Meta-Moderator: Empowering Multi-Agent Debate with Meta-Cognition

Wentao Hu[♠] **Zhuoyue Wan**[♠] **Jinhao Shen**[♠]
Chen Jason Zhang[♠] **Xiaoyong Wei**[♡,♠,*] **Qing Li**[♠]
[♠]The Hong Kong Polytechnic University [♡]Sichuan University
wayne-wt.hu@connect.polyu.hk
{jason-c.zhang, cs007.wei}@polyu.edu.hk

Abstract

Multi-agent debate can improve large language model reasoning by eliciting diverse hypotheses and critiques, yet its performance is often constrained by weak moderation. Common pipelines rely on fixed budgets, agreement-based stopping, or untrained judges, leading to redundant deliberation and unreliable evidence aggregation. We cast moderation as a meta-cognitive process, monitoring debate utility, controlling deliberation, and adjudicating a final answer, and introduce **Meta-Moderator**, a learnable framework that dynamically regulates debate and decides when to finalize an answer. Meta-Moderator is trained independently of the debaters via outcome-driven policy optimization, making debate regulation an explicit capability rather than an incidental effect of prompting. Across five benchmarks, Meta-Moderator outperforms widely used decision layers and transfers across tasks and system configurations. Further analyses show that it allocates debate more selectively and reduces mis-aggregation after informative hypotheses appear.[1]

1 Introduction

Large language models (LLMs) often achieve stronger reasoning when multiple agents are allowed to interact, critique one another, and iteratively refine their answers (Du et al., 2024; Liang et al., 2024; Chan et al., 2024). This multi-agent debate (MAD) paradigm has produced consistent gains across tasks such as mathematical reasoning, multi-hop question answering, and factual verification, motivating the common belief that structured interaction yields more reliable reasoning than a single agent (Zhang and Xiong, 2025; Hu et al., 2025b).

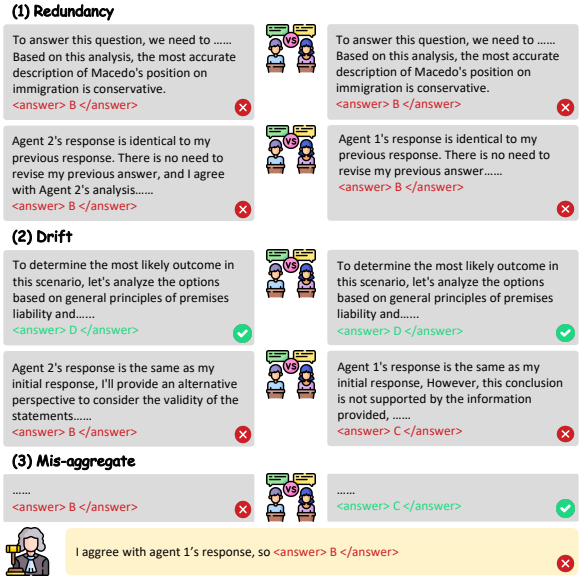

Figure 1: Three common issues in standard multi-agent debate: redundancy, drift, and mis-aggregation.

Despite its promise, MAD is far from reliably beneficial. As shown in Figure 1, debates can (i) become **redundant**, expending additional rounds that contribute little new information; (ii) **drift** away from the key uncertainty, accumulating plausible but irrelevant details; or (iii) **mis-aggregate** high-quality arguments into an incorrect final decision, for instance due to superficial consensus, last-round bias, or voting failures under dispersed opinions (Becker, 2024; Zheng et al., 2024; Choi et al., 2025; Cui et al., 2025; Hong et al., 2025). These failure modes arise naturally when collective deliberation is run without reliable mechanisms for assessing progress, allocating deliberation, and committing to a final decision.

We argue that these issues point to a central, under-modeled bottleneck in MAD: the *moderator*. In real-world adversarial deliberation (e.g., courtroom argumentation), advocates focus on local evidence and persuasive reasoning, while a neutral adjudicator maintains global context, tracks what

[*] Corresponding author
[1] Our code available at https://github.com/Huenao/Meta-Moderator.

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 26435–26451
October 24-29, 2026 ©2026 Association for Computational Linguistics

has been resolved, and decides when the discussion is ready for a decision. By contrast, most MAD systems treat moderation as a heuristic add-on: interaction is governed by fixed budgets, shallow consensus checks, or voting-based aggregation (Choi et al., 2025; Cui et al., 2025). Recent work has begun to explore measurement-driven controllers that stop when debate signals plateau or when judge dynamics stabilize (Chang and Chang, 2025; Hu et al., 2025a). However, these approaches still rely on predefined signals and hand-chosen stopping criteria, rather than learning a moderation policy optimized for end-task accuracy. Some approaches introduce an LLM judge, but it typically operates as a prompted black-box evaluator rather than a policy optimized for regulating debate dynamics (Liang et al., 2024; Hong et al., 2026). As a result, the very capabilities that determine whether debate translates into accuracy gains are left implicit and fragile.

This motivates a meta-cognitive view of moderation. Drawing inspiration from meta-reasoning and metacognition, which study how cognitive processes are monitored and regulated under limited resources (Ackerman and Thompson, 2017; Ball and Richardson, 2025; Gao et al., 2024; De Sabbata et al., 2024), we characterize effective moderation as a tightly coupled loop of three functions: **Monitoring** whether deliberation is making substantive progress versus repeating or drifting; **Control** allocating deliberation by deciding whether to prolong deliberation or commit to a decision (Wei and Yang, 2013); and **Adjudication** synthesizing a final answer from the debate content in a way that is robust to superficial agreement and aggregation biases. Under this perspective, moderation is not merely another round of object-level reasoning; it is a distinct, policy-like capability that must decide *when* further deliberation is worthwhile and *when* to commit to a final decision.

Motivated by this framing, we introduce **Meta-Moderator**, a framework for learnable moderation that treats debate regulation as a meta-level decision problem. Meta-Moderator learns an explicit moderation policy that decides, at each round, whether further deliberation is warranted or whether to stop and commit to a final answer. We train this policy independently from debaters with outcome-driven reinforcement learning, so that moderation becomes a dedicated, learnable competence rather than an incidental outcome of prompting or heuristic aggregation.

Our contributions are threefold:

- We identify moderation as a central bottleneck in multi-agent debate and formalize MAD as a meta-level regulation problem characterized by a monitoring-control-adjudication loop.
- We propose **Meta-Moderator**, a framework for *learnable moderation* that instantiates an explicit moderation policy to decide whether to continue deliberation or stop and commit to a final answer.
- Across multiple reasoning tasks, we show that learned meta-level regulation improves final answer accuracy compared to common MAD baselines.

2 Related Work

2.1 Multi-Agent Debate

Multi-agent debate (MAD) aims to improve LLM reasoning by having agents propose answers, critique one another, and refine arguments (Du et al., 2024). Subsequent work promotes divergent thinking, structured argumentation, and stronger verification in multimodal and retrieval-augmented settings (Liang et al., 2024; Zheng et al., 2024; Hu et al., 2025b).

However, recent analyses suggest that debate is not consistently beneficial. Gains are often attributable to diversity in initial answers rather than multi-round deliberation, and debate trajectories do not reliably increase correctness (Choi et al., 2025). Other work further reports conformity and bias amplification, where consensus-seeking dynamics suppress minority hypotheses and propagate shared errors (Cui et al., 2025). These findings imply that debate effectiveness depends not only on debaters' object-level reasoning, but also on how debate is monitored, budgeted, and aggregated into a final decision.

Most MAD pipelines rely on fixed debate lengths, agreement-based stopping, or untrained judges. Measurement-driven stopping moves beyond fixed rounds but still depends on predefined signals and hand-tuned criteria (Chang and Chang, 2025; Hu et al., 2025a). We address this gap by learning a moderation policy optimized for end-task accuracy.

