LEARNING FROM SELF-DEBATE: PREPARING REASONING MODELS FOR MULTI-AGENT DEBATE

Anonymous authors
Paper under double-blind review

ABSTRACT

The reasoning abilities of large language models (LLMs) have been substantially improved by reinforcement learning with verifiable rewards (RLVR). At test time, collaborative reasoning through Multi-Agent Debate (MAD) has emerged as a promising approach for enhancing LLM performance. However, RLVR typically trains isolated solvers, while multi-agent RL often optimizes MAD as a system. We therefore argue that improving MAD first requires strengthening each participant's ability to evaluate competing reasoning and decide when to revise or retain its answer. We show that reassessment improves average correctness when corrections of wrong answers outweigh changes from correct to incorrect answers, and that maximizing correctness rewards also maximizes this net accuracy gain. Guided by this analysis, we propose *Self-Debate Reinforcement Learning* (SDRL), which post-trains a single LLM to solve problems independently and reason over peer responses during multi-agent debate. SDRL builds debate contexts from sampled solutions, with frequency-based pairing prioritizing the most prevalent answer disagreements, and generates responses conditioned on these contexts. Finally, SDRL jointly optimizes both the initial and debate-conditioned responses, yielding a model that is effective as both a standalone solver and a debate participant. Experiments across multiple base models and reasoning benchmarks show that SDRL strengthens single-model reasoning and improves MAD performance across diverse debate protocols and agent configurations.

1 INTRODUCTION

Large Language Models (LLMs) have recently demonstrated strong reasoning capabilities, achieving impressive performance on tasks such as mathematics, programming, and tool use Guo et al. (2025a); Yang et al. (2025); Liu et al. (2025a). Reinforcement Learning with Verifiable Rewards (RLVR) plays a crucial role in enabling these capabilities by leveraging tasks with automatically checkable outcomes to provide reliable supervision for scaling reasoning. To further enhance LLM performance, a growing line of work Wang et al. (2022); Xiong et al. (2025b); Wang et al. (2025b); Zheng et al. (2025b) studies test-time scaling strategies to further improve the reasoning ability of LLMs. Among these approaches, collaborative reasoning Du et al. (2023); Liang et al. (2024) through Multi-Agent Debate (MAD) has emerged as a promising paradigm: multiple LLM agents propose solutions to a shared question and update their answers in response to their peers Choi et al. (2025); Yi et al. (2025).

Despite this promise, effective methodologies for preparing reasoning models for MAD interaction remain underexplored. Existing RLVR methods typically train LLMs to solve problems independently, while MAD also asks them to reason over competing solutions. This mismatch between training and inference can limit LLM effectiveness in MAD settings Choi et al. (2025); Kumaran et al. (2025); Li & Goyal (2025). Recent multi-agent RL approaches often optimize MAD as a system, requiring resource-intensive training of several agents Liao et al. (2025); Liu et al. (2025d); Zhang et al. (2025b); Li et al. (2025). System-level gains do not necessarily improve each participant's ability to assess competing reasoning. We therefore argue that improving MAD first requires strengthening each participant's ability to evaluate competing reasoning and decide when to revise or retain its answer.

In this work, we study how to post-train a single reasoning model for both independent problem solving and participation in MAD. Extending the Dirichlet Compound Multinomial (DCM) framework of Choi et al. (2025), we show how reassessing peer reasoning changes average agent correctness. We

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

quantify this change through *net accuracy gain*: how often debate corrects a wrong answer minus how often it turns a correct answer into a wrong one. On fixed debate contexts, rewarding correct answers optimizes this gain. We further show that equal progress on more frequent disagreements yields greater overall net accuracy gain, motivating training on response pairs with the two most frequent distinct answers. Building on this analysis, we propose *Self-Debate Reinforcement Learning* (SDRL), a learning paradigm that enables a single model to learn from its own reasoning conflicts, effectively preparing it for both standalone problem-solving and multi-agent interactions. Given a prompt, SDRL first samples multiple candidate solutions from the current policy and evaluates them using verifiable rewards. SDRL then constructs debate pairs from these solutions, with frequency-based pairing focusing training on the dominant competing answers. Conditioned on a debate pair, the model generates second-round responses that deliberate over both candidates to produce a final answer. Finally, SDRL jointly optimizes both the initial and second-round responses under verifiable rewards. This joint objective encourages the model to produce accurate first-pass solutions while also learning when and how to revise its reasoning during debate-style interaction, enabling a single trained model to serve as a highly competent standalone solver and MAD participant.

Overall, our **contributions** can be summarized as follows:

1. We model reasoning-dependent reassessment and characterize population drift, connect net accuracy gain to correctness-reward optimization, and establish a frequency-based rule for prioritizing answer disagreements.
2. We propose *Self-Debate Reinforcement Learning* (SDRL), a framework that equips a *single* reasoning model with both independent problem-solving ability and productive revision behavior by jointly training initial and debate-conditioned responses with correctness rewards.
3. Extensive experiments across multiple reasoning benchmarks demonstrate that SDRL consistently enhances multi-agent debate performance while simultaneously strengthening single-agent accuracy.

2 PRELIMINARIES

Multi-Agent Debate. We formulate the debate process as an iterative response-revision process among a set of N language model agents, denoted by $\Pi = \{\pi_1, \ldots, \pi_N\}$. Let $\mathcal{Q}$ represent the input space and $\mathcal{O}$ the output space. Each agent π_i generates an independent response $o_{i,0} \sim \pi_i(q)$ for a given input $q \in \mathcal{Q}$.

In the *Majority Voting* setting Zhao et al. (2025a), these independent outputs are immediately aggregated via a voting function $\mathcal{V} : \mathcal{O}^N \to \mathcal{O}$ to produce the final prediction:

$$o_0 = \mathcal{V}(o_{1,0}, \ldots, o_{N,0}). \tag{1}$$

Multi-Agent Debate (MAD) extends this static ensemble approach by introducing a T round communication framework. Under the simultaneous-talk protocol Chan et al. (2023); Choi et al. (2025), agents revise their responses in parallel over discrete rounds $t \in \{1, \ldots, T\}$. At step t, agent π_i conditions its update on the set of peer responses generated in the previous round, formally defined as:

$$\mathcal{O}_i^{(t-1)} = \{o_{j,t-1} \mid j \in \mathcal{N}(i)\}. \tag{2}$$

where $\mathcal{N}(i) \subseteq \{1, \ldots, N\}$ denotes the neighborhood observable to agent π_i, including itself. The agent then refines its answer using a debate protocol $\mathcal{D}$, which integrates the original query q and the peer context $\mathcal{O}_i^{(t-1)}$ to produce the updated response:

$$o_{i,t} = \mathcal{D}\left(q; \mathcal{O}_i^{(t-1)}\right). \tag{3}$$

This iterative refinement can be viewed as a functional composition over T rounds. The state of agent π_i at the final round is expressed as:

$$o_{i,T} = (\mathcal{D} \circ \cdots \circ \mathcal{D})(q; \mathcal{O}_i) = \mathcal{D}^{(T)}(q; \mathcal{O}_i). \tag{4}$$

Finally, the system-level output is derived by aggregating the refined responses from the terminal state:

$$o_T = \mathcal{V}(o_{1,T}, \ldots, o_{N,T}). \tag{5}$$

Following previous works Choi et al. (2025); Yi et al. (2025), we focus on homogeneous agent settings, where all π_i share identical architectures.

Reinforcement Learning. To eschew the computational overhead of a separate critic network, we adopt the Group Relative Policy Optimization (GRPO) framework (Shao et al., 2024; Guo et al., 2025a). For a specific query q, the behavior policy $\pi_{\theta_{\text{old}}}$ samples a group of n responses $\{o_i\}_{i=1}^{n}$. The advantage for the i-th response is estimated by normalizing the reward r_i against the group statistics:

$$\hat{A}_{i,t} = \frac{r_i - \text{mean}(\{r_i\}_{i=1}^{n})}{\text{std}(\{r_i\}_{i=1}^{n})}. \tag{6}$$

Building on this, we employ Decoupled Clip and Dynamic sAmpling Policy Optimization (DAPO) (Yu et al., 2025) to stabilize updates for long chain-of-thought reasoning. DAPO mitigates entropy collapse and reward noise through asymmetric clipping and dynamic sampling constraints. The policy is optimized via the following token-level gradient objective:

$$\begin{aligned} \mathcal{J}_{\text{DAPO}}(\theta) = \quad & \mathbb{E}_{(q,o^*)\sim\mathcal{Q},\{o_i\}_{i=1}^{n}\sim\pi_{\theta_{\text{old}}}(\cdot|q)} \left[\frac{1}{\sum_{i=1}^{n}|o_i|} \sum_{i=1}^{n}\sum_{t=1}^{|o_i|} \right. \\ & \left. \min\Big(r_{i,t}(\theta)\hat{A}_{i,t},\ \text{clip}\Big(r_{i,t}(\theta), 1-\varepsilon_{\text{low}}, 1+\varepsilon_{\text{high}}\Big)\hat{A}_{i,t}\Big)\right] \\ & \text{s.t.} \quad 0 < \Big|\{o_i \mid \texttt{is_equivalent}(o^*, o_i)\}\Big| < n, \end{aligned} \tag{7}$$

where $r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q,o_{i,<t})}$ is the importance ratio, and o^* denotes the ground-truth answer.

3 Theoretical Analysis

Building on the Dirichlet Compound Multinomial (DCM) framework of Choi et al. (2025), we characterize when debate improves population accuracy, connect net accuracy gain to correctness rewards, and derive a rule for selecting debate contexts. Proofs appear in Appendix G.

3.1 Reassessment and Population Drift

The basic DCM update of Choi et al. (2025) uses peer answer counts, preserving expected population accuracy under neutrality conditions. Yet evaluating peers' reasoning can reveal errors or validate alternative solutions. We introduce effective evidence β to capture this reassessment and characterize when it improves accuracy. Fix a question with answer set $\mathcal{A} = \{1, \ldots, K\}$, where answer 1 is correct. Agent $i \in \{1, \ldots, N\}$ has DCM parameters $\alpha_{i,t} \in (0, \infty)^K$ at round t. Given the history $\mathcal{F}_t$ before sampling that round's answers, its correctness probability is $p_{i,t} := \Pr(y_{i,t} = 1 \mid \mathcal{F}_t) = \alpha_{i,t}^{(1)} / \|\alpha_{i,t}\|_1$. Under fully connected communication, including each agent's own response, all agents observe counts $c_t^{(k)} = \sum_{j=1}^{N} \mathbf{1}\{y_{j,t-1} = k\}$ and update by

$$\alpha_{i,t} = \alpha_{i,t-1} + wc_t + \beta_{i,t-1}, \qquad w \geq 0, \quad \beta_{i,t-1} \geq 0, \quad \|\beta_{i,t-1}\|_1 = m_\beta. \tag{8}$$

The term $\beta_{i,t-1}$ represents effective evidence from reassessing the full reasoning context. It may depend on the same responses as c_t. The fixed mass $m_\beta \geq 0$ sets its scale. Assume $\|\alpha_{i,0}\|_1 = S_0 > 0$ for all agents, so their total mass is $S_t = S_0 + tC$, with $C = wN + m_\beta$. Define the *reassessment advantage* $\delta_{i,t} := \mathbb{E}[\beta_{i,t}^{(1)} \mid \mathcal{F}_t] - m_\beta p_{i,t}$ and its mean $\bar{\delta}_t := N^{-1} \sum_i \delta_{i,t}$.

Theorem 3.1 (Population drift)**.** *For update equation 8 with full connectivity, common w, m_β, and equal initial masses S_0, the mean correctness $\bar{p}_t = N^{-1} \sum_i p_{i,t}$ satisfies*

$$\mathbb{E}[\bar{p}_t \mid \mathcal{F}_{t-1}] - \bar{p}_{t-1} = \frac{\bar{\delta}_{t-1}}{S_0 + tC}. \tag{9}$$

Its conditional drift is positive exactly when $\bar{\delta}_{t-1} > 0$. If $\bar{\delta}_{t-1} = 0$ at every round, $(\bar{p}_t)_{t\geq 0}$ is a martingale.

The key cancellation is $\sum_i (\bar{p}_{t-1} - p_{i,t-1}) = 0$: social influence redistributes beliefs, while reassessment determines population drift. Setting $m_\beta = 0, w = 1$ recovers the neutral population process of Choi et al. (2025, Appendix D.4).

For correctness indicators X, Y before and after a debate step, define the *net accuracy gain*

$$G := \mathbb{E}[Y - X] = \Pr(X = 0, Y = 1) - \Pr(X = 1, Y = 0). \tag{10}$$

This identity holds for any transition law. For a uniformly selected agent in the DCM process, $G = \mathbb{E}[\bar{\delta}_{t-1}/S_t]$. Thus net accuracy gain measures the benefit of reassessment without estimating β. The accuracy of the final vote additionally depends on the joint answer distribution (Appendix G.6).

3.2 Training Reassessment with Correctness Rewards

To train reassessment, fix the context law $D^{\mathsf{S}}_{\text{old}}$ induced by the behavior policy and selection rule S. A context $c = (q, o^1, o^2)$ contains the question and an ordered pair of full responses. Here, $X(c)$ is the correctness of the response in the initial position. Let Y_θ denote the correctness of a response from $\pi_\theta(\cdot \mid c)$, and define $G_{\mathsf{S}}(\theta) = \mathbb{E}_{D^{\mathsf{S}}_{\text{old}},\, \pi_\theta}[Y_\theta - X(c)]$.

Proposition 3.2 (Reward alignment and advantage invariance)**.** *On $D^{\mathsf{S}}_{\text{old}}$, the correctness reward $r = 2Y_\theta - 1$ satisfies*

$$J_{\text{corr}}(\theta) := \mathbb{E}[r] = 2G_{\mathsf{S}}(\theta) + 2\mathbb{E}[X(c)] - 1. \tag{11}$$

The context-specific baseline $2X(c) - 1$ also cancels in group-normalized advantages (Appendix G.3).

