MATCHCREDIT: STATE-MATCHED CREDIT ASSIGNMENT FOR ADAPTIVE MULTI-AGENT REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-agent reasoning (MAR) can improve reasoning reliability through additional collaboration, but adaptive routing requires determining which collaboration operator is worthwhile at each intermediate state. Existing query-level labels or trajectory-level returns provide only coarse supervision for this decision, as the effect of an operator is confounded by the reached interaction state and subsequent routing choices. We propose **MatchCredit**, a state-matched credit assignment framework for efficient adaptive MAR. MatchCredit expands alternative operators from the same reached state and evaluates their downstream continuations under a shared prefix, thereby reducing prefix-state confounding. Terminal correctness and additional suffix cost are then converted into state-local pairwise operator preferences to train a lightweight router. Collaboration trees are used only for offline supervision construction, while test-time reasoning follows a single adaptive trajectory. Experiments on six reasoning benchmarks show that MatchCredit substantially reduces inference cost while preserving or modestly improving reasoning accuracy, yielding a favorable accuracy–cost trade-off over representative MAR methods. Averaged across the six benchmarks, MatchCredit achieves 86.66% accuracy with 2,193 tokens per query.

1 INTRODUCTION

Large language model based multi-agent reasoning (MAR) enables multiple agents to exchange complementary analyses, refine intermediate solutions, and aggregate diverse opinions for complex reasoning tasks (Qian et al., 2024; Hong et al., 2024; Chen et al., 2024). By combining diverse reasoning capabilities with structured interaction, MAR systems improve robustness and solution quality over isolated inference. However, these gains often require additional model calls, message exchanges, and agent coordination, resulting in substantial inference overhead that increases with the scale and structural complexity of collaboration. Moreover, the benefit of additional collaboration is state-dependent: some reasoning states may already support a reliable solution, whereas others may benefit from further interaction or a different collaboration operator. Efficient MAR therefore requires adaptive policies that determine when additional collaboration is worthwhile and which collaboration operator should be invoked at the current state.

To address these limitations, recent studies have shifted from predefined collaboration workflows toward adaptive routing and orchestration. MasRouter (Yue et al., 2025) and DAAO (Su et al., 2026) dynamically configure collaboration modes, agent roles, model assignments, or workflow depth according to query characteristics. MaAS (Zhang et al., 2025a) and BAMAS (Yang et al., 2026a) further construct query-dependent agentic architectures under performance and budget constraints. Graph-based approaches adapt communication structures by learning interaction topologies, sparsifying edges, or controlling agent participation and information visibility (Zhuge et al., 2024; Zhang et al., 2025b; Yang et al., 2026b; Wang et al., 2026a; Lu et al., 2026). More recent methods make collaboration decisions sequentially during execution: AnyMAC (Wang et al., 2025b) iteratively selects the next agent and relevant prior context, iMAD (Fan et al., 2026) adaptively determines when additional debate is needed, Router-R1 (Zhang et al., 2025c) formulates model invocation and aggregation as multi-round routing, and meta-policy approaches learn high-level collaboration actions conditioned on evolving agent states (Yang & Thomason, 2026). Collectively, these methods demonstrate the potential of adaptive orchestration to improve reasoning efficiency under inference

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

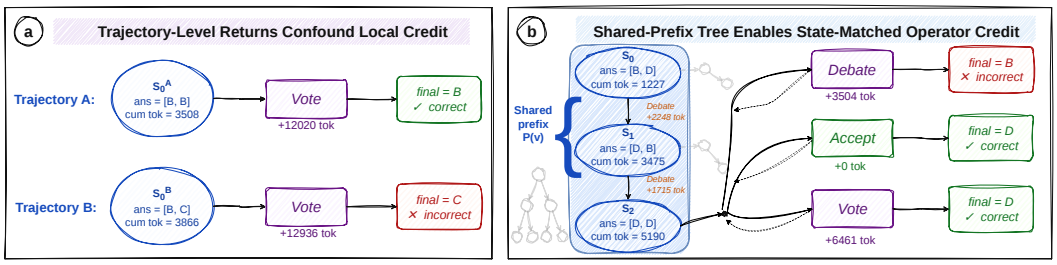


Figure 1: Motivation for state-matched operator credit. (a) The same operator yields different outcomes across states. (b) Different operators at the same state produce different correctness and costs.

constraints. This shift toward sequential, state-dependent routing raises a fundamental learning problem: *how should credit be assigned to an intermediate collaboration decision?*

This question is closely related to recent efforts on fine-grained credit assignment in collaborative LLM systems, including credit attribution to individual messages or agent contributions (Chen et al., 2026; Li et al., 2026c). Adaptive MAR routing, however, requires assigning credit to an intermediate collaboration operator under a particular multi-agent state. The difficulty is that the final outcome of a MAR trajectory is jointly determined by the collaboration state produced by previous decisions, the operator selected at the current state, and the continuation that follows. Consequently, query-level labels or trajectory-level returns provide only indirect supervision for individual routing decisions: a successful trajectory may contain unnecessary or harmful choices, while a failed trajectory may still contain locally beneficial ones. Moreover, operators observed under different execution histories are evaluated from different reached states, making their downstream outcomes difficult to compare directly. The key challenge is therefore to construct operator-specific supervision through *state-matched operator comparisons*, where alternative operators are evaluated from the same reached collaboration state while preserving the delayed consequences of the current decision.

Our key insight is to derive operator-level supervision through *state-matched downstream comparisons*, rather than directly attributing trajectory-level outcomes to preceding routing decisions. As illustrated in Figure 1, applying different operators to the same intermediate state can lead to substantially different downstream correctness and inference costs, indicating that the preferred operator depends on the reached collaboration state. MatchCredit therefore expands candidate operators from a shared execution prefix, holding the query, interaction history, and accumulated computation fixed, and evaluates each alternative through its complete downstream continuation. The resulting supervision is defined over *runtime collaboration operators*: each comparison asks which admissible operator is preferred at the same reached multi-agent state according to its downstream correctness and additional inference cost. Shared-prefix trees provide the matched states needed to construct these comparisons offline, while complete-suffix evaluation preserves the delayed consequences of each operator. These state-matched comparisons yield state-local operator preferences for training a router that performs single-trajectory adaptive inference.

Based on this insight, we propose **MatchCredit**, a state-matched credit assignment framework for adaptive multi-agent reasoning. MatchCredit constructs shared-prefix collaboration trees by expanding admissible operators as sibling alternatives from the same sampled intermediate state. Each state-operator pair is assigned a correctness-prioritized suffix credit based on the terminal correctness and cumulative additional cost of its evaluated downstream continuation. The resulting credits are converted into state-local pairwise preferences, prioritizing correct continuations and using lower cost to distinguish alternatives with equal correctness. A lightweight pairwise router is then trained from these preferences to select among admissible operators at inference time. The collaboration trees are used only for offline supervision construction, while test-time reasoning follows a single adaptive trajectory.

Our contributions are threefold. First, we formulate intermediate routing supervision in adaptive MAR as a **state-matched credit assignment** problem for collaboration operators, where alternative decisions are evaluated from the same reached multi-agent state to obtain comparable supervision for state-dependent routing. Second, we introduce **MatchCredit**, a state-matched credit assignment framework that constructs state-local operator preferences by comparing the downstream correctness

and additional cost of complete continuations from the same reached state. Third, experiments across diverse reasoning benchmarks show that MatchCredit reduces inference cost while preserving overall reasoning accuracy.

2 Related Work

Efficient and Adaptive Multi-Agent Collaboration. Existing methods improve multi-agent efficiency by adapting interaction depth, model allocation, agent participation, or communication structure. Free-MAD reduces redundant consensus seeking, while iMAD determines whether additional debate is warranted (Cui et al., 2026; Fan et al., 2026). At the query and workflow level, MasRouter, DAAO, MaAS, and MAS-Orchestra construct query-dependent model combinations, agentic architectures, or collaboration workflows, while BAMAS additionally considers computation budgets (Yue et al., 2025; Su et al., 2026; Zhang et al., 2025a; Ke et al., 2026; Yang et al., 2026a). Structure-oriented methods optimize communication graphs, edge activation, agent roles, or model mixtures (Zhang et al., 2025b; 2026; Yang et al., 2026b; Li et al., 2026b; Yao et al., 2026; Li et al., 2026a). During execution, DyTopo, Router-R1, Optimal-Agent-Selection (STRMAC), OI-MAS, and RUMAD adapt communication topology, agent or model selection, and debate behavior according to the evolving interaction state (Lu et al., 2026; Wang et al., 2026a; Zhang et al., 2025c; Wang et al., 2025a; 2026b). MatchCredit builds on this direction by treating heterogeneous collaboration mechanisms as callable operators and learning which operator to invoke at each reached state.