2.2 Metacognition for LLMs

Recent work on metacognition in LLMs mainly focuses on self-monitoring and self-regulation (Gao

et al., 2024). Typical operationalizations include confidence estimation, uncertainty awareness, and failure prediction (Ji-An et al., 2025). Although such self-evaluative signals can be elicited, they are often miscalibrated and only weakly aligned with models' true knowledge limitations (Wang et al., 2025; Wang and Zhao, 2024). Complementary work frames meta-reasoning as a control mechanism over actions such as tool use, retrieval, or allocating additional computation (Li et al., 2025; Alazraki and Rei, 2025; De Sabbata et al., 2024).

In contrast, we study metacognition as coordination in multi-agent deliberation: a moderator monitors agents' externalized reasoning traces and regulates whether further debate is worthwhile. This motivates a learnable moderation policy for MAD.

2.3 Reinforcement Learning for LLMs

Reinforcement learning (RL) is increasingly used to shape LLM behavior beyond supervised fine-tuning (Wu et al., 2025b; Wan et al., 2025), including improving multi-step reasoning and learning decision policies for LLM-based agents (Shao et al., 2024; Guo et al., 2025; Chen et al., 2026). RL has also been applied to higher-level control, such as planning, tool selection, and allocating search or computation, making it a natural approach for learning meta-level policies over hierarchical reasoning processes (Jin et al., 2025; Chen et al., 2025; Qian et al., 2025; Wu et al., 2025a).

However, prior work largely optimizes a single agent's reasoning or tool-use policy, leaving the moderation policy, when to stop and how to adjudicate multi-agent evidence, underexplored. We address this gap by training a meta-level moderator with GRPO to directly optimize end-task accuracy.

3 Meta-Moderator

3.1 Problem Formulation

Let $\mathcal{X}$ denote the input space (e.g., questions) and $\mathcal{Y}$ the output space (e.g., free-form or multiple-choice answers). Given an instance $x \in \mathcal{X}$ with gold label $y^\star \in \mathcal{Y}$, we consider a multi-agent debate system composed of N LLM-based debaters $\mathcal{D} = \{d_1, \ldots, d_N\}$.

Agent responses. Each debater produces a textual response at round t, denoted by $r^t_{d_i}$. In our setting, $r^t_{d_i}$ contains both *reasoning* and a *final answer*; we denote the extracted answer by $y^t_{d_i} = f_{\text{ans}}(r^t_{d_i})$. At initialization ($t = 0$), each debater independently generates an initial response:

$$r^0_{d_i} \sim \texttt{Init}(x), \quad i \in \{1, \ldots, N\}, \tag{1}$$

where `Init` samples an initial response conditioned on x.

Debate dynamics. The debate runs for at most $T_{\max}$ rounds. At each round $t \geq 1$, debater d_i observes the set of other debaters' responses from the previous round,

$$\mathcal{R}^t_{-d_i} = \{\, r^{t-1}_{d_j} \mid j \in \{1, \ldots, N\},\, j \neq i \,\}, \tag{2}$$

and updates its response via a one-round debate operator `Debate`:

$$r^t_{d_i} \sim \texttt{Debate}\big(x; \mathcal{R}^t_{-d_i}\big). \tag{3}$$

All debaters update in parallel using the responses from the previous round.

Let $\mathcal{H}_t$ denote the debate transcript from initialization through round t:

$$\mathcal{H}_t = \{r^\tau_{d_i} \mid i \in \{1, \ldots, N\},\ \tau \in \{1, \ldots, t\}\}. \tag{4}$$

Decision layer. A moderator (decision layer) observes the debate and may terminate at any round $t \leq T_{\max}$. Formally, the system outputs

$$\hat{y} = \mathcal{F}(\mathcal{H}_t), \tag{5}$$

where t is the (data-dependent) stopping time chosen by the moderator. The function $\mathcal{F}$ abstracts the *decision layer* of debate: it can be instantiated as majority voting over $\{y^t_{d_i}\}_{i=1}^N$, or as a judge model conditioning on $\mathcal{H}_t$.

Objective. Our goal is to maximize end-task correctness of the overall debate system by improving the decision layer:

$$\max\ \mathbb{E}\big[\mathbf{1}\{\hat{y} = y^\star\}\big], \tag{6}$$

where the expectation is over task instances and the stochasticity of debater generation and debate updates. Next, we will parameterize $\mathcal{F}$ as a *learnable* moderator policy that decides when to stop and how to adjudicate, enabling the system to better exploit debate evidence when producing $\hat{y}$.

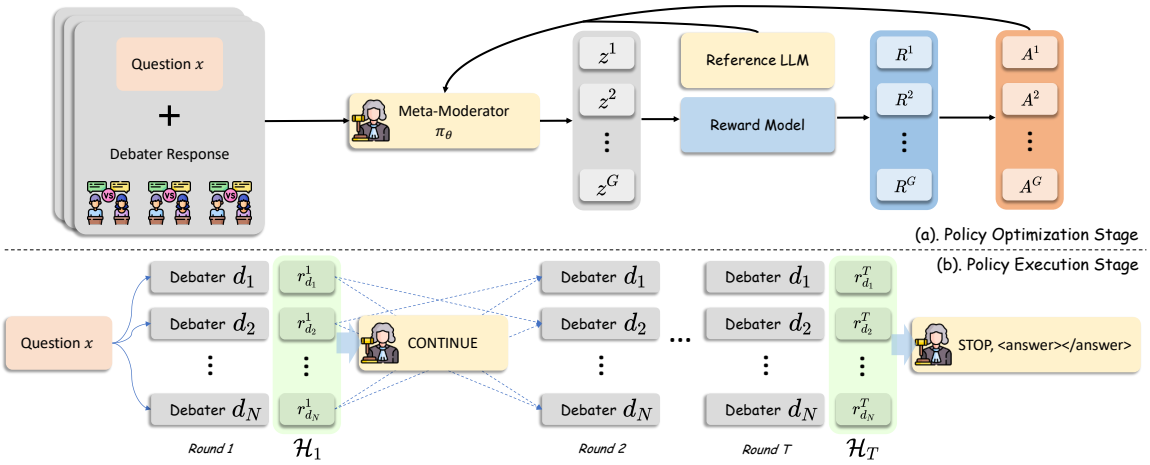

Figure 2: Overview of Meta-Moderator. The moderator monitors debate states and adjudicates a final answer when the debate stops; it is trained via outcome-driven policy optimization and used as a drop-in decision layer.

3.2 Meta-Moderator Policy

The formulation above separates the debate layer, which generates multi-agent evidence through interaction, from the decision layer $\mathcal{F}$, which maps debate evidence to a single prediction and an adaptive stopping point. Most existing MAD systems instantiate $\mathcal{F}$ with static heuristics or a prompted judge. In contrast, we parameterize $\mathcal{F}$ as a **Meta-Moderator**: a learnable meta-level policy that performs monitoring, control (adaptive stopping), and adjudication over the evolving debate.

Moderator state. At round t, the moderator observes a debate state

$$s_t = \phi(\mathcal{H}_t), \tag{7}$$

where $\phi(\cdot)$ specifies the moderator's input view. In general, ϕ may expose the full transcript or a compressed representation. In this work, we use a round-level view:

$$s_t = \phi\left(x, \{r^t_{d_i}\}_{i=1}^N\right), \tag{8}$$

constructed from the question and the current-round debater responses. Importantly, debaters' responses at round t may include summaries of prior discussion, allowing s_t to encode relevant history without explicitly concatenating $\mathcal{H}_t$.

Moderator policy and structured output. The moderator is implemented as a conditional language-model policy $\pi_\theta(\cdot \mid s_t)$ that produces a structured moderation output.

Monitoring. Monitoring refers to assessing whether the current debate state contains sufficient, non-redundant evidence for a reliable decision. Rather than introducing an explicit scalar monitor, we treat monitoring as an implicit internal signal represented by the policy when producing its control decision at round t. Formally, the control distribution $\pi_\theta(a_t \mid s_t)$ depends on s_t and thus implicitly captures whether additional deliberation is expected to be beneficial under the current debate evidence.