Under equation 8 with $N = 2$, suppose π_θ induces effective evidence β_θ while initial beliefs, counts, and m_β remain fixed. Let $p(c)$ be the originating agent's pre-debate correctness probability and $\delta_\theta(c) := \mathbb{E}[\beta^{(1)}_\theta \mid c] - m_\beta p(c)$. Then

$$G_{\mathsf{S}}(\theta) - G_{\mathsf{S}}(\theta_{\text{old}}) = \frac{\mathbb{E}[\beta^{(1)}_\theta - \beta^{(1)}_{\theta_{\text{old}}}]}{S_t} = \frac{\mathbb{E}[\delta_\theta(c) - \delta_{\theta_{\text{old}}}(c)]}{S_t}. \tag{12}$$

By equation 11, maximizing correctness reward on these fixed contexts maximizes expected correct-answer evidence and reassessment advantage. Under Theorem 3.1, positive mean advantage yields positive drift.

3.3 Prioritizing Dominant Answer Disagreements

To avoid enumerating all $\binom{n}{2}$ response pairs, we prioritize the most prevalent disagreement. Fix an indexed pool $\mathcal{O}$ of n responses containing at least two distinct answers. Let n_k be the count of answer k, and $a_{(1)}, a_{(2)}$ the two most frequent distinct answers, with ties resolved consistently. Parenthetic indices denote frequency rank.

Proposition 3.3 (Dominant-disagreement selection)**.** *Sampling two distinct response indices uniformly without replacement gives unordered disagreement probability $\widehat{\omega}_{k\ell} = 2n_k n_\ell/[n(n-1)]$ for $k \neq \ell$. The pair $\{a_{(1)}, a_{(2)}\}$ maximizes both $\widehat{\omega}_{k\ell}$ and the covered answer mass $(n_k + n_\ell)/n$ over distinct answer pairs.*

Consider the reference task that samples such a pair, randomizes its order, and generates a debate response. Let $g_{k\ell}(\theta; \mathcal{O})$ be its conditional net accuracy gain given answer pair $\{k, \ell\}$, averaging over full texts, order, and generated responses. Then

$$G_{\text{ref}}(\theta; \mathcal{O}) = \sum_{k \leq \ell} \widehat{\omega}_{k\ell} g_{k\ell}(\theta; \mathcal{O}), \qquad \widehat{\omega}_{kk} = \frac{n_k(n_k - 1)}{n(n-1)}, \tag{13}$$

omitting zero-weight terms. Improving only $g_{k\ell}$ by ε increases G_{ref} by $\widehat{\omega}_{k\ell}\varepsilon$. Hence equal progress on the most frequent disagreement has the largest aggregate effect among disagreement types. Frequency-based pairing allocates training to these dominant conflicts, while correctness rewards determine the direction of revision. Appendix G.5 further quantifies the roles of learning progress and transfer across contexts.

4 Methodology

We propose Self-Debate Reinforcement Learning (SDRL), which jointly post-trains a *single* policy for independent problem solving and debate-conditioned revision. Section 4.1 trains standalone reasoning. Section 4.2 constructs debate contexts and applies the reward alignment in Proposition 3.2 to train productive revision.

Algorithm 1 Self-Debate Reinforcement Learning (SDRL)

Input: Policy π_θ, prompt set $\mathcal{D} = \{q_i\}_{i=1}^N$, initial responses n, second-round responses n_d, training steps K
Output: Updated policy $\pi_{\theta_{\text{updated}}}$

for $t = 1, \ldots, K$ **do**
  Sample mini-batch $\mathbf{q} \subseteq \mathcal{D}$ and initialize $\mathbf{B} \leftarrow \emptyset$
  for each $q \in \mathbf{q}$ **do**
    Sample initial responses $\{o_i\}_{i=1}^n \sim \pi_\theta(\cdot \mid q)$ and compute advantages $\{A_i\}_{i=1}^n$
    $\mathbf{B} \leftarrow \mathbf{B} \cup \{(q, \{o_i\}_{i=1}^n, \{A_i\}_{i=1}^n)\}$
    Select debate pair (o^1, o^2) from $\{o_i\}_{i=1}^n$ and construct debate prompt q_d
    Sample second-round responses $\{o_j^d\}_{j=1}^{n_d} \sim \pi_\theta(\cdot \mid q_d)$ and compute advantages $\{A_j^d\}_{j=1}^{n_d}$
    $\mathbf{B} \leftarrow \mathbf{B} \cup \{(q_d, \{o_j^d\}_{j=1}^{n_d}, \{A_j^d\}_{j=1}^{n_d})\}$
  end for
  Update π_θ using the RL optimizer on $\mathbf{B}$
end for
return $\pi_{\theta_{\text{updated}}}$

4.1 SINGLE-AGENT REASONING

To improve the model's reasoning ability in the single-agent setting, SDRL follows the GRPO framework Shao et al. (2024) and optimizes the policy using verifiable rewards. For each prompt q, SDRL samples n independent responses $\{o_i\}_{i=1}^n \sim \pi_\theta(\cdot \mid q)$ from the current policy and assigns a sparse outcome reward $r_j \in \{+1, -1\}$ based on the correctness of the final answer. We then compute advantages for each response using Eq. 6. To prepare candidate responses for debate training, we follow DAPO Yu et al. (2025) to oversample prompts and retain mixed-correctness groups as in Eq. 7. These pools contain distinct final answers for debate pairing.

4.2 DEBATE TRAINING

Alongside standalone reasoning, SDRL trains the same policy to reassess an initial solution using a peer response. The net accuracy gain objective in Eq. 10 rewards corrections and penalizes changes from correct to incorrect answers. We construct each context from two initial responses, making answer-pair selection the basic unit of training allocation analyzed in Section 3.3.

Debate pair construction. Let $\mathcal{O}(q) = \{o_i\}_{i=1}^n$ denote the initial responses, which we use as the candidate pool, and let $a(o_i)$ be the extracted final answer of response o_i. SDRL constructs a debate pair $\mathcal{P}(o^1, o^2)$ by selecting two candidates from $\mathcal{O}(q)$ using one of two pairing rules:

1. *Random pairing*. It samples two responses uniformly from $\mathcal{O}(q)$. This exposes the model to both agreement and disagreement between sampled solutions, corresponding to the answer-pair components in Eq. 13.
2. *Frequency-based pairing*. It first identifies the most common final answer $a_{(1)}$ and the second most common final answer $a_{(2)}$ among the sampled responses. It then selects one response corresponding to each answer to form a debate pair. By Proposition 3.3, this choice maximizes disagreement frequency and covered answer mass. Correctness rewards then guide revision toward correct answers.

We evaluate both constructions and use the same downstream prompting and optimization for either choice.

Debate prompt formulation. Given a selected pair $\mathcal{P}(o^1, o^2)$, we build a debate prompt q_d by serializing the two candidate responses into a two-turn conversation, and then prompt the model to deliberate over both responses and produce a final answer. The detailed conversation format and prompt template are shown in the Appendix. For *frequency-based pairing*, we randomize the order of the selected responses so that each answer class can occupy either position. In the notation of Section 3.2, $c = q_d$ and $X(c)$ is the correctness of the response in the initial assistant position.

Second-round responses and joint optimization. For each debate prompt q_d, we sample n_d second-round responses and score them with the same verifiable reward used for the initial responses. We compute advantages by normalizing rewards across responses to the same q_d. We then combine the initial and second-round responses into one training batch and update the same policy using DAPO.

5 EXPERIMENTS

5.1 IMPLEMENTATION DETAILS

5.1.1 TRAINING DETAILS

Following recent works (Zheng et al., 2025a; Cheng et al., 2025; Zheng et al., 2025c), We employ Qwen2.5-3B Qwen et al. (2025), Qwen3-4B-Base and Qwen3-8B-Base Yang et al. (2025) as our backbone models. Following works (Yu et al., 2025; Cheng et al., 2025; Cui et al., 2025), we utilize the DAPO-Math-17K dataset (Yu et al., 2025) for training. Given its demonstrated stability and superiority over vanilla GRPO (Yu et al., 2025; Cheng et al., 2025), we adopt the DAPO algorithm Yu et al. (2025) as both our baseline and the core optimization method for our SDRL approach.

We configure the learning rate at 1×10^{-6} with a linear warm-up over the first 10 rollout steps. The rollout phase utilizes a prompt batch size of 256, generating 8 responses per prompt. For each debate pair, we generate 4 responses for Qwen2.5-3B and 8 responses for Qwen3-4B-Base and Qwen3-8B-Base. The correctness component of the reward assigns $+1$ to correct solutions and -1 otherwise. The overlength penalty is specified in the Appendix. All training experiments are implemented within the verl framework (Sheng et al., 2024). Additional configuration details are provided in the Appendix.

5.1.2 EVALUATION

Benchmarks. Following prior work Choi et al. (2025); Zhao et al. (2025b); Li & Goyal (2025), we conduct evaluations on the mathematical reasoning benchmarks MATH500 (Hendrycks et al., 2021), AMC 2023, and AIME 2024/2025.

Multi-Agent Debate. Consistent with established MAD literature (Choi et al., 2025), we primarily evaluate under the *Decentralized MAD* setting Du et al. (2023); Choi et al. (2025), where each agent observes all peer responses from the previous round. Additional ablation studies on more debate frameworks are provided in Section 5.4. We benchmark performance against the strong *Majority Voting* baseline, which selects the most common answer from the initial round without debate. Our main experiments utilize $N = 5$ agents and $T = 1$ debate round. Ablations on N and T are provided separately. In decentralized MAD, Δ measures the change in majority-vote accuracy, while G in Eq. 10 measures the change in mean agent correctness. Inference is conducted with a rollout temperature of 1.0 and top-p sampling ($p = 0.9$). For all MAD experiments, we report the average score across 5 independent runs.

Single-Agent Reasoning. Following recent RL reasoning literature (Guo et al., 2025a; Liu et al., 2025b), we evaluate model performance without debate. For each prompt, we sample K independent responses, and report the average accuracy ($mean@K$) and majority-vote accuracy ($maj@K$), setting $K = 32$ for the AMC and AIME datasets, and $K = 4$ for the MATH500 benchmarks. Further evaluation details are available in the Appendix.

5.2 MULTI-AGENT DEBATE

Table 1 compares SDRL with the DAPO baseline under the decentralized MAD setting with N=5 agents and one debate round. Because the debate protocol is fixed, this comparison tests whether training the individual model changes the outcome of MAD. SDRL-rand denotes random pairing, and SDRL-freq denotes frequency-based pairing for debate pair construction. The main observation is that SDRL consistently improves multi-agent debate across model architectures. The strongest results are achieved by SDRL-freq, which gives the best average *Debate* accuracy for Qwen3-4B-Base and Qwen3-8B-Base, and matches the best result for Qwen2.5-3B.

Frequency-based pairing achieves stronger overall debate performance than random pairing. For Qwen3-4B-Base, SDRL-freq improves average *Debate* accuracy from 55.6 with SDRL-rand to 57.7. This is consistent with Section 3.3, which shows that equal progress on a more frequent disagreement produces a larger overall net accuracy gain. By concentrating on the dominant competing answers, SDRL-freq trains the model to resolve choices it frequently encounters in its own reasoning.

Table 1: Multi-agent debate performance comparison of different model architectures between the DAPO baseline and SDRL across math reasoning benchmarks. The debate system contains 5 agents. *Maj* denotes the majority-vote accuracy of the agents' direct responses to the question. *Debate* denotes the performance of the decentralized multi-agent system after debate round 1. $\Delta = \textit{Debate} - \textit{Maj}$ is the change in system accuracy. Best results are bolded.

Method	MATH500			AMC23			AIME24			AIME25			Avg.		
	Maj	Debate	Δ	Maj	Debate	Δ	Maj	Debate	Δ	Maj	Debate	Δ	Maj	Debate	Δ
Qwen2.5-3B															
DAPO	70.9	71.1	0.2	49.0	50.0	1.0	7.3	10.0	2.7	4.0	3.3	-0.7	32.8	33.6	0.8
+ SDRL-rand	70.7	71.4	**0.7**	49.2	**53.8**	**4.6**	8.7	11.7	3.0	**5.6**	**6.1**	0.5	33.6	**35.8**	**2.2**
+ SDRL-freq	**71.4**	**71.7**	0.3	**52.5**	53.5	1.0	**9.3**	**13.3**	**4.0**	3.3	4.7	**1.4**	**34.1**	**35.8**	1.7
Qwen3-4B-Base															
DAPO	86.9	83.1	-3.8	**73.0**	76.0	3.0	26.7	28.3	1.6	24.7	24.0	-0.7	52.8	52.9	0.1
+ SDRL-rand	**88.8**	**86.6**	-2.2	72.5	**80.0**	**7.5**	**30.8**	31.7	0.9	22.7	24.0	**1.3**	53.7	55.6	1.9
+ SDRL-freq	87.0	85.9	**-1.1**	72.5	79.0	6.5	28.7	**36.0**	**7.3**	**28.7**	**30.0**	**1.3**	**54.2**	**57.7**	**3.5**
Qwen3-8B-Base															
DAPO	86.6	85.3	-1.3	79.0	77.5	-1.5	28.0	26.7	-1.3	23.3	24.7	1.4	54.2	53.6	-0.6
+ SDRL-rand	86.2	86.5	**0.3**	**81.5**	86.5	5.0	31.0	34.8	3.8	**26.7**	32.0	5.3	56.4	60.0	3.6
+ SDRL-freq	**88.0**	**88.2**	0.2	80.0	**88.0**	**8.0**	**33.3**	**37.9**	**4.6**	25.6	**32.2**	**6.6**	**56.7**	**61.6**	**4.9**

Table 2: Single-agent performance comparison of different model architectures between the DAPO baseline and SDRL across several math reasoning benchmarks. We report the average accuracy $\mathrm{mean@}K$ and the majority-vote accuracy $\mathrm{maj@}K$ over K independent runs. Best results are bolded.

