RUMAD: Reinforcement-Unifying Multi-Agent Debate

AAAI Track

Chao Wang*
Tsinghua University
Shenzhen, China
wangchao23@mails.tsinghua.edu.cn

Han Lin*
Zhejiang University
Hangzhou, China
Tsinghua University
Shenzhen, China
hlin@zju.edu.cn

Huaze Tang
Tsinghua University
Shenzhen, China
thz21@mails.tsinghua.edu.cn

Huijing Lin
Tsinghua University
Shenzhen, China
linhj24@mails.tsinghua.edu.cn

Wenbo Ding†
Tsinghua University
Shenzhen, China
ding.wenbo@sz.tsinghua.edu.cn

ABSTRACT

Multi-agent debate (MAD) systems leverage collective intelligence to enhance reasoning capabilities, yet existing approaches struggle to simultaneously optimize accuracy, consensus formation, and computational efficiency. Static topology methods lack adaptability to task complexity variations, while external LLM-based coordination risks introducing privileged knowledge that compromises debate neutrality. This work presents **RUMAD (Reinforcement-Unifying Multi-Agent Debate)**, a novel framework that formulates dynamic communication topology control in MAD as a reinforcement learning (RL) problem.

RUMAD employs a content-agnostic observation scheme that captures high-level debate dynamics avoiding access to raw agent reasoning content. RUMAD uses a multi-objective reward to model solution quality, cohesion and efficiency. A PPO-trained controller dynamically adjusts edge weights in the communication graph, while a dual-threshold mechanism enables fine-grained control over both agent activation and information visibility.

Experimental evaluation across MMLU, GSM8K, and GPQA benchmarks demonstrates that RUMAD achieves substantial efficiency gains—reducing token costs by over 80%—while still improving reasoning accuracy compared to single LLM model and multiple MAD baselines. Notably, RUMAD trained exclusively on MMLU exhibits robust zero-shot generalization to out-of-domain (OOD) tasks, indicating that the learned communication strategies capture task-independent principles of effective multi-agent coordination. These results establish RUMAD as a efficient and robust approach for deploying multi-agent reasoning application with practical resource constraints.

KEYWORDS

Multi-agent Debate; LLM; Reinforcement Learning

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/CBJO8409

ACM Reference Format:
Chao Wang*, Han Lin*, Huaze Tang, Huijing Lin, and Wenbo Ding†. 2026. RUMAD: Reinforcement-Unifying Multi-Agent Debate: AAAI Track. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/CBJO8409

1 INTRODUCTION

Multi-agent debate (MAD) frameworks harness the collective intelligence of multiple large language models (LLMs) [1, 5, 6, 9, 13, 21, 25] to achieve superior reasoning and decision-making beyond what any single model can attain. However, the effectiveness and practicality of MAD systems fundamentally depend on how agents communicate and exchange information during debate [11].

Adaptivity: Most existing MAD approaches utilize static sparse topologies, such as Sparse MAD (S-MAD) [10], Group Debate (GD) [12] or Selective Sparse MAD (S^2-MAD) [22] using ring, star, or fixed grouped/hierarchical structures, to reduce the communication burden compared to fully connected networks [4, 8, 19]. Such methods inherently lack flexibility. Fixed topologies cannot adapt to the evolving diversity, difficulty, or semantic style changes present across different tasks (e.g., allocating the same budget to an easy and a hard problem). As a result, they easily either cut off valuable information paths for complex tasks or retain redundant, low-value connections on simple tasks, ultimately hampering both efficiency and collective reasoning.

Privacy: Some adaptive methods introduce external LLM-based "judger" or "summarizer" roles, where a larger often more powerful agent (e.g GPT-4) supervises the debate. Context within a debate is exposed to external models. External knowledge in return suppresses the diversity and emergent collaboration of internal agent swarm.

Efficiency:A further critical limitation of existing MAD approaches is the lack of explicit modeling of token cost. Most previous MAD designs focus solely on accuracy or consensus, overlooking the real-world computational and financial constraints imposed by large-scale LLM deployments. Without a principled framework for managing communication budgets, these systems either waste

*These authors contributed equally to this work.
††Corresponding author: ding.wenbo@sz.tsinghua.edu.cn

Table 1: Different types of MAD methods are designed from different strategies. Mark • indicates corresponding metrics are taken into design target. Mark ○ indicates methods may have side-effect on corresponding metrics but don't focus on them. Mark - means methods fail on corresponding aspects.

	Static Topology Methods	External-LLM Methods	**RUMAD**
Accuracy	•	•	•
Consensus	○	•	•
External Opinion Independent	•	-	•
Question Independent	•	-	•
Communication Cost Control	○	○	•

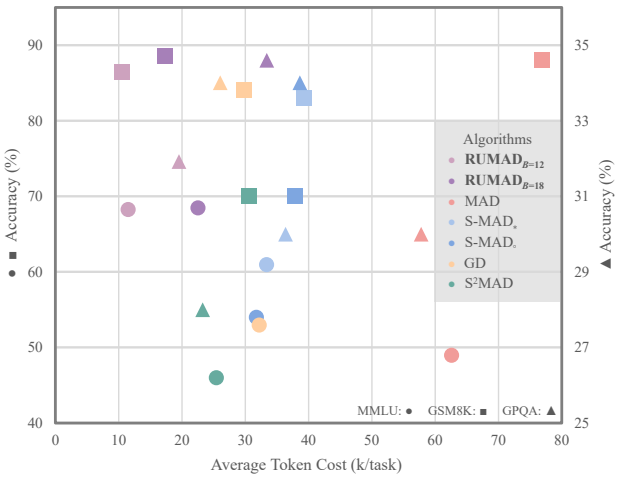

Figure 1: Comparison of accuracy and average token cost. Accuracy for MMLU (•) and GSM8K (■) corresponds to the left Y-axis, while GPQA (▲) corresponds to the right Y-axis. Points closer to the top-left corner represent a superior accuracy-efficiency trade-off. The results show that RUMAD variants consistently outperform baselines by achieving higher accuracy at a lower token cost.

excessive resources or struggle to generalize under practical constraints. Static methods typically impose a hard upper bound on communication rounds by restricting the debate topology.