Control. Control uses the monitoring assessment to regulate deliberation depth. The moderator selects a control decision

$$a_t \in \{\texttt{CONTINUE}, \texttt{STOP}\}, \tag{9}$$

where CONTINUE allocates an additional debate round and STOP terminates deliberation. While moderation can in principle include richer interventions that shape the next-step information flow (e.g., requesting counterexamples or verification), we focus on this minimal control space to isolate the core question: *is further interaction beneficial given s_t?*

Adjudication. When the moderator decides to stop, it produces a single final prediction by aggregating multi-agent evidence in the current state. This is realized as conditional generation under the same policy:

$$\hat{y}_t \sim \pi_\theta(\cdot \mid s_t, a_t = \texttt{STOP}), \tag{10}$$

yielding a learnable instantiation of $\mathcal{F}$ that may adjudicate at an intermediate round rather than relying on a fixed horizon. Unlike surface-level consensus, learnable adjudication can weigh competing rationales and identify unresolved inconsistencies before committing to $\hat{y}_t$.

Overall, Meta-Moderator performs inter-agent meta-level regulation by monitoring debate utility, controlling whether to continue deliberation, and adjudicating a final answer from accumulated debate evidence.

3.3 Training Data Construction

We train Meta-Moderator from an offline dataset of debate states. For each training instance $x \in \mathcal{X}'$, we run a fixed multi-agent debate protocol to obtain a multi-round transcript $\mathcal{H}_t$ up to $T_{\max}$. We then extract, for each round t, the moderator-observable state

$$s_t = \phi\big(x, \{r_{d_i}^t \mid i \in \{1, \dots, N\}\}\big), \tag{11}$$

where $\phi(\cdot)$ is a deterministic formatting function that maps the question and the current-round debater responses into the moderator's input.

Each training example is a tuple $(s_t, y^\star, \eta_t)$. Here $\eta_t \in \{0, 1\}$ is an offline auxiliary signal indicating whether a correct candidate answer already appears among debaters at round t. We compute η_t by automatically extracting each debater's predicted answer from $r_{d_i}^t$, followed by dataset-specific normalization and matching against $y^\star$. This signal is used only to shape rewards during training; at inference time, Meta-Moderator observes only s_t and never accesses gold labels.

3.4 Reinforcing Meta-Cognition in Debate

We train the Meta-Moderator to exhibit *meta-cognition* over multi-agent debate: it must *monitor* whether the current debate state already contains sufficient evidence for a correct answer, and *control* the process by either continuing deliberation or committing to a final answer. Given a debate state s_t, the moderator generates a textual decision $z_t \sim \pi_\theta(\cdot \mid s_t)$. We constrain the output space to two executable forms:

$$z_t \in \Big\{\texttt{CONTINUE},\ \texttt{STOP} \,\|\, \langle\texttt{answer}\rangle \hat{y}_t \langle/\texttt{answer}\rangle\Big\}, \tag{12}$$

which is parsed into an action-answer pair $(a_t, \hat{y}_t)$.

Reward. We use three additive reward components to reinforce (i) protocol-compliant decisions, (ii) sufficiency monitoring for debate termination, and (iii) correctness upon commitment:

$$\begin{aligned} R(s_t, z_t) = R_{\text{fmt}}(z_t) + R_{\text{ctrl}}(s_t, a_t) \\ + R_{\text{ans}}(y^\star, a_t, \hat{y}_t). \end{aligned} \tag{13}$$

Format reward R_{fmt} enforces an *observable* and machine-parseable control protocol: `CONTINUE` alone, or `STOP` with a properly delimited $\langle\texttt{answer}\rangle$ span. We assign 1 for a valid format; for `STOP` without an answer span we give partial credit (0.5); otherwise 0.

Control reward R_{ctrl} reinforces *sufficiency monitoring*: the moderator should stop when the debate already contains a gold-matching proposal, and continue otherwise. We use an offline indicator $\eta_t \in \{0, 1\}$, where $\eta_t = 1$ means that at round t at least one debater proposes a gold-matching answer (estimated by automatic answer extraction and normalization). We reward `STOP` when $\eta_t = 1$, and reward `CONTINUE` when $\eta_t = 0$; otherwise the reward is 0. This gold-derived signal is used only for offline reward shaping during training and is never provided as an inference-time input.

Answer reward R_{ans} reinforces *commitment correctness* and is only applied at termination:

$$R_{\text{ans}}(y^\star, a_t, \hat{y}_t) = \begin{cases} \mathbb{I}[\hat{y}_t = y^\star], & a_t = \texttt{STOP}, \\ 0, & a_t = \texttt{CONTINUE}. \end{cases} \tag{14}$$

Dataset-specific matching is used for correctness.

GRPO optimization. For each state s_t, we sample G candidate decisions $\{z_t^{(i)}\}_{i=1}^G$ from the old policy and compute total rewards $\{R^{(i)}\}_{i=1}^G$. We construct group-relative advantages by normalizing rewards within the group:

$$A^{(i)} = \frac{R^{(i)} - \mu_{\{R\}}}{\sigma_{\{R\}}}, \tag{15}$$

and update π_θ with a clipped surrogate objective and a KL regularizer, following GRPO (Shao et al., 2024). More details are presented in Appendix C.

3.5 Inference-Time Moderation Protocol

At test time, Meta-Moderator is used as a drop-in regulator for any fixed set of debaters. Given an instance x, the system iterates rounds $t = 0, 1, \dots, T_{\max}$: (i) debaters produce responses $\{r_{d_i}^t\}_{i=1..N}$ under the debate protocol; (ii) the moderator state $s_t = \phi(x, \{r_{d_i}^t\}_{i=1..N})$ is constructed; (iii) the moderator samples or decodes a decision z_t from $\pi_\theta(\cdot \mid s_t)$ and parses it into $(a_t, \hat{y}_t)$. If $a_t = \texttt{CONTINUE}$, the debate proceeds to the next round. If $a_t = \texttt{STOP}$, the system terminates and returns $\hat{y}_t$ as the final answer. If no stop decision is made by $T_{\max}$, the system terminates at $T_{\max}$ using the moderator's final output.

Method	Use Moderator	GSM8K	AMC	MATH500	StrategyQA	MMLU
Single-agent			*Llama-3.1-8B-Instruct*			
Naive	-	22.20	7.23	3.80	68.20	57.00
Reflection (Shinn et al., 2023)	-	30.40	3.61	7.40	57.80	34.20
CoT (Wei et al., 2022)	-	**83.40**	16.87	33.60	67.40	62.60
Multi-Agent Debate			*Llama-3.1-8B-Instruct*			
+ Majority-Voting (Cui et al., 2025)	✘	79.00	25.20	27.80	66.40	61.80
+ Consensus (Du et al., 2024)	✘	81.80	22.89	27.20	**69.80**	**64.00**
+ LLM-as-Judge (Liang et al., 2024)	✔	80.00	**25.30**	33.60	63.00	62.00
+ **Meta-Moderator*** **(Ours)**	✔	79.80	19.28	**33.80**	68.80	61.60
+ **Meta-Moderator (Ours)**	✔	**83.80**	**27.71**	**35.80**	**72.00**	**67.20**
Single-agent			*Qwen-2.5-7B-Instruct*			
Naive	-	20.60	15.66	4.00	65.20	69.60
Reflection (Shinn et al., 2023)	-	89.20	20.48	18.80	69.80	68.60
CoT (Wei et al., 2022)	-	90.20	21.69	15.40	**70.00**	61.20
Multi-Agent Debate			*Qwen-2.5-7B-Instruct*			
+ Majority-Voting (Cui et al., 2025)	✘	90.80	26.51	14.40	67.20	68.80
+ Consensus (Du et al., 2024)	✘	90.60	26.51	14.40	67.20	68.80
+ LLM-as-Judge (Liang et al., 2024)	✔	90.80	**43.37**	**19.00**	67.40	69.80
+ **Meta-Moderator*** **(Ours)**	✔	**90.80**	39.76	13.80	67.60	**70.60**
+ **Meta-Moderator (Ours)**	✔	**91.20**	**42.96**	**29.40**	**70.40**	**69.80**

Table 1: Overall results of **Meta-Moderator** and baselines on five benchmarks under two backbone settings.**Meta-Moderator*** denotes the *untrained* (prompted) moderator. Blue and light blue denote the best and second-best performances, respectively. All methods are evaluated under the same debate protocol and decoding configuration.