Method	MATH500		AMC23		AIME24		AIME25		Avg.	
	mean@4	maj@4	mean@32	maj@32	mean@32	maj@32	mean@32	maj@32	mean	maj
Qwen2.5-3B										
DAPO	64.5	70.4	48.5	56.8	7.2	13.5	3.1	4.2	30.8	36.2
+ SDRL-rand	65.1	70.6	48.1	57.4	**9.9**	13.9	**4.8**	**8.0**	32.0	**37.5**
+ SDRL-freq	**65.5**	**71.0**	**49.9**	**58.0**	9.7	**15.5**	4.1	4.9	**32.3**	37.4
Qwen3-4B-Base										
DAPO	82.9	84.4	71.9	81.4	24.8	31.1	24.9	27.8	51.1	56.2
+ SDRL-rand	**83.7**	85.8	72.3	**84.1**	25.9	**35.1**	20.4	27.6	50.6	58.2
+ SDRL-freq	82.2	**86.2**	**74.2**	83.0	**26.6**	34.4	**26.6**	**37.2**	**52.4**	**60.2**
Qwen3-8B-Base										
DAPO	83.2	85.4	77.7	86.8	26.1	33.0	23.6	26.6	52.7	58.0
+ SDRL-rand	83.4	86.6	78.3	90.5	30.1	38.4	**27.3**	**35.8**	54.8	62.8
+ SDRL-freq	**85.4**	**88.4**	**80.6**	**93.6**	**32.6**	**43.4**	26.6	34.4	**56.3**	**65.0**

The gains from SDRL-freq are larger on stronger backbones. It improves average *Debate* accuracy from 33.6 to 35.8 on Qwen2.5-3B, compared with 53.6 to 61.6 on Qwen3-8B-Base. This pattern suggests that stronger models can better distinguish correct reasoning from errors in peer responses. SDRL trains them to apply this ability when comparing competing solutions, allowing their stronger reasoning capacity to translate into more effective revision during debate.

The gains also depend on task difficulty. On MATH500, high initial accuracy leaves less room for correction, leading to smaller improvements than on AMC23 and AIME. For Qwen3-8B-Base, SDRL-freq improves *Debate* accuracy from 85.3 to 88.2 on MATH500, compared with 26.7 to 37.9 on AIME24. This pattern supports the intuition that SDRL is particularly useful when initial reasoning is more uncertain. In these cases, evaluating alternative solutions offers more opportunities to identify errors and revise an incorrect answer.

5.3 SINGLE-AGENT REASONING

We follow prior reasoning LLM evaluation protocols Yu et al. (2025) and evaluate each model as a single agent. This assesses SDRL in its standalone role using the same trained policy evaluated in MAD. Table 2 reports both $\mathrm{mean@}K$ and $\mathrm{maj@}K$ across benchmarks. Overall, SDRL improves

Table 3: Agent-level transition statistics after one debate round on Qwen3-4B-Base. Both transition frequencies are percentages of all evaluated responses. The empirical net accuracy gain G is Incorrect→Correct minus Correct→Incorrect, reported in percentage points. Best results are bolded within each benchmark.

Benchmark	Method	Correct→Incorrect	Incorrect→Correct	G
MATH500	*DAPO*	7.5	5.8	-1.6
	+ SDRL-freq	**4.9**	**6.1**	**1.2**
AMC23	*DAPO*	3.5	8.5	5.0
	+ SDRL-freq	**1.5**	**14.5**	**13.0**
AIME24	*DAPO*	1.3	4.7	3.3
	+ SDRL-freq	**0.7**	**10.7**	**10.0**
AIME25	*DAPO*	**0.7**	2.0	1.3
	+ SDRL-freq	1.5	**6.2**	**4.7**
Avg.	*DAPO*	3.3	5.3	2.0
	+ SDRL-freq	**2.2**	**9.4**	**7.2**

single-agent reasoning across all three model architectures. Together with the MAD results, these gains show that SDRL improves both standalone reasoning and debate performance.

The advantage of frequency-based pairing extends to standalone reasoning. For Qwen3-4B-Base, SDRL-freq improves average $\mathrm{mean@}K$ from 50.6 with SDRL-rand to 52.4. The frequent disagreements emphasized by our theory arise from the model's own sampled solutions and reflect recurring alternatives in its reasoning. Learning to distinguish correct solutions from these alternatives can therefore improve answers generated without peer context.

The gains are more pronounced on stronger backbones. SDRL-freq improves average $\mathrm{maj@}K$ from 36.2 to 37.4 on Qwen2.5-3B, compared with 58.0 to 65.0 on Qwen3-8B-Base. Stronger models can better identify errors and recognize valid reasoning in competing solutions, making training comparisons more informative. These results suggest that learning to evaluate peer reasoning also helps these models construct more reliable solutions independently.

As in multi-agent debate, harder tasks offer more opportunities for improvement. For Qwen3-8B-Base, SDRL-freq improves $\mathrm{mean@}K$ from 83.2 to 85.4 on MATH500, compared with 26.1 to 32.6 on AIME24. High accuracy on MATH500 leaves less room for improvement. The larger gains on AIME24 suggest that comparing competing solutions during training is especially useful when the model's reasoning is uncertain, helping it recognize misleading paths and select sound reasoning when solving a problem independently.

5.4 Further Analysis

As SDRL with frequency-based pairing yields better overall performance than random pairing, we conduct additional experiments using SDRL+freq. More experiments, including debate-round ablations and critique quality evaluation, can be found in the Appendix.

Net Accuracy Gain. We evaluate the net accuracy gain objective in Eq. 10 using Qwen3-4B-Base. Table 3 reports agent-level transitions after the first debate round. Both transition frequencies use all evaluated responses as their denominator, so their difference is the empirical G. SDRL-freq increases Incorrect→Correct transitions on all four benchmarks and reduces Correct→Incorrect transitions on three benchmarks. The average net accuracy gain G increases from 2.0 to 7.2 percentage points. Thus, the trained policy produces a larger response-level accuracy gain after incorporating peer context.

Number of Agents. We report experimental results for the decentralized MAD system with varying numbers of agents under a single debate round, as shown in Table 4. Overall, SDRL consistently outperforms the DAPO baseline across all agent settings. Increasing the number of agents from 3 to 5 leads to consistent improvements in average debate performance. However, in the 7-agent setting on AMC23, AIME24, and AIME25, performance drops because we reduce the max response tokens during generation to keep the debate sequence within the model context window, which results in truncated and incomplete responses.

Table 4: Ablation on the number of agents N in decentralized MAD setting. *Maj* denotes the majority-vote accuracy of the agents' direct responses to the question. *Debate* denotes the performance of the decentralized multi-agent system after debate round 1. $\Delta = \textit{Debate} - \textit{Maj}$ is the change in system accuracy.

Method	MATH500			AMC23			AIME24			AIME25			Avg.		
	Maj	Debate	Δ	Maj	Debate	Δ	Maj	Debate	Δ	Maj	Debate	Δ	Maj	Debate	Δ
Qwen2.5-3B															
DAPO (N=3)	67.5	68.2	0.7	48.0	50.0	2.0	4.0	6.7	2.7	1.3	1.3	0.0	30.2	31.6	1.4
DAPO (N=5)	70.9	71.1	0.2	49.0	50.0	1.0	7.3	10.0	2.7	4.0	3.3	-0.7	32.8	33.6	0.8
DAPO (N=7)	72.4	71.9	-0.5	49.0	51.0	2.0	6.7	9.3	2.7	3.3	2.7	-0.7	32.9	33.7	0.9
+SDRL (N=3)	66.2	67.3	1.1	51.0	55.0	4.0	11.3	13.3	2.0	1.3	4.7	3.3	32.5	35.1	2.6
+SDRL (N=5)	71.4	71.7	0.3	52.5	53.5	1.0	9.3	13.3	4.0	3.3	4.7	1.4	34.1	35.8	1.7
+SDRL (N=7)	71.5	71.9	0.4	53.0	55.0	2.0	12.0	14.7	2.7	4.7	6.0	1.3	35.3	36.9	1.6
Qwen3-4B-Base															
DAPO (N=3)	85.2	82.8	-2.4	68.0	70.5	2.5	20.7	25.3	4.7	24.7	23.3	-1.3	49.6	50.5	0.9
DAPO (N=5)	86.9	83.1	-3.8	73.0	76.0	3.0	26.7	28.3	1.6	24.7	24.0	-0.7	52.8	52.9	0.1
DAPO (N=7)	87.7	84.0	-3.7	72.0	76.0	4.0	23.3	26.0	2.7	23.3	24.7	1.3	51.6	52.7	1.1
+SDRL (N=3)	84.4	83.6	-0.7	70.0	76.7	6.7	24.7	34.0	9.3	20.0	23.3	3.3	49.8	54.4	4.7
+SDRL (N=5)	87.0	85.9	-1.1	72.5	79.0	6.5	28.7	36.0	7.3	28.7	30.0	1.3	54.2	57.7	3.5
+SDRL (N=7)	88.1	85.9	-2.2	70.0	81.7	11.7	23.3	32.0	8.7	22.7	28.7	6.0	51.0	57.1	6.1

Table 5: Multi-agent debate performance under different debate settings. The debate system contains 5 agents. *Maj* denotes the majority-vote accuracy of the agents' direct responses to the question. *Debate* denotes system accuracy after debate round 1. $\Delta = \textit{Debate} - \textit{Maj}$ is the change in system accuracy.

Method	MATH500			AMC23			AIME24			AIME25			Avg.		
	Maj	Debate	Δ	Maj	Debate	Δ	Maj	Debate	Δ	Maj	Debate	Δ	Maj	Debate	Δ
Qwen2.5-3B															
DAPO (sparse)	71.4	70.8	-0.6	47.5	44.2	-3.3	10.0	6.7	-3.3	3.3	3.3	0.0	33.1	31.3	-1.8
DAPO (centralized)	62.5	63.2	0.7	38.3	42.5	4.2	6.7	4.4	-2.2	3.3	5.6	2.2	27.7	28.9	1.2
+SDRL (sparse)	70.5	71.0	0.5	48.3	52.5	4.2	6.7	8.9	2.2	4.4	4.4	0.0	32.5	34.2	1.7
+SDRL (centralized)	64.5	65.5	1.0	45.0	52.5	7.5	4.4	10.0	5.6	4.4	8.9	4.4	29.6	34.2	4.6
Qwen3-4B-Base															
DAPO (sparse)	86.5	86.0	-0.6	70.5	78.0	7.5	22.7	24.7	2.0	24.0	25.3	1.3	50.9	53.5	2.6
DAPO (centralized)	80.7	79.5	-1.2	64.0	72.0	8.0	23.3	27.3	4.0	21.4	22.0	0.6	47.4	50.2	2.9
+SDRL (sparse)	86.7	86.6	-0.1	69.5	78.5	9.0	28.7	36.0	7.3	28.7	33.3	4.6	53.4	58.6	5.2
+SDRL (centralized)	80.5	81.0	0.5	66.5	78.0	11.5	25.3	34.0	8.7	25.3	26.7	1.4	49.4	54.9	5.5

Additional Debate Frameworks. Following Choi et al. (2025), we evaluate how the learned revision policy transfers across communication structures. These include Sparse MAD Li et al. (2024b), a variant of decentralized MAD that uses a sparse communication topology to improve efficiency, and Centralized MAD Guo et al. (2024), where a central agent aggregates peers' responses and produces an updated response at each round. The results are reported in Table 5. Overall, SDRL outperforms the DAPO baseline in both debate accuracy and the improvement brought by debate across all settings, demonstrating the robustness of SDRL-trained models under different debate frameworks.

6 CONCLUSION

In this work, we studied how to post-train individual reasoning models for both standalone problem-solving and participation in multi-agent debate. We developed a theoretical analysis that links improvements in debate to net accuracy gain, connects this objective to correctness rewards, and motivates training on frequent answer disagreements. Guided by this analysis, we proposed *Self-Debate Reinforcement Learning* (SDRL), which enables a single model to learn from its own reasoning conflicts. We constructed debate contexts from the model's sampled solutions, using frequency-based pairing to focus on the dominant competing answers. We then jointly optimized the initial and debate-conditioned responses with verifiable rewards, training the model to solve problems independently and revise its answers using peer reasoning. Through experiments across multiple model backbones, reasoning benchmarks, debate protocols, and agent configurations, we demonstrated that SDRL consistently improves both multi-agent debate performance and single-agent accuracy.

AI USE STATEMENT

We used generative AI tools to assist with manuscript writing and editing. The authors take full responsibility for the final content, including its accuracy and originality.

ETHICS STATEMENT

SDRL aims to improve reasoning reliability by training models to compare solutions and correct errors using peer feedback. Our experiments use publicly available models and benchmarks. When debate models are used to support decisions, we recommend checking their answers against reliable evidence and having a human review the final decision.

REPRODUCIBILITY STATEMENT

We describe the training procedure in Algorithm 1 and provide model choices, datasets, hyperparameters, evaluation protocols, and prompts in Section 5.1 and Appendices B–D. Theoretical assumptions are stated in Section 3, with complete proofs in Appendix G.

REFERENCES

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201*, 2023.

Mingyang Chen, Linzhuang Sun, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z Pan, Wen Zhang, Huajun Chen, et al. Learning to reason with search for llms via reinforcement learning. *arXiv preprint arXiv:2503.19470*, 2025a.

Xiusi Chen, Gaotang Li, Ziqi Wang, Bowen Jin, Cheng Qian, Yu Wang, Hongru Wang, Yu Zhang, Denghui Zhang, Tong Zhang, et al. Rm-r1: Reward modeling as reasoning. *arXiv preprint arXiv:2505.02387*, 2025b.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. *arXiv preprint arXiv:2506.14758*, 2025.

Hyeong Kyu Choi, Xiaojin Zhu, and Sharon Li. Debate or vote: Which yields better decisions in multi-agent large language models? *arXiv preprint arXiv:2508.17536*, 2025.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first International Conference on Machine Learning*, 2023.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025a.

Jiaxin Guo, Zewen Chi, Li Dong, Qingxiu Dong, Xun Wu, Shaohan Huang, and Furu Wei. Reward reasoning model. *arXiv preprint arXiv:2505.14674*, 2025b.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*, 2024.

Shanshan Han, Qifan Zhang, Yuhang Yao, Weizhao Jin, and Zhaozhuo Xu. Llm multi-agent systems: Challenges and open problems. *arXiv preprint arXiv:2402.03578*, 2024.