Major challenges arise in balancing accuracy, diversity and efficiency(communication budget) of MAD, as show in Table 1. To address these challenges, we propose **RUMAD** (***Reinforcement-Unifying Multi-Agent Debate***), a novel framework that formulates dynamic agent coordination and topology control as a reinforcement learning (RL) problem, without external LLM utilization.

RUMAD formulates dynamic communication topology control as a reinforcement learning (RL) problem that is fundamentally independent of debate content. Specifically, we represent the communication topologies among agents as a dynamic weighted directed graph, where a non-LLM RL agent dynamically adjusts the edge weights in response to the evolving debate context. Crucially, the RL controller is **content-agnostic**: it does not intervene in the reasoning process or rely on privileged knowledge, but instead observes **high-level structural dynamics** such as semantic similarity scalars, answer agreement, debate progress, and communication cost. By modeling observations in this manner, RUMAD preserves the neutrality of dynamic debate management and effectively mitigates the risk of introducing external opinions or privileged information into the reasoning process.

RUMAD constructs a multi-objective reward function that explicitly balances accuracy, consensus, and token efficiency, while strictly enforcing communication budget constraints. We employ the PPO algorithm to train the topology control agent, which can achieve these trade-offs after only lightweight training. More importantly, the content-agnostic and neutral design of RUMAD endows it with strong zero-shot generalization capability, enabling robust performance across diverse tasks and domains without the need for task-specific finetuning.

Our experiments on standard benchmarks, including **MMLU** [7], **GSM8K** [3], and **GPQA** [16], demonstrate that RUMAD achieves a superior balance between collective accuracy and token usage compared to both fully-connected and static sparse baselines. As shown in Figure 1, take GSM8K for example, RUMAD$_{B=12}$ requires only 0.121k tokens per percentage point of accuracy, and RUMAD$_{B=18}$ requires 0.195k, compared to the 0.354k tokens required by the second-best baseline (GD). Concretely, RUMAD reduces token cost by 80% while improving accuracy on MMLU, and robustly generalizes to new tasks (GPQA, GSM8K) with similar performance.

Our contributions are summarized as follows:

- We propose RUMAD, a novel reinforcement learning framework that dynamically controls multi-agent debate topologies to optimize both accuracy and efficiency.
- We design a unified, content-agnostic observation and reward scheme that enables robust and generalizable learning of communication strategies across tasks and agent configurations.
- We empirically demonstrate that RUMAD outperforms both fully connected and static sparse MAD baselines, achieving superior trade-offs between performance and cost on multiple challenging benchmarks.

2 PRELIMINARIES

2.1 Multi-Agent Debate

Multi-agent debate (MAD) is an emergent paradigm for collective reasoning, where multiple language model agents iteratively exchange their intermediate thoughts and conclusions in order to reach a more accurate and robust consensus than any single agent

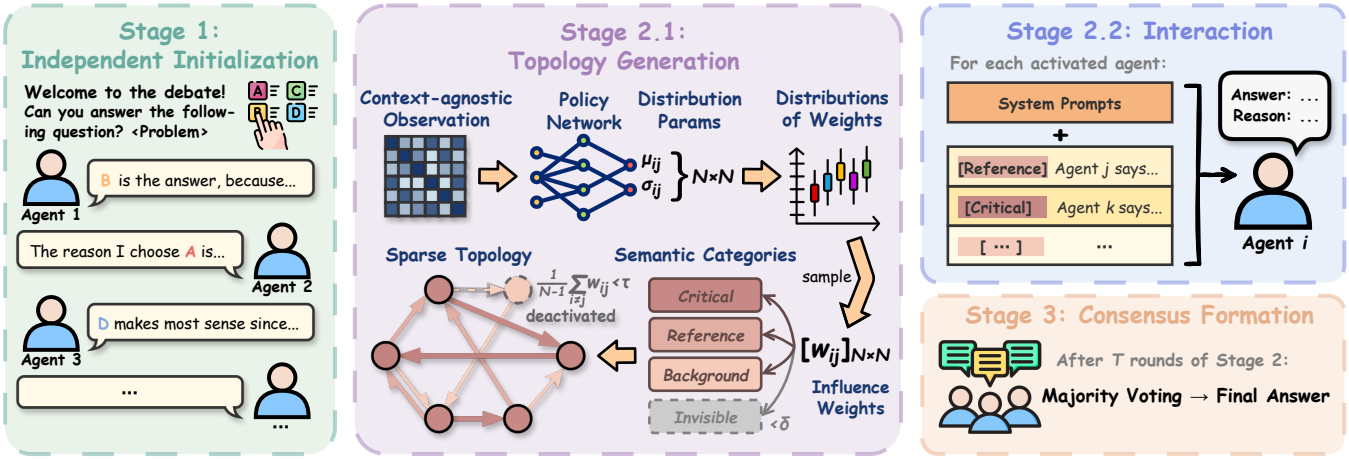

Figure 2: The process pipeline of RUMAD. In the first stage, all agents gives the initial response. In the second stage, RUMAD organizes topology in stage 2.1 and the selected agents debate with each other in stage 2.2. In the last stage, the final decision is obtained via majority voting.

alone. A central research focus within MAD is the design of communication topologies—that is, how agents share information and interact across debate rounds.