4 Experiments

4.1 Experimental Settings

Baselines. We compare **Meta-Moderator** with single-agent prompting and multi-agent debate baselines under a unified setup. For single-agent methods, we include **Naive**, **CoT** (Wei et al., 2022), and **Reflection** (Shinn et al., 2023). For MAD, we use a standard protocol where debaters iteratively revise responses conditioned on peers' previous-round outputs (Du et al., 2024), and vary only the decision layer: **Majority Voting** (Cui et al., 2025), **Consensus** (Du et al., 2024), or **LLM-as-a-Judge** (Liang et al., 2024).

Datasets and Metrics. We evaluate on five benchmarks covering mathematical reasoning, logical reasoning, and general knowledge: **GSM8K** (Cobbe et al., 2021), **AMC**, **MATH500** (Hendrycks et al., 2021), **StrategyQA** (Geva et al., 2021), and **MMLU** (Hendrycks et al., 2020). For large test sets, we randomly sample 500 instances; for smaller benchmarks, we use the full test set (e.g., **AMC** has 83 instances). We report **accuracy**. To train Meta-Moderator, we sample 500 instances each from the training splits of **GSM8K** and **MMLU** to construct per-round moderation decisions.

Implementation Details. We instantiate both debaters and the moderator with instruction-tuned LLMs, using Llama-3.1-8B-Instruct and Qwen-2.5-7B-Instruct as backbones (Dubey et al., 2024; Bai et al., 2023). Unless otherwise specified, we use N=2 debaters and a maximum debate budget of $T_{\max}$=5 rounds. All methods use the same debater protocol and decoding configuration; only the decision layer differs. Additional backbone settings are reported in Appendix H.

4.2 Main Results

We evaluate **Meta-Moderator** against single-agent prompting and multi-agent debate baselines across five benchmarks. As shown in Table 1, Meta-Moderator generally improves over common MAD decision layers and achieves the strongest overall performance across the evaluated settings, supporting the value of *learnable* debate regulation. By contrast, heuristic strategies (e.g., majority voting and consensus-based stopping) are often brittle and can underperform strong single-agent CoT prompting, indicating that additional interaction alone does not reliably yield accuracy gains.

Meta-Moderator is particularly beneficial where heuristic aggregation is brittle, including StrategyQA, MATH500, and MMLU, while prompted LLM-as-a-judge remains competitive on AMC.

Method	GSM8K	AMC	MATH500	StrategyQA	MMLU
Multi-Agent Debate	*Llama-3.1-8B-Instruct & Llama-3.2-3B-Instruct*				
+ LLM-as-Judge (Liang et al., 2024)	**79.80**	**26.51**	**29.60**	**64.80**	**62.80**
+ **Meta-Moderator*** **(Ours)**	79.40	**26.51**	27.80	60.60	61.40
+ **Meta-Moderator (Ours)**	**83.60**	22.89	**28.00**	**71.20**	**65.60**

Table 2: Results with decoupled debater and moderator backbones: debaters use Llama-3.1-8B-Instruct while the moderator uses Llama-3.2-3B-Instruct.

Method	GSM8K	AMC	MATH500	StrategyQA	MMLU
Multi-Agent Debate	*Llama-3.1-8B-Instruct*				
+ Majority-Voting (Cui et al., 2025)	77.40	18.07	23.60	68.20	62.80
+ Consensus (Du et al., 2024)	**79.00**	20.48	23.80	**72.00**	63.40
+ LLM-as-Judge (Liang et al., 2024)	78.20	**24.10**	**31.60**	69.60	**63.80**
+ **Meta-Moderator*** **(Ours)**	78.40	**24.10**	29.40	67.80	63.60
+ **Meta-Moderator (Ours)**	**83.80**	22.89	**35.60**	**72.60**	**66.00**

Table 3: Results with more debaters (N=3) under the same debate budget.

Relative to voting-based aggregation and prompted LLM-as-a-judge, Meta-Moderator more reliably converts useful debate evidence into final-answer accuracy, suggesting that learning *when* to stop and *how* to adjudicate helps turn debate into gains. The learned moderator is also robust across backbone families and model strengths: while heuristic MAD variants can degrade through error propagation and conformity, Meta-Moderator remains competitive with strong single-agent baselines, implying that meta-level regulation can mitigate imperfect object-level reasoning.

Finally, the untrained variant **Meta-Moderator*** is consistently weaker than the trained moderator and can underperform heuristic alternatives on multiple datasets. This gap suggests that effective moderation is not merely a byproduct of role prompting; rather, it requires learning a stable policy via outcome-driven optimization.

4.3 Generalization

We evaluate the generalization of Meta-Moderator along three dimensions.

Cross-task generalization. Trained only on GSM8K and MMLU, Meta-Moderator improves accuracy on unseen benchmarks (AMC, MATH500, and StrategyQA; Table 1), suggesting that it learns transferable signals of debate utility and decision readiness rather than dataset-specific heuristics.

Cross-backbone generalization. To decouple debater and moderator backbones, we pair Llama-3.1-8B-Instruct debaters with a smaller Llama-3.2-3B-Instruct moderator (Table 2). Meta-Moderator remains effective and outperforms prompted LLM-as-a-Judge on several benchmarks, suggesting that gains come from the learned policy rather than moderator scale.

Robustness to the number of debaters. Increasing the number of debaters from N=2 to N=3 while keeping the backbone and debate budget fixed (Table 3), Meta-Moderator remains competitive on most benchmarks, showing robustness beyond a specific debate size.

Overall, these results support moderation as a policy-like capability that transfers across tasks, heterogeneous system instantiations, and debate group sizes.

4.4 Ablation: Decoupling Stopping and Adjudication

To disentangle adaptive stopping from learned adjudication, we evaluate two decoupled variants under the same debaters, protocol, budget $T_{\max}$, and decoding: (i) *Adjudication-only* fixes the debate length to 3 and uses Meta-Moderator only for the final answer; (ii) *Stopping-only* uses Meta-Moderator for `CONTINUE/STOP` but applies a fixed heuristic aggregator upon stopping. Table 4 shows that neither component alone consistently recovers the full model's gains, while combining adaptive

Method	GSM8K	AMC	MATH500	StrategyQA	MMLU
Multi-Agent Debate			*Llama-3.1-8B-Instruct*		
+ **Meta-Moderator*** **(Ours)**	79.80	19.28	33.80	68.80	61.60
+ **Meta-Moderator (Ours)**	**83.80**	**27.71**	**35.80**	**72.00**	**67.20**
+ Adjudication Only	81.80	24.10	28.00	62.00	63.60
+ Adaptive Stopping Only (Majority-Voting)	79.40	24.10	29.40	66.20	62.40
+ Adaptive Stopping Only (LLM-as-Judge)	80.40	27.71	33.60	66.80	62.20

Table 4: Ablation separating adaptive stopping and learned adjudication under the same debate protocol.