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

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

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*, 2023.

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

Dharshan Kumaran, Stephen M Fleming, Larisa Markeeva, Joe Heyward, Andrea Banino, Mrinal Mathur, Razvan Pascanu, Simon Osindero, Benedetto De Martino, Petar Velickovic, et al. How overconfidence in initial choices and underconfidence under criticism modulate change of mind in large language models. *arXiv preprint arXiv:2507.03120*, 2025.

Aochong Oliver Li and Tanya Goyal. Off-trajectory reasoning: Can llms collaborate on reasoning trajectory? *arXiv preprint arXiv:2510.06410*, 2025.

Haoran Li, Ziyi Su, Yun Xue, Zhiliang Tian, Yiping Song, and Minlie Huang. Advancing collaborative debates with role differentiation through multi-agent reinforcement learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 22655–22666, 2025.

Xinyi Li, Sai Wang, Siqi Zeng, Yu Wu, and Yi Yang. A survey on llm-based multi-agent systems: workflow, infrastructure, and challenges. *Vicinagearth*, 1(1):9, 2024a.

Yunxuan Li, Yibing Du, Jiageng Zhang, Le Hou, Peter Grabowski, Yeqing Li, and Eugene Ie. Improving multi-agent debate with sparse communication topology. *arXiv preprint arXiv:2406.11776*, 2024b.

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

Junwei Liao, Muning Wen, Jun Wang, and Weinan Zhang. Marft: Multi-agent reinforcement fine-tuning. *arXiv preprint arXiv:2504.16129*, 2025.

Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. Deductive verification of chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 36:36407–36433, 2023.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025a.

Chenxi Liu, Junjie Liang, Yuqi Jia, Bochuan Cao, Yang Bai, Heng Huang, and Xun Chen. Explore data left behind in reinforcement learning for reasoning language models. *arXiv preprint arXiv:2511.04800*, 2025b.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864*, 2025c.

Shuo Liu, Tianle Chen, Zeyu Liang, Xueguang Lyu, and Christopher Amato. Llm collaboration with multi-agent reinforcement learning. *arXiv preprint arXiv:2508.04652*, 2025d.

Tongxuan Liu, Xingyu Wang, Weizhe Huang, Wenjiang Xu, Yuting Zeng, Lei Jiang, Hailong Yang, and Jing Li. Groupdebate: Enhancing the efficiency of multi-agent debate using group discussion. *arXiv preprint arXiv:2409.14051*, 2024.

Xiaoyuan Liu, Tian Liang, Zhiwei He, Jiahao Xu, Wenxuan Wang, Pinjia He, Zhaopeng Tu, Haitao Mi, and Dong Yu. Trust, but verify: A self-verification approach to reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2505.13445*, 2025e.

Chiyu Ma, Enpei Zhang, Yilun Zhao, Wenjun Liu, Yaning Jia, Peijun Qing, Lin Shi, Arman Cohan, Yujun Yan, and Soroush Vosoughi. Judging with many minds: Do more perspectives mean less prejudice? *arXiv preprint arXiv:2505.19477*, 2025.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*, 2023.

Jianing Qi, Xi Ye, Hao Tang, Zhigang Zhu, and Eunsol Choi. Learning to reason across parallel samples for llm reasoning. *arXiv preprint arXiv:2506.09014*, 2025.

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

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL `https://arxiv.org/abs/2412.15115`.

Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1073–1083, 2017.

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

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

Andries Smit, Paul Duckworth, Nathan Grinsztajn, Thomas D Barrett, and Arnu Pretorius. Should we be going mad? a look at multi-agent debate strategies for llms. *arXiv preprint arXiv:2311.17371*, 2023.

Yashar Talebirad and Amirhossein Nadiri. Multi-agent collaboration: Harnessing the power of intelligent llm agents. *arXiv preprint arXiv:2306.03314*, 2023.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, 2019.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025a.

Xiyao Wang, Chunyuan Li, Jianwei Yang, Kai Zhang, Bo Liu, Tianyi Xiong, and Furong Huang. Llava-critic-r1: Your critic model is secretly a strong policy model. *arXiv preprint arXiv:2509.00676*, 2025b.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Tianyi Xiong, Yi Ge, Ming Li, Zuolong Zhang, Pranav Kulkarni, Kaishen Wang, Qi He, Zeying Zhu, Chenxi Liu, Ruibo Chen, et al. Multi-crit: Benchmarking multimodal judges on pluralistic criteria-following. *arXiv preprint arXiv:2511.21662*, 2025a.

Tianyi Xiong, Xiyao Wang, Dong Guo, Qinghao Ye, Haoqi Fan, Quanquan Gu, Heng Huang, and Chunyuan Li. Llava-critic: Learning to evaluate multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 13618–13628, 2025b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Xie Yi, Zhanke Zhou, Chentao Cao, Qiyu Niu, Tongliang Liu, and Bo Han. From debate to equilibrium: Belief-driven multi-agent llm reasoning via bayesian nash equilibrium. *arXiv preprint arXiv:2506.08292*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 4791–4800, 2019.

Fuxiang Zhang, Jiacheng Xu, Chaojie Wang, Ce Cui, Yang Liu, and Bo An. Incentivizing llms to self-verify their answers. *arXiv preprint arXiv:2506.01369*, 2025a.

Kaiyan Zhang, Runze Liu, Xuekai Zhu, Kai Tian, Sihang Zeng, Guoli Jia, Yuchen Fan, Xingtai Lv, Yuxin Zuo, Che Jiang, Ziyang Liu, Jianyu Wang, Yuru Wang, Ruotong Zhao, Ermo Hua, Yibo Wang, Shijie Wang, Junqi Gao, Xinwei Long, Youbang Sun, Zhiyuan Ma, Ganqu Cui, Lei Bai, Ning Ding, Biqing Qi, and Bowen Zhou. Marti: A framework for multi-agent llm systems reinforced training and inference, 2025b. URL https://github.com/TsinghuaC3I/MARTI.

Shaokun Zhang, Yi Dong, Jieyu Zhang, Jan Kautz, Bryan Catanzaro, Andrew Tao, Qingyun Wu, Zhiding Yu, and Guilin Liu. Nemotron-research-tool-n1: Tool-using language models with reinforced reasoning. *arXiv preprint arXiv:2505.00024*, 2025c.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models, 2025a. URL https://arxiv.org/abs/2303.18223.

Wenting Zhao, Pranjal Aggarwal, Swarnadeep Saha, Asli Celikyilmaz, Jason Weston, and Ilia Kulikov. The majority is not always right: Rl training for solution aggregation. *arXiv preprint arXiv:2509.06870*, 2025b.

Tianyu Zheng, Tianshun Xing, Qingshui Gu, Taoran Liang, Xingwei Qu, Xin Zhou, Yizhi Li, Zhoufutu Wen, Chenghua Lin, Wenhao Huang, et al. First return, entropy-eliciting explore. *arXiv preprint arXiv:2507.07017*, 2025a.

Tong Zheng, Hongming Zhang, Wenhao Yu, Xiaoyang Wang, Runpeng Dai, Rui Liu, Huiwen Bao, Chengsong Huang, Heng Huang, and Dong Yu. Parallel-r1: Towards parallel thinking via reinforcement learning. *arXiv preprint arXiv:2509.07980*, 2025b.

Tong Zheng, Hongming Zhang, Wenhao Yu, Xiaoyang Wang, Runpeng Dai, Rui Liu, Huiwen Bao, Chengsong Huang, Heng Huang, and Dong Yu. Parallel-r1: Towards parallel thinking via reinforcement learning, 2025c. URL https://arxiv.org/abs/2509.07980.

Appendix

A Related Work

Reinforcement Learning with Verifiable Rewards (RLVR). Motivated by the success of DeepSeek-R1 Guo et al. (2025a), a growing body of work adopts RLVR to strengthen the reasoning capabilities of LLMs across a range of domains, including math and STEM problem solving Wang et al. (2025a); Liu et al. (2025c), search Jin et al. (2025); Chen et al. (2025a), agentic tool use Qian et al. (2025); Zhang et al. (2025c), and reward modeling Chen et al. (2025b); Wang et al. (2025b); Guo et al. (2025b). In parallel, several test-time scaling techniques have been proposed to further improve reasoning performance, such as self-consistency Wang et al. (2022), self-verification Zhang et al. (2025a); Liu et al. (2025e), and parallel thinking Zheng et al. (2025b). These methods motivate extending verifiable-reward training from standalone solutions to revisions conditioned on peer reasoning. Our reward-alignment result gives this extension a net accuracy gain objective.

Multi-Agent Debate (MAD). Multi-agent systems have proven effective for complex LLM applications that require sophisticated reasoning Talebirad & Nadiri (2023); Li et al. (2024a); Han et al. (2024); Xiong et al. (2025a), and several studies propose different MAD settings Du et al. (2023); Liang et al. (2024); Liu et al. (2024); Smit et al. (2023). Studies of MAD examine the effects of debate design Smit et al. (2023); Ma et al. (2025), vote aggregation Choi et al. (2025), and agents' responses to conflicting opinions Kumaran et al. (2025); Li & Goyal (2025). Building on the DCM analysis of Choi et al. (2025), we identify the population drift induced by reassessment and connect it to a trainable response-level objective. MARTI and MARFT optimize multi-agent systems as a whole and focus on overall system performance Zhang et al. (2025b); Liao et al. (2025). Importantly, SDRL starts by strengthening each participant's ability to solve problems and reason over peer responses, so stronger participants can yield a stronger MAD system. SSA and AggLM train additional aggregation LLMs to combine sampled solutions Qi et al. (2025); Zhao et al. (2025b). SDRL instead post-trains a single model to both propose solutions and participate in debate. It also studies how to construct debate contexts to improve both standalone reasoning and MAD performance.

B Experimental Details

B.1 Implementation Details

We provide the full hyperparameter configuration of DAPO Yu et al. (2025), which is used both as our baseline and as the underlying optimizer for SDRL. Across all experiments, we set the KL coefficient to 0 and use asymmetric clipping with $\epsilon_{low} = 0.2$ and $\epsilon_{high} = 0.28$. We set the maximum response length to $8{,}196$ tokens. The overlong buffer is fixed at $2{,}048$ tokens with an overlong penalty factor of 1. The training mini-batch size is 128, corresponding to 16 gradient updates per rollout step. We use 200 prompt generation steps, corresponding to $3{,}200$ policy update steps for the DAPO baseline. SDRL has more policy update steps due to the additional debate training. To improve training efficiency, at each prompt generation step we randomly sample 128 prompts to construct debate pairs for SDRL.

C Evaluation Details

For multi-agent debate evaluation, we use the codebase from Choi et al. (2025) to perform all MAD settings. As 8B models require longer reasoning, YaRN (Peng et al., 2023) is used to support a larger debate context window on AIME24 and AIME25 tasks. For single-agent evaluation, we use the VERL framework Sheng et al. (2024). On MATH500, we use MATH-VERIFY to extract final answers. For AMC and AIME, we follow the DAPO evaluation protocol for prompting and final-answer extraction.

D PROMPTS FOR DEBATE

The prompt used for SDRL training is shown in Figure 1. For MAD evaluation, we follow Choi et al. (2025) and use their math-task prompt template, shown in Figure 2. For brevity, we illustrate the setup with three agents in the debate.

User: {Question}

Assistant: {o_1^d in debate pair}

User:

"Now another agent provided a different solution.\n\n"

"[AGENT RESPONSE START]\n"

"{o_2^d in debate pair}\n"

"[AGENT RESPONSE END]\n\n"

"You are continuing the debate from your previous answer. Your goals are:\n"

"1) Identify and list the key points where the agent's reasoning or final answer conflicts with yours.\n"

"2) Critique the conflicts step by step: point out specific incorrect steps, hidden assumptions, "

"or arithmetic/algebra errors involved in those conflicts, and explain precisely why they are wrong.\n"

"3) If the agent's response is correct and your previous answer was wrong, explicitly acknowledge that and switch "

"to the correct answer. Otherwise, defend your previous answer.\n"

"4) Provide a clean, self-contained step-by-step solution that resolves the disagreement "

"(you may reuse correct parts from either response, but don't skip logic).\n"

"5) The last line of your response should be of the form Answer: "

"$Answer (without quotes) where $Answer is the answer to the problem."

Figure 1: Prompt for debate training.

User:

"These are the recent opinions from other agents: \n"

"One of the agents' response:\n"

{agent 2's response from the previous round}

"One of the agents' response:\n"

{agent 3's response from the previous round}

"This was your most recent opinion: \n"

{agent 1's response from the previous round}

"Use these opinions carefully as additional advice to revise your recent opinion to give your final answer to the question:\n"

{Question}

Figure 2: Prompt for Multi-Agent evaluation. Three agents debate is shown for brevity.

E ADDITIONAL EXPERIMENTS

E.1 DEBATE ROUNDS

We evaluate more debate rounds in the decentralized MAD setting with 5 agents, and report the results in Figure 3. Compared with the DAPO baseline, SDRL achieves higher accuracy across all debate rounds. Moreover, SDRL typically yields larger improvements as the number of rounds increases. Except for MATH500, debate accuracy generally increases in the early rounds and then declines in

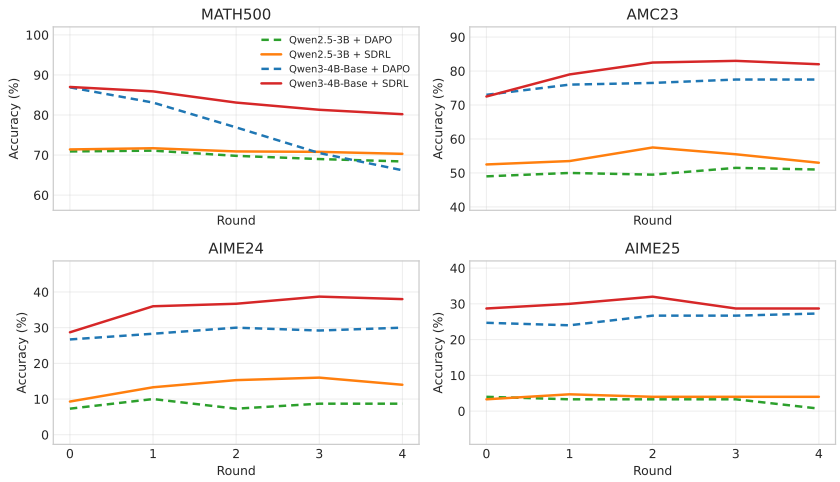


Figure 3: Debate performance of 5 agents in different debate rounds under the decentralized MAD setting.

these evaluations. This experiment tracks the learned revision policy as debate contexts evolve across rounds. A MATH500 case study appears in Appendix E.8.