Early studies typically employed fully connected topologies, allowing all agents to access each other's responses at every round. However, such dense communication can be inefficient and costly, particularly as the number of agents grows. To address this, subsequent works have explored various forms of sparse and structured topologies. For instance, S-MAD [10] adopts fixed sparse patterns such as rings (only adjacent agents are visible) or stars (only centric agent are visible), enabling each agent to communicate only with a subset of neighbors, thereby reducing token consumption. Other approaches, like Group Debate (GD) [12], partition agents into static groups, which debate internally before group-level aggregation. Similarly, two-stage topologies such as S^2-MAD [22] first organize local debates within small groups and then aggregate results through a secondary global round if group opinions differ. These strategies all aim to balance the trade-off consensus formation, and computational efficiency.

2.2 Static Workflow Generation and Dynamic Topology Control

Multi-agent debate (MAD) systems rely on efficient inter-agent communication topologies. This challenge shares many similarities with recent research about automatic generation of agent workflows. such as MACNET [15], AFlow [24], and MaAS [23], which primarily focus on optimizing the static workflow or agentic pipeline. For example, MACNET analyzes the performance of predefined static topologies (e.g., chain, mesh) . AFlow uses MCTS to search for a single, optimal static workflow for a given task domain. MaAS refines this by using a controller to select a single, query-dependent static workflow at the beginning of a task. Despite above advances, however, most existing multi-agent LLM frameworks rely on static or manually specified topologies, which cannot adapt to the dynamic needs or varying complexity of different tasks.

This motivates the development of adaptive and principled methods for controlling agent interactions. Reinforcement learning (RL) is an end-to-end solution which is capable of both modeling and scheduling temporal dynamics during debate. Hierarchical RL frameworks like FMH [2] use multi-faceted rewards, where the objective is semantic task decomposition via manager-worker subgoaling.

Our work addresses a fundamentally different problem. RUMAD is not concerned with generating a static, task-level workflow. Instead, we focus on dynamic, round-by-round communication management within a debate. Our RL controller is invoked at each step to adaptively reconfigure the communication graph based on the evolving state of the debate (e.g., agent agreement, semantic similarity). This allows RUMAD to achieve fine-grained efficiency, such as pruning connections to agents that have already reached consensus, a capability not present in static workflow models. Considering hierarchical reward modeling, RUMAD's multi-objective reward function is novel in its content-agnostic nature, balancing the system-level trade-offs between solution quality, group cohesion, and computational cost.

2.3 Markov Decision Process

A Markov Decision Process (MDP) is a mathematical framework for modeling sequential decision-making problems, where an agent interacts with an environment over discrete time steps. At each step, the agent observes the current state, selects an action, and receives a reward and a new state from the environment. The objective is to find a policy that maximizes the expected cumulative reward [20]. In our multi-agent debate (MAD) setting, this framework allows the controller to base its topology decisions solely on the observable debate state at each round, ensuring a neutral and adaptive decision process that does not require access to agents' internal semantic content.

2.4 Proximal Policy Optimization

Proximal Policy Optimization (PPO) [18] is a widely used reinforcement learning algorithm for training policies in high-dimensional and dynamic environments. PPO employs an actor-critic framework, where a policy network proposes actions and a value network estimates the expected return for each state. By optimizing a clipped surrogate objective, PPO achieves stable and efficient policy updates, and has demonstrated robust performance across various sequential decision-making tasks. In our MAD framework, the value network provides reliable estimates of the debate state's long-term potential, which in turn stabilizes and guides the learning of RUMAD about adaptive topology control strategies. Additionally, we design a multi-objective reward model to capture the diverse goals of the debate process, and employ Generalized Advantage Estimation (GAE) [17] to further improve learning efficiency and convergence.

3 METHOD

We propose a reinforcement learning (RL) framework for adaptive topology control in multi-agent debate (MAD), in which a centralized controller dynamically regulates inter-agent communication to maximize collective reasoning performance under communication constraints. Our approach is characterized by the following innovations: (1) a content-agnostic observation scheme that leverages only interaction-relevant features without accessing agent-level semantics, (2) a multi-objective reward formulation balancing accuracy, consensus, efficiency, and sparsity, (3) a soft and differentiable action space modeling each edge's communication weight as a stochastic variable, and (4) a dual mechanism for communication efficiency combining budget constraints and agent pruning. The full pipeline consists of three stages: initial independent agent reasoning, iterative topology-controlled sparse debate, and consensus aggregation via majority voting.

3.1 Framework Pipeline

To illustrate the operation process of RUMAD, we decompose our framework into three sequential stages, each responsible for distinct aspects of multi-agent debate, as show in Figure 2.

Stage 1: Independent Initialization. All agents start by independently generating their initial responses to the given problem. Each agent i produces an initial response r_i^0 and conclusion c_i^0 based solely on the input prompt, without knowledge of other agents' perspectives, ensuring diverse hypotheses that span the solution space.

Stage 2: Adaptive Sparse Debate. In this stage, agents iteratively interact under dynamically controlled topologies. We divide the process into two focused steps:

(1) *Topology Generation:* The PPO policy network ingests the current observations and outputs the influence weights w_{ij}^t that determines how agents exchange information. Agents with insufficient incoming influence will be masked and reuse their previous responses to save tokens. Thus, we obtain the active nodes and the directed edges with different weights, which together constitute the interaction topology.

(2) *Interaction:* For each active agent i, the debate environment construct customized prompts by grouping neighbors into categories based on weight magnitude (Critical, Reference, Background, Invisible). This interaction step propagates information along pruned links, enabling focused debate under a communication budget.

Stage 3: Consensus Formation. After T adaptive rounds, the final conclusions $\{c_i^T\}$ are aggregated via majority voting. The consensus answer emerges as the label with the highest agent support, reflecting the collective reasoning of the ensemble rather than any single participant.

