anirban Roy. Consilience: Conformally calibrated communication control for hidden-profile multi-agent reasoning, 2026. URL https://arxiv.org/abs/2608.20564.

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

Claudette Cayrol and Marie-Christine Lagasquie-Schiex. On the acceptability of arguments in bipolar argumentation frameworks. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 3571 of *Lecture Notes in Computer Science*, pp. 378–389. Springer, 2005. doi: 10.1007/11518655_33. URL https://link.springer.com/chapter/10.1007/11518655_33.

Hyeong Kyu Choi, Xiaojin Zhu, and Sharon Li. Debate or vote: Which yields better decisions in multi-agent large language models? In *Advances in Neural Information Processing Systems*, volume 38, pp. 101732–101764, 2025. doi: 10.52202/085713-3405. URL https://papers.neurips.cc/paper_files/paper/2025/hash/934252acd87f254d5d4672fbde283bd2-Abstract-Conference.html.

Yu Cui, Hang Fu, Haibin Zhang, Licheng Wang, and Cong Zuo. Free-MAD: Consensus-free multi-agent debate. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 31977–31997, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1600. URL https://aclantology.org/2026.findings-acl.1600/.

Mark J. Dixon and Stuart G. Coles. Modelling association football scores and inefficiencies in the football betting market. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 46 (2):265–280, 1997. doi: 10.1111/1467-9876.00065.

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

Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995. doi: 10.1016/0004-3702(94)00041-X.

Sugyeong Eo, Hyeonseok Moon, Evelyn Hayoon Zi, Chanjun Park, and Heuiseok Lim. Debate only when necessary: Adaptive multiagent collaboration for efficient LLM reasoning. *arXiv preprint arXiv:2504.05047*, 2025. doi: 10.48550/arXiv.2504.05047. URL https://arxiv.org/abs/2504.05047v2.

Andrew Estornell, Jean-François Ton, Yuanshun Yao, and Yang Liu. ACC-Collab: An actor-critic approach to multi-agent LLM collaboration. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/e187897ed7780a579a0d76fd4a35d107-Abstract-Conference.html.

Wei Fan, JinYi Yoon, and Bo Ji. iMAD: Intelligent multi-agent debate for efficient and accurate LLM inference. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(35):29403–29411, 2026. doi: 10.1609/aaai.v40i35.40181. URL https://ojs.aaai.org/index.php/AAAI/article/view/40181.

Weijun Gao, Xiang Ding, Haoyang Liu, and Tiancheng Xing. Where reasoning diverges: Localized multi-agent debate for multi-hop question answering. *arXiv preprint arXiv:2608.01463*, 2026. doi: 10.48550/arXiv.2608.01463. URL https://arxiv.org/abs/2608.01463v2.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63:3–42, 2006. doi: 10.1007/s10994-006-6226-1. URL https://link.springer.com/article/10.1007/s10994-006-6226-1.

Andreas Groll, Christophe Ley, Gunther Schauberger, and Hans Van Eetvelde. A hybrid random forest to predict soccer matches in international tournaments. *Journal of Quantitative Analysis in Sports*, 15(4):271–287, 2019. doi: 10.1515/jqas-2018-0060. URL https://doi.org/10.1515/jqas-2018-0060.

Nicholas Hay, Stuart Russell, David Tolpin, and Solomon Eyal Shimony. Selecting computations: Theory and applications. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, 2012. URL https://arxiv.org/abs/1207.5879.

Xingyuan Hua, Sheng Yue, Xinyi Li, Yizhe Zhao, Jinrui Zhang, and Ju Ren. Context learning for multi-agent discussion. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2602.02350.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/8b4add8b0aa8749d80a34ca5d941c355-Abstract-Conference.html.

Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. HoVer: A dataset for many-hop fact extraction and claim verification. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 3441–3460, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.309. URL https://aclanthology.org/2020.findings-emnlp.309/.

Woojeong Jin, Rahul Khanna, Suji Kim, Dong-Ho Lee, Fred Morstatter, Aram Galstyan, and Xiang Ren. ForecastQA: A question answering challenge for event forecasting with temporal text data. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4636–4650. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.357. URL https://aclanthology.org/2021.acl-long.357/.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Xinglin Wang, Bin Sun, Heda Wang, and Kan Li. Escape sky-high cost: Early-stopping self-consistency for multi-step reasoning. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/3fe2a777282299ecb4f9e7ebb531f0ab-Abstract-Conference.html.