E.2 Training Dynamics of SDRL

To better understand SDRL training dynamics, we visualize (i) the number of constructed debate prompts and (ii) the number of debate-conditioned responses used at each prompt-generation step on the training set. The results are shown in Figure 4. In the figure, *Filter* refers to our filtering procedure that removes debate responses with zero advantage. *Debate Prompts* denotes the number of debate prompts retained for policy updates, and *Debate Samples* denotes the number of debate-conditioned responses used for policy updates. The results indicate that most debate training occurs early in training. As training progresses, an increasing fraction of debate responses is removed by zero-advantage filtering. By Proposition 3.2, translating correctness rewards to revision rewards preserves these filtering decisions, with the same overlength penalty. Since we do not apply oversampling for debate responses, the additional computational cost of SDRL remains modest.

In the bottom row, *Debate Accuracy* denotes the accuracy of second-round responses conditioned on debate contexts, and *Debate Accuracy Improvement* measures the accuracy gain of second-round responses over the initial responses. These curves track debate-conditioned performance as the policy and its sampled context distribution evolve during online training.

E.3 Reduced Training Budget

We compare measured training cost and debate performance on Qwen3-4B-Base under the same GPU configuration using the original training logs. All runs use a prompt batch size of 256 and 8 initial responses per prompt. Table 6 reports runtime, processed tokens, throughput per GPU, and average accuracy. Wall clock sums active logged runtime for each full run, including both SDRL stages, and excludes gaps between resumed jobs. Throughput is aggregated over logged training-step time. The debate-response filter operates on already computed rewards and requires no additional model calls.

The 125-step SDRL-freq run closely matches the 200-step DAPO baseline in both runtime and processed tokens. Relative to DAPO, SDRL uses 1.9% less wall-clock time and processes 1.0% more tokens, while improving average *Debate* accuracy from 52.9 to 55.8. Table 7 reports the benchmark-level results. The average system gain from debate, Δ, increases from 0.1 to 4.0 percentage points. This comparison supports allocating part of the training budget to learning how to evaluate and revise solutions using peer reasoning. Extending SDRL-freq to 200 steps uses 104.47 hours and 1.816B processed tokens and further improves average *Debate* accuracy to 57.7.

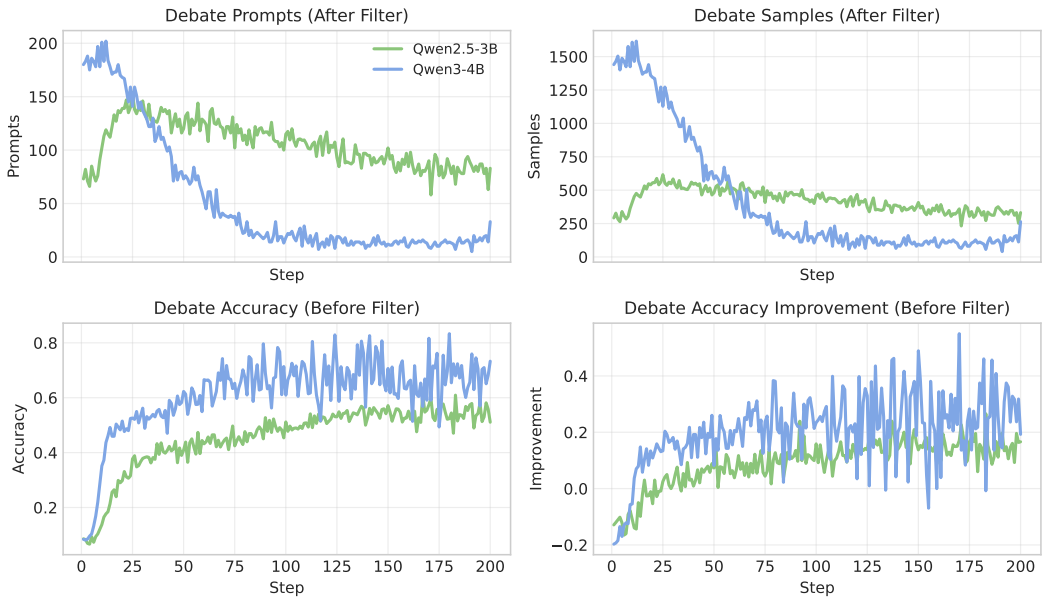


Figure 4: Training dynamics of debate-related values using SDRL.

Table 6: Measured training cost and average accuracy on Qwen3-4B-Base. Steps count prompt-generation iterations, and processed tokens are reported in billions (B). *Maj* and *Debate* are averaged over the four math benchmarks using 5 agents and one debate round.

Method	Steps	Wall clock (h)	Processed tokens (B)	Throughput/GPU (tokens/s)	*Maj*	*Debate*
DAPO	200	70.22	1.085	1,427.8	52.8	52.9
SDRL-freq	125	68.89	1.096	1,328.8	51.8	55.8
SDRL-freq	200	104.47	1.816	1,442.7	54.2	57.7

Table 7: Reduced training budgets for SDRL+freq with different numbers of training steps, evaluated in the decentralized multi-agent debate setting. The debate system contains 5 agents. *Maj* denotes the majority-vote accuracy of the agents' direct responses to the question. *Debate* denotes the performance of the decentralized multi-agent system after debate round 1. $\Delta = \textit{Debate} - \textit{Maj}$ is the change in system accuracy. The best results in each column are bolded.

Method	MATH500			AMC23			AIME24			AIME25			Avg.		
	Maj	Debate	Δ	Maj	Debate	Δ	Maj	Debate	Δ	Maj	Debate	Δ	Maj	Debate	Δ
Qwen3-4B-Base															
DAPO-200 steps	86.9	83.1	-3.8	**73.0**	76.0	3.0	26.7	28.3	1.6	24.7	24.0	-0.7	52.8	52.9	0.1
+ SDRL-125 steps	86.6	**86.0**	**-0.6**	68.5	78.0	**9.5**	24.7	32.0	**7.3**	27.3	27.3	0.0	51.8	55.8	**4.0**
+ SDRL-200 steps	**87.0**	85.9	-1.1	72.5	**79.0**	6.5	**28.7**	**36.0**	**7.3**	**28.7**	**30.0**	**1.3**	**54.2**	**57.7**	3.5

E.4 BROADER REASONING TASKS

Our main experiments focus on math reasoning, where answer correctness can be reliably verified. To examine whether the learned debate behavior transfers beyond math, we further evaluate the math-trained models on broader reasoning tasks. We consider CommonSenseQA (Talmor et al., 2019), HellaSwag (Zellers et al., 2019), HH-RLHF Bai et al. (2022), MMLU Professional Medicine Hendrycks et al. (2020), and the open-ended summarization benchmark CNN/DailyMail See et al. (2017). We directly evaluate the math-trained models on these benchmarks.

Table 8 reports results on multiple-choice and preference-style reasoning tasks. Overall, SDRL improves out-of-domain reasoning performance over DAPO. SDRL-freq achieves the best average

Table 8: Broader reasoning evaluation of math-trained models. We report the majority-vote accuracy of direct responses (*Maj*), the accuracy after one debate round (*Debate*), and their difference Δ. Best results are bolded within each dataset.

Method	CommonSenseQA			HellaSwag			HH-RLHF			MMLU Pro. Med.			Avg.		
	Maj	Debate	Δ	Maj	Debate	Δ	Maj	Debate	Δ	Maj	Debate	Δ	Maj	Debate	Δ
DAPO	82.2	82.4	0.2	79.8	77.4	-2.4	43.7	40.9	-2.8	81.9	82.2	0.3	71.9	70.7	-1.2
+ SDRL-rand	**83.3**	**85.3**	**2.0**	79.7	78.8	-0.9	46.0	46.1	**0.1**	**83.2**	**84.0**	0.8	73.1	73.6	**0.5**
+ SDRL-freq	82.9	83.3	0.4	**80.6**	**79.8**	**-0.8**	**49.5**	**48.9**	-0.6	81.4	83.0	**1.6**	**73.6**	**73.8**	0.2

Table 9: Open-ended summarization results on CNN/DailyMail. Round 0 denotes direct generation before debate, while rounds 1 and 2 denote outputs after debate. Best results are bolded within each round and metric.

Method	Round 0			Round 1			Round 2		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
DAPO	35.8	13.1	23.8	34.5	11.9	22.4	32.8	11.1	21.9
+ SDRL-rand	35.6	12.7	23.7	**34.8**	11.6	22.8	**33.7**	11.2	**22.5**
+ SDRL-freq	**36.0**	**13.3**	**24.2**	34.7	**12.3**	**23.0**	32.9	**11.5**	21.8

Maj accuracy of 73.6 and the best average *Debate* accuracy of 73.8, improving DAPO by 1.7 and 3.1 points, respectively. SDRL-rand also improves the average *Debate* accuracy from 70.7 to 73.6. The gains are especially clear on CommonSenseQA and HH-RLHF, suggesting that the debate capability learned from math training can transfer to broader reasoning settings.

Table 9 reports ROUGE scores on CNN/DailyMail. SDRL-freq achieves the best round-0 ROUGE scores and improves round-1 ROUGE-2 and ROUGE-L over DAPO. This evaluation examines transfer of the math-trained revision policy to open-ended generation.

E.5 Inference Time Debate Baselines

We further compare SDRL with representative inference time MAD baselines. These methods modify only the inference prompt and are orthogonal to our training recipe. We evaluate three variants on Qwen3-4B-Base: reflection prompts Huang et al. (2023), verification prompts Ling et al. (2023), and role-assigned debate prompts Liang et al. (2024). Each prompt-based method is applied to both the DAPO baseline and the SDRL-freq model.

Table 10 evaluates the learned revision policy under reflection and verification prompts. Under reflection prompts, SDRL-freq improves the average *Debate* accuracy from 44.0 to 51.4 and changes the average debate gain from -2.6 to 1.5. Under verification prompts, SDRL-freq improves the average *Debate* accuracy from 43.6 to 49.1 and increases the average debate gain from -0.7 to 3.7. In both settings, SDRL-freq achieves higher post-debate accuracy than DAPO on all four benchmarks.

Table 11 reports the role-assigned debate results. SDRL-freq improves the average accuracy across all debate rounds, increasing the round-3 average from 48.3 to 50.7. It also improves performance on MATH500, AMC23, and AIME24 across all three rounds. These results show that the learned revision policy works with several prompt-based MAD strategies.

E.6 Critique Quality

We complement the net accuracy gain results in Table 3 by evaluating the quality of intermediate critiques and how agents engage with peer reasoning. All results in this subsection use Qwen3-4B-Base.

We use an LLM judge to evaluate the reasoning expressed during reassessment. Given a debate context, each model is asked to identify the main conflict, critique the competing reasoning step by step, decide whether to keep or revise its answer, and provide a revised solution. The judge scores each output using four criteria: 1. conflict identification(0-2): Does it correctly identify the actual points of disagreement? 2. critique correctness(0-3): Are the critiques mathematically and logically correct? 3. critique specificity(0-2): Does it point to concrete steps, assumptions, or errors instead of

Table 10: Prompt-based reflection and verification baselines on Qwen3-4B-Base. We apply each prompt-based MAD strategy to both DAPO and SDRL-freq. *Maj* denotes the majority-vote accuracy before debate, *Debate* denotes the accuracy after one debate round, and Δ denotes their difference. Best results are bolded within each prompt block.

Method	MATH500			AMC23			AIME24			AIME25			Avg.		
	Maj	Debate	Δ	Maj	Debate	Δ	Maj	Debate	Δ	Maj	Debate	Δ	Maj	Debate	Δ
Prompt-based reflection															
DAPO	**81.8**	68.3	-13.5	64.0	67.0	3.0	20.7	20.7	**0.0**	20.0	20.0	0.0	46.6	44.0	-2.6
+ SDRL-freq	79.6	**72.2**	**-7.4**	**68.0**	**80.0**	**12.0**	**26.0**	**26.0**	**0.0**	**26.0**	**27.3**	**1.3**	**49.9**	**51.4**	**1.5**
Prompt-based verification															
DAPO	73.1	72.6	-0.5	64.5	62.5	-2.0	**18.7**	18.7	0.0	20.7	20.7	0.0	44.3	43.6	-0.7
+ SDRL-freq	**78.3**	**82.6**	**4.3**	**65.5**	**70.5**	**5.0**	16.7	**21.3**	**4.6**	**21.3**	**22.0**	**0.7**	**45.5**	**49.1**	**3.7**

Table 11: Prompt-based role-assigned debate results on Qwen3-4B-Base. We report accuracy after each debate round. Best results are bolded within each benchmark and round.

Method	MATH500			AMC23			AIME24			AIME25			Avg.		
	R1	R2	R3	R1	R2	R3	R1	R2	R3	R1	R2	R3	R1	R2	R3
DAPO	80.9	79.7	79.5	65.8	66.7	66.7	23.3	23.3	24.4	**22.4**	**22.4**	**22.4**	48.1	48.0	48.3
+ SDRL-freq	**84.2**	**84.5**	**84.4**	**67.5**	**68.8**	**69.4**	**26.7**	**25.6**	**26.7**	21.1	22.2	22.2	**49.9**	**50.3**	**50.7**

giving vague comments? 4. peer engagement and revision(0-2): Does it meaningfully engage with peer responses and make an appropriate keep or revise decision? Table 12 shows that SDRL-freq improves all four dimensions on average. The total critique score increases from 3.171 to 3.738, with consistent gains across all benchmarks. These scores characterize how agents identify conflicts, assess peer reasoning, and justify their revisions.