3.2 Content-Agnostic Observation

At each debate round t, the controller observes a global state o_t constructed without reference to raw agent content. For each agent i, let r_i^t denote its reasoning embedding and c_i^t its current answer. The embedded observable state for actor and critic is identically defined by:

$$o_t = \mathrm{MLP}([\mathrm{sim}(i,j)]_{N\times N}) \tag{1}$$

where the $N \times N$ similarity matrix is given by MAD environment:

$$\mathrm{sim}(i,j) = \lambda \cos(r_i^t, r_j^t) + (1-\lambda)\,\mathbb{I}(c_i^t = c_j^t) \tag{2}$$

with hyperparameter $\lambda \in [0,1]$, cosine similarity $\cos(\cdot,\cdot)$ between reasoning embeddings, and $\mathbb{I}(\cdot)$ the indicator function for answer agreement. The observation thus encodes only answer-level agreement and high-level semantic structure, ensuring content-agnostic control. RUMAD scheduler can only observe an $N \times N$ matrix with floating-point similarity number within $[0,1]$ without accessing the raw text (nor embedding of text) of agent reasoning.

3.3 Action Space and Edge Weight Distribution Modeling

In our framework, the action space at each debate round consists of a set of continuous communication weights, where each possible directed edge (i,j) is assigned a real-valued weight $w_{ij}^t \in (0,1)$ reflecting the influence of agent j on agent i. To enable principled policy optimization via stochastic gradient methods, we adopt a probabilistic action parameterization: the policy network (actor) outputs, for each edge, the parameters $(\mu_{ij}^t, \sigma_{ij}^t)$ of a Gaussian distribution, from which the edge weight is sampled as

$$w_{ij}^t \sim \mathrm{Sigmoid}\big(\mathcal{N}(\mu_{ij}^t,\ {\sigma_{ij}^t}^2)\big). \tag{3}$$

This formulation satisfies the philosophy of PPO. The use of a Gaussian distribution, followed by a Sigmoid transformation to enforce the valid range, provides a flexible and fully differentiable mechanism for modeling continuous actions. This allows for efficient exploration of the topology space, mitigates overfitting to deterministic patterns, and enables robust credit assignment across complex, high-dimensional action spaces. Such stochastic action parameterizations are standard in modern RL for continuous control (e.g., PPO).

3.4 Communication Budget Constraint

To achieve joint optimization of reasoning accuracy and communication efficiency, we introduce a dual-threshold mechanism that

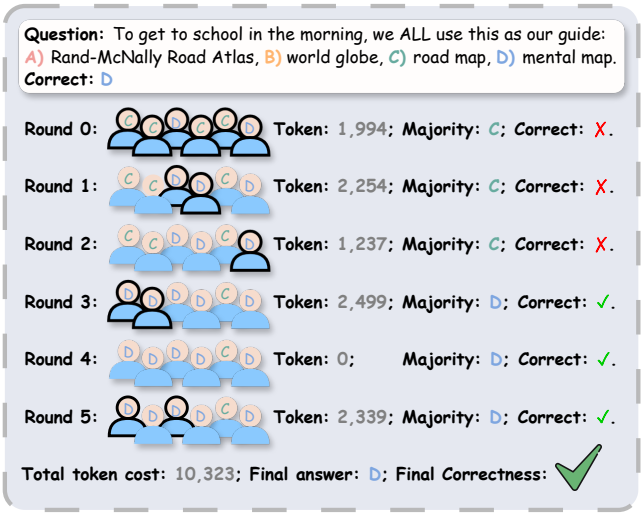

Figure 3: An example debate controlled by RUMAD.

directly links the learned edge weights to both token consumption and semantic information flow within the debate process.

Specifically, at each round, the sampled weight matrix $W^t = [w_{ij}^t]_{N\times N}$ not only determines the semantic relevance of each neighbor's response for prompt construction (edge-wise), but also governs whether an agent actively updates its opinion (node-wise), thereby incurring token cost.

For each agent i, we compute its average external influence (in-degree) as

$$\bar{w}_i^t = \frac{1}{N-1}\sum_{j\neq i} w_{ij}^t, \tag{4}$$

where N is the total number of agents. An activation threshold τ is applied: if $\bar{w}_i^t < \tau$, agent i is pruned for the current round and reuses its previous response. For simplicity, we utilize w_{ii} as activation threshold τ. The $\tau = w_{ii}^t$ design dynamically models implicit "confidence" of each agent. An agent is confident enough while assigned with higher weight on its own opinion than the average of the external, $\bar{w}_i^t < w_{ii}^t$. As shown by example round 4 in Figure 3, it results in zero token usage this round that all agents are assigned as confident. In the sub-sequential round 5, different weights are sampled from distributions defined in Equation 3 thus more communication occurs. This threshold thus directly controls the expected number of agents generating new tokens per round, providing a fine-grained handle on collective communication expenditure.

The computed weights determine not only the influence strength but also how information is presented to each agent. We implement a three-tier prompt priority filter:

$$P_{ij}^t = \begin{cases} \text{[Critical]} & \text{if } w_{ij}^t > 0.40 \\ \text{[Reference]} & \text{if } w_{ij}^t > 0.25 \\ \text{[Background]} & \text{if } w_{ij}^t > 0.10 \end{cases} \tag{5}$$

Agents prioritize debate context with tags P_{ij}^t quantized from computed influence weights w_{ij}^t, while maintaining the natural language interaction paradigm of LLMs.