Credit Assignment for Multi-Agent Orchestration. Learning adaptive orchestration requires attributing downstream outcomes to intermediate decisions. Existing routing methods typically optimize trajectory-level objectives that combine task performance with model usage, budget, or communication cost (Yang et al., 2026a; Zhang et al., 2025c; Wang et al., 2026a), but such signals couple the effects of multiple decisions within a trajectory. C3 isolates *message-level* contributions by comparing alternative messages under matched interaction contexts (Chen et al., 2026). CCPO focuses on *agent-level* attribution in multi-agent reinforcement learning, estimating each agent's marginal contribution through counterfactual trajectories and using the resulting agent-specific advantages for policy optimization (Li et al., 2026c). MatchCredit instead targets *runtime collaboration operators*, comparing alternative operators from the same reached multi-agent state and using their downstream correctness and additional inference cost to construct state-local routing preferences.

3 Formalization

Dynamic Multi-Agent Reasoning. Given a query q, multi-agent reasoning evolves over collaboration states $s_t = (q, H_t, B_t)$, where H_t is the observable interaction history and B_t is the remaining inference budget. Let $\mathbb{O} = \{o_1, \ldots, o_L\}$ denote a finite catalog of collaboration operators. At state s_t, the router selects an admissible operator o_t,

$$o_t \sim \pi_\theta(\cdot \mid s_t), \qquad o_t \in \mathbb{O}(s_t) \subseteq \mathbb{O}, \tag{1}$$

where $\mathbb{O}(s_t)$ is protocol-dependent. Executing o_t incurs cost $c(s_t, o_t) \geq 0$ and yields,

$$s_{t+1} \sim P(\cdot \mid s_t, o_t), \qquad B_{t+1} = B_t - c(s_t, o_t). \tag{2}$$

A complete trajectory τ, its final answer, and total inference cost are given by:

$$\tau = (s_0, o_0, \ldots, o_{T-1}, s_T), \qquad \hat{y}_\tau = f(s_T), \qquad C(\tau) = \sum_{t=0}^{T-1} c(s_t, o_t), \tag{3}$$

where s_T is terminal. For the studied protocol, we instantiate the operator catalog as

$$\mathbb{O} = \{o_{\text{acc}}, o_{\text{deb}}, o_{\text{vote}}\}, \tag{4}$$

representing three basic collaboration functions: *termination*, which accepts the current answer; *interaction*, which performs another debate round to obtain additional reasoning evidence; and *aggregation*, which combines candidate answers through voting. This instantiation provides distinct choices between stopping, acquiring additional collaborative information.

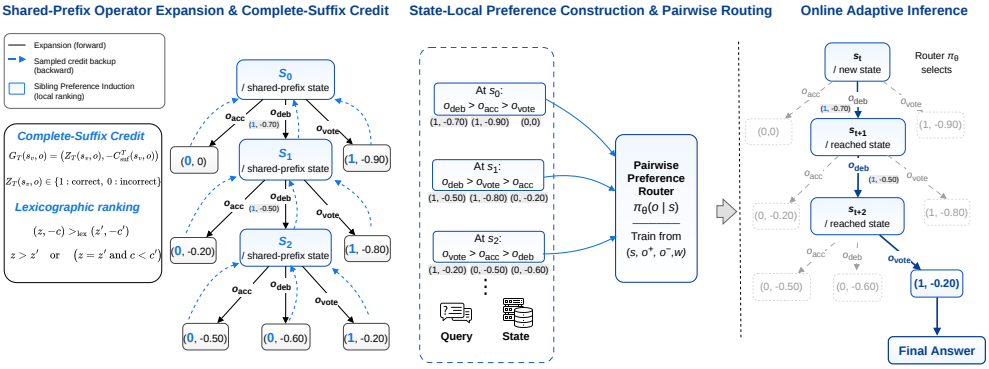


Figure 2: Overview of MatchCredit. Shared-prefix collaboration trees are expanded offline to assign correctness-prioritized suffix credits and extract state-local operator preferences. These preferences train a pairwise state router that follows a single adaptive trajectory during inference.

System Objective and Credit Assignment. Let $\mathcal{D} = \{(q_i, y_i)\}_{i=1}^{M}$ denote a labeled benchmark and $U(\hat{y}, y) = \mathbb{I}[\hat{y} = y] \in \{0, 1\}$ the correctness indicator. Given a reference policy π_{ref}, we define its accuracy as

$$A_{\text{ref}} = \mathbb{E}_{(q,y)\sim\mathcal{D},\, \tau\sim\pi_{\text{ref}}} \left[U(\hat{y}_\tau, y)\right]. \tag{5}$$

Adaptive multi-agent reasoning seeks to reduce inference cost while preserving accuracy within an allowed tolerance $\epsilon \geq 0$:

$$\begin{aligned} \min_{\theta} \quad & \mathbb{E}_{(q,y)\sim\mathcal{D},\, \tau\sim\pi_\theta}[C(\tau)] \\ \text{s.t.} \quad & \mathbb{E}_{(q,y)\sim\mathcal{D},\, \tau\sim\pi_\theta}[U(\hat{y}_\tau, y)] \geq A_{\text{ref}} - \epsilon. \end{aligned} \tag{6}$$

This constrained formulation specifies the system-level accuracy–cost objective; the router is trained using the state-local supervision introduced in Section 4. A terminal outcome evaluates the complete trajectory but does not isolate the contribution of an individual state–operator decision, since it also reflects the reached collaboration state and subsequent decisions. MatchCredit addresses this credit-assignment problem by comparing alternative operators from the same reached state and evaluating their complete downstream continuations.

4 MatchCredit: State-Matched Credit Assignment

As shown in Figure 2, MatchCredit consists of four modules: *Shared-Prefix Operator Expansion*, *Correctness-Prioritized Suffix Credit*, *State-Local Preference Construction*, and *Pairwise Preference Routing*. Its key principle is *state-matched complete-suffix comparison*: alternative operators are expanded from the same reached state and evaluated through their represented downstream continuations. The trees are used only for offline supervision; inference follows a single adaptive trajectory.

Shared-Prefix Operator Expansion. For each labeled example (q_i, y_i), MatchCredit constructs K independently sampled collaboration trees $\mathcal{T}_i^{(k)}$,

$$\mathcal{T}_i^{(k)} = (\mathcal{V}_i^{(k)}, \mathcal{E}_i^{(k)}), \qquad k = 1, \ldots, K, \tag{7}$$

where node v represents state $s_v = (q_i, H_v, B_v)$. At each non-terminal node, every admissible operator is expanded from the same realized state:

$$(s_v, o) \xrightarrow{\text{Exec}} \begin{cases} \text{terminal state}, & o \in \{o_{\text{acc}}, o_{\text{vote}}\}, \\ \text{child state } s_u, & o = o_{\text{deb}}. \end{cases} \tag{8}$$

Defining $r(v)$ as the number of debate rounds to reach v. Debate is allowed only when $r(v) < R_{\max}$; at $R_{\max}$ it is removed from the admissible set, so every branch terminates through Accept or Vote.

Let $P(v)$ be the trajectory prefix ending at s_v. A branch initiated by operator o is $\tau(v, o)$,

$$\tau(v, o) = P(v) \oplus o \oplus S(v, o), \tag{9}$$

where $S(v, o)$ is the remaining state–operator sequence to termination. Sibling branches therefore share the same query, interaction history, remaining budget, and accumulated computation. For additive inference cost,

$$\begin{aligned}&[C_{\mathrm{pre}}(v) + C_{\mathrm{suf}}(s_v, o_a)] - [C_{\mathrm{pre}}(v) + C_{\mathrm{suf}}(s_v, o_b)]\\ &\quad = C_{\mathrm{suf}}(s_v, o_a) - C_{\mathrm{suf}}(s_v, o_b).\end{aligned} \tag{10}$$

Thus, the shared prefix provides the state-matched comparison, while the suffix retains the downstream consequence of the selected operator.

Correctness-Prioritized Suffix Credit. For each expanded state–operator pair in sampled tree $\mathcal{T}$, define

$$G_{\mathcal{T}}(s_v, o) = \left(Z_{\mathcal{T}}(s_v, o), -C^{\mathcal{T}}_{\mathrm{suf}}(s_v, o)\right), \tag{11}$$

where $Z_{\mathcal{T}}(s_v, o) \in \{0, 1\}$ is terminal correctness, and $C^{\mathcal{T}}_{\mathrm{suf}}(s_v, o)$ is the total additional cost from s_v to termination, including the cost of o.

Credits are ordered lexicographically:

$$(z, -c) \succ_{\mathrm{lex}} (z', -c') \iff [z > z'] \vee [z = z' \wedge c < c']. \tag{12}$$

Thus, correctness is prioritized, while cost is used only to distinguish continuations with identical correctness.