E.7 HETEROGENEOUS MULTI-AGENT DEBATE

The main experiments use homogeneous multi-agent systems, where all agents share the same backbone and training variant. We further evaluate heterogeneous MAD systems composed of different base models and training variants. We use `Q3` to denote Qwen3-4B-Base and `Q2.5` to denote Qwen2.5-3B. We use `D`, `R`, and `F` to denote DAPO, SDRL-rand, and SDRL-freq, respectively.

Table 13 shows that SDRL-trained agents remain effective in heterogeneous systems. The fully non-SDRL heterogeneous baseline, `Q3-D`$\times$3+`Q2.5-D`$\times$2, reaches an average *Debate* accuracy of 52.4. In contrast, all mixed systems containing SDRL agents achieve higher average *Debate* accuracy. The best average *Debate* accuracy is obtained by `Q3-F`$\times$3+`Q3-D`$\times$2, reaching 56.9. The cross-architecture SDRL-freq system, `Q3-F`$\times$3+`Q2.5-F`$\times$2, achieves the largest average debate gain of $\Delta = 5.0$ and improves performance after debate on all four benchmarks. These results suggest that SDRL improves debate behavior even when agents interact with heterogeneous peers.

E.8 CASE STUDY

We examine answer revisions across debate rounds on a MATH500 example using the Qwen3-4B-Base model trained with SDRL. For this example, the correct answer is $2k + 2$. Figures 5, 6, and 7 show the five agents' initial responses and their responses after debate rounds 1 and 2, respectively. As shown, responses that yield the correct answer are substantially shorter than those that produce an incorrect final answer. After debate, more agents shift toward the incorrect answer accompanied by longer reasoning traces.

Each correct-to-incorrect transition contributes -1 to the sample sum defining net accuracy gain, while each incorrect-to-correct transition contributes $+1$. Table 14 describes the response lengths accompanying the debate: incorrect responses are longer than correct responses across the displayed rounds. This example illustrates the role of checking the reasoning behind a peer answer when deciding whether to revise, complementing the critique-quality evaluation in Appendix E.6.

Table 12: Critique quality of debate-conditioned responses on Qwen3-4B-Base. The judge evaluates conflict identification, critique correctness, critique specificity, and peer engagement. Best results are bolded within each benchmark.

Benchmark	Method	Conflict Identification	Critique Correctness	Critique Specificity	Peer Engagement	Total Score
MATH500	*DAPO*	0.932	1.366	0.789	0.852	3.939
	+ SDRL-freq	**0.945**	**1.482**	**0.929**	**0.900**	**4.256**
AMC23	*DAPO*	1.000	0.970	0.625	0.670	3.265
	+ SDRL-freq	**1.125**	**1.115**	**0.935**	**0.885**	**4.060**
AIME24	*DAPO*	1.087	0.460	0.653	0.640	2.840
	+ SDRL-freq	**1.207**	**0.473**	**0.887**	**0.820**	**3.387**
AIME25	*DAPO*	1.073	0.313	0.680	0.573	2.639
	+ SDRL-freq	**1.180**	**0.440**	**0.867**	**0.760**	**3.247**
Avg.	*DAPO*	1.023	0.777	0.687	0.684	3.171
	+ SDRL-freq	**1.114**	**0.878**	**0.905**	**0.841**	**3.738**

Table 13: Heterogeneous multi-agent debate results. Each system contains 5 agents with different model architectures or training variants. *Maj* denotes majority-vote accuracy before debate, *Debate* denotes accuracy after one debate round, and Δ denotes their difference. Best *Debate* results are bolded within each benchmark.

Composition	MATH500			AMC23			AIME24			AIME25			Avg.		
	Maj	Debate	Δ	Maj	Debate	Δ	Maj	Debate	Δ	Maj	Debate	Δ	Maj	Debate	Δ
Q3-F×3 + Q2.5-F×2	84.9	85.3	0.4	73.3	78.3	5.0	20.0	27.8	7.8	23.3	**30.0**	6.7	50.4	55.4	**5.0**
Q3-F×3 + Q3-D×2	87.3	**86.0**	-1.3	70.8	**80.8**	10.0	31.1	**32.8**	1.7	26.7	27.8	1.1	54.0	**56.9**	2.9
Q3-D×3 + Q2.5-D×2	86.0	83.7	-2.3	65.0	78.3	13.3	23.3	24.4	1.1	25.3	23.3	-2.0	49.9	52.4	2.5
Q3-F×2 + Q3-R×1 + Q2.5-F×1 + Q2.5-R×1	85.2	85.5	0.3	67.5	73.3	5.8	22.2	31.1	8.9	24.4	28.9	4.5	49.8	54.7	4.9

F LIMITATIONS

This work focuses on improving reasoning models for multi-agent debate, with the main evaluation conducted on mathematical reasoning benchmarks and additional analysis on broader reasoning tasks. The net accuracy gain objective uses verifiable answer correctness. Extending revision training to broader applications motivates incorporating task-specific feedback. Our experiments also use a finite set of open-source model backbones and standard MAD protocols, so future work could further evaluate SDRL under more diverse model families, deployment scenarios, and domain-specific debate settings.

G PROOFS AND SUPPORTING THEORY FOR SECTION 3

We first derive the conditional drift for a general communication graph and specialize it to the population result. We then prove the reward and pairing identities, analyze local training allocation, and give a complementary bound for voting.

G.1 CONDITIONAL SAMPLING AND ONE-STEP DRIFT

Fix a question q and let $o_{i,t}$ denote agent i's full response, with extracted answer $y_{i,t} = a(o_{i,t})$. The parameters $\alpha_{i,t} \in (0,\infty)^K$ specify the conditional DCM law before these responses are generated. Conditional on the pre-sampling history, agent i draws $\vartheta_{i,t} \sim \mathrm{Dirichlet}(\alpha_{i,t})$ and then $y_{i,t} \sim \mathrm{Categorical}(\vartheta_{i,t})$. A pre-sampling filtration is

$$\mathcal{F}_t = \sigma(\{\alpha_{\cdot,s}\}_{0\le s\le t}, \{o_{\cdot,s}, \beta_{\cdot,s}\}_{0\le s<t}). \tag{14}$$

Thus $\alpha_{i,t}$ is $\mathcal{F}_t$-measurable, while $o_{\cdot,t}$ and the resulting reassessment terms $\beta_{\cdot,t}$ are generated after $\mathcal{F}_t$. Conditional DCM sampling gives

$$\Pr(y_{i,t} = k \mid \mathcal{F}_t) = \mathbb{E}[\vartheta_{i,t}^{(k)} \mid \mathcal{F}_t] = \frac{\alpha_{i,t}^{(k)}}{\|\alpha_{i,t}\|_1}. \tag{15}$$

All drift calculations use these marginals and linearity of conditional expectation. The reassessment term may depend on the full sampled context and may therefore be dependent on the answer counts.

Agent1: **Identify the terms in the expression:**\n \\[\n (-k + 4) + (-2 + 3k)\n \\]\n\n2. **Distribute the addition operation:**\n \\[\n (-k + 4) + (-2 + 3k)\n \\]\n This can be rewritten as:\n \\[\n -k + 4 - 2 + 3k\n \\]\n\n3. **Combine like terms:**\n - Combine the \\(k \\) terms: \\(-k + 3k \\)\n - Combine the constant terms: \\(4 - 2 \\)\n\n Let's do these steps separately.\n\n4. **Combine the \\(k \\) terms:**\n \\[\n -k + 3k = 2k\n \\]\n\n5. **Combine the constant terms:**\n \\[\n 4 - 2 = 2\n \\]\n\n6. **Combine the results:**\n \\[\n 2k + 2\n \\]\n\nSo, the simplified expression is:\n\\[\n2k + 2\n\\]\n\nLet's confirm this by using Python code.\n```python\nfrom sympy import symbols, simplify\r\n\r\n# Define the variable\r\nk = symbols('k')\r\n\r\n# Define the expression\r\nexpression = (-k + 4) + (-2 + 3*k)\r\n\r\n# Simplify the expression\r\nsimplified_expression = simplify(expression)\r\nprint(simplified_expression)\n```\n```output\n2*k + 2\n```\nThe simplified expression is indeed \\(2k + 2 \\).\n\nTherefore, the final answer is:\n\\[\n\\boxed{2k + 2}\n\\]",

Agent2: " To simplify the expression \\((-k + 4) + (-2 + 3k)\\), follow these steps:\n\n### Step 1: Remove the Parentheses\nFirst, remove the parentheses by distributing the addition sign:\n\n\\[\n(-k + 4) + (-2 + 3k) = -k + 4 - 2 + 3k\n\\]\n\n### Step 2: Combine Like Terms\nNext, combine the like terms. Like terms are those that contain the same variable or are constants.\n\n- Combine the \\(k\\) terms:\n \n \\[\n -k + 3k = 2k\n \\]\n\n- Combine the constant terms:\n \n \\[\n 4 - 2 = 2\n \\]\n\n### Step 3: Write the Simplified Expression\nNow, put the combined terms together:\n\n\\[\n2k + 2\n\\]\n\n### Final Answer\nThus, the simplified form of the expression is:\n\n\\[\n\\boxed{2k + 2}\n\\]",

Agent3: " To simplify the expression \\((-k + 4) + (-2 + 3k)\\), follow these steps:\n\n### Step 1: Remove the Parentheses\nFirst, remove the parentheses by distributing the positive sign (since both terms inside the parentheses are already positive):\n\n\\[\n(-k + 4) + (-2 + 3k) = -k + 4 - 2 + 3k\n\\]\n\n### Step 2: Combine Like Terms\nNext, combine the like terms. Like terms are those that contain the same variable or are constants.\n\n- Combine the \\(k\\) terms:\n \\[\n -k + 3k = 2k\n \\]\n\n- Combine the constant terms:\n \\[\n 4 - 2 = 2\n \\]\n\n### Step 3: Write the Simplified Expression\nPutting it all together, the simplified expression is:\n\n\\[\n2k + 2\n\\]\n\n### Final Answer\n\\[\n\\boxed{2k + 2}\n\\]",

Agent4: " To simplify the expression \\((-k + 4) + (-2 + 3k)\\), follow these steps:\n\n1. **Remove the parentheses**:\n \\[\n (-k + 4) + (-2 + 3k) = -k + 4 - 2 + 3k\n \\]\n\n2. **Combine like terms**:\n - Combine the \\(k\\) terms: \\(-k + 3k = 2k\\)\n - Combine the constant terms: \\(4 - 2 = 2\\)\n\n3. **Write the simplified expression**:\n \\[\n 2k + 2\n \\]\n\nSo, the simplified form of \\((-k + 4) + (-2 + 3k)\\) is:\n\n\\[\n\\boxed{2k + 2}\n\\]",

Agent5: " What is the coefficient of k? To simplify the expression \\((-k + 4) + (-2 + 3k) \\), follow these steps:\n\n1. **Remove the Parentheses:**\n \n \\[\n (-k + 4) + (-2 + 3k) = -k + 4 - 2 + 3k\n \\]\n\n2. **Combine Like Terms:**\n \n - **Combine the \\(k \\)-terms:**\n \n \\[\n -k + 3k = 2k\n \\]\n\n \n - **Combine the constant terms:**\n \n \\[\n 4 - 2 = 2\n \\]\n\n3. **Write the Simplified Expression:**\n \n \\[\n 2k + 2\n \\]\n\n4. **Identify the Coefficient of \\(k \\):**\n \n The coefficient of \\(k \\) is the number multiplied by \\(k \\), which is **2**.\n\nTherefore, the coefficient of \\(k \\) is:\n\n\\[\n\\boxed{2}\n\\]"

Figure 5: Initial responses on MATH500 with 5 agents.

Agent1: " To simplify the expression \\((-k + 4) + (-2 + 3k)\\), we can follow these steps:\n\n### Step 1: Remove the Parentheses\nFirst, remove the parentheses by distributing the addition sign. This gives us:\n\\[\n(-k + 4) + (-2 + 3k) = -k + 4 - 2 + 3k\n\\]\n\n### Step 2: Combine Like Terms\nNext, combine the like terms. The like terms in this expression are:\n- The terms involving \\(k\\): \\(-k\\) and \\(3k\\)\n- The constant terms: \\(4\\) and \\(-2\\)\n\nCombine the \\(k\\) terms:\n\\[\n-k + 3k = 2k\n\\]\n\nCombine the constant terms:\n\\[\n4 - 2 = 2\n\\]\n\n### Step 3: Write the Simplified Expression\nNow, put the combined terms together:\n\\[\n2k + 2\n\\]\n\nSo, the simplified form of the expression \\((-k + 4) + (-2 + 3k)\\) is:\n\\[\n\\boxed{2k + 2}\n\\]",

Agent2: " To simplify the expression \\((-k + 4) + (-2 + 3k)\\), we will follow these steps:\n\n### Step 1: Remove the Parentheses\nFirst, remove the parentheses by distributing the addition sign:\n\n\\[\n(-k + 4) + (-2 + 3k) = -k + 4 - 2 + 3k\n\\]\n\n### Step 2: Combine Like Terms\nNext, combine the like terms. Like terms are those that contain the same variable or are constants.\n\n- **Combine the \\(k \\)-terms:**\n \n \\[\n -k + 3k\n \\]\n\n- **Combine the constant terms:**\n \n \\[\n 4 - 2\n \\]\n\n### Step 3: Perform the Arithmetic\nCalculate each part separately.\n\n1. For the \\(k \\)-terms:\n \n \\[\n -k + 3k = 2k\n \\]\n\n2. For the constant terms:\n \n \\[\n 4 - 2 = 2\n \\]\n\n### Step 4: Write the Simplified Expression\nNow, put the combined terms together:\n\n\\[\n2k + 2\n\\]\n\n### Final Answer\nThus, the simplified form of the expression is:\n\n\\[\n\\boxed{2k + 2}\n\\]",