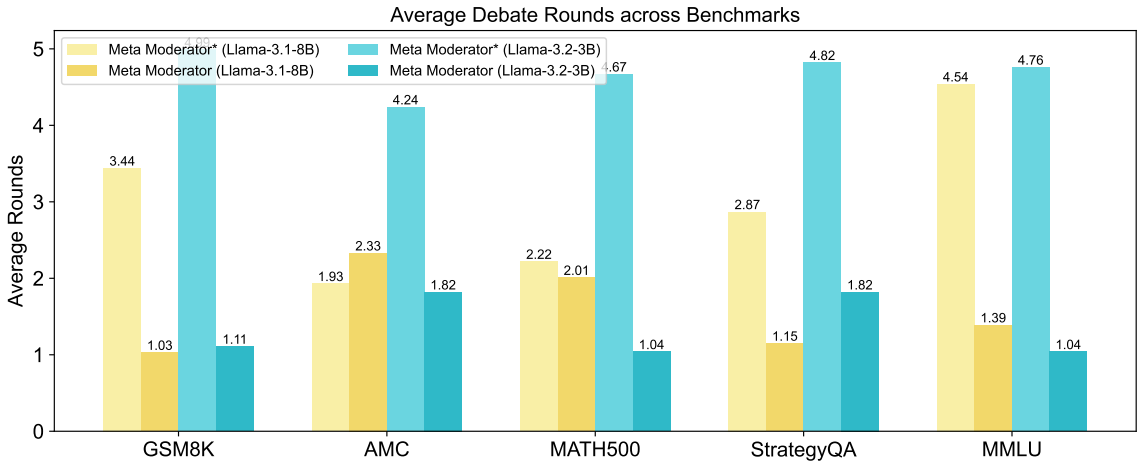

Figure 3: Average debate rounds across benchmarks, comparing the untrained Meta-Moderator* and the trained Meta-Moderator under the same maximum debate budget.

	GSM8K	AMC	MATH	StratQA	MMLU
Meta-Moderator*					
$y^* \in \mathcal{H}_t(\uparrow)$	**432**	21	180	**386**	**385**
$\hat{y} \neq y^* \ldots (\downarrow)$	33	8	25	42	77
Meta-Moderator					
$y^* \in \mathcal{H}_t(\uparrow)$	431	**24**	**187**	369	351
$\hat{y} \neq y^* \ldots (\downarrow)$	**12**	**1**	**8**	**9**	**15**

Table 5: Analysis of debate coverage and adjudication.

stopping with learned adjudication performs best across all five benchmarks.

4.5 Analysis of Debate Coverage and Adjudication

To understand why Meta-Moderator helps, we decompose outcomes into *coverage*: whether a correct hypothesis ever appears; *adjudication*: whether it is selected once available. For each instance, let $\mathcal{H}_t$ be the set of debaters' answers observed up to the moderator's stopping time t, and let y^* denote the gold answer. We count *oracle availability* when the gold answer appears in the debate, $y^* \in \mathcal{H}_t$, and *mis-aggregation* when the final prediction is still incorrect despite oracle availability, $\hat{y} \neq y^* \wedge y^* \in \mathcal{H}_t$.

As shown in Table 5, training does not uniformly increase oracle availability: coverage improves on AMC and MATH500, remains nearly unchanged on GSM8K, and decreases on StrategyQA and MMLU. In contrast, the trained Meta-Moderator consistently reduces mis-aggregation across all five benchmarks (from 33 to 12 on GSM8K, 8 to 1 on AMC, 25 to 8 on MATH500, 42 to 9 on StrategyQA, and 77 to 15 on MMLU). Since $\mathcal{H}_t$ is defined at the method-specific stopping time, lower coverage may partly reflect earlier termination rather than poorer hypothesis generation. These results suggest that the accuracy gains primarily arise from more reliable finalization when useful debate evidence is available, rather than from uniformly increasing debate coverage.

4.6 Analysis of Debate Rounds

We analyze stopping behavior by reporting the average number of debate rounds used by Meta-Moderator and the untrained Meta-Moderator* in Figure 3 (maximum budget $T_{\max}$=5). Meta-

Moderator* tends to over-deliberate, often consuming nearly the full budget, especially with the smaller backbone. After training, Meta-Moderator uses substantially fewer rounds across benchmarks while remaining adaptive: it reduces redundant debate on easier instances but allocates more rounds when helpful on harder benchmarks (Figure 3).

Overall, Meta-Moderator learns adaptive debate budgeting rather than a fixed stopping heuristic.

5 Conclusion

We identify moderation as a central bottleneck in multi-agent debate and argue that effective debate requires a meta-cognitive loop of monitoring, control, and adjudication. We therefore propose **Meta-Moderator**, a learnable framework that regulates deliberation by deciding at each round whether to continue debating or to stop and commit to a final answer. The moderation policy is trained independently of debaters via outcome-driven reinforcement learning. Across five reasoning benchmarks and two backbone settings, Meta-Moderator consistently improves accuracy over common debate decision layers and generalizes across tasks, heterogeneous moderator/debater backbones, and debate group sizes. Analyses further show that it budgets debate adaptively, increases the likelihood that a correct hypothesis emerges, and reduces misaggregation once it does, enabling more reliable and efficient deliberation.

Limitations

A key limitation of our approach is the additional training and inference cost of learning and running a dedicated moderator in conjunction with multi-agent debate. Although Meta-Moderator often reduces debate rounds at test time, it still requires extra model calls for monitoring and adjudication, and outcome-driven RL training further increases compute relative to purely prompted decision layers.

Moreover, our moderator has a restricted control interface due to our data construction: we only consider the minimal `continue`/`stop` action space. While this design isolates the effect of learnable stopping and adjudication, it does not capture richer interventions that may further improve debate quality.

Future work could reduce overhead by distilling the moderator to a smaller model, amortizing monitoring signals, or pruning agents adaptively. It would also be valuable to expand the moderation action space and develop objectives and evaluation suites beyond final-answer accuracy, enabling moderators that learn not only *when* to stop, but also *how* to steer deliberation.

Acknowledgement

This work was partially supported by NSFC/RGC Joint Research Scheme (N_PolyU5179/25), Hong Kong RGC (PolyU25600624), Innovation Technology Fund (ITS/052/23MX, PRP/009/22FX), industrial sponsors and RMGS (P0045948, P0048183, P0048191, P0046453, P0060272). This work was also supported by the National Natural Science Foundation of China (Grant No.: 62372314). The experimental part of this work was supported by The Centre for Large AI Models (CLAIM) of The Hong Kong Polytechnic University.

References

Rakefet Ackerman and Valerie A Thompson. 2017. Meta-reasoning: Monitoring and control of thinking and reasoning. *Trends in cognitive sciences*, 21(8):607–617.

Lisa Alazraki and Marek Rei. 2025. Meta-reasoning improves tool use in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 7885–7897, Albuquerque, New Mexico. Association for Computational Linguistics.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Linden J Ball and Beth H Richardson. 2025. Metareasoning: Theoretical and methodological developments.

Jonas Becker. 2024. Multi-agent large language models for conversational task-solving. *arXiv preprint arXiv:2410.22932*.

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

Edward Y Chang and Ethan Y Chang. 2025. Multi-agent collaborative intelligence: Dual-dial control for reliable llm reasoning. *arXiv preprint arXiv:2510.04488*.

Mingyang Chen, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, and

Weipeng Chen. 2025. Research: Learning to reason with search for llms via reinforcement learning. *Preprint*, arXiv:2503.19470.

Xiuwei Chen, Wentao Hu, Hanhui Li, Yongxin Wang Jun Zhou, Zisheng Chen, Meng Cao, Yihan Zeng, Kui Zhang, Yu-Jie Yuan, Jianhua Han, Hang Xu, and Xiaodan Liang. 2026. Syncloop: A multimodal dual-loop framework for self-improving mathematical reasoning. *Preprint*, arXiv:2507.16518.

Hyeong Kyu Choi, Jerry Zhu, and Sharon Li. 2025. Debate or vote: Which yields better decisions in multiagent large language models? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

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. *arXiv preprint arXiv:2110.14168*.

Yu Cui, Hang Fu, Haibin Zhang, Licheng Wang, and Cong Zuo. 2025. Free-mad: Consensus-free multi-agent debate. *arXiv preprint arXiv:2509.11035*.

C Nicolò De Sabbata, Theodore R Sumers, Badr AlKhamissi, Antoine Bosselut, and Thomas L Griffiths. 2024. Rational metareasoning for large language models. *arXiv preprint arXiv:2410.05563*.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Peizhong Gao, Ao Xie, Shaoguang Mao, Wenshan Wu, Yan Xia, Haipeng Mi, and Furu Wei. 2024. Meta reasoning for large language models. *arXiv preprint arXiv:2406.11698*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

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. *arXiv preprint arXiv:2501.12948*.