Yunxuan Li, Yibing Du, Jiageng Zhang, Le Hou, Peter Grabowski, Yeqing Li, and Eugene Ie. Improving multi-agent debate with sparse communication topology. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7281–7294, Miami, Florida, USA, November 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.427. URL https://aclanthology.org/2024.findings-emnlp.427/.

Hanmeng Liu, Jian Liu, Leyang Cui, Zhiyang Teng, Nan Duan, Ming Zhou, and Yue Zhang. LogiQA 2.0—an improved dataset for logical reasoning in natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2947–2962, 2023. doi: 10.1109/TASLP.2023.3293046. URL https://ieeexplore.ieee.org/document/10174688/.

Yexiang Liu, Jie Cao, Zekun Li, Ran He, and Tieniu Tan. Breaking mental set to improve reasoning through diverse multi-agent debate. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/3de667dab3b3d812583abc0a786139a0-Paper-Conference.pdf.

Jie Ma, Zhitao Gao, Qi Chai, Wangchun Sun, Pinghui Wang, Hongbin Pei, Jing Tao, Lingyun Song, Jun Liu, Chen Zhang, and Lizhen Cui. Debate on graph: A flexible and reliable reasoning framework for large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(23):24768–24776, 2025. doi: 10.1609/aaai.v39i23.34658. URL https://ojs.aaai.org/index.php/AAAI/article/view/34658.

Terufumi Morishita, Gaku Morio, Atsuki Yamaguchi, and Yasuhiro Sogawa. Enhancing reasoning capabilities of LLMs via principled synthetic logic corpus. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2340. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/8678da90126aa58326b2fc0254b33a8c-Abstract-Conference.html.

Debjit Paul, Mete Ismayilzada, Maxime Peyrard, Beatriz Borges, Antoine Bosselut, Robert West, and Boi Faltings. REFINER: Reasoning feedback on intermediate representations. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1100–1126. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.eacl-long.67. URL https://aclanthology.org/2024.eacl-long.67/.

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. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Mina Rezaei and S. Yaser Samadi. Predicting the 2026 FIFA world cup with sufficient dimension reduction of Elo rating histories, 2026. URL https://arxiv.org/abs/2606.24171v1.

Michael Rizvi-Martel, Satwik Bhattamishra, Neil Rathi, Guillaume Rabusseau, and Michael Hahn. Benefits and limitations of communication in multi-agent reasoning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/e9372abc370b8302459abdce4bdce5a5-Paper-Conference.pdf.

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.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/1b623663fd9b874366f3ce019fdfdd44-Abstract-Conference.html.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3621–3634, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.317. URL https://aclanthology.org/2021.findings-acl.317/.

Hongduan Tian, Xiao Feng, Ziyuan Zhao, Xiangyu Zhu, Rolan Yan, and Bo Han. Multi-agent debate with memory masking. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/09425891e393e64b0535194a81ba15b7-Abstract-Conference.html.

Chao Wang, Han Lin, Huaze Tang, Huijing Lin, and Wenbo Ding. RUMAD: Reinforcement-unifying multi-agent debate. In *Proceedings of the 25th International Conference on Autonomous Agents and Multiagent Systems*, pp. 3695–3703. International Foundation for Autonomous Agents and Multiagent Systems, 2026. doi: 10.65109/CBJO8409. URL https://ifaamas.org/Proceedings/aamas2026/pdfs/CBJO8409.pdf. AAAI Track.

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

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

Hao Wu, Shoucheng Song, Chang Yao, Haoyu Wang, Huaiyu Wan, Youfang Lin, and Kai Lv. Group perspective matters: Regulating debate relationships can mitigate blind conformity in multi-agent debate. *arXiv preprint arXiv:2608.03648*, 2026. URL https://arxiv.org/abs/2608.03648.

Ruoyu Xu, Gaoxiang Li, and Victor S. Sheng. GAVEL: Evidence-contract debate with mechanized scrutiny for provenance-grounded fact-checking. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 35907–35920, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1789. URL https://aclanthology.org/2026.findings-acl.1789/.