To regularize overall communication, we introduce a global budget parameter B representing a soft upper bound on the total number of active communication links per round. Formally, the set of active links is defined as $\{(i,j) : w_{ij}^t > \delta\}$(where $\delta = 0.10$ as minimum visible weight shown in Equation 5), and a budget penalty is incorporated into the policy loss whenever this set exceeds B:

$$\mathcal{L}_{\text{budget}}(\theta) = \max\left(0, \sum_{i,j}\mathbb{I}(w_{ij}^t > \delta) - B\right), \tag{6}$$

where $\mathbb{I}(\cdot)$ is the indicator function. Crucially, B acts as a **training-time prior** via the loss function, encouraging the policy to internalize a preference for sparsity, rather than acting as a **hard constraint at inference**. This budget constraint reflects a prior preference for overall communication compactness, while leaving the allocation of communication opportunities across agents to be adaptively learned by the policy to best balance informativeness and efficiency.

Through this budget-regularized design, RUMAD enables precise, learnable trade-offs between debate effectiveness and communication cost, granting the controller the capacity to both minimize unnecessary token usage and adaptively maintain critical informational flows.

3.5 Multi-Objective Reward

To guide the RL agent in navigating the complex trade-offs of debate, we designed a hierarchical, multi-objective reward function. Our design is structured around three core objectives:

Maximizing Solution Quality. As major rewarding signal, $\text{Accuracy}_{t/ep}$ are 0/1 binary rewards reflecting whether the majority answers match the ground truth. Progress_t applies a round-driven-decaying reward amplification to Accuracy_t when round answer is correct. Thus correctness in early-stage is encouraged. $\text{Improvement}_{t/ep}$ are also binary signals rewarding shift from wrong answer to correct answer across adjacent rounds or between round 0 and final response.

Promoting Group Cohesion. $\text{Consensus}_{t/ep}$ measures the proportion of agent answers that match current majority answer, together with their semantic similarity, rewarding more unified and coherent group responses.

Optimizing Resource Efficiency. Efficiency_t incentivizes reduced token consumption relative to a baseline answer-length budget. In addition, a penalty signal, Sparsity_t, is designed as indicator of "schedule expense". Expenses are assigned corresponding to semantic magnitude filtered in Equation 5.

Formally, at each debate round t, the controller receives a composite reward R_t that integrates six key factors:

$$\begin{aligned} R_t = {} & \alpha_1\,\text{Accuracy}_t + \alpha_2\,\text{Consensus}_t + \alpha_3\,\text{Progress}_t \\ & + \alpha_4\,\text{Efficiency}_t + \alpha_5\,\text{Improvement}_t - \alpha_6\,\text{Sparsity}_t. \end{aligned} \tag{7}$$

where Progress_t and Sparsity_t are designed as in-progress-only signals. At the end of each debate episode, we compute a terminal reward R_{ep}:

$$\begin{aligned} R_{ep} = {} & \beta_1\,\text{Accuracy}_{ep} + \beta_2\,\text{Consensus}_{ep} \\ & + \beta_3\,\text{Efficiency}_{ep} + \beta_4\,\text{Improvement}_{ep} \end{aligned} \tag{8}$$

R_t and R_{ep} are roughly homogeneous with different preference. Parameters of R_t and R_{ep} are distributed similarly[1], except Improvement$_t$ are emphasized with higher weight in R_t as encouragement of runtime adjustment. This two-tier reward mechanism enables the controller to balance immediate step-wise gains (e.g., rapid consensus or token savings) with ultimate debate-level objectives, facilitating robust, generalizable policy learning for efficient multi-agent reasoning under budget constraints. Structural necessity of this hierarchical design is supported by ablation in Section 4.3. And parameter robustness is shown through 0-shot generalization revealed in Section 4.2.

3.6 RL Dynamic Topology Control

The controller is trained using the Proximal Policy Optimization (PPO) framework. The policy loss is given by

$$\mathcal{L}_\pi(\theta) = -\mathbb{E}_t \left[\min \left(r_t \hat{A}_t,\ \text{clip}(r_t, 1-\varepsilon, 1+\varepsilon)\hat{A}_t \right) \right] \tag{9}$$

where $r_t = \frac{\pi_\theta(a_t|o_t)}{\pi_{\theta_{\text{old}}}(a_t|o_t)}$ is the probability ratio, and $\hat{A}_t$ is the advantage computed via Generalized Advantage Estimation (GAE) using the composite reward.

The PPO value network ϕ is trained to minimize the squared error:

$$\mathcal{L}_v(\phi) = \mathbb{E}_t \left[(V_\phi(o_t) - V_t^{\text{target}})^2 \right] \tag{10}$$

with $V_\phi(o_t)$ as predicted expectation of future return with current observation o_t versus V_t^{target} as the bootstrapped return at time step t.

The total loss for joint PPO training is

$$\mathcal{L}(\theta, \phi) = \mathcal{L}_\pi(\theta) + c_1\, \mathcal{L}_v(\phi) + c_2\, \mathcal{L}_{\text{budget}}(\theta) \tag{11}$$

where $c_1, c_2 \geq 0$ balance loss signals among policy, value and regularization into comparable orders of magnitude, according to PPO style.

Training proceeds by iteratively sampling debate episodes, updating the controller with the composite loss, and periodically evaluating performance under fixed communication budgets. At deployment, the controller deterministically selects edge weights using the mean of the learned Gaussian distributions.

4 EXPERIMENTS

4.1 Experimental Setup

We conduct experiments on three challenging and complementary benchmarks: **MMLU** [7], **GSM8K** [3], and **GPQA** [16]. The PPO-based topology controller of RUMAD is exclusively trained on the MMLU dev dataset[1], leveraging its broad coverage of multi-domain reasoning tasks. For evaluation, we report results on the MMLU test set as well as on GSM8K (arithmetic reasoning) and GPQA (graduate-level science QA). Notably, for GSM8K and GPQA, we apply RUMAD controller in a zero-shot transfer setting without any further fine-tuning, directly assessing the generalization capability of the learned communication policy.