For terminal operators,

$$G_{\mathcal{T}}(s_v, o) = \left(U(\hat{y}_{s_v,o}, y_i), -c(s_v, o)\right), \quad o \in \{o_{\mathrm{acc}}, o_{\mathrm{vote}}\}. \tag{13}$$

If o_{deb} reaches successor state s_u, define

$$G^{\star}_{\mathcal{T}}(s_u) = \operatorname*{lex\,max}_{o' \in \mathbb{O}(s_u)} G_{\mathcal{T}}(s_u, o') = \left(Z^{\star}_{\mathcal{T}}(s_u), -C^{\star}_{\mathcal{T}}(s_u)\right), \tag{14}$$

and recursively assign

$$G_{\mathcal{T}}(s_v, o_{\mathrm{deb}}) = \left(Z^{\star}_{\mathcal{T}}(s_u), -[c(s_v, o_{\mathrm{deb}}) + C^{\star}_{\mathcal{T}}(s_u)]\right). \tag{15}$$

Because Debate is inadmissible at $R_{\max}$, the recursion terminates. The backup in Eq. (15) is intentionally optimistic: it captures the *sampled continuation potential*, i.e., the best downstream outcome represented in the tree, rather than the expected value under the current router. This is particularly suitable for non-terminal operators such as Debate, whose utility lies in opening promising future paths that the router is trained to recover through state-local preferences at subsequent states.

State-Local Preference Construction. Each sampled tree induces within-state operator preferences:

$$o_a \succ^{\mathcal{T}}_{s_v} o_b \iff G_{\mathcal{T}}(s_v, o_a) \succ_{\mathrm{lex}} G_{\mathcal{T}}(s_v, o_b). \tag{16}$$

Pooling all strictly ordered sibling comparisons yields

$$\mathcal{D}_{\mathrm{pref}} = \{(s_j, o^+_j, o^-_j, w_j)\}_{j=1}^{J}, \tag{17}$$

where $o^+_j \succ^{\mathcal{T}}_{s_j} o^-_j$. $w_j > 0$ is a weight that emphasizes correctness-changing comparisons and balances opposite preference directions within each unordered operator pair. Trees are evaluated independently for credit construction; only their resulting pairwise samples are pooled. Since supervision is state-local, credits need not be numerically comparable across different states.

Pairwise Preference Routing. The router encodes each state s_v as

$$\mathbf{x}_v = \phi(s_v) \in \mathbb{R}^d, \tag{18}$$

where $\phi(\cdot)$ combines a query representation with observable collaboration-state features, and predicts operator logits $u_\theta(s_v, o)$,

$$\mathbf{u}_\theta(s_v) = \mathrm{MLP}_\theta(\mathbf{x}_v) \in \mathbb{R}^{|\mathbb{O}|}, \qquad u_\theta(s_v, o) = [\mathbf{u}_\theta(s_v)]_o. \tag{19}$$

For pair (s_j, o_j^+, o_j^-, w_j), the difference $\Delta_{\theta,j}$ is defined as:

$$\Delta_{\theta,j} = u_\theta(s_j, o_j^+) - u_\theta(s_j, o_j^-). \tag{20}$$

The router is trained with the weighted Bradley–Terry loss $\mathcal{L}_{\text{pref}}(\theta)$,

$$\mathcal{L}_{\text{pref}}(\theta) = \frac{\sum_{j=1}^{J} w_j \log(1 + \exp(-\Delta_{\theta,j}))}{\sum_{j=1}^{J} w_j}. \tag{21}$$

This objective directly learns within-state operator rankings without requiring globally calibrated values across heterogeneous states.

At inference step t, MatchCredit greedily selects the operator o_t,

$$o_t = \arg\max_{o \in \mathbb{O}(s_t)} u_\theta(s_t, o), \tag{22}$$

where o_{deb} continues reasoning and o_{acc} or o_{vote} terminates it. Thus, offline tree expansion provides state-matched supervision, while online execution remains a single adaptive trajectory.

5 Experiments

Datasets: We evaluate MatchCredit on six benchmarks: MMLU (Hendrycks et al., 2021a) and GPQA-Diamond (Rein et al., 2024) for general and scientific knowledge, GSM8K (Cobbe et al., 2021) and MATH-500 (Hendrycks et al., 2021b) for mathematical reasoning, MedQA (Jin et al., 2021) for medical question answering, and LogiQA2.0 (Liu et al., 2023) for logical reasoning. The standard evaluation split and answer-extraction protocol are used for each benchmark.

Models and Collaboration Setting. Our core experiments use GPT-4.1-mini and DeepSeek-V3-0324 as a heterogeneous agent pair. Single-agent baselines evaluate each model independently, whereas all collaboration and routing methods use the same pair.

Baselines. For single-agent reasoning, we include Chain-of-Thought (CoT) (Wei et al., 2022), Self-Refine (Madaan et al., 2023), and five-sample Self-Consistency (Wang et al., 2023) for both backbones. For collaboration, we compare with fixed Multi-Agent Debate (MAD) (Du et al., 2024) and the rule-based progressive HCP-MAD Liu et al. (2026) workflow. We further include DAAO (Su et al., 2026) as a query-level router and BAMAS (Yang et al., 2026a) and AnyMAC (Wang et al., 2025b) as adaptive-routing baselines.

Evaluation Metrics. We report task accuracy and token consumption per query. Avg. Acc. is the unweighted mean of the six benchmark accuracies, while Avg. Tokens is the unweighted mean of the token averages.

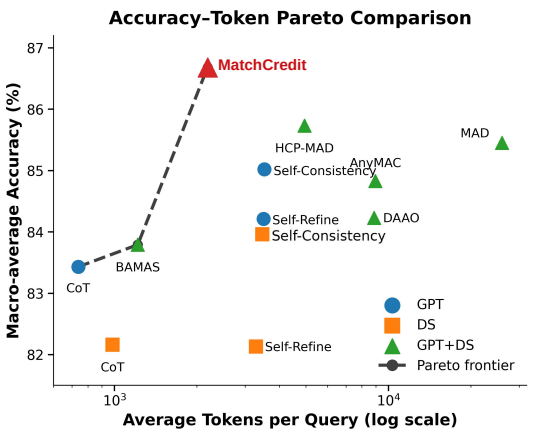
Implementation Details. For each benchmark, we randomly sample 100 labeled questions from the official split and divide them into 80 training and 20 validation questions; evaluation is performed on a disjoint test set. For each sampled question, we construct $K{=}2$ independently decoded collaboration trees, yielding 200 trees per benchmark. Shared-prefix expansion forks *Accept*/*Debate*/*Vote* continuations from intermediate states, producing approximately 10.3 operator trajectories per question on average, with at most $R_{\max}{=}5$ debate rounds. Preferences from the training subset are used to fit a benchmark-specific router, while the validation subset is used for model selection. The router is a two-layer MLP with hidden size 128 and a 398-dimensional input, consisting of a frozen 384-dimensional `all-MiniLM-L6-v2` query embedding and a 14-dimensional collaboration-state representation. We train each router with three independent random seeds and report mean $\pm$ sample standard deviation; these training seeds are distinct from the decoding seeds used for tree construction.

5.1 Main Results

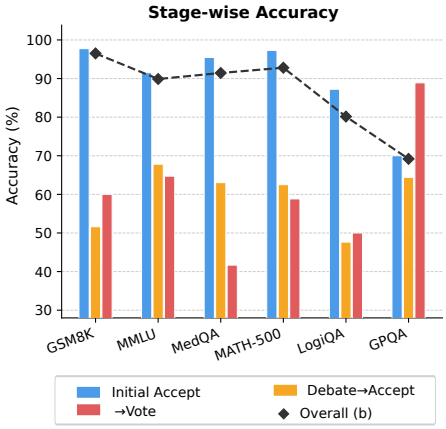
We compare MatchCredit against representative single-agent, multi-agent, and learned-routing methods across six reasoning benchmarks, with the main results summarized in Table 1. Among single-agent baselines, Chain-of-Thought requires the fewest tokens but achieves relatively lower accuracy. On GPT-4.1-mini, Self-Consistency increases the average accuracy from 83.43% to 85.02%, at the cost of raising token consumption from 738 to 3,529. In comparison, MatchCredit achieves a higher

Table 1: Comparison across six benchmarks. The best result is shown in **bold**, and the second-best is underlined. GPT and DS denote GPT-4.1-mini and DeepSeek-V3-0324, respectively; all collaboration and routing methods use the same GPT+DS heterogeneous pair. MatchCredit results are reported as mean ± sample standard deviation over three seeds. BAMAS* explicitly optimizes collaboration under a constrained inference budget and therefore represents a lower-compute operating point.