Haocheng Yang, Fengxiang Cheng, Tianjun Yao, Mengyue Yang, Jiajun Chai, Xiaohan Wang, Guojun Yin, Wei Lin, Soummya Kar, Fenrong Liu, Haoxuan Li, and Yisen Wang. MAD-Logic: Multi-agent debate enhances symbolic translation and reasoning. In *International Conference on Learning Representations*, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/db5d2bcbd77637c2b92d21be9e990384-Paper-Conference.pdf.

Wei Yang, Shixuan Li, Heng Ping, Peiyu Zhang, Paul Bogdan, and Jesse Thomason. Auditing multi-agent LLM reasoning trees outperforms majority vote and LLM-as-judge. *arXiv preprint arXiv:2602.09341*, 2026b. doi: 10.48550/arXiv.2602.09341. URL https://arxiv.org/abs/2602.09341v2.

Sukwon Yun, Jie Peng, Pingzhi Li, Wendong Fan, Jie Chen, James Y Zou, Guohao Li, and Tianlong Chen. Graph-of-Agents: A graph-based framework for multi-agent LLM collaboration. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/21a87fb07e9ab0ecc4d9d1b940676229-Paper-Conference.pdf.

Achim Zeileis. Football meets machine learning: Forecasting the 2026 FIFA world cup. Research forecast and published match-probability matrix, June 2026. URL https://www.zeileis.org/news/fifa2026/.

Yuting Zeng, Weizhe Huang, Lei Jiang, Tongxuan Liu, XiTai Jin, Chen Tianying Tiana, Jing Li, and Xiaohua Xu. S^2-MAD: Breaking the token barrier to enhance multi-agent debate efficiency. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 9393–9408. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-long.475. URL https://aclanthology.org/2025.naacl-long.475/.

Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Yu, and Tianlong Chen. Cut the crap: An economical communication pipeline for LLM-based multi-agent systems. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/bbc461518c59a2a8d64e70e2c38c4a0e-Paper-Conference.pdf.

Xiaochen Zhu, Caiqi Zhang, Yizhou Chi, Tom Stafford, Nigel Collier, and Andreas Vlachos. Demystifying multi-agent debate: The role of confidence and diversity. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 33909–33930. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.1694. URL https://aclanthology.org/2026.findings-acl.1694/.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. GPTSwarm: Language agents as optimizable graphs. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62743–62767. PMLR, 2024. URL https://proceedings.mlr.press/v235/zhuge24a.html.

A CORRECTION ACCOUNTING AND LOCAL CONTROL

A.1 OPENING-RELATIVE OUTCOMES

Let $C_0 = \{d_i^0\}_{i=1}^m$ contain the opening commitments modulo task equivalence. With the same readout at opening and endpoint,

$$\begin{aligned} P_t &= \mathbf{1}\{\hat{y}_0 \not\equiv y,\ \hat{y}_t \equiv y,\ y \in C_0\}, \\ A_t &= \mathbf{1}\{\hat{y}_0 \not\equiv y,\ \hat{y}_t \equiv y,\ y \notin C_0\}, \\ D_t &= \mathbf{1}\{\hat{y}_0 \equiv y,\ \hat{y}_t \not\equiv y\}. \end{aligned} \tag{11}$$

The four opening–endpoint correctness combinations give $u(\hat{y}_t, y) - u(\hat{y}_0, y) = P_t + A_t - D_t$. Promotion and acquisition partition successful corrections by opening coverage. For a selector restricted to C_0, $\mathbf{1}\{\hat{y}_t \equiv y\} \le \mathbf{1}\{y \in C_0\}$ pointwise, yielding the candidate-coverage ceiling after taking expectations.

A.2 LOCAL LABELS AND PATH ACCOUNTING

With the fixed native readout ν, the local labels of Equation 6 satisfy, writing $\mathrm{res}_t = \mathrm{res}(s_t, a_t, s_{t+1})$ and $\mathrm{dam}_t = \mathrm{dam}(s_t, a_t, s_{t+1})$,

$$\mathrm{res}_t - \mathrm{dam}_t = u(\nu(s_{t+1}), y) - u(\nu(s_t), y). \tag{12}$$

Summing over completed exchanges telescopes:

$$\sum_{t<\tau} (\mathrm{res}_t - \mathrm{dam}_t) = u(\nu(s_\tau), y) - u(\nu(s_0), y). \tag{13}$$

Weighting damage or subtracting call cost gives the corresponding cost-sensitive path value. Local events can recur along a trajectory, whereas opening-relative outcomes are counted once at the endpoint.