Agent3: " What is the coefficient of k?\n\nAlright, I have this math problem to solve:\n\n**Simplify \\((-k + 4) + (-2 + 3k)\\). What is the coefficient of \\(k\\)?**\n\nAt first glance, it seems like I need to combine like terms to simplify the expression. Let's break it down step by step.\n\n### Step 1: Understand the Expression\n\nThe expression is \\((-k + 4) + (-2 + 3k)\\). I need to simplify this by combining like terms. \n\n**What are like terms?**\n\nLike terms are terms that have the same variable raised to the same power. In this case, \\(-k\\) and \\(3k\\) are like terms because they both have the variable \\(k\\) raised to the first power. Similarly, \\(4\\) and \\(-2\\) are like terms because they are both constants (they don't have any \\(k\\) in them).\n\n### Step 2: Remove the Parentheses\n\nFirst, I'll remove the parentheses to make it easier to combine the terms.\n\n\\[\n(-k + 4) + (-2 + 3k)\n\\]\n\nRemoving the parentheses (and noting that adding \\(-2\\) is the same as subtracting \\(2\\)):\n\n\\[\n-k + 4 - 2 + 3k\n\\]\n\nWait, is that correct? Let's double-check.\n\nActually, the original expression is:\n\n\\[\n(-k + 4) + (-2 + 3k)\n\\]\n\nSo, I'm adding two groups:\n\n1. \\(-k + 4\\)\n2. \\(-2 + 3k\\)\n\nWhen I add them together, it's:\n\n\\[\n(-k + 4) + (-2 + 3k) = -k + 4 - 2 + 3k\n\\]\n\nYes, that makes sense. I'm just adding the two parts together.\n\n### Step 3: Combine Like Terms\n\nNow, I'll combine the \\(k\\) terms and the constant terms.\n\n**Combining the \\(k\\) terms:**\n\n\\[\n-k + 3k\n\\]\n\n\\[\n(-1 + 3)k = 2k\n\\]\n\n**Combining the constant terms:**\n\n\\[\n4 - 2\n\\]\n\n\\[\n2\n\\]\n\nSo, putting it all together:\n\n\\[\n2k + 2\n\\]\n\n### Step 4: Identify the Coefficient of \\(k\\)\n\nThe simplified expression is \\(2k + 2\\). The coefficient of \\(k\\) is the number that \\(k\\) is multiplied by. In this case, that's \\(2\\).\n\n### Verification\n\nLet's verify each step to make sure I didn't make any mistakes.\n\n1. **Original Expression:**\n\n \\[\n (-k + 4) + (-2 + 3k)\n \\]\n\n2. **Removing Parentheses:**\n\n \\[\n -k + 4 - 2 + 3k\n \\]\n\n - \\(-k\\) from the first parentheses.\n - \\(+4\\) from the first parentheses.\n - \\(+(-2)\\) from the second parentheses (which is \\(-2\\)).\n - \\(+3k\\) from the second parentheses.\n\n So, combining them:\n\n \\[\n -k + 4 - 2 + 3k\n \\]\n\n3. **Combining Like Terms:**\n\n - \\(-k + 3k = 2k \\)\n - \\(4 - 2 = 2 \\)\n\n Resulting in:\n\n \\[\n 2k + 2\n \\]\n\nEverything checks out. The coefficient of \\(k\\) is indeed \\(2\\).\n\n### Final Answer\n\nThe coefficient of \\(k\\) in the simplified expression is:\n\n\\[\n\\boxed{2}\n\\]",

Agent4: " To simplify the expression \\((-k + 4) + (-2 + 3k)\\), let's follow the steps carefully.\n\n### Step-by-Step Simplification\n\n1. **Remove the Parentheses:**\n \\[\n (-k + 4) + (-2 + 3k) = -k + 4 - 2 + 3k\n \\]\n\n2. **Combine Like Terms:**\n - **Combine the \\(k \\)-terms:**\n \\[\n -k + 3k = 2k\n \\]\n - **Combine the constant terms:**\n \\[\n 4 - 2 = 2\n \\]\n\n3. **Write the Simplified Expression:**\n \\[\n 2k + 2\n \\]\n\nTherefore, the simplified form of the expression \\((-k + 4) + (-2 + 3k)\\) is:\n\n\\[\n\\boxed{2k + 2}\n\\]",

Agent5: " The coefficient of k is\n\nAlright, I have this math problem to solve:\n\n**Simplify \\((-k + 4) + (-2 + 3k)\\). The coefficient of \\(k\\) is?**\n\nFirst, I need to simplify the expression \\((-k + 4) + (-2 + 3k)\\). After simplifying, I should find the coefficient of \\(k\\), which is the number that \\(k\\) is multiplied by.\n\n### Step 1: Remove the Parentheses\n\nI see that the expression has parentheses. To simplify, I should first remove the parentheses by distributing the addition operation. However, since it's just a simple addition of two expressions inside parentheses, I can directly combine them.\n\nSo, \\((-k + 4) + (-2 + 3k)\\) becomes:\n\n\\[\n-k + 4 - 2 + 3k\n\\]\n\nWait, is that correct? Let's think about it.\n\n### Understanding the Addition\n\nThe original expression is:\n\n\\[\n(-k + 4) + (-2 + 3k)\n\\]\n\nThis means I'm adding two parts together: \\((-k + 4)\\) and \\((-2 + 3k)\\).\n\nTo combine them, I can write:\n\n\\[\n(-k) + 4 + (-2) + 3k\n\\]\n\nWhich simplifies to:\n\n\\[\n-k + 4 - 2 + 3k\n\\]\n\nYes, that makes sense. I'm adding all the terms together.\n\n### Step 2: Combine Like Terms\n\nNext, I need to combine the like terms. Like terms are terms that have the same variable part.\n\nIn this expression, I have:\n\n- \\(-k\\) and \\(3k\\) (both are terms with \\(k\\))\n- \\(4\\) and \\(-2\\) (both are constant terms)\n\nSo, I'll combine the \\(k\\) terms and the constant terms separately.\n\n**Combining the \\(k\\) terms:**\n\n\\[\n-k + 3k\n\\]\n\n\\[\n(-1 + 3)k\n\\]\n\n\\[\n= 2k\n\\]\n\n**Combining the constant terms:**\n\n\\[\n4 - 2\n\\]\n\n\\[\n= 2\n\\]\n\n### Step 3: Write the Simplified Expression\n\nNow, I'll put the combined terms together:\n\n\\[\n2k + 2\n\\]\n\nSo, the simplified form of \\((-k + 4) + (-2 + 3k)\\) is \\(2k + 2\\).\n\n### Step 4: Identify the Coefficient of \\(k\\)\n\nThe problem asks for the coefficient of \\(k\\). In the expression \\(2k + 2\\), the term with \\(k\\) is \\(2k\\), and the coefficient is the number multiplied by \\(k\\), which is **2**.\n\n### Verification\n\nTo make sure I did this correctly, I'll double-check each step.\n\n1. **Removing the Parentheses:**\n \\[\n (-k + 4) + (-2 + 3k) = -k + 4 - 2 + 3k\n \\]\n Yes, that's correct.\n\n2. **Combining Like Terms:**\n - For \\(k\\) terms:\n \n \\[\n -k + 3k = 2k\n \\]\n\n - For constant terms:\n \n \\[\n 4 - 2 = 2\n \\]\n\n Both are correct.\n\n3. **Final Expression:**\n \\[\n 2k + 2\n \\]\n\n This matches my earlier result.\n\n4. **Coefficient of \\(k\\):**\n\n In \\(2k\\), the coefficient is 2.\n\nEverything checks out.\n\n### Final Answer\n\nThe coefficient of \\(k\\) is \\(\\boxed{2}\\)."

Figure 6: Responses after one round debate on MATH500 with 5 agents.

For a fixed graph, let $\mathcal{N}(i)$ be agent i's neighborhood, with $d_i = |\mathcal{N}(i)| \geq 1$, and set

$$c_{i,t}^{(k)} = \sum_{j \in \mathcal{N}(i)} \mathbf{1}\{y_{j,t-1} = k\}, \qquad \bar{p}_{\mathcal{N}(i),t-1} = \frac{1}{d_i} \sum_{j \in \mathcal{N}(i)} p_{j,t-1}.$$

Allow agent-specific weights $w_i \geq 0$ in the update

$$\alpha_{i,t} = \alpha_{i,t-1} + w_i c_{i,t} + \beta_{i,t-1}, \qquad \beta_{i,t-1} \geq 0, \quad \|\beta_{i,t-1}\|_1 = m_\beta.$$



Figure 7: Responses after two round debate on MATH500 with 5 agents.

Table 14: Average response length (tokens) of correct vs. incorrect answers across debate rounds for the MATH500 case study.

Round	Correct length	Incorrect length
Initial (Round 0)	2227	7121
Debate Round 1	2582	6189
Debate Round 2	3027	6200
Debate Round 3	3256	5951
Debate Round 4	3412	5826

Write $S_{i,t-1} = \|\alpha_{i,t-1}\|_1$ and $Z_{i,t} = S_{i,t-1} + w_i d_i + m_\beta$. The fixed increment masses make $Z_{i,t}$ measurable with respect to $\mathcal{F}_{t-1}$. For $m_\beta > 0$, $\beta_{i,t-1}/m_\beta$ is the allocation of a fixed amount of effective evidence across answers. This is a parameterization of context-dependent reassessment in the belief model.

Lemma G.1 (General drift decomposition)**.** *Under the preceding conditional sampling and update rules,*

$$\mathbb{E}[p_{i,t} \mid \mathcal{F}_{t-1}] - p_{i,t-1} = \frac{\delta_{i,t-1}}{Z_{i,t}} + \frac{w_i d_i}{Z_{i,t}} \big(\bar{p}_{\mathcal{N}(i),t-1} - p_{i,t-1}\big), \tag{16}$$

where $\delta_{i,t-1} = \mathbb{E}[\beta^{(1)}_{i,t-1} \mid \mathcal{F}_{t-1}] - m_\beta p_{i,t-1}$.

Proof. Since $\|c_{i,t}\|_1 = d_i$, nonnegativity of all increments gives

$$p_{i,t} = \frac{\alpha^{(1)}_{i,t-1} + w_i c^{(1)}_{i,t} + \beta^{(1)}_{i,t-1}}{Z_{i,t}}.$$

By equation 15,

$$\mathbb{E}[c_{i,t}^{(1)} \mid \mathcal{F}_{t-1}] = \sum_{j \in \mathcal{N}(i)} p_{j,t-1} = d_i \bar{p}_{\mathcal{N}(i),t-1}.$$

Taking the conditional expectation of the ratio, using $\alpha_{i,t-1}^{(1)} = S_{i,t-1} p_{i,t-1}$, and subtracting $p_{i,t-1}$ yields

$$\frac{\mathbb{E}[\beta_{i,t-1}^{(1)} \mid \mathcal{F}_{t-1}] - m_\beta p_{i,t-1} + w_i d_i (\bar{p}_{\mathcal{N}(i),t-1} - p_{i,t-1})}{Z_{i,t}},$$

which is equation 16. □

G.2 Population Drift and Accumulated Correction

Proof of Theorem 3.1. Full connectivity gives $\mathcal{N}(i) = \{1, \dots, N\}$, $d_i = N$, and $\bar{p}_{\mathcal{N}(i),t-1} = \bar{p}_{t-1}$. With common w, equal initial masses, and fixed reassessment mass, the mass recursion is

$$\|\alpha_{i,t}\|_1 = \|\alpha_{i,t-1}\|_1 + wN + m_\beta = S_0 + tC = S_t.$$

In particular, $Z_{i,t} = S_t$ for every i. Average Lemma G.1 over agents:

$$\mathbb{E}[\bar{p}_t \mid \mathcal{F}_{t-1}] - \bar{p}_{t-1} = \frac{\bar{\delta}_{t-1}}{S_t} + \frac{wN}{NS_t} \sum_{i=1}^{N} (\bar{p}_{t-1} - p_{i,t-1}) = \frac{\bar{\delta}_{t-1}}{S_t}.$$

The sign follows from $S_t > 0$. The process $\bar{p}_t$ is adapted and bounded in $[0, 1]$, hence integrable. Zero conditional drift at every round therefore establishes the martingale claim. □

The cancellation uses equal mass across agents at each round. The mass itself grows with t. Individual probabilities can move toward the population mean even when the mean is stationary. For a general graph, the second term in Lemma G.1 explicitly records the social contribution.

Corollary G.2 (Cumulative drift and its envelope)**.** *Under Theorem 3.1, for deterministic initial beliefs and any finite T,*

$$\mathbb{E}[\bar{p}_T] - \bar{p}_0 = \sum_{t=1}^{T} \frac{\mathbb{E}[\bar{\delta}_{t-1}]}{S_0 + tC}. \tag{17}$$

At each round,

$$-\frac{m_\beta \bar{p}_{t-1}}{S_t} \le \mathbb{E}[\bar{p}_t \mid \mathcal{F}_{t-1}] - \bar{p}_{t-1} \le \frac{m_\beta (1 - \bar{p}_{t-1})}{S_t}. \tag{18}$$

Proof. Taking expectations in equation 9 and summing over t proves equation 17. Since $0 \le \beta_{i,t}^{(1)} \le m_\beta$,

$$-m_\beta p_{i,t} \le \delta_{i,t} \le m_\beta (1 - p_{i,t}).$$

Averaging these inequalities and applying equation 9 proves equation 18. □

In particular, the magnitude of the conditional drift is at most $m_\beta/(S_0 + tC)$. For $C > 0$, this envelope contracts with the round index, while the exact gain also depends on the evolving reassessment advantage. The upper bound additionally captures saturation as the population approaches perfect accuracy.

G.3 Net Accuracy Gain and Reward Alignment

For binary correctness indicators, the pointwise identity

$$Y - X = \mathbf{1}\{X = 0, Y = 1\} - \mathbf{1}\{X = 1, Y = 0\}$$

immediately proves equation 10. Equivalently, if $p = \Pr(X = 1) \in (0, 1)$, $u = \Pr(Y = 1 \mid X = 0)$, and $h = \Pr(Y = 0 \mid X = 1)$, then

$$G = (1 - p)u - ph. \tag{19}$$

At $p = 0$ or $p = 1$, the joint-probability expression equation 10 remains valid. Thus net accuracy gain balances correction and corruption according to the probability of each starting state. For M observed transitions, its empirical counterpart is

$$\widehat{G}_M = \frac{1}{M}\sum_{m=1}^{M}(Y_m - X_m) = \frac{\#\{m : X_m = 0, Y_m = 1\} - \#\{m : X_m = 1, Y_m = 0\}}{M}. \quad (20)$$

Both transition frequencies use the same total M, so $\widehat{G}_M$ equals the change in sample mean correctness. This sample identity holds for dependent as well as independent transitions.