	Models / Methods	Accuracy (%)						Overall	
		MMLU	GPQA-D	GSM8K	MATH-500	MedQA	LogiQA2.0	Avg. Acc.	Avg. Tok.
GPT	CoT	87.04	64.65	95.60	87.17	90.02	76.08	83.43	738
GPT	Self-Refine	87.11	67.68	94.47	88.60	90.97	76.40	84.21	3,501
GPT	Self-Consistency	87.92	68.18	95.83	91.00	90.57	76.59	85.02	3,529
DS	CoT	86.50	61.62	96.13	88.55	85.39	74.75	82.16	984
DS	Self-Refine	86.70	60.61	95.52	86.00	88.14	75.83	82.13	3,284
DS	Self-Consistency	88.81	61.11	**96.89**	92.00	85.62	79.33	83.96	3,466
GPT+DS	MAD	88.09	65.61	96.59	91.60	91.12	79.70	85.45	25,961
GPT+DS	HCP-MAD	88.37	66.16	96.66	92.20	91.20	79.77	85.73	4,938
GPT+DS	DAAO	88.65	65.66	96.74	86.00	90.57	77.85	84.25	8,856
GPT+DS	BAMAS*	86.70	61.42	95.38	87.40	91.20	80.09	83.70	1,217
GPT+DS	AnyMAC	86.91	68.69	95.60	90.60	86.72	**80.47**	84.83	8,964
	MatchCredit (Ours)	**89.88** ±0.08	**69.19** ±0.51	96.51 ±0.16	**92.80** ±0.20	**91.44** ±0.13	80.15 ±0.23	**86.66** ±0.11	2,193 ±186



(a) Accuracy–cost comparison over six benchmarks.



(b) Stage-wise routing behavior of MatchCredit.

Figure 3: Accuracy–cost performance and routing behavior of MatchCredit.

average accuracy of 86.66 % while reducing token consumption to 2,193. These results suggest that state-adaptive collaboration provides a more favorable accuracy-cost trade-off than repeatedly sampling independent solutions from a single agent.

MatchCredit also compares favorably with conventional multi-agent reasoning methods. Fixed Multi-Agent Debate achieves an average accuracy of 85.45% with 25,961 tokens, while HCP-MAD substantially reduces the cost to 4,938 tokens and obtains 85.73% accuracy. Relative to HCP-MAD, MatchCredit further improves average accuracy by 0.93 percentage points while reducing token consumption by 55.59%. These results indicate that state-conditioned routing preserves the reasoning benefits of multi-agent collaboration while avoiding unnecessary debate and aggregation.

Among learned routing approaches, DAAO and AnyMAC obtain average accuracies of 84.25% and 84.83% with 8,856 and 8,964 tokens, respectively, whereas MatchCredit reaches 86.66 % with only 2,193 tokens. BAMAS achieves the lowest inference cost at 1,217 tokens, but its average accuracy is 83.70%, 2.96 percentage points below MatchCredit. Its low cost is associated with a conservative collaboration policy that uses approximately one to two model calls per query, which limits additional reasoning on difficult instances.

Accuracy–Cost Comparison. Figure 3a compares average accuracy and token consumption, with better operating points toward the upper-left. MatchCredit lies on the empirical Pareto frontier

and achieves the highest macro-average accuracy among the evaluated methods. Under the tested configurations, it strictly dominates Multi-Agent Debate, HCP-MAD, DAAO, AnyMAC, and Self-Consistency, which are both less accurate and more costly. Chain-of-Thought and BAMAS require fewer tokens but achieve lower accuracy; for example, BAMAS saves 976 tokens per query while trailing MatchCredit by 2.96 percentage points. Overall, MatchCredit provides a favorable empirical accuracy–cost operating point, combining high reasoning accuracy with substantially lower inference cost than most multi-agent and adaptive-routing baselines.

Progressive Routing Behavior. Figure 3b shows that MatchCredit terminates most queries via initial Accept at round 0, including 97.27% of GSM8K, 93.21% of MMLU, 88.22% of MedQA, and 88.20% of MATH-500 instances. These early-accepted subsets are highly reliable, with conditional accuracies of 97.74%, 91.52%, 95.46%, and 97.28%, respectively. More difficult benchmarks require broader collaboration: GPQA-D routes 29.79% of queries through Debate and 4.55% to Vote, while LogiQA2.0 routes 17.49% through Debate; MATH-500 invokes Vote for 10.20% of its queries. Because later decisions apply only to instances unresolved at the initial round, their conditional accuracies characterize harder router-selected subsets rather than the intrinsic quality of Debate or Vote. Overall, MatchCredit achieves 69.19%–96.51% accuracy while terminating reliable cases early and reserving additional collaboration for harder queries.

Analysis of Correctness Transitions. Table 2 compares the initial reference prediction with the final MatchCredit prediction for every query. When the two heterogeneous agents agree at the initial round, their shared answer is used as the initial reference; otherwise, Agent A's answer is used. Across the six datasets, MatchCredit corrects 1.44%–8.08% of all queries that are initially incorrect, while changing only 0.30%–2.36% from correct to incorrect. The correction rate is higher on challenging datasets such as GPQA-D and LogiQA2.0, whereas smaller rates on high-accuracy datasets mainly reflect fewer initial errors. Since $\times \to \checkmark$ exceeds $\checkmark \to \times$ on every benchmark, MatchCredit yields a positive net accuracy gain across all six datasets.

Table 2: Correctness transitions of MatchCredit.

Dataset	$\times \to \checkmark$ (%)	$\checkmark \to \times$ (%)	$\checkmark \to \checkmark$ (%)	$\times \to \times$ (%)
GPQA-D	8.08	1.01	61.11	29.80
LogiQA2.0	6.11	1.46	74.05	18.38
MedQA	5.03	2.36	86.41	6.21
MMLU	4.38	1.18	85.50	8.94
MATH-500	3.60	1.00	89.20	6.20
GSM8K	1.44	0.30	95.07	3.18

5.2 Ablation Study

We conduct two complementary ablation analyses to disentangle where the gains of MatchCredit originate. First, Figure 4 compares alternative supervision strategies under the same model pair, operator space, and offline data budget, varying only the construction of router supervision. Second, Table 3 keeps the state-matched supervision fixed while removing individual design components, allowing us to isolate the contributions of correctness-prioritized suffix credit and state-conditioned routing.

Outcome-Supervised Trajectory. This variant assigns positive or negative labels to all actions according to the final trajectory outcome, without comparing operators from the same state. Compared with MatchCredit, it lowers GPQA-D and LogiQA2.0 accuracy by 1.15 and 0.89 points while increasing token usage by 258 and 330. Because successful trajectories may contain redundant actions and failed ones also may include useful intermediate decisions, terminal labels provide unreliable local credit. MatchCredit instead compares sibling operators under a shared prefix to obtain state-matched supervision.

Trajectory-Level Preference. Trajectory-Level Preference ranks complete trajectories by final correctness and total cost, but still compares entire action sequences rather than operators at a shared state. Relative to MatchCredit, it improves GPQA-D accuracy by 0.49 points while using 324 more tokens; on LogiQA2.0, it lowers accuracy by 0.63 points and adds 614 tokens. Because trajectories may differ at multiple routing steps, their overall ranking cannot identify which decision caused the gain or loss. It may also favor a long correct trajectory even when earlier acceptance would achieve the same correctness at lower cost. MatchCredit avoids this confounding by comparing candidate operators after an identical shared prefix.

Terminal-Reward Preference. Terminal-Reward Preference ranks complete trajectories using a scalar reward that combines final correctness and token cost with coefficient λ, testing whether

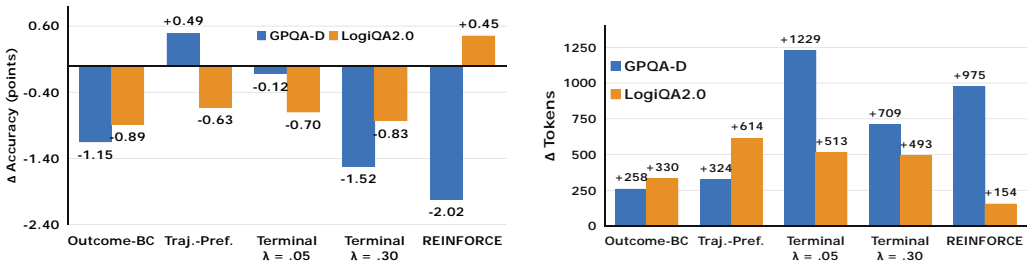


Figure 4: Supervision ablations relative to MatchCredit on GPQA-D and LogiQA2.0. Each bar reports Variant vs. MatchCredit. Positive ΔAccuracy indicates higher accuracy, whereas positive ΔTokens indicates higher inference cost.

scalarization can replace MatchCredit's correctness-prioritized suffix credit. With $\lambda = 0.05$, accuracy on GPQA-D and LogiQA2.0 decreases by 0.12 and 0.70 points, while token usage increases by 1,229 and 513. Raising λ to 0.30 saves 520 and 20 tokens, yet causes additional accuracy drops of 1.40 and 0.13 points. It therefore remains both less accurate and more expensive than MatchCredit. These results show that scalar rewards are sensitive to λ: small values under-penalize cost, while larger values sacrifice correctness before matching MatchCredit's efficiency. Moreover, tuning λ changes only the global accuracy–cost trade-off and does not resolve intermediate credit assignment.