B IMPLEMENTATION AND EVALUATION DETAILS

B.1 DATASET SPLITS AND ROLE ASSIGNMENT

Three of the reasoning domains that supply fitting data also appear among the evaluated benchmarks, so Table A1 states the pool assignment for every dataset. Head fitting reads FIT only. Scheduler tuning selects the ranker coefficients on CAL, and STOP tuning selects the gate threshold on the trajectories that the selected ranker produces on those same tasks; no evaluation case enters either stage. Recomputing all FIT, CAL, and evaluation groups from normalized task inputs with one identical function yields no shared task identity and no shared premise or evidence group between the fitting pools and any evaluation cohort. That check is exact normalized-input matching rather than semantic near-duplicate detection, and three tasks are excluded from validation pools permanently by dataset and task identity across seeds. FIFA supplies no fitting or tuning case: its heads, coefficients, and gate are transferred unchanged.

B.2 CONTROLLER SPECIFICATION

Each rescue and damage predictor contains 96 extremely randomized trees and uses the same 88-dimensional state–action vector. Features describe public answers, rationale and graph structure, the proposed focus and participants, operation, nominal call cost, position, and action history. The training and validation sets contain 1,139 and 341 task–seed identities, respectively. Gold correctness supplies the offline branch labels only.

We first tune the scheduler on the validation set, run the selected scheduler on that validation set, and then tune the STOP gate on the resulting validation trajectories.

Table A1: Dataset splits and role assignment. Counts are registered task identities; evaluation counts are registered cases (one seed for the reasoning cohorts, 104 fixtures with three seeds for FIFA). Head fitting reads FIT only, scheduler and STOP tuning read CAL only, and no evaluation case enters either stage. A dash marks a dataset that supplies no data for that role.

Dataset	Head fitting	Scheduler tuning	STOP tuning	Final test
ProofWriter	312	98	98	400
LogiQA2	319	97	97	500
FOLIO	269	87	87	–
HotpotQA	239	59	59	–
FLDx2	–	–	–	500
ForecastQA	–	–	–	500
TriviaQA	–	–	–	500
GPQA	–	–	–	198
FEVEROUS	–	–	–	500
HOVER	–	–	–	500
FIFA 2026	–	–	–	312
Total	1,139	341	341	3,598 (+312)

Fitting and tuning pools use the registered ProofWriter, LogiQA2, FOLIO, and HotpotQA corpora. Evaluation cohorts use the official ProofWriter test file, the LogiQA2 development file, the FLDx2 test file, the ForecastQA development file, TriviaQA validation questions, GPQA-Diamond, the FEVEROUS development challenges, and the HOVER development release. Scheduler and STOP tuning reuse the same 341 CAL tasks at two stages: the ranker coefficients are selected first, and the gate threshold is then selected on the trajectories the selected ranker produces on those tasks.

Let $c(a)$ denote the nominal number of model calls required by action a, with $c(\mathrm{Reflect}) = 1$ and $c(\mathrm{Resolve}) = 2$. The final configuration is

$$q_r = \hat{p}_{\mathrm{res}} - 0.10c, \tag{14}$$

$$q_g = \hat{p}_{\mathrm{res}} - 0.10\hat{p}_{\mathrm{dam}} - 0.025c, \tag{15}$$

$$\theta = -0.01. \tag{16}$$

In the selected configuration, action ranking uses rescue predictions and nominal call cost, while the STOP gate additionally incorporates predicted damage; the damage coefficient in q_r is zero. Ranking uses q_r, and the gate accepts its top action when $q_g > \theta$. Both Reflect and Resolve are legal when their prerequisites hold. Reflect costs one nominal call, Resolve costs two, and the controller executes at most six exchanges.

The fitted heads are reused without refitting. The same coefficients are used for reasoning and FIFA.