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

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Mengze Hong, Wailing Ng, Chen Jason Zhang, and Di Jiang. 2025. Qualbench: Benchmarking chinese llms with localized professional qualifications for vertical domain evaluation. *Preprint*, arXiv:2505.05225.

Mengze Hong, Xia Zeng, Zeyang Lei, Sheng Wang, Chen Jason Zhang, Di Jiang, Taiming Fu, Jinfeng Huang, Mengqiao Liu, Qinghe Chang, Haosheng Zou, Qiongyi Zhou, Sijun He, Simonjmdeng, Haojing Huang, Zijian Li, Lucas Mu Li, Fubao Zhang, Mona Zhou, Wei Ma, Yuan Hua, Qi Zhu, Shuo Jiang, Chenxuan Ma, Yuanmeng Zhang, Jian Song, Minlong Peng, Di Liang, and Davey Chen. 2026. Uxbench: Benchmarking user experience in ai assistants. *Preprint*, arXiv:2606.09570.

Tianyu Hu, Zhen Tan, Song Wang, Huaizhi Qu, and Tianlong Chen. 2025a. Multi-agent debate for LLM judges with adaptive stability detection. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Wentao Hu, Wengyu Zhang, Yiyang Jiang, Chen Jason Zhang, Xiaoyong Wei, and Li Qing. 2025b. Removal of hallucination on hallucination: Debate-augmented RAG. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15839–15853, Vienna, Austria. Association for Computational Linguistics.

Li Ji-An, Hua-Dong Xiong, Robert Wilson, Marcelo G Mattar, and Marcus K. Benna. 2025. Language models are capable of metacognitive monitoring and control of their internal activations. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.

Wenjun Li, Dexun Li, Kuicai Dong, Cong Zhang, Hao Zhang, Weiwen Liu, Yasheng Wang, Ruiming Tang, and Yong Liu. 2025. Adaptive tool use in large language models with meta-cognition trigger. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13346–13370, Vienna, Austria. Association for Computational Linguistics.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17889–17904, Miami, Florida, USA. Association for Computational Linguistics.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. 2025. Toolrl: Reward is all tool learning needs. *arXiv preprint arXiv:2504.13958*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652.

Zhuoyue Wan, Yuanfeng Song, Shuaimin Li, Chen Jason Zhang, and Raymond Chi-Wing Wong. 2025. Datavist5: A pre-trained language model for jointly understanding text and data visualization. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*, pages 1704–1717.

Guoqing Wang, Wen Wu, Guangze Ye, Zhenxiao Cheng, Xi Chen, and Hong Zheng. 2025. Decoupling metacognition from cognition: a framework for quantifying metacognitive ability in llms. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press.

Yuqing Wang and Yun Zhao. 2024. Metacognitive prompting improves understanding in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1914–1926, Mexico City, Mexico. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Xiao-Yong Wei and Zhen-Qun Yang. 2013. Coaching the exploration and exploitation in active learning for interactive video retrieval. *IEEE Transactions on Image Processing*, 22(3):955–968.

Junlin Wu, Xianrui Zhong, Jiashuo Sun, Bolian Li, Bowen Jin, Jiawei Han, and Qingkai Zeng. 2025a. Structure-r1: Dynamically leveraging structural knowledge in llm reasoning through reinforcement learning. *arXiv preprint arXiv:2510.15191*.

Qian Wu, Zheyao Gao, Longfei Gou, and Qi Dou. 2025b. DDxTutor: Clinical reasoning tutoring system with differential diagnosis-based structured reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30934–30957, Vienna, Austria. Association for Computational Linguistics.

Shaowei Zhang and Deyi Xiong. 2025. Debate4MATH: Multi-agent debate for fine-grained reasoning in math. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16810–16824, Vienna, Austria. Association for Computational Linguistics.

Changmeng Zheng, Dayong Liang, Wengyu Zhang, Xiao-Yong Wei, Tat-Seng Chua, and Qing Li. 2024. A picture is worth a graph: A blueprint debate paradigm for multimodal reasoning. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 419–428.

A More Implementation Details

For large benchmarks, we evaluate all methods on the same fixed subset of 500 instances sampled once with a shared random seed, ensuring identical evaluation sets across methods. We use deterministic decoding (temperature=0) for all model components.

For inference, Qwen-3-30B-Instruct was run on two NVIDIA RTX A6000 GPUs, while all other models used one RTX A6000. For GRPO training, the Llama-3.1-8B-Instruct, Llama-3.2-3B-Instruct, and Qwen-2.5-7B-Instruct models were each trained on one RTX A6000.

B Algorithm

Algorithm 1 describes our inference-time moderation protocol for multi-agent debate.

C Learning Meta-Moderator with Reinforcement Learning

The GRPO update maximizes a PPO-style clipped surrogate objective with a KL regularizer:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{s_t \sim P(S),\, \{z_t^{(i)}\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot|s_t)} \Bigg[\frac{1}{G} \sum_{i=1}^{G} \min \Big(\rho_i(\theta)\, A^{(i)}, \; \text{clip}\big(\rho_i(\theta), 1-\epsilon, 1+\epsilon\big) A^{(i)} \Big) - \beta D_{\text{KL}}\left(\pi_\theta(\cdot \mid s_t) \| \pi_{\text{ref}}(\cdot \mid s_t)\right) \Bigg] \tag{16}$$

where $\rho_i(\theta) = \frac{\pi_\theta(z_t^{(i)}|s_t)}{\pi_{\theta_{\text{old}}}(z_t^{(i)}|s_t)}$ is the importance ratio. The clipping operator constrains the update magnitude within $[1-\epsilon, 1+\epsilon]$ to stabilize learning, while the KL term regularizes the policy toward a reference model π_{ref} with strength β. In all experiments, we use a group size of G=8, a clipping ratio of ϵ=0.2, and a KL coefficient of β=0, and optimize the moderator for 3000 gradient-update steps. This group-relative, clipped update yields stable policy improvements under sparse, outcome-driven rewards and directly optimizes the moderator's regulation behavior.

D Prompt Templates

This section presents the prompt templates used in our implementation for both the debaters and the Meta-Moderator. We adopt a lightweight tag-based protocol for parsing: each debater ends with

Algorithm 1 Meta-Moderator Inference-Time Moderation for Multi-Agent Debate

Require: Input x, debaters $\mathcal{D} = \{d_1, \ldots, d_N\}$, init operator Init, debate operator Debate, moderator policy π_θ, state formatter $\phi(\cdot)$, max rounds $T_{\max}$.

Ensure: Final answer $\hat{y}$.

Initialize round $t \leftarrow 0$
Initialize transcript $\mathcal{H} \leftarrow \emptyset$
for $i = 1$ to N **do**
  Initial response $r_{d_i}^0 \sim \text{Init}(x)$
end for
Update transcript $\mathcal{H} \leftarrow \mathcal{H} \cup \{r_{d_i}^0\}_{i=1}^N$
while $t \leq T_{\max}$ **do**
  Construct moderator state $s_t \leftarrow \phi\big(x, \{r_{d_i}^t\}_{i=1}^N\big)$
  Moderator decision $z_t \sim \pi_\theta(\cdot \mid s_t)$
  Parse $(a_t, \hat{y}_t) \leftarrow \text{ParseDecision}(z_t)$ ▷ $z_t \in \{$`CONTINUE`, `STOP` ‖ `<answer>`$\hat{y}_t$`</answer>`$\}$
  if $a_t =$ `STOP` **then**
    return $\hat{y}_t$
  end if
  if $t = T_{\max}$ **then** ▷ No stop decision within budget; terminate with final moderator output
    $\hat{y} \leftarrow \text{FinalizeAtBudget}(s_t, \pi_\theta)$
    return $\hat{y}$
  end if
  ▷ One-round parallel debate update
  $t \leftarrow t + 1$
  for $i = 1$ to N **in parallel do**
    Observe peers $\mathcal{R}_{-d_i}^t \leftarrow \{r_{d_j}^{t-1} \mid j \neq i\}$
    Update response $r_{d_i}^t \sim \text{Debate}\big(x; \mathcal{R}_{-d_i}^t\big)$
  end for
  Update transcript $\mathcal{H} \leftarrow \mathcal{H} \cup \{r_{d_i}^t\}_{i=1}^N$
end while

<answer>...</answer>, and the Meta-Moderator outputs either CONTINUE or STOP with an optional <answer> span.