REINFORCE Router. To isolate the effect of credit assignment from routing architecture, we construct a controlled sequential-routing baseline based on REINFORCE. It uses the same router architecture, state representation, operator space, and depth constraint as MatchCredit, while replacing state-matched supervision with delayed trajectory-level rewards. Compared with MatchCredit, REINFORCE reduces GPQA-D accuracy by 2.02 points while consuming 975 more tokens; on LogiQA2.0, it improves accuracy by 0.45 points but requires 154 additional tokens. These results suggest that trajectory-level returns provide less effective supervision for intermediate routing decisions, as all actions in a trajectory receive the same delayed signal regardless of their local contribution. In contrast, MatchCredit directly compares sibling operators at the same state, providing more targeted supervision for selecting *Accept*, *Debate*, or *Vote*.

Design Components. Table 3 further isolates the contributions of two key components within the shared-prefix framework. Removing correctness-prioritized credit produces the largest performance drop: accuracy on GPQA-D and LogiQA2.0 decreases by 4.24 and 1.52 points, respectively, while token consumption increases by 466 and 237. Without the correctness-first ordering, lower-cost actions may be preferred even when they reduce the likelihood of reaching a correct answer. Removing state-conditioned routing also lowers accuracy by 1.15 and 1.08 points and increases token usage by 1,261 and 354. Because this variant cannot adapt to evolving agreement, confidence, and interaction history, it often continues reasoning in states that could already be terminated. Together, these results show that correctness-prioritized credit preserves answer quality, while state-conditioned routing enables efficient, instance-specific allocation of collaboration.

Table 3: MatchCredit component ablation.

Variant	GPQA-D		LogiQA2.0	
	Acc.	Tok.	Acc.	Tok.
MatchCredit	**69.19**	**4,107**	**80.15**	**1,812**
w/o Correctness Priority	64.95	4,573	78.63	2,049
w/o State Conditioning	68.04	5,368	79.07	2,166

6 Conclusion

We propose MatchCredit, a shared-prefix credit assignment framework that derives correctness-prioritized, state-local operator preferences to train a lightweight router for adaptive multi-agent reasoning. Across six benchmarks, MatchCredit achieves a favorable accuracy–cost trade-off by improving reasoning performance while substantially reducing token usage. Current limitations include its restricted Accept–Debate–Vote operator set and reliance on benchmark-specific labeled data; future work will explore richer operators, cross-domain routing, and uncertainty-aware credit estimation.

REFERENCES

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=EHg5GDnyq1.

Yanjun Chen, Yirong Sun, Hanlin Wang, Xinming Zhang, Xiaoyu Shen, Wenjie Li, and Wei Zhang. Contextual counterfactual credit assignment for multi-agent reinforcement learning in LLM collaboration. *CoRR*, abs/2603.06859, 2026. doi: 10.48550/ARXIV.2603.06859. URL https://doi.org/10.48550/arXiv.2603.06859.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

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, San Diego, California, United States, July 2-7, 2026*, pp. 31977–31997. Association for Computational Linguistics, 2026. URL https://aclanthology.org/2026.findings-acl.1600/.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11733–11763. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/du24e.html.

Wei Fan, JinYi Yoon, and Bo Ji. imad: Intelligent multi-agent debate for efficient and accurate LLM inference. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 29403–29411. AAAI Press, 2026. doi: 10.1609/AAAI.V40I35.40181. URL https://doi.org/10.1609/aaai.v40i35.40181.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021a. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

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 Joaquin Vanschoren and Sai-Kit Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021b. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. Metagpt: Meta programming for A multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=VtmBAGCN7o.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021. doi: 10.3390/app11146421.

Zixuan Ke, Yifei Ming, Austin Xu, Ryan Chin, Xuan-Phi Nguyen, Prathyusha Jwalapuram, Jiayu Wang, Semih Yavuz, Caiming Xiong, and Shafiq Joty. Mas-orchestra: Understanding and improving multi-agent reasoning through holistic orchestration and controlled benchmarks, 2026. URL https://arxiv.org/abs/2601.14652.

Haoran Li, Shulun Chen, Shaoyuan Sun, and Hanchen Wang. Multi-agent coordination adaptation via structure-guided orchestration, 2026a. URL https://arxiv.org/abs/2605.25746.

Shijun Li, Hilaf Hasson, and Joydeep Ghosh. OMAC: A holistic optimization framework for LLM-based multi-agent collaboration. In *Forty-third International Conference on Machine Learning*, 2026b. URL https://openreview.net/forum?id=6C4YQcq8YX.

Zhongyi Li, Wan Tian, Yikun Ban, Jinju Chen, Huiming Zhang, Yang Liu, and Fuzhen Zhuang. Counterfactual credit policy optimization for multi-agent collaboration. *CoRR*, abs/2603.21563, 2026c. doi: 10.48550/ARXIV.2603.21563. URL https://doi.org/10.48550/arXiv.2603.21563.

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.

Yiqing Liu, Hantao Yao, Wu Liu, Allen He, and Yongdong Zhang. Hcp-mad:heterogeneous consensus-progressive reasoning for efficient multi-agent debate, 2026. URL https://arxiv.org/abs/2604.09679.

Yuxing Lu, Yucheng Hu, Xukai Zhao, and Jiuxin Cao. Dytopo: Dynamic topology routing for multi-agent reasoning via semantic matching, 2026. URL https://arxiv.org/abs/2602.06039.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/91edff07232fb1b55a505a9e9f6c0ff3-Abstract-Conference.html.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. Chatdev: Communicative agents for software development. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 15174–15186. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.810. URL https://doi.org/10.18653/v1/2024.acl-long.810.

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.

Jinwei Su, Qizhen Lan, Yinghui Xia, Lifan Sun, Weiyou Tian, Tianyu Shi, and Lewei He. Difficulty-aware agentic orchestration for query-specific multi-agent workflows. In Hakim Hacid, Yoelle Maarek, Francesco Bonchi, Ido Guy, and Emine Yilmaz (eds.), *Proceedings of the ACM Web Conference 2026, WWW 2026, Dubai, United Arab Emirates, originally scheduled for April 13-17, 2026, rescheduled for June 29 - July 3, 2026*, pp. 2060–2070. ACM, 2026. doi: 10.1145/3774904.3792240. URL https://doi.org/10.1145/3774904.3792240.

Chao Wang, Han Lin, Huaze Tang, Huijing Lin, and Wenbo Ding. RUMAD: reinforcement-unifying multi-agent debate. In Viviana Mascardi, John Thangarajah, Chris Amato, and Louise A. Dennis (eds.), *Proceedings of the 25th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2026, Paphos, Cyprus, May 25-29, 2026*, pp. 3695–3703. ACM, 2026a. doi: 10.65109/CBJO8409. URL https://dl.acm.org/doi/10.65109/CBJO8409.

Jingbo Wang, Sendong Zhao, Haochun Wang, Yuzheng Fan, Lizhe Zhang, Yan Liu, and Ting Liu. Optimal-agent-selection: State-aware routing framework for efficient multi-agent collaboration, 2025a. URL https://arxiv.org/abs/2511.02200.

Jingbo Wang, Sendong Zhao, Jiatong Liu, Haochun Wang, Wanting Li, Bing Qin, and Ting Liu. Orchestrating intelligence: Confidence-aware routing for efficient multi-agent collaboration across multi-scale models, 2026b. URL https://arxiv.org/abs/2601.04861.

Song Wang, Zhen Tan, Zihan Chen, Shuang Zhou, Tianlong Chen, and Jundong Li. Anymac: Cascading flexible multi-agent collaboration via next-agent prediction. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025b.

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, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Liming Yang, Junyu Luo, Xuanzhe Liu, Yiling Lou, and Zhenpeng Chen. BAMAS: structuring budget-aware multi-agent systems. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 29802–29810. AAAI Press, 2026a. doi: 10.1609/AAAI.V40I35.40226. URL https://doi.org/10.1609/aaai.v40i35.40226.