B.3 Evaluation and cost

The eight reasoning cohorts and their sizes are listed in Section 5; pooled results weight cases rather than dataset percentages. Task-native classification or mathematical-equivalence scorers evaluate every registered case. Terminal failures, including cases whose opening prediction was correct, remain in the evaluation denominator and are scored by the registered readout. All MAD baselines are adapted to our common retry and task-readout rules. kTok sums returned input and output tokens for openings, executed revisions, answer repairs, and the FIFA terminal head when used. Calls counts returned model requests, while actions count completed revision exchanges.

B.4 Matched opening accuracy and cost

Table A2 compares AAS-MAD with its own zero-action opening under the same cases and task readout. FIFA applies the identical probability head, prior fusion, and draw adjustment at both endpoints.

Table A2: Matched opening controls. Accuracies and known returned kTok per item use the same cases and task readout at opening and after AAS-MAD. FIFA includes the same terminal probability head at both endpoints.

		Accuracy (%)			kTok	
Dataset	n	Opening	AAS-MAD	Δ (pp)	Opening	AAS-MAD
ProofWriter	400	84.25	87.25	+3.00	16.74	24.27
LogiQA2	500	85.20	85.60	+0.40	3.76	6.19
FLDx2	500	80.40	84.60	+4.20	7.21	17.97
ForecastQA	500	62.60	62.00	-0.60	2.20	4.37
TriviaQA	500	60.60	62.00	+1.40	3.31	6.25
GPQA	198	68.18	68.18	+0.00	18.68	72.11
FEVEROUS	500	57.40	57.80	+0.40	22.46	42.65
HOVER	500	58.20	60.40	+2.20	18.40	26.35
FIFA 2026	312	68.27	68.59	+0.32	22.64	27.10

B.5 Accuracy gains from each method's opening

Table A3 gives the numerical values shown in Figure 3.

Table A3: Changes from each method's own opening, corresponding to Figure 3. ΔAcc is accuracy gain in percentage points; ΔkTok is additional known returned input+output tokens per case in thousands. Pooled values weight all 3,598 cases. Bold marks the largest accuracy gain and light green the lowest additional token use in each row.

	Vanilla MAD		Sparse MAD		FREE-MAD		MAD-M^2		AAS-MAD	
Dataset	ΔAcc	ΔkTok	ΔAcc	ΔkTok	ΔAcc	ΔkTok	ΔAcc	ΔkTok	ΔAcc	ΔkTok
ProofWriter	+2.8	+17.25	**+4.2**	+14.57	+3.2	+19.31	**+4.2**	+21.22	+3.0	+7.53
LogiQA2	**+1.4**	+20.40	+0.4	+16.88	-0.8	+22.92	+1.2	+23.17	+0.4	+2.44
FLDx2	+0.2	+25.99	-0.4	+21.32	+1.0	+27.62	+0.4	+33.44	**+4.2**	+10.76
ForecastQA	+0.6	+11.17	-2.0	+9.55	-3.0	+14.28	**+1.6**	+9.69	-0.6	+2.17
TriviaQA	-0.4	+8.59	+0.2	+7.60	+0.4	+12.27	+0.6	+6.63	**+1.4**	+2.94
GPQA	+1.5	+83.24	+1.0	+62.61	**+2.0**	+81.24	+1.5	+85.05	0.0	+53.43
FEVEROUS	**+1.2**	+23.50	-0.6	+21.83	-1.8	+26.09	0.0	+36.27	+0.4	+20.18
HOVER	+2.6	+31.01	**+3.0**	+27.66	+1.2	+33.96	+1.0	+42.73	+2.2	+7.94
Pooled	+1.17	+23.26	+0.61	+19.63	+0.06	+25.68	+1.22	+28.15	**+1.45**	+10.23

B.6 Component-ablation protocol

The component study uses a 300-case slice spanning twelve reasoning domains (24 cases each) and AIME 2024/2025 (six cases each). All scheduled variants use the controller in Appendix B.2.

Random action samples from the legal menu under the same STOP gate. Reflect-only and Resolve-only restrict the menu by operation; one-Reflect also stops after its first accepted revision. No-graph-input refits both heads without graph-derived features while retaining the graph-based menu and deliberation context. The two no-scheduler controls give all four agents one synchronous revision, with and without graph context. No-answer-format-retry disables task-level answer repair. All nine rows retain the same 300-case denominator.