To maximize the heterogeneity and dynamic interactions within the debate, we instantiate a diverse agent pool comprising three different LLM models: **LLaMA-3.1-8B-Instruct** [21], **ChatGLM-4-9B** [5], and **Deepseek-Math-7B-Instruct** [6]. Each architecture is represented by two agents, forming a six-member debate group per trial. For each agent, there are 4 non-similar agents and 1 similar agent. This multi-architecture composition is designed to capture a spectrum of reasoning styles and knowledge bases.

For computational efficiency and scalability, all LLMs are deployed using **4-bit groupwise quantization (Q4_K_M)** with mixed precision. Embedding model used for RUMAD to compute similarity among texts, defined in Equation 2, is **Nomic-Embed-v1** [14]. This configuration enables concurrent inference of all agents on a single NVIDIA GeForce RTX 3090 GPU, facilitating systematic ablation and large-scale evaluation. Thus, an efficiency-focused **0-shot, 4-bit** setup is crucial for fair baseline comparison. For instance, the LLaMA-3.1-8B-Instruct is officially reported with

[1]Check reward weights and training hyper parameters in Table 5,6 in Appendix.A of https://github.com/chaozju2016/RUMAD/blob/main/RUMAD.pdf

Table 2: Performance of RUMAD and baselines across three evaluation datasets. Token cost is calculated as average of each topic debated. (6,6) is general mark of MAD systems indicating 6-agent, 6-round configuration. RUMAD significantly reduces token cost with comparable or improved accuracy. S-MAD with different structures are denoted as * (Star) and o (Ring). Since RUMAD is trained upon MMLU dev partition, MMLU test partition is used for evaluation. RUMAD evaluated upon GPQA-main and GSM8K is zero-shot migrated, without any finetuning with new datasets. RUMAD shows superior cross-domain generality. Note: All baseline accuracies are measured under identical 4-bit quantization and 0-shot settings for a fair comparison.

Task	Method (6,6)	ACC	Token Cost (k/task)	Cost Saving
MMLU	**RUMAD**$_{B=12}$	68%	**11.43**	**-81.74%**
	RUMAD$_{B=18}$	**68%**	22.46	-64.11%
	MAD	49%	62.58	N/A
	S-MAD$_*$	61%	33.32	-46.75%
	S-MAD$_\circ$	54%	31.70	-49.34%
	GD	53%	32.17	-48.59%
	S^{2}MAD	46%	25.36	-58.48%
GPQA	**RUMAD**$_{B=12}$	32%	**19.53**	**-66.21%**
	RUMAD$_{B=18}$	**35%**	33.39	-42.24%
	MAD	30%	57.80	N/A
	S-MAD$_*$	30%	36.37	-37.08%
	S-MAD$_\circ$	34%	38.59	-33.24%
	GD	34%	26.05	-54.93%
	S^{2}MAD	28%	23.26	-59.76%
GSM8K	**RUMAD**$_{B=12}$	86%	**10.46**	**-86.40%**
	RUMAD$_{B=18}$	**89%**	17.28	-77.53%
	MAD	88%	76.90	N/A
	S-MAD$_*$	83%	39.38	-48.78%
	S-MAD$_\circ$	70%	37.82	-50.81%
	GD	84%	29.76	-61.30%
	S^{2}MAD	70%	30.51	-60.32%

69.4% accuracy on MMLU under **5-shot, bf16-precision** conditions, while actually achieving 57.9% under **0-shot, 4-bit** setting.

We benchmark RUMAD primarily against MAD, a fully-connected debate protocol that omits both visibility control and prompt structuring, thereby preserving complete information exchange among all agents throughout the discussion. In addition, we evaluate against S-MAD, GD, and S^2-MAD using each method's reported optimal configuration.

Our evaluation focuses on two key metrics: (1) solution **accuracy** across diverse problem categories, and (2) computational **efficiency** as measured by total token consumption. And we evaluate generality of RUMAD by comparing its performance across different benchmarks. Scripts for training and evaluating, along with RUMAD PPO checkpoint weight files, will be provided in supplementary files.

4.2 Results and Analysis

As shown in Table 2, our proposed RUMAD framework achieves a substantial reduction in communication cost while maintaining, or even enhancing, answer accuracy across all evaluated benchmarks. On both MMLU and GSM8K, RUMAD reduces average token usage per task by over 80% compared to the fully connected MAD baseline, yet achieves much better accuracy (e.g., 68% vs 49% on MMLU). RUMAD also provides best token cost saving on GPQA, while higher accuracy is available if assigned more communication budget (12 to 18 times per round). This demonstrates the efficacy of reinforcement learning-based dynamic topology control of RUMAD in identifying and preserving the most informative agent interactions under tight communication budgets. The results highlight that significant efficiency gains are attainable without sacrificing group reasoning performance.

*B **as configurable prior budget**.* A salient advantage of our framework lies in the explicit budget parameter B utilized in budget loss defined in Equation 6, which directly governs the maximum expected communication load per round. By adjusting B, practitioners can specify a clear prior on resource consumption, aligning debate efficiency with practical deployment constraints. Our results show that even at relatively low budget levels (e.g., $B = 12$ for six agents), the controller learns to allocate communication judiciously, maintaining high accuracy with a fraction of the original token cost.

We configured a comparison with $B = 18$, indicating user prior preference is communication of 3 times per agent per round in identical 6-agent system. RUMAD is capable to allocate this budget adaptively and get higher accuracy performance on all benchmarks, with corresponding trade-off on token-cost. This interpretability and controllability of B enhances the usability of our approach in real-world multi-agent systems. As we can see, on MMLU tasks, $B = 12$ budget is sufficient for RUMAD to coordinate debate and achieve comparable accuracy with $B = 18$ budget (68.30% vs 68.48%), while on GPQA and GSM8K, RUMAD utilize more budget to provide better performance.

As defined in Equation 6, B acts as a **training-time prior** via the loss function, encouraging the policy to internalize a preference for sparsity, rather than a **hard constraint at inference** applied to scheduler. Utilization of B provides better alignment of human prior preference without increasing cost of hyperparameter tuning.