Peiting Yang, Jiahao Shi, Caiyi Xu, Ming Liu, Yanxia Wu, and Rongsheng Li. Agenttailor: A semantic-aware llm-based multi-agent system with actor-critic structure. In *Forty-third International Conference on Machine Learning*, 2026b.

Wei Yang and Jesse Thomason. Learning to deliberate: Meta-policy collaboration for agentic llms with multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

Tianjun Yao, Zhaoyi Li, and Zhiqiang Shen. Hieramas: Optimizing intra-node llm mixtures and inter-node topology for multi-agent systems, 2026. URL https://arxiv.org/abs/2602.20229.

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 Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 15549–15572. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.ACL-LONG.757. URL https://doi.org/10.18653/v1/2025.acl-long.757.

Guibin Zhang, Luyang Niu, Junfeng Fang, Kun Wang, Lei Bai, and Xiang Wang. Multi-agent architecture search via agentic supernet. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*, pp. 75834–75852. PMLR / OpenReview.net, 2025a. URL https://proceedings.mlr.press/v267/zhang25bi.html.

Guibin Zhang, Yanwei Yue, Xiangguo Sun, Guancheng Wan, Miao Yu, Junfeng Fang, Kun Wang, Tianlong Chen, and Dawei Cheng. G-designer: Architecting multi-agent communication topologies via graph neural networks. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien,

Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025b. URL https://proceedings.mlr.press/v267/zhang25cu.html.

Haozhen Zhang, Tao Feng, and Jiaxuan You. Router-r1: Teaching llms multi-round routing and aggregation via reinforcement learning. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025c. URL http://papers.nips.cc/paper_files/paper/2025/hash/ceaa137fce916aba5c65fceb1309088b-Abstract-Conference.html.

Zhen Zhang, Wanjing Zhou, Juncheng Li, Hao Fei, Jun Wen, and Wei Ji. Radar: Redundancy-aware diffusion for multi-agent communication structure generation, 2026. URL https://arxiv.org/abs/2605.09907.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. Gptswarm: Language agents as optimizable graphs. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62743–62767. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/zhuge24a.html.

LIMITATIONS

Despite the effectiveness of MatchCredit, our work has three key limitations: (1) The evaluation is mainly conducted on reasoning and question-answering benchmarks; more diverse interactive and open-ended tasks should be explored to further verify its generalization. (2) MatchCredit requires offline shared-prefix trees to estimate the state-conditional utility of candidate operators. Although this training process does not increase test-time inference cost, constructing these trees requires additional LLM calls and computation. (3) The current implementation considers a fixed operator set and agent pool. Its performance may depend on the available operators, backbone models, and training distribution, and extending MatchCredit to larger or dynamically changing collaboration spaces remains an important direction.

ETHICAL STATEMENT

MatchCredit learns routing preferences from offline collaboration trajectories and executes only the selected reasoning path during inference. Its shared-prefix credit assignment does not directly alter the knowledge or safety alignment of the underlying language models. Nevertheless, biases, factual errors, and harmful content present in the training trajectories may be transferred to the learned router, while efficiency-oriented routing may repeatedly favor an unreliable operator in underrepresented states.

AI USE STATEMENT

Large language models (LLMs) were used as general-purpose assistance tools in this work, primarily for polishing the manuscript to improve clarity, readability, and presentation.

A ADDITIONAL METHOD DETAILS

This section specifies the observable state representation, preference weighting, and operator execution used in MatchCredit.

A.1 STATE REPRESENTATION

At decision step t, let $\mathbf{a}_t = (a_t^A, a_t^B)$ denote the extracted answers of the two debate agents and $\mathbf{r}_t = (r_t^A, r_t^B)$ their complete responses. MatchCredit maintains the cumulative histories

$$\mathcal{H}_t^a = (\mathbf{a}_0, \ldots, \mathbf{a}_t), \qquad \mathcal{H}_t^r = (\mathbf{r}_0, \ldots, \mathbf{r}_t). \tag{23}$$

Let T_t^{cum} denote the cumulative tokens consumed by the initial responses and preceding debate rounds, T_t^{step} the tokens consumed by the latest generation step, and L_q the number of characters in query q.

The router input is

$$\mathbf{x}_t = [\mathbf{e}_q; \boldsymbol{\phi}(s_t)] \in \mathbb{R}^{398}, \tag{24}$$

where $\mathbf{e}_q \in \mathbb{R}^{384}$ is a frozen query embedding and $\boldsymbol{\phi}(s_t) \in \mathbb{R}^{14}$ contains observable collaboration-state features. All features are available at inference time and do not use the ground-truth answer.

For the lexical features, each complete response is lowercased and converted into a set of tokens containing letters, digits, or underscores. Punctuation is ignored.Let $\mathcal{T}(r)$ denote the resulting token set for response r. The lexical similarity between two responses is computed as

$$J(r, r') = \frac{|\mathcal{T}(r) \cap \mathcal{T}(r')|}{|\mathcal{T}(r) \cup \mathcal{T}(r')|}. \tag{25}$$

We set $J(r, r') = 0$ when both token sets are empty.

Table 4: Router input representation. The complete input is $\mathbf{x}_t = [\mathbf{e}_q; \boldsymbol{\phi}(s_t)] \in \mathbb{R}^{398}$, consisting of 14 observable collaboration-state features and a frozen 384-dimensional query embedding.

Idx.	Observable feature	Definition and domain
1	`round_fraction` $t/R_{\max}$	Current debate round divided by $R_{\max}$. $[0, 1]$
2	`answer_a_valid` v_t^A	Whether Agent A produces a parsable answer. $\{0, 1\}$
3	`answer_b_valid` v_t^B	Whether Agent B produces a parsable answer. $\{0, 1\}$
4	`consensus` c_t	Whether both current answers are valid and identical. $\{0, 1\}$
5	`previous_consensus` c_{t-1}	Whether both agents agreed in the preceding round; 0 at $t = 0$. $\{0, 1\}$
6	`answer_a_changed` δ_t^A	Whether Agent A changes its answer from the preceding round; 0 at $t = 0$. $\{0, 1\}$
7	`answer_b_changed` δ_t^B	Whether Agent B changes its answer from the preceding round; 0 at $t = 0$. $\{0, 1\}$
8	`pair_repeated` p_t	Whether the ordered answer pair is unchanged from the preceding round; 0 at $t = 0$. $\{0, 1\}$
9	`disagreement_streak` ρ_{dis}	$\min\{d_t/(R_{\max}+1), 1\}$, where d_t counts trailing states in which the agents disagree. $[0, 1]$
10	`cross_reasoning_jaccard` J_{cross}	$J(r_t^A, r_t^B)$, the lexical overlap between the two current responses. $[0, 1]$
11	`reasoning_novelty` ν_{reason}	$1 - \frac{1}{2}[J(r_t^A, r_{t-1}^A) + J(r_t^B, r_{t-1}^B)]$; 0 at $t = 0$. $[0, 1]$
12	`cumulative_token_scale` $\widetilde{T}_{\text{cum}}$	$\min\{\log(1 + T_t^{\text{cum}})/\log(1 + 5000), 2\}$. $[0, 2]$
13	`last_step_token_scale` $\widetilde{T}_{\text{step}}$	$\min\{\log(1 + T_t^{\text{step}})/\log(1 + 5000), 2\}$. $[0, 2]$
14	`question_length_scale` $\widetilde{L}_q$	$\min\{L_q/2000, 2\}$, where L_q is the number of characters in the complete query. $[0, 2]$
–	`query_embedding` $\mathbf{e}_q$	L2-normalized 384-dimensional embedding from the frozen `all-MiniLM-L6-v2` encoder; $\|\mathbf{e}_q\|_2 = 1$.

A.2 Preference Weighting

Following Eq. (16) of the main paper, MatchCredit retains every strictly ordered pair of admissible sibling operators:

$$G(s_j, o_j^+) \succ_{\text{lex}} G(s_j, o_j^-).$$

Each pair is first weighted according to whether it distinguishes correct and incorrect continuations:

$$b_j = \begin{cases} 2, & z_j^+ \neq z_j^-, \\ 1, & z_j^+ = z_j^-, \end{cases} \tag{26}$$

where z_j^+ and z_j^- are the correctness components of the corresponding suffix credits. Thus, correctness-changing comparisons receive twice the weight of comparisons distinguished only by inference cost.

To reduce directional imbalance, let $n(a, b)$ denote the number of training pairs that prefer operator a over operator b. For pair j, we multiply its base weight by

$$\min\left\{\frac{n(o_j^+, o_j^-) + n(o_j^-, o_j^+)}{2n(o_j^+, o_j^-)}, 3\right\}. \tag{27}$$

This approximately equalizes the contributions of opposite preference directions within each unordered operator pair, while limiting the amplification factor to 3.