C Action outcomes and cost sensitivity

C.1 Net correction and its depth-dependent components

The registered complete six-action paths use the dual-head ranker with Resolve and the same 3,598 opening cases. At each depth, rescue is the number of initially incorrect cases currently correct, including promotion and acquisition; damage is the number of initially correct cases currently incorrect. A case can enter or leave either set as its answer changes.

Table A4: Opening-relative outcomes along complete six-action diagnostic paths. The denominator is 3,598 at every depth. Net rescue equals promotion plus acquisition minus damage. These continuations can differ from the final policy's observed paths.

Depth K	0	1	2	3	4	5	6
Promotion	0	123	175	184	185	184	187
Acquisition	0	0	3	5	6	7	7
Rescue	0	123	178	189	191	191	194
Damage	0	76	127	142	145	151	150
Net rescue	0	47	51	47	46	40	44

Table A4 shows an early peak followed by erosion and a partial recovery. After step two, the additional 16 rescued endpoints are outweighed by 23 additional damaged endpoints. Rescue growth slows markedly: it adds 55 cases from step one to two, but only 11, two, zero, and three over successive later steps. From depth two to six, rescued endpoints increase by 16, whereas damaged endpoints increase by 23, reducing net correction from 51 to 44. The relevant imbalance is whether further correction offsets further damage and computation.

The dataset trajectories shown in Figures 4 and 5 reveal different sources of this imbalance. On ProofWriter, rescue rises from 22 to 23 between steps two and six while damage rises from 11 to 16, reducing net rescue from 11 to seven. On FLDx2, rescue remains at 42 while damage rises from 26 to 30. FEVEROUS exhibits an earlier damaging shift: from step one to two, rescue rises from 11 to 20 but damage rises from nine to 24, turning a net gain of two into a loss of four. HOVER instead improves from a net gain of three at step one to 20 at step six. These contrasts motivate a controller that responds to the current reasoning state, rather than a single early cutoff chosen from the pooled curve.

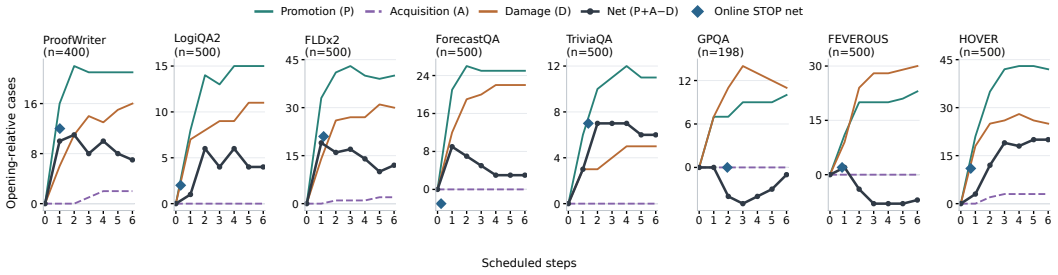

Figure 4: Opening-relative promotion, acquisition, damage, and net correction by depth on the complete diagnostic paths. The marked adaptive endpoint is shown at its mean action count; it is a separate policy summary, not an intermediate fixed-depth observation. The policies share openings but can follow different continuations.

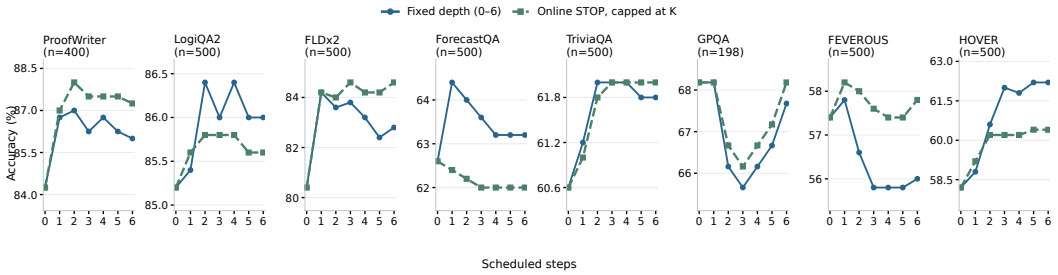

Figure 5: Accuracy over zero to six scheduled steps for complete paths (blue) and observed online-STOP prefixes (green dashed); vertical scales differ by panel. The complete-path and online-STOP curves are separate policy summaries, not a matched stopping ablation: they share openings but can follow different continuations. Online STOP yields 349 correct decisions on ProofWriter and 289 on FEVEROUS, versus 344 and 280 after six fixed steps; on HOVER, it yields 302 versus 311.