Table 3: Ablation study of RUMAD's key components. We report accuracy (ACC, %) and average token cost (TC, in k/-task). The best result for each metric is in bold. We set $B = 12$ for all variants.

Ablation Module	MMLU		GPQA		GSM8K	
	ACC	TC	ACC	TC	ACC	TC
RUMAD	68.3	**11.4**	**31.9**	**19.5**	**86.4**	**10.5**
β distribution	65.1	18.7	29.2	28.7	84.5	17.2
w/o R_{ep}	66.9	15.6	31.0	24.4	85.1	14.3
w/o R_t	67.2	22.2	30.1	32.6	86.0	20.2
w/o agent activation	65.4	51.9	31.7	67.5	86.1	45.0
w/o $\mathcal{L}_{\text{budget}}$	**70.0**	17.4	27.7	26.8	85.8	15.6

RUMAD zero-shot generality. We observe that the RUMAD controller, trained solely on the MMLU development set, generalizes robustly to out-of-domain benchmarks such as GPQA and GSM8K in a zero-shot setting. It's notable that MMLU is single-choice task while GSM8K needs arithmetic answers. Despite no exposure to these domains during training, RUMAD maintains competitive accuracy and cost savings, underscoring the generality and transferability of the learned communication policy. This strong out-of-domain performance also serves as key evidence of **hyperparameter robustness**. The same model trained on MMLU under reward model with parameterα_k, β_k transferred directly to GSM8K and GPQA without modification, indicating the learned policy is not brittle or overfitted to the training domain, a common concern with complex reward structures. This result suggests that our method is not tightly coupled to any single task distribution or agent type, facilitating practical deployment across diverse collaborative reasoning scenarios.

4.3 Ablation Study

We conduct a comprehensive ablation study to investigate the contribution of each key component of RUMAD. To ensure a fair comparison, all variants are trained exclusively on the MMLU dev dataset and operate under a consistent 6-agent, 6-round configuration. The results, summarized in Table 3, are evaluated on MMLU, GPQA, and GSM8K, reporting both accuracy and average token cost. The findings demonstrate that each module is integral to the model's performance, and the full RUMAD framework achieves the best overall trade-off between accuracy and efficiency.

Change to β distribution. RUMAD models agent connections using a Gaussian distribution to avoid overfitting. Replacing this with a β distribution results in a significant performance drop: accuracy falls across all benchmarks (e.g., 68.3% to 65.1% on MMLU), and the token cost substantially increases, underscoring the superior modeling capability of the Gaussian parameterization.

Ablation of R_{ep} and R_t. Our two-tier reward system is critical for balancing immediate and long-term goals. It combines the per-round reward R_t with an episode reward R_{ep} applied only at the final round. In thew/o R_{ep} variant, we replace the final round's R_{ep} with the standard R_t, which causes a notable drop in accuracy

Table 4: Scalability analysis with varying agent numbers. We report Accuracy (ACC) and average Token Cost (TC, k/task) on MMLU and GSM8K.

Agents (N)	Budget (B)	MMLU		GSM8K	
		ACC	TC	ACC	TC
3	6	66.6%	2.3	85.0%	2.0
6	12	68.3%	11.4	86.4%	10.5
6	18	68.5%	22.5	**88.6%**	17.3
9	18	**70.3%**	17.6	88.5%	15.9

and a ∼37% increase in token cost on MMLU (11.4k to 15.6k). Conversely, in the w/o R_t variant, we broadcast the final R_{ep} value to all preceding rounds. This results in a severe impact on efficiency, nearly doubling the token cost. This highlights that both reward signals are necessary for learning an efficient policy.

Ablation of agent activation. The agent activation mechanism is paramount for efficiency. It conditionally decides if an agent generates a new response, enabling true communication sparsification. Removing it (w/o agent activation) means that even with efficient pruning of communication edges, all agents are still forced to generate a new response in every round. This inability to dynamically silence agents causes a catastrophic increase in token cost (e.g., a ∼4.5x increase on MMLU), while accuracy also suffers. This confirms that dynamic agent activation, not just edge pruning, is the primary driver of RUMAD's token savings.

Ablation of $\mathcal{L}_{\text{budget}}$. The budget loss $\mathcal{L}_{\text{budget}}$ acts as a crucial regularizer for both efficiency and generalization. Without it, the controller fails to converge to an efficient policy, incurring higher token costs. Interestingly, while this variant achieves higher accuracy on the in-domain MMLU task (70.0%), its performance drops significantly on the out-of-domain benchmarks GPQA (27.7% vs. 31.9%) and GSM8K (85.8% vs. 86.4%). This strongly suggests that $\mathcal{L}_{\text{budget}}$ not only enforces efficiency but also guides the RL algorithm to learn more generalizable communication strategies.

4.4 Scalability Analysis

We investigate RUMAD's scalability and robustness by evaluating its performance across varying agent group sizes ($N \in \{3, 6, 9\}$) and budgets ($B \in \{6, 12, 18\}$), with results summarized in Table 4. For a fair comparison, all hyperparameters except B (e.g., learning rate, reward coefficients) are kept identical, demonstrating the model's robustness.

First, the results demonstrate a clear positive correlation between the number of agents and reasoning performance. When scaling the agent count N and communication budget B proportionally ($N \in \{3, 6, 9\}$ while $B = 2 \times N$), we observe consistent improvements in accuracy—rising from 66.6% to 70.3% on MMLU and from 85.0% to 88.5% on GSM8K. This trend indicates that RUMAD effectively harnesses the increased diversity and collective intelligence of larger groups. While the token cost naturally increases with the number of agents, the framework remains efficient relative to the performance gains.