D.1 Prompt for Meta-Moderator (STOP/CONTINUE)

Prompt for Meta-Moderator (STOP/CONTINUE)

You are a moderator in a debate. Your task is to decide whether to CONTINUE or STOP the debate based on the arguments presented by debaters.
If STOP, output the STOP first and then the final answer between <answer> and </answer> tags with no explanations or additional text.
If CONTINUE, output the CONTINUE only.

Question:
{question}

Other agents' responses:
{agents_responses}

D.2 Prompt for Meta-Moderator (Forced Answer at $T_{\max}$)

Prompt for Meta-Moderator (Forced Answer at $T_{\max}$)

You are a moderator in a debate competition. Your task is to determine the correct final answer based on the arguments presented by debaters.
Output only the final answer with no explanations or additional text.

Question:
{question}

Other agents' responses:
{agents_responses}

D.3 Debater Prompts

Prompt for Debater Agent (Initial Round)

Answer the question based on your own knowledge. Explain your answer step by step, and put the answer between <answer> and </answer> tags at the end of your response.
{optional confidence instruction}

Question:
{question}

Prompt for Debater Agent (Debate Round)

I will give the answers and arguments to this question from other agents. Use their solution as additional advice; note that they may be wrong. If you disagree with the other agents, please give your reasons and answer; otherwise, revise your previous answer. Explain your answer step by step, and put the answer between <answer> and </answer> tags at the end of your response.

Question: {question}
Other agents' responses:
{other_agents_responses}

E Training Template

Training data generation. We construct the Meta-Moderator training set by first generating multi-agent debate traces using the standard Multi-Agent Debate (MAD) pipeline, with Llama3.1-8B-Instruct as the underlying generator. For each question, MAD produces debater responses per round. In our data construction, we only retain two debaters ($N = 2$) and use their responses to form the debate state; any additional agent outputs are discarded. Each training instance is a single-round snapshot consisting of the original question and the two agents' responses at that round, paired with a Meta-Moderator decision label (CONTINUE or STOP) and (when applicable) a target final answer.

Template. We train the Meta-Moderator using a single prompt template constructed from the current-round debate state. The input consists of: (i) an instruction that constrains the output space to CONTINUE or STOP (with <answer>...</answer>

when stopping), and (ii) the question plus the two debaters' responses at the current round.

Training Template for Meta-Moderator

You are a moderator in a debate. Your task is to decide whether to CONTINUE or STOP the debate based on the arguments presented by debaters. If STOP, output the STOP first and then the final answer between <answer> and </answer> tags with no explanations or additional text. If CONTINUE, output the CONTINUE only.

Question: {question}
Other agents' responses:
Agent 1: {response_1}
Agent 2: {response_2}

F Efficiency and Cost Accounting

We report the training and inference overhead of Meta-Moderator for completeness.

F.1 Training cost

Meta-Moderator requires constructing offline debate trajectories and performing RL optimization. The total generation cost for constructing training data is approximately 11.25M tokens (9.78M input + 1.46M output). The moderator is trained once and reused across tasks and backbone configurations; debaters are not fine-tuned.

F.2 Inference cost, latency, and memory

We additionally report inference token usage for all methods in Table 7 to quantify the accuracy–compute trade-off. At inference time, all methods share the same debate protocol, backbone, maximum round budget (T_{max}), and decoding configuration; the only difference is the decision layer. Meta-Moderator introduces one additional moderator call per round. However, it typically reduces the average number of debate rounds (Figure 3), thereby lowering the number of debater-side generation tokens. As a result, the net inference token cost is comparable and is often reduced due to adaptive early stopping. The memory overhead is limited to loading one additional moderator model, which can be smaller than the debaters (as demonstrated by the decoupled debater-moderator setting in Table 2). From an implementation perspective, our framework only replaces the decision layer and does not modify the debate protocol itself. Importantly, our claim is not that Meta-Moderator is universally cheaper than all prompt-based baselines, but that it improves the accuracy-compute trade-off by allocating deliberation adaptively.

G Trained Single-Agent GRPO Baseline

To isolate whether performance gains come from GRPO training alone, we implement a trained single-agent baseline in Table 6. We use the same backbone (Llama-3.1-8B-Instruct) and the same GRPO optimizer. The reward is based on final-answer correctness with dataset-specific matching. We train on the same training splits used to construct Meta-Moderator training data (500 instances each from GSM8K and MMLU).

Overall, Naive+GRPO can improve over naive prompting on some benchmarks, but it remains below Meta-Moderator. This suggests that the gains are not explained by GRPO training alone, and that meta-level regulation of multi-agent debate (adaptive stopping and adjudication) provides additional benefits.

H Additional Results with Different Backbone Settings

Shown in Table 8, we report additional backbone settings for the debate system beyond the main results. We evaluate three settings: (i) we use **Llama-3.2-3B-Instruct** (Dubey et al., 2024) as both the debaters and the moderator; (ii) we use **Qwen-2.5-14B-Instruct** (Bai et al., 2023) as the debaters, while using **Llama-3.1-8B-Instruct** as the moderator; and (iii) we use **Qwen-3-30B-Instruct** (Bai et al., 2023) as the debaters, while using **Llama-3.1-8B-Instruct** as the moderator. Following our main setup, we construct the moderator state using responses from **two** debaters ($N = 2$). All numbers are accuracies (%).

Discussion. Across both settings, using a learned Meta-Moderator consistently improves over heuristic aggregation baselines (Majority-Voting / Consensus) and is competitive with (or better than) an LLM-as-Judge. Notably, in the 7B setting, Meta-Moderator yields clear gains on MATH500 and StrategyQA compared to heuristic aggregation, indicating better debate termination and answer selection.

Method	GSM8K	AMC	MATH500	StrategyQA	MMLU
Naive	22.20	7.23	3.80	68.20	57.00
Naive+GRPO	81.60	19.28	4.60	68.80	62.80
Meta-Moderator (Ours)	**83.80**	**27.71**	**35.80**	**72.00**	**67.20**

Table 6: Trained single-agent GRPO baseline (Naive+GRPO) under Llama-3.1-8B-Instruct.

Method	GSM8K	AMC	MATH500	StrategyQA	MMLU
Single-agent			*Llama-3.1-8B-Instruct*		
Naive	147.89	210.96	252.53	87.38	175.95
Reflection	2408.09	3573.96	3223.21	2118.11	2763.92
CoT	336.06	614.43	516.29	354.07	485.06
Multi-Agent Debate			*Llama-3.1-8B-Instruct*		
+ Majority-Voting	10972.33	17785.76	16442.11	18460.87	13222.93
+ Consensus	10034.28	16890.30	15809.21	18740.09	12238.40
+ LLM-as-Judge	9401.76	15335.23	14399.28	14614.69	11084.35
+ **Meta-Moderator***	12329.51	8598.46	10060.58	7008.93	22350.50
+ **Meta-Moderator**	3522.08	12802.40	10668.56	4192.31	4017.78

Table 7: Inference token statistics (average tokens per instance; input+output) across five benchmarks under the same debate protocol, backbone, round budget, and decoding configuration. Token counts include all LLM calls used by each method (e.g., debaters and the decision component).

I Case Study

We present a qualitative case study illustrating how moderation failures can negate the potential benefits of multi-agent debate, and how a trained moderator mitigates these issues. Figure 4 shows an instance where the untrained **Meta-Moderator*** fails to stop in time and eventually commits to an incorrect answer due to late-round drift, whereas the trained **Meta-Moderator** stops earlier and selects the correct hypothesis.