Local correction on the final policy's paths. The first four executed depths yield 94/401, 38/289, 10/266, and 2/233 rescues among currently incorrect decisions (23.44%, 13.15%, 3.76%, and 0.86%).

Damage rates also fall: 51/330, 30/263, 12/251, and 4/221 (15.45%, 11.41%, 4.78%, and 1.81%). At depths five and six, rescues are 3/220 and 7/207, against damages of 2/212 and 2/206. Thus the initial decline in correction opportunity is not strictly monotonic through the entire horizon. The ledger contains 3,099 completed revisions on 731 cases, averaging 0.86 actions per registered case. The remaining 2,867 have no recorded revision: 2,863 end at the opening and four are retained terminal failures. Completed revisions yield 154 local rescues and 101 local damages, for a net gain of 53. One additional terminal failure changes an initially correct case to an incorrect endpoint under the evaluation rule, yielding the reported net gain of 52. All four terminal failures remain in the evaluation denominator. Local events may recur, whereas the 136 promotions and 84 opening-relative damages count each endpoint once. GPQA illustrates a remaining allocation limitation: 53.43k additional tokens per case yield no net accuracy gain.

C.2 Operation choice and cost sensitivity

Table A5: Action outcomes and cost sensitivity on the final policy's paths. Top: observed action outcomes by dataset, where R/W is rescues per incorrect pre-action decision, D/C is damages per correct pre-action decision, and kTok is mean returned tokens per completed action. Bottom: Resolve availability on the same executed-state menus, where the no-cost column removes only the ranking call penalty; no unexecuted outcomes are imputed.

Dataset	Action	n	R/W	D/C	kTok
ProofWriter	Reflect	398	21/99	8/299	7.32
ProofWriter	Resolve	8	0/3	1/5	12.44
LogiQA2	Reflect	160	6/78	5/82	7.43
LogiQA2	Resolve	5	1/3	0/2	5.97
FLDx2	Reflect	551	44/210	22/341	8.52
FLDx2	Resolve	30	1/9	2/21	22.80
ForecastQA	Reflect	119	5/73	8/46	9.03
ForecastQA	Resolve	3	1/2	0/1	4.14
TriviaQA	Reflect	672	10/498	4/174	2.11
TriviaQA	Resolve	14	1/10	0/4	3.72
GPQA	Reflect	377	15/212	15/165	27.59
GPQA	Resolve	6	0/2	0/4	29.51
FEVEROUS	Reflect	396	19/237	16/159	24.70
FEVEROUS	Resolve	19	0/12	1/7	16.47
HOVER	Reflect	336	30/164	19/172	11.64
HOVER	Resolve	5	0/4	0/1	12.06

Dataset	Menus	Resolve legal	Selected	No-cost first
ProofWriter	406	406	8	80
LogiQA2	165	165	5	56
FLDx2	581	581	30	156
ForecastQA	122	122	3	56
TriviaQA	686	686	14	294
GPQA	383	381	6	128
FEVEROUS	415	409	19	115
HOVER	341	341	5	109

Action outcomes. Table A5 summarizes the 3,099 revisions executed by the final policy. Rescue rates use currently incorrect decisions as their denominator; damage rates use currently correct decisions. Operation-level rows characterize the states selected by the policy.

Reflect revises one agent's reasoning using the public state; Resolve adds a targeted challenge–response exchange. Resolve rescues four of 45 currently incorrect decisions (8.89%), compared with 150 of 1,571 for Reflect (9.55%). These rates are conditional on policy-selected states and do not constitute a matched comparison of the two operations. Resolve costs 15.87k tokens and

approximately two calls per action on average, versus 11.76k and approximately one call for Reflect. It is selected 90 times, compared with 3,009 Reflect actions.