A more compelling insight emerges when scaling the agent pool under a fixed budget. By increasing the agent count from $N = 6$ to $N = 9$ while keeping the budget at $B = 18$, we observe a simultaneous increase in MMLU accuracy (68.5% → 70.3%) and a significant decrease in token cost for both benchmarks (e.g., 22.5k → 17.6k on MMLU). This counter-intuitive result strongly demonstrates the adaptability of our RL controller. Given a larger pool of agents, the controller learns to allocate its fixed budget more judiciously, selecting a more specialized or diverse subset of agents to participate. This intelligent selection leads to a more efficient path to the solution, enhancing performance while reducing redundant communication.

Across experiments upon different swarm, RUMAD shows stable ability to find a more efficient and accurate policy without requiring extensive hyperparameter tuning. It suggests RUMAD learns a generalizable communication policy rather than overfitting to a specific setup, highlighting its hyperparameter insensitivity and flexibility for deployment in diverse environments.

5 CONCLUSION

This paper introduces RUMAD, a principled reinforcement learning framework for adaptive topology control in multi-agent debate systems. By leveraging a content-agnostic observation design and a multi-objective reward scheme, RUMAD enables dynamic, efficient, and neutral management of agent interactions—successfully balancing accuracy, consensus, and computational efficiency under strict communication budgets. RUMAD also exhibits robust zero-shot generalization to diverse tasks and domains, showcasing strong transferability and practical deployment potential. Extensive experiments across MMLU, GSM8K, and GPQA demonstrate that RUMAD reduces token consumption by over 80% compared to fully connected baselines, while maintaining or even improving collective reasoning accuracy against all baselines.

These results establish RUMAD as a promising solution for scalable, cost-effective multi-agent reasoning, and open avenues for future research on principled **coordination** and **communication mechanisms** in large-scale **multi-agent systems**. While RUMAD demonstrates strong performance, we acknowledge existing limitations. The centralized PPO controller, while efficient for typical MAD groups (6-8 agents), may face scalability challenges in scenarios with hundreds of agents. Future exploration about decentralized or hierarchical Multi-agent RL controllers is worth studying.

ACKNOWLEDGMENTS

This work was supported by Shenzhen Science and Technology Program (No. KJZD20240903100905008).

REFERENCES

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report.
[2] Sanjeevan Ahilan and Peter Dayan. 2019. Feudal Multi-Agent Hierarchies for Cooperative Reinforcement Learning. https://doi.org/10.48550/arXiv.1901.08492 arXiv:1901.08492 [cs]
[3] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems.
[4] Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent

debate.
[5] Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, et al. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools.
[6] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.
[7] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding.
[8] Jennifer Hill, W Randolph Ford, and Ingrid G Farreras. 2015. Real conversations with artificial intelligence: A comparison between human–human online conversations and human–chatbot conversations. , 245–250 pages.
[9] Juyong Jiang, Fan Wang, Jiasi Shen, Sungju Kim, and Sunghun Kim. 2024. A Survey on Large Language Models for Code Generation.
[10] Yunxuan Li, Yibing Du, Jiageng Zhang, Le Hou, Peter Grabowski, Yeqing Li, and Eugene Ie. 2024. Improving Multi-Agent Debate with Sparse Communication Topology.
[11] Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2023. Encouraging divergent thinking in large language models through multi-agent debate.
[12] Tongxuan Liu, Xingyu Wang, Weizhe Huang, Wenjiang Xu, Yuting Zeng, Lei Jiang, Hailong Yang, and Jing Li. 2024. Groupdebate: Enhancing the efficiency of multi-agent debate using group discussion.
[13] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2023. A comprehensive overview of large language models.
[14] Zach Nussbaum, John X. Morris, Brandon Duderstadt, and Andriy Mulyar. 2025. Nomic Embed: Training a Reproducible Long Context Text Embedder. https://doi.org/10.48550/arXiv.2402.01613 arXiv:2402.01613 [cs]
[15] Chen Qian, Zihao Xie, YiFei Wang, Wei Liu, Kunlun Zhu, Hanchen Xia, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, Zhiyuan Liu, and Maosong Sun. 2025. Scaling Large Language Model-based Multi-Agent Collaboration. https://doi.org/10.48550/arXiv.2406.07155 arXiv:2406.07155 [cs]
[16] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. GPQA: A Graduate-Level Google-Proof Q&A Benchmark. https://doi.org/10.48550/arXiv.2311.12022 arXiv:2311.12022 [cs]
[17] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2018. High-Dimensional Continuous Control Using Generalized Advantage Estimation. https://doi.org/10.48550/arXiv.1506.02438 arXiv:1506.02438 [cs]
[18] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. https://doi.org/10.48550/arXiv.1707.06347 arXiv:1707.06347 [cs]
[19] Qiushi Sun, Zhangyue Yin, Xiang Li, Zhiyong Wu, Xipeng Qiu, and Lingpeng Kong. 2023. Corex: Pushing the boundaries of complex reasoning through multi-model collaboration.
[20] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, Massachusetts.
[21] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models.
[22] Yuting Zeng, Weizhe Huang, Lei Jiang, Tongxuan Liu, Xitai Jin, Chen Tianying Tiana, Jing Li, and Xiaohua Xu. 2025. S^2-MAD: Breaking the Token Barrier to Enhance Multi-Agent Debate Efficiency.
[23] Guibin Zhang, Luyang Niu, Junfeng Fang, Kun Wang, Lei Bai, and Xiang Wang. 2025. Multi-Agent Architecture Search via Agentic Supernet. https://doi.org/10.48550/arXiv.2502.04180 arXiv:2502.04180 [cs]
[24] Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. 2025. AFlow: Automating Agentic Workflow Generation. https://doi.org/10.48550/arXiv.2410.10762 arXiv:2410.10762 [cs]
[25] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models.