The implementation additionally balances the total contribution of different training questions. Let S_q be the sum of the correctness- and direction-weighted pairs generated from question q, and let $\overline{S}$ be the average of S_q over questions that generate at least one preference pair. The final weight is

$$w_j = b_j \min\left\{\frac{n(o_j^+, o_j^-) + n(o_j^-, o_j^+)}{2n(o_j^+, o_j^-)}, 3\right\} \text{clip}\left(\frac{\overline{S}}{S_{q(j)}}, \frac{1}{3}, 3\right), \tag{28}$$

Table 5: Task-specific data used by MatchCredit.

Dataset	Training pool	Test size
MMLU	Official `auxiliary_train`	1,474
GPQA-D	GPQA excluding the Diamond subset	198
GSM8K	Official training split	1,319
MATH-500	MATH excluding the MATH-500 items	500
MedQA	Official training split	1,273
LogiQA2.0	Official training split	1,572

where $q(j)$ denotes the question associated with pair j. The weighted Bradley–Terry loss is normalized by $\sum_j w_j$, as defined in Eq. (21) of the main paper.

A.3 Collaboration Operators

At round $t = 0$, two heterogeneous agents independently produce the initial responses and candidate answers. When DEBATE is selected, both agents are invoked in parallel and receive the query, the two complete responses from the immediately preceding round, and the compact answer trajectories of both agents; earlier full responses are not repeated. The newly generated responses and extracted answers are appended to the interaction history and become the current state. ACCEPT is admissible when the two current answers are valid and identical. VOTE is admissible at every state and invokes five new agents, each of which independently answers the original query without access to the debate history or current candidate answers. Their valid answers are aggregated by equal-weight simple majority vote; ties are resolved deterministically by selecting the lexicographically smallest answer.

B Experimental Details

B.1 Data Splits and Training Budget

All training examples are drawn from auxiliary pools disjoint from the reported test sets. For GPQA-D, the pool contains GPQA examples outside the Diamond subset; for MATH-500, it contains MATH examples excluding the MATH-500 evaluation items. We use the official training splits for the remaining datasets. MatchCredit uses 80 unique questions for training and 20 disjoint questions for validation on each benchmark, and then constructs $K = 2$ independently decoded trees for each question. Trainable baselines use 200 unique questions sampled from the same task-specific pool, whereas inference-only methods use no training data. Training, validation, and test question IDs are separated.

Table 7 reports the offline supervision cost and test-time efficiency of the routing methods. Training tokens and cost include all LLM calls used to construct the training supervision, while test accuracy, tokens, and wall time are macro-averaged over the six benchmarks. Router latency includes query encoding and router inference but excludes LLM calls. MatchCredit uses 100 unique training questions per benchmark, for 600 questions in total, and constructs $K{=}2$ independently decoded shared-prefix trees for each question, yielding 1,200 trees overall. Shared prefixes are generated once and reused across sibling continuations, rather than being repeatedly counted for each root-to-leaf trajectory. A dash indicates that the corresponding quantity is not applicable.

B.2 Implementation Details

As shown in Table 6, the main model pair is GPT-4.1-mini and DeepSeek-V3-0324. The frozen `all-MiniLM-L6-v2` encoder applies its default mean pooling followed by L2 normalization. The trainable router contains two 128-dimensional ReLU hidden layers, no dropout, and a three-logit output head.

After each epoch, we evaluate the weighted Bradley–Terry loss on the validation preference set and retain the checkpoint with the lowest validation loss. Training stops after 10 epochs without

Table 6: Router optimization settings.

Item	Setting
Router architecture	$398 \to 128 \to 128 \to 3$
Activation / dropout	ReLU / 0
Optimizer	AdamW
Learning rate / weight decay	3×10^{-4} / 0
Batch size / maximum epochs	256 / 80
Random seed / gradient clip	42 / 10.0

Table 7: Training and test-time efficiency across six datasets. Accuracy is macro-averaged.

Offline training

Method	Train/val Q.	Tokens	Cost ($)	Router architecture
DAAO	1,200	41.35M	32.11	VAE–MoE/FFN
BAMAS	1,200	**6.19M**	**4.71**	MLP
AnyMAC	1,200	52.38M	29.16	NAP/NLP (GPT-2)
MatchCredit	600 (1,200)	19.31M	17.35	MLP

Test-time inference

Method	Acc. (%)	Tokens	Wall time (s)	Router latency (ms) / params (M)
DAAO	84.25	8,856	38.38	0.72 / 0.388
BAMAS	83.70	**1,217**	**10.84**	3.09 / **0.035**
AnyMAC	84.83	8,964	49.23	11.94 / 117
MatchCredit	**86.66**	2,193	21.79	**0.15** / 0.068

improvement, and the best checkpoint is restored for evaluation. At inference time, the router directly selects the highest-scoring admissible operator.

B.3 Training and Inference Efficiency

Table 7 reports the offline supervision cost and test-time efficiency of the routing methods. For MatchCredit, each benchmark uses 100 unique questions from the official training split, comprising 80 training and 20 validation questions. We construct $K=2$ independently decoded shared-prefix trees for each question, giving 200 trees per benchmark and 1,200 trees across the six benchmarks. Offline tokens and cost include all LLM calls used to construct supervision for both the training and validation subsets. Shared prefixes are generated once and reused across sibling continuations, rather than being repeatedly counted for each root-to-leaf trajectory. Test accuracy, token consumption, and wall time are measured on disjoint test sets and macro-averaged over the six benchmarks. Router latency includes query encoding and MLP inference but excludes LLM calls.

C Additional Results

C.1 Backbone Generalization

As shown in Table 8, we further replace the main backbones with `meta-llama/llama-4-maverick` and `qwen3-235b-a22b`. Single-agent rows evaluate each backbone independently; all collaborative methods use the same Qwen–Llama pair.

Table 8: Results with the Qwen3-235B-A22B and Llama-4-Maverick heterogeneous pair. Qwen and Llama denote the two respective backbones. Avg. Acc. is the macro-average across the six benchmarks, and Avg. Tokens denotes the average token consumption per query.

Models	Methods	Accuracy (%)						Overall	
		MMLU	GPQA-D	GSM8K	MATH-500	MedQA	LogiQA2.0	Avg. Acc.	Avg. Tokens
Qwen	CoT	86.39	59.60	93.92	87.15	86.76	80.11	82.32	**794**
Llama	CoT	88.36	66.67	95.67	85.14	87.79	75.59	83.20	943
Qwen+Llama	BAMAS	85.54	61.11	96.29	88.20	88.21	80.90	83.38	823
	AnyMAC	88.28	66.15	94.77	90.20	82.80	79.76	83.66	8,011
	DAAO	89.15	65.66	96.21	92.20	**91.04**	78.82	85.51	2,965
	HCP-MAD	88.37	67.17	96.29	90.60	89.54	**82.35**	85.72	3,435
	MatchCredit	**89.65**	**68.18**	**96.36**	**92.40**	89.36	80.75	**86.12**	2,380

Table 9: Comparison between task-specific and One-for-All Router MatchCredit routers. Both settings are evaluated on identical sampled test IDs.

Training regime	Routers	MMLU	GPQA-D	GSM8K	MATH-500	MedQA	LogiQA2.0	Avg. Acc. / Tokens
Task-specific routers	6	89.67	69.19	98.50	92.50	92.00	79.50	86.89 / 2,257
One-for-All Router	1	88.33	67.68	98.00	91.50	90.50	80.00	86.00 / 2,897

Table 10: Single-source cross-dataset transfer of MatchCredit. Routers trained on the target dataset or MATH-500 are evaluated on a subject-stratified 400-question MMLU subset and the full 198-question GPQA-D test set.

Target	Training source	Acc.	Tokens
MMLU	MMLU	88.75	1,330
MMLU	MATH-500	89.00	1,858
GPQA-D	GPQA-D	69.19	4,107
GPQA-D	MATH-500	68.69	3,934
LogiQA2.0	LogiQA2.0	80.15	1,812
LogiQA2.0	MATH-500	80.27	2,572

C.2 One-for-All Router

To control evaluation cost, all live tests use fixed sampled subsets rather than full test sets.

To examine whether a single router can serve multiple tasks, we compare task-specific routers with a shared router trained on pooled data from all six datasets. To limit evaluation cost, both settings are evaluated on fixed test subsets sampled with seed 42. GPQA-D uses the full 198-question test set; MMLU uses a subject-stratified sample of 300 questions; and GSM8K, MATH-500, MedQA, and LogiQA2.0 each use 200 uniformly sampled questions. Both settings are evaluated on identical test IDs.