Oracle availability but mis-aggregation. In the early rounds, both debaters independently propose the correct option (`B. friendship`), meaning the gold hypothesis is already present in the debate history (i.e., oracle availability holds, $y^* \in \mathcal{H}_t$). However, **Meta-Moderator*** chooses to `continue`, spending additional rounds despite diminishing marginal utility. As the debate proceeds, one agent introduces a spurious line of reasoning (arguing that none of the options perfectly match and then selecting `C. luxury` by elimination), which shifts the discussion away from the previously consistent correct hypothesis. The moderator then `stops` and commits to the wrong answer, constituting a mis-aggregation error despite oracle availability ($\hat{y} \neq y^* \wedge y^* \in \mathcal{H}_t$).

Why learned moderation helps. This example highlights two failure modes of untrained moderation: (i) *over-deliberation*, where redundant interaction consumes budget after a correct hypothesis has emerged, and (ii) *problem drift*, where additional rounds introduce distracting arguments that increase the risk of mis-aggregation. In contrast, the trained Meta-Moderator recognizes early decision readiness and terminates the debate before drift dominates, selecting `B. friendship`. This behavior is consistent with our quantitative findings that learning a moderation policy reduces unnecessary rounds and lowers mis-aggregation once informative hypotheses appear (Sections 4.5 and Figure 3).

Method	Use Moderator	GSM8K	AMC	MATH500	StrategyQA	MMLU
Single-agent			*Llama-3.2-3B-Instruct*			
Naive	-	4.80	3.61	2.60	66.00	32.80
Reflection	-	17.80	2.41	16.20	47.00	39.00
CoT	-	74.20	14.46	25.20	57.60	40.00
Multi-Agent Debate			*Llama-3.2-3B-Instruct*			
+ Majority-Voting	✘	28.80	3.61	13.60	44.80	32.40
+ Consensus	✘	33.40	2.41	16.20	47.00	39.00
+ LLM-as-Judge	✔	66.20	7.23	17.40	24.40	41.40
+ **Meta-Moderator***	✔	63.40	8.43	17.40	44.80	39.80
+ **Meta-Moderator**	✔	74.60	18.07	23.40	56.40	42.40
Single-agent			*Qwen-2.5-14B-Instruct*			
Naive	-	28.80	9.64	7.20	69.80	67.40
Reflection	-	92.40	33.73	47.60	77.60	67.00
CoT	-	92.40	16.87	44.00	77.00	64.20
Multi-Agent Debate	**Debaters:** *Qwen-2.5-14B-Instruct* + **Moderator:** *Llama-3.1-8B-Instruct*					
+ Majority-Voting	✘	92.00	44.58	55.00	77.60	67.80
+ Consensus	✘	92.00	44.58	55.60	77.40	68.20
+ LLM-as-Judge	✔	92.20	44.58	53.56	76.80	68.60
+ **Meta-Moderator***	✔	92.40	42.71	52.20	77.60	69.40
+ **Meta-Moderator**	✔	92.60	44.58	56.00	77.80	69.00
Single-agent			*Qwen-3-30B-Instruct*			
Naive	-	52.20	13.25	14.20	73.00	78.40
Reflection	-	93.00	7.23	33.00	73.00	76.60
CoT	-	93.80	13.25	37.80	76.00	76.80
Multi-Agent Debate	**Debaters:** *Qwen-3-30B-Instruct* + **Moderator:** *Llama-3.1-8B-Instruct*					
+ Majority-Voting	✘	93.00	27.71	30.40	73.40	75.20
+ Consensus	✘	93.40	24.10	28.20	69.80	68.60
+ LLM-as-Judge	✔	93.40	36.14	47.00	73.40	78.40
+ **Meta-Moderator***	✔	92.80	36.14	47.40	72.40	76.60
+ **Meta-Moderator**	✔	93.60	39.76	59.60	75.60	79.80

Table 8: Additional results under different backbone choices for debaters and moderators.

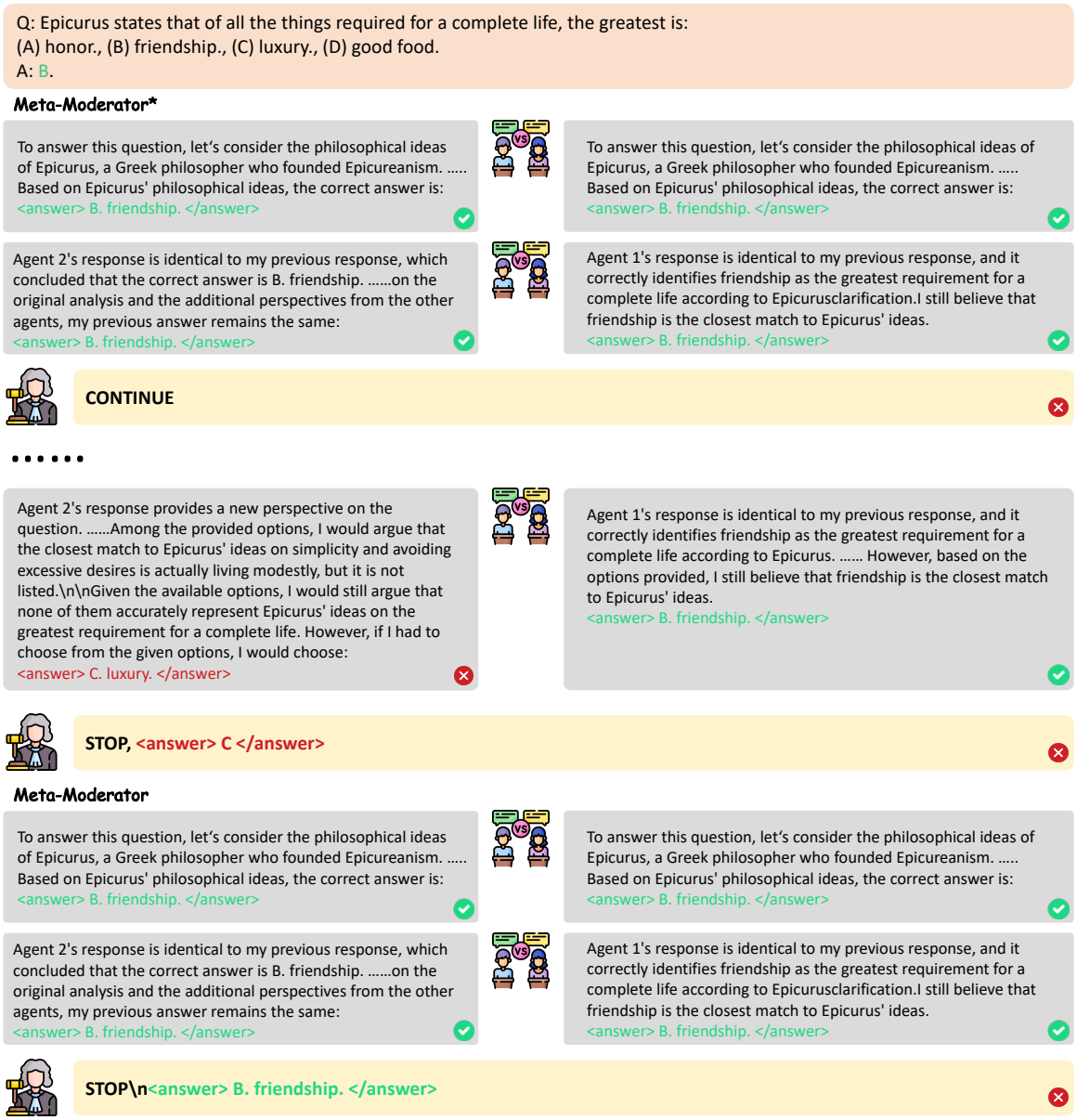

Figure 4: Case study comparing a failure of the untrained Meta-Moderator* with a successful decision by the trained Meta-Moderator.