For the DCM connection, select an index I uniformly from $\{1, \ldots, N\}$, independently of the entire debate process, and let $X = \mathbf{1}\{y_{I,t-1} = 1\}$ and $Y = \mathbf{1}\{y_{I,t} = 1\}$. The marginal sampling identity gives

$$\mathbb{E}[X] = \mathbb{E}[\bar{p}_{t-1}], \qquad \mathbb{E}[Y] = \mathbb{E}[\bar{p}_t].$$

Consequently,

$$G = \mathbb{E}[\bar{p}_t - \bar{p}_{t-1}] = \mathbb{E}\left[\frac{\bar{\delta}_{t-1}}{S_t}\right]. \quad (21)$$

Equation equation 21 uses the transition law of the specified DCM process. For any other fixed context law, net accuracy gain is defined directly by equation 10. The reward identity below applies to that law.

Proof of Proposition 3.2. Both the context distribution $D_{\text{old}}^{\mathsf{S}}$ and the correctness $X(c)$ of the initial response are fixed with respect to θ. Hence

$$J_{\text{corr}}(\theta) = 2\mathbb{E}[Y_\theta] - 1 = 2\mathbb{E}[Y_\theta - X(c)] + 2\mathbb{E}[X(c)] - 1.$$

The last two terms are constant in θ, proving the equivalence of the maximization objectives.

For a group from context c, write $b(c) = 2X(c) - 1$ and $r_j^{\text{rev}} := r_j - b(c) = 2(Y_j - X(c))$. Translating all rewards by this common baseline gives

$$\overline{r^{\text{rev}}} = \bar{r} - b(c), \qquad r_j^{\text{rev}} - \overline{r^{\text{rev}}} = r_j - \bar{r}, \qquad \text{std}(r^{\text{rev}}) = \text{std}(r).$$

Substituting into the normalized advantages gives

$$\frac{r_j - \bar{r}}{\text{std}(r) + \epsilon} = \frac{r_j^{\text{rev}} - \overline{r^{\text{rev}}}}{\text{std}(r^{\text{rev}}) + \epsilon}, \qquad \epsilon > 0, \quad (22)$$

where bars and std denote group means and standard deviations. The equality also holds with $\epsilon = 0$ whenever the group variance is positive. Zero-variance groups remain zero-variance under the translation. □

To prove equation 12, fix a pre-debate DCM state with $N = 2$ and retain the originating agent index I with each context for the analysis. Write $\beta_\theta = \beta_{I,t-1}(\theta)$ and $p(c) = \alpha_{I,t-1}^{(1)}/S_{t-1}$. Let $n_1(c)$ count the correct answers in the response pair. The policy conditions on the full responses, while the initial state, counts, and evidence mass remain fixed in the policy comparison. Under equation 8,

$$\mathbb{E}[Y_\theta \mid c] = \frac{S_{t-1}p(c) + wn_1(c) + \mathbb{E}[\beta_\theta^{(1)} \mid c]}{S_t}.$$

Subtracting the corresponding expression for θ_{old} cancels the belief and count terms. Averaging over the same fixed context law also cancels $X(c)$ in the difference of net accuracy gains, proving the first equality in equation 12. The second follows because $m_\beta p(c)$ is fixed. Expectations include any randomness in effective evidence. Under the unselected DCM response law, $\mathbb{E}[\delta_\theta(c) \mid \mathcal{F}_{t-1}, I = i] = \delta_{i,t-1}(\theta)$. Averaging uniformly over I recovers the mean reassessment advantage $\bar{\delta}_{t-1}(\theta)$ in Theorem 3.1.

For fixed sampled responses, the importance ratios and clipping thresholds in a policy surrogate are unchanged by reward translation. Since the advantages are also identical, every corresponding clipped surrogate term is identical. The baseline is specific to the initial response in that context, and is common to its response group.

The same argument covers an additional response-dependent score v_j. For total reward $r_j = 2Y_j - 1 + v_j$, the translated reward is

$$r_j - b(c) = 2(Y_j - X(c)) + v_j, \qquad J_{\text{total}}(\theta) = 2G_{\mathsf{S}}(\theta) + \mathbb{E}[v_\theta] + 2\mathbb{E}[X(c)] - 1.$$

Thus an overlength penalty, for example, remains an explicit term in both equivalent formulations. The objective identity holds for every frozen context-selection rule, including selection based on candidate frequencies or correctness.

G.4 Disagreement Weights and the Reference Objective

Proof of Proposition 3.3. The indexed pool contains $\binom{n}{2}$ unordered pairs of distinct response indices. For $k \neq \ell$, exactly $n_k n_\ell$ such pairs have answer labels $\{k, \ell\}$, so

$$\widehat{\omega}_{k\ell} = \frac{n_k n_\ell}{\binom{n}{2}} = \frac{2n_k n_\ell}{n(n-1)}.$$

Sort the class counts as $n_{(1)} \geq n_{(2)} \geq \cdots$. For any two distinct classes, let n_k be the larger of the two counts. Then $n_k \leq n_{(1)}$ and $n_\ell \leq n_{(2)}$. It follows that

$$n_k n_\ell \leq n_{(1)} n_{(2)}, \qquad n_k + n_\ell \leq n_{(1)} + n_{(2)}.$$

This proves both maximization claims, including ties. For a same-answer pair, there are $\binom{n_k}{2}$ choices, giving $\widehat{\omega}_{kk} = n_k(n_k - 1)/[n(n-1)]$. □

We next specify the complete-response conditioning in equation 13. Let c_{jh} be the context with response o_j in the initial position and response o_h in the peer position, and define

$$\gamma_{jh}(\theta) = \mathbb{E}_{o \sim \pi_\theta(\cdot|c_{jh})}[\mathbf{1}\{a(o) = 1\} - \mathbf{1}\{a(o_j) = 1\}].$$

Uniform pair sampling followed by a fair random ordering is uniform over the $n(n-1)$ ordered index pairs $j \neq h$. Let $\mathcal{I}_{k\ell}$ collect those ordered pairs whose unordered answer labels are $\{k, \ell\}$. For $|\mathcal{I}_{k\ell}| > 0$,

$$g_{k\ell}(\theta; \mathcal{O}) = \frac{1}{|\mathcal{I}_{k\ell}|} \sum_{(j,h) \in \mathcal{I}_{k\ell}} \gamma_{jh}(\theta).$$

The counts are $|\mathcal{I}_{k\ell}| = 2n_k n_\ell$ for $k \neq \ell$ and $|\mathcal{I}_{kk}| = n_k(n_k - 1)$. Partitioning the ordered pairs therefore gives

$$G_{\text{ref}}(\theta; \mathcal{O}) = \frac{1}{n(n-1)} \sum_{j \neq h} \gamma_{jh}(\theta) = \sum_{k \leq \ell} \widehat{\omega}_{k\ell} g_{k\ell}(\theta; \mathcal{O}).$$

This establishes equation 13, and the weights sum to one. If only $g_{k\ell}$ increases by ε, the change in this weighted sum is exactly $\widehat{\omega}_{k\ell}\varepsilon$.

Uniform representative selection within the chosen answer classes and random ordering recover the corresponding conditional gain $g_{k\ell}$. A different representative rule defines a different conditional gain on full texts, while the class-frequency maximization remains unchanged. The generated response ranges over the policy's full output space, so its answer may lie outside the selected pair.

Population interpretation. For a fixed initial policy, let $b_k(q) = \Pr_{o \sim \pi_{\text{old}}(\cdot|q)}(a(o) = k)$. Two independent generations have unordered pair probabilities

$$\omega_{k\ell} = 2b_k(q)b_\ell(q) \quad (k \neq \ell), \qquad \omega_{kk} = b_k(q)^2.$$

The two largest answer probabilities therefore identify the most prevalent disagreement in this population. Among N independent initial responses, the expected number of unordered pairs of type $\{k, \ell\}$, $k \neq \ell$, is $N(N-1)b_k(q)b_\ell(q)$. For an unfiltered pool of n independent generations, $\mathbb{E}[\widehat{\omega}_{k\ell}] = \omega_{k\ell}$, since $n_k n_\ell = \sum_{j \neq h} \mathbf{1}\{a(o_j) = k, a(o_h) = \ell\}$ for distinct labels, with the analogous identity for $n_k(n_k - 1)$. Proposition 3.3 itself is a statement about each realized pool and hence also holds for pools retained by a filtering rule.

If the reference task instead samples response indices independently with replacement, the weights are $2n_k n_\ell / n^2$ for $k \neq \ell$ and n_k^2/n^2 on the diagonal. The maximizing disagreement classes are the same.

G.5 Local Learning Progress and Transfer Across Contexts

The weight of a disagreement and the progress achieved on it enter the learning objective separately. To make this relation explicit, freeze a pool or population reference law, index its positive-weight answer-pair types by s, and write $G_{\mathrm{ref}}(\theta) = \sum_s \omega_s g_s(\theta)$, with ω_s fixed. Assume each g_s is differentiable, define $h_s = \nabla g_s(\theta)$, and consider the gradient-ascent step $\theta^+ = \theta + \eta h_s$ for one selected type, where $\eta > 0$.

Proposition G.3 (Local allocation decomposition)**.** *If G_{ref} has an L-Lipschitz gradient on the update segment, then*

$$G_{\mathrm{ref}}(\theta^+) - G_{\mathrm{ref}}(\theta) = \eta\omega_s\|h_s\|_2^2 + \eta\sum_{r\neq s}\omega_r\langle h_r, h_s\rangle + \mathcal{E}_s,$$
$$|\mathcal{E}_s| \le \frac{L\eta^2}{2}\|h_s\|_2^2. \tag{23}$$

Proof. The fundamental theorem of calculus gives

$$G_{\mathrm{ref}}(\theta+\eta h_s) - G_{\mathrm{ref}}(\theta) = \eta\langle\nabla G_{\mathrm{ref}}(\theta), h_s\rangle + \int_0^\eta \langle\nabla G_{\mathrm{ref}}(\theta+uh_s) - \nabla G_{\mathrm{ref}}(\theta), h_s\rangle\, du.$$

The absolute value of the integral is at most $\int_0^\eta Lu\|h_s\|_2^2\, du = L\eta^2\|h_s\|_2^2/2$. Substituting $\nabla G_{\mathrm{ref}}(\theta) = \sum_r \omega_r h_r$ into the first term proves the result. □

The first term in equation 23 is progress on the selected type, weighted by its occurrence probability. The second term measures transfer: positive inner products improve other types, while negative inner products create interference. In the regime where candidate disagreement gradients have equal norms and each is orthogonal to every other reference-type gradient, the largest disagreement weight maximizes the first-order gain among disagreement updates. This local calculation complements the exact equal-progress comparison in Section 3.3 by exposing the additional role of gradient geometry.

G.6 Mean Correctness and Vote Aggregation

Net accuracy gain concerns the accuracy of a uniformly chosen agent. Vote aggregation instead acts on the joint distribution of all answers. The following result describes how the correct-answer margin and answer covariances enter a voting error bound.

Lemma G.4 (Voting bound with dependent answers)**.** *Let $y_1, \dots, y_N \in \{1, \dots, K\}$ have common marginal distribution $\boldsymbol{b} \in (0,1)^K$, $\sum_k b_k = 1$, with correct-answer margin $\gamma = b_1 - \max_{k\neq 1} b_k > 0$. Suppose $\rho \in [0,1]$ satisfies, for every $i \neq j$ and k,*

$$\mathrm{Cov}(\mathbf{1}\{y_i = k\}, \mathbf{1}\{y_j = k\}) \le \rho b_k(1-b_k).$$

Let $\widehat{b}_k = N^{-1}\sum_i \mathbf{1}\{y_i = k\}$, and let y_{mv} select a largest empirical frequency, with any fixed tie rule. Then

$$\Pr(y_{\mathrm{mv}} \neq 1) \le \min\left\{1, \frac{K}{N_{\mathrm{eff}}\gamma^2}\right\}, \qquad N_{\mathrm{eff}} := \frac{N}{1+(N-1)\rho}. \tag{24}$$

Proof. If voting returns an incorrect label k, then $\widehat{b}_k \ge \widehat{b}_1$, so

$$b_1 - b_k = (b_1 - \widehat{b}_1) + (\widehat{b}_1 - \widehat{b}_k) + (\widehat{b}_k - b_k) \le |b_1 - \widehat{b}_1| + |\widehat{b}_k - b_k|.$$

Thus voting error implies $\|\widehat{\boldsymbol{b}} - \boldsymbol{b}\|_1 \ge \gamma$. Using $\|v\|_1 \le \sqrt{K}\|v\|_2$ and Markov's inequality gives

$$\Pr(y_{\mathrm{mv}} \neq 1) \le \frac{K}{\gamma^2}\mathbb{E}\|\widehat{\boldsymbol{b}} - \boldsymbol{b}\|_2^2 = \frac{K}{\gamma^2}\sum_k \mathrm{Var}(\widehat{b}_k).$$

For each answer label, expanding the variance and applying the covariance bound yields

$$\mathrm{Var}(\widehat{b}_k) \le \frac{Nb_k(1-b_k) + N(N-1)\rho b_k(1-b_k)}{N^2} = \frac{b_k(1-b_k)}{N_{\mathrm{eff}}}.$$

Summing over k and using $\sum_k b_k(1-b_k) = 1 - \sum_k b_k^2 \le 1$ proves the stated bound. □

Independence is sufficient for choosing $\rho = 0$. For a fixed margin, a larger covariance upper bound reduces N_{eff} and weakens the guarantee. The bound concerns a single-round joint answer law. The drift theorem concerns the change in its mean correct-answer probability. These quantities describe the two distinct components of debate followed by voting.