Ranking-cost sensitivity. Removing only the ranking call penalty makes Resolve the top action in 994 of the same 3,099 recorded menus rather than 90, changing 904 operation choices. The term therefore materially affects operation selection.

D FIFA HEAD AND TRANSFER EVALUATION

The controller transfers unchanged to 104 FIFA fixtures with three seeds. Qwen3-Flash reads the complete opening, completed revisions, native vote, and frozen pre-match base rates with thinking disabled, temperature zero, and top-p one. Its probability vector p is fused with prior π as

$$\widetilde{p} = 0.71p + 0.29\pi. \tag{17}$$

The margin-based draw adjustment uses parameters $(0.08, 0.25)$ and renormalizes the result. The matched opening passes through the identical probability transformation. The rescue and damage predictors and both controller scoring rules are reused unchanged. The football-specific terminal-readout settings, probability-fusion weight, and draw-adjustment parameters are selected on historical football matches (pre-2014 qualifiers and 2006–2022 World Cup and qualifier fixtures) and fixed for the FIFA evaluation; no FIFA 2026 outcome or gold label enters their selection.

Table A6 compares stopping under the same ranker, opening, and probability head.

Table A6: FIFA stopping control: the final dual-head ranker, opening, and probability head are fixed. Costs include the complete pipeline; all arms retain 312 cases.

Controller	Correct	Acc. (%)	Brier	kTok	Calls
Opening	213/312	<u>68.27</u>	0.505	22.64	5.00
Adaptive STOP	214/312	**68.59**	0.506	27.10	5.39
Fixed six actions	213/312	<u>68.27</u>	0.509	84.72	11.00

The fixed-six-action control disables learned stopping while retaining the ranking rule, opening, action menu, horizon, and probability head. Its trajectory matches the adaptive arm through the adaptive stopping point. Evaluation uses a frozen pre-match evidence snapshot. The three seeds are repeated evaluations of the same 104 fixtures; $n = 312$ denotes fixture–seed evaluations rather than 312 independent fixtures.

Non-MAD football baselines. Table A7 reports football-specific reference forecasts evaluated using their native outputs and information sets. Elo with multinomial calibration and independent Poisson follow the baseline formulations of Rezaei & Samadi (2026); Dixon–Coles uses time decay and low-score correction (Dixon & Coles, 1997). These models and the eight-feature adaptation of the hybrid random forest (Groll et al., 2019) are fitted on historical matches ending before the tournament and evaluated once on each of the 104 fixtures. The adapted forest is distinct from the full Groll/Zeileis system, whose author-published match-probability matrix is rescored on those fixtures (Zeileis, 2026).

The SDR SIR-2 (M9) and SAVE-2 (M11) rows use the author-published Table 5 probabilities for 72 group-stage fixtures (Rezaei & Samadi, 2026), with rounded probabilities renormalized before scoring. Every row uses its native probabilities and information set, without our terminal LLM reader, prior fusion, or draw adjustment. These are forecasting references rather than matched controller interventions; their LLM token and call costs are not applicable.

Table A7: Non-MAD football forecasting baselines (native outputs). These rows use each model's native probabilities and information set, without the terminal LLM reader, prior fusion, or draw adjustment.

Method	n	Acc. (%)	Brier ↓	NLL ↓	RPS ↓
Elo + multinomial	104	**62.50**	0.525	0.882	0.159
Independent Poisson	104	**62.50**	0.528	0.895	0.161
Dixon-Coles	104	59.62	0.560	0.941	0.180
Hybrid RF (8 features)	104	59.62	0.549	0.913	0.171
Groll/Zeileis hybrid	104	<u>60.58</u>	0.516	0.877	0.158
SDR SIR-2 (M9)	72	56.94	0.572	0.955	0.184
SDR SAVE-2 (M11)	72	54.17	0.572	0.955	0.184
AAS-MAD	312	**68.59**	0.506	0.874	0.151

SDR covers 72 group-stage fixtures; all other rows cover 104 fixtures. These are forecasting references rather than matched controller interventions, so LLM token and call costs are not applicable. AAS-MAD is included for comparison rather than as a native-output forecast: it uses the full pipeline, including the terminal LLM reader, prior fusion, and draw adjustment, over the same 104 fixtures with three seeds ($n = 312$).