As shown in Table 9, the shared router retains comparable average accuracy, decreasing from 86.89% to 86.00%. Its higher token consumption, however, indicates that task-specific training provides more efficient routing decisions. This one-model solution avoids maintaining and selecting separate routers for different datasets, demonstrating the cross-task applicability of the learned routing policy, although it incurs moderately higher inference cost.

C.3 Single-Source Cross-Dataset Transfer

We further evaluate whether a router learned on one dataset can be directly applied to unseen tasks. A router trained only on cached MATH-500 trees is evaluated on MMLU and GPQA-D without target-domain training labels or additional tree generation.

As shown in Table 10, the MATH-500 router matches the target-trained routers on both targets: it slightly improves MMLU accuracy and remains within 0.5 points on GPQA-D. These results provide preliminary evidence that the learned routing policy captures transferable collaboration patterns rather than being fully dataset-specific.

C.4 Sensitivity to Tree Sampling and Debate Rounds

Table 11 examines sensitivity to the number of sampled trees K and the maximum number of debate rounds $R_{\max}$. Increasing K from 1 to 2 improves accuracy on both benchmarks, whereas $K = 3$ provides no further gain despite slightly lower token consumption. Increasing $R_{\max}$ generally improves accuracy by exposing the router to deeper continuations. The relatively high cost at $R_{\max} = 1$ results from more frequent escalation to the expensive *Vote* operator. Since $R_{\max}$ only specifies an upper bound, a larger value does not necessarily increase inference cost because the router may terminate earlier.

Table 11: Sensitivity to K and $R_{\max}$.

Analysis	K	$R_{\max}$	GPQA-D		LogiQA2.0	
			Acc.	**Tok.**	**Acc.**	**Tok.**
Tree samples	1	5	66.67	4,043	79.83	2,057
	2	**5**	**69.19**	4,107	**80.15**	1,812
	3	5	68.69	**3,819**	79.64	**1,796**
Max. rounds	2	1	67.17	5,057	77.67	3,520
	2	3	67.68	4,354	77.93	**1,613**
	2	**5**	**69.19**	**4,107**	**80.15**	1,812

D Case Studies

We present representative trajectories illustrating successful adaptive routing and a remaining failure mode. Router probabilities are obtained from the live-test checkpoint, and suffix credits are computed from sibling branches of the corresponding shared-prefix tree.

Case 1: Debate Corrects an Initial Error — MMLU

ID: `val:clinical_knowledge_val:8`
Question: Digital rectal examination anatomy (abridged).
Ground truth: D.

State s_0: Agent A predicts B and Agent B predicts D. The admissible operators are Debate and Vote. The router assigns $P(\text{Debate}) = 0.53$ and $P(\text{Vote}) = 0.47$, and selects Debate. The debate adds 1,923 tokens.

State s_1: Agent A revises B→D, yielding the correct consensus (D, D). The router assigns $P(\text{Accept}) = 0.66$ and $P(\text{Vote}) = 0.34$, and terminates with D.

Sibling suffix credits at s_0:

Operator	Correct	Suffix tokens	Preference
Debate	1	1,923	preferred
Vote	1	3,842	rejected

Result: Correct (D).
Analysis: Both continuations are correct, but the shared-prefix credit prefers Debate because it reaches the same outcome with substantially lower additional cost.

Case 2: Voting Selects the Correct Answer after Persistent Disagreement — MMLU

ID: `val:moral_disputes_val:13`
Question: Reiman raises objections to the arguments presented by Nathanson, Kant, abolitionists, or van den Haag (abridged).
Ground truth: D (van den Haag).

State s_0: Agent A predicts D and Agent B predicts B. Because the agents disagree, the admissible operators are Debate and Vote. The router assigns $P(\text{Debate}) = 0.52$ and

$P(\text{Vote}) = 0.48$, and selects Debate. The debate adds approximately 1,900 tokens, but neither agent revises its answer.

State s_1: The answers remain (D, B). The router assigns $P(\text{Debate}) = 0.47$ and $P(\text{Vote}) = 0.53$, and selects Vote. The vote stage adds approximately 4,130 tokens and returns D as the majority decision.

Sibling suffix credits at s_1:

Operator	Correct	Suffix tokens	Preference
Debate	0	∼4,500	rejected
Vote	1	∼4,130	preferred

Result: Correct (D). The complete execution consumes 6,924 tokens.

Analysis: Agent A retains the correct answer, but Debate does not resolve the disagreement. From the same state, the complete Vote continuation terminates correctly, whereas the Debate continuation ends incorrectly. The shared-prefix credit therefore prefers Vote primarily for correctness and secondarily for its lower suffix cost.

Case 3: Wrong Consensus Causes Early Acceptance — GPQA-D

ID: `diamond:106`
Question: In-silico screening of a molecule with multiple chirality and tautomeric configurations (abridged).
Ground truth: A.

State s_0: Both agents independently predict D. The router assigns $P(\text{Accept}) = 0.81$, $P(\text{Debate}) = 0.08$, and $P(\text{Vote}) = 0.11$, and terminates immediately with D.

Sibling suffix credits at s_0:

Operator	Correct	Suffix tokens	Preference
Accept	0	0	preferred
Debate	0	2,366	rejected
Vote	0	4,715	rejected

Result: Incorrect (D); total token consumption is 1,522.
Analysis: All sampled continuations remain incorrect, so the correctness-prioritized credit falls back to cost and favors Accept. This case illustrates that state-matched routing cannot recover an answer absent from all available continuations.

E PROMPT TEMPLATES

All stages use user-only API messages. The templates below show the multiple-choice format used for MMLU, GPQA-D, MedQA, and LogiQA2.0. For numerical and mathematical benchmarks, only the required final-answer format is changed.

The initial reasoning and voting stages use deterministic decoding with temperature 0.0. During Debate, each agent receives the original question, the two most recent responses, and compact answer trajectories. Earlier full responses are not repeatedly appended. The voting stage reuses the initial independent-reasoning prompt exactly.

Initial Independent Reasoning

Input: {User_Query} **Temperature:** 0.0

```
You are an assistant. Please solve the following problem
independently.

PROBLEM:
{User_Query}
```

```
Give your reasoning and final answer in the following format:

Thoughts: [step-by-step reasoning with relevant intermediate results]

Answer: (X), where X is chosen from [A,B,C,D].

The response must end with ``Answer: (X)''.
```

Heterogeneous Debate

Input: original question, both preceding responses, and both compact answer trajectories **Temperature:** provider/model default

```
You are participating in round {Round_Index} of a two-agent reasoning
debate.

PROBLEM:
{User_Query}

YOUR PREVIOUS RESPONSE:
{My_Previous_Response}

OTHER AGENT'S PREVIOUS RESPONSE:
{Other_Previous_Response}

YOUR ANSWER TRAJECTORY:
R0:{a0} -> R1:{a1} -> ···

OTHER AGENT'S ANSWER TRAJECTORY:
R0:{b0} -> R1:{b1} -> ···

STATUS: {AGREE or DISAGREE}

Audit both responses from first principles. If the answers disagree,
revise your answer only when the other response reveals a concrete
error in your reasoning. Do not change your answer merely to conform
to the other agent.

Return exactly:

Decision: KEEP or REVISE

Critique: [the decisive error, or ``no decisive error found'']

Thoughts: [concise updated reasoning]

Answer: (X), where X is chosen from [A,B,C,D].
```

Figure 5: Prompt templates for initial independent reasoning and heterogeneous debate. The initial stage uses deterministic decoding, whereas Debate uses the provider/model default decoding temperature.

Independent Voting

Input: {User_Query} **Temperature:** 0.0

Each voter receives exactly the same user message as in the Initial Independent Reasoning stage.

```
You are an assistant. Please solve the following problem
independently.

PROBLEM:
{User_Query}

Give your reasoning and final answer in the following format:

Thoughts: [step-by-step reasoning with relevant intermediate results]

Answer: (X), where X is chosen from [A,B,C,D].

The response must end with ``Answer: (X)''.
```

External Vote Aggregation

The voting prompt contains no debate history, previous answer, routing state, or other voter output. After all independent responses are generated, the final answer labels are extracted and aggregated outside the language model:

$$\hat{a}_{\text{vote}} = \arg\max_{a \in \mathcal{A}} \sum_{j=1}^{N_{\text{vote}}} \mathbb{I}[\hat{a}_j = a] \,.$$

Here, $\hat{a}_j$ denotes the answer produced by voter j, and $\mathcal{A}$ denotes the answer space. Thus, Vote broadens the evidence pool through history-independent reasoning rather than extending the preceding debate.

Figure 6: Independent-voting prompt and external aggregation procedure. Each voter independently solves the original query using the same prompt as the initial reasoning stage. Voters cannot observe the debate history or one another's outputs, and their extracted answers are combined by simple majority voting outside the language model.