SELF-IMPROVEMENT OF LANGUAGE MODELS BY POST-TRAINING ON MULTI-AGENT DEBATE

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-improvement, where models improve beyond their current performance without external supervision, remains a challenge. The core difficulty is sourcing a training signal stronger than what the model itself can currently produce. Majority voting has been shown to provide such a signal by aggregating over multiple samples, helping mitigate some of the inconsistencies in LM reasoning. In this work, we study multi-agent debate—where models collaborate and exchange reasoning over multiple rounds—as an alternative to single-round majority voting for constructing self-improvement signals. We introduce Multi-Agent Consensus Alignment (MACA), which uses reinforcement learning (RL) to post-train models on debate-derived signals. We study scalar consensus rewards and preference learning over majority and minority reasoning traces, comparing them with SFT on consensus-supporting traces. MACA improves (1) multi-agent debate accuracy (up to +26.87% on MATH), (2) single-agent accuracy (up to +21.51% on MathQA), and (3) self-consistency (up to +27.6% on GSM8K). We also observe gains on unseen benchmarks, including +16.3% on GPQA and +11.6% on CommonsenseQA. In a separate rollout-matched study, we evaluate online self-debate GRPO, in which one policy debates among its own rollouts during RL. The resulting policy improves held-out GPQA accuracy by 3.39% over the zero-shot base model.

1 INTRODUCTION

Self-improvement in language models, where models train on self-generated signals without external supervision, is challenging. LMs suffer from inconsistent reasoning, often producing contradictory solutions to the same problem under exploratory sampling (Holtzman et al., 2020; Wang et al., 2023). While inference-time aggregation methods such as majority voting and multi-agent debate help improve the reliability of the final response (Wang et al., 2023; Irving et al., 2018), they do not address the core problem: the internal reasoning instability that produces these inconsistencies in the first place. We argue that self-consistency, the ability to generate consistent outputs across diverse reasoning paths (Elazar et al., 2021; Wang et al., 2023), is a property of well-aligned reasoning systems. In the brain, consistency emerges through dedicated control mechanisms in prefrontal and anterior cingulate regions that resolve conflicts between competing neural activations, directing cognition toward coherent solutions (Miller & Cohen, 2001; Friston, 2010; Botvinick et al., 2004; Shenhav et al., 2013; Zhang et al., 2025). Language models, by contrast, lack mechanisms to align their diverse reasoning paths toward consensus.

Prior work shows that majority-vote signals can be an effective self-improvement signal (Wang et al., 2023; Zuo et al., 2025; Prasad et al., 2025). However, poor self-consistency can weaken this signal: a noisy majority can produce the wrong pseudo-label. This is a problem for the small models we study, where statistical aggregation can compound low-quality reasoning and debate can yield weak consensus.

We introduce Multi-Agent Consensus Alignment (MACA), a reinforcement learning framework that post-trains models on debate-derived signals. For each prompt, model clones first answer independently, then revise their responses after seeing anonymized peer reasoning. The final majority answer partitions the responses into consensus-supporting and dissenting trajectories for post-training. Unlike single-round majority voting, these trajectories retain the peer context that produced each revised response. MACA trains models to favor reasoning patterns that converge across peers.

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.

Updated policies can then generate debate data for the next training iteration (Fig. 1). Across our multi-agent experiments, MACA improves collaborative performance (up to +26.87% on MATH), individual reasoning (up to +21.51% on MathQA), and self-consistency (up to +27.6% on GSM8K), with gains on unseen reasoning domains, including +16.3% on GPQA and +11.6% on CommonsenseQA.

Given these debate-derived labels, we study scalar rewards through majority-vote GRPO (Shao et al., 2024) and preference labels for DPO (Rafailov et al., 2024) and KTO (Ethayarajh et al., 2024), comparing them with an MV-SFT baseline trained on consensus-supporting demonstrations. Beyond improving the multi-agent system, we ask whether debate can also drive single-agent self-improvement with online RL. In this setting, one shared policy samples multiple initial responses, generates follow-up responses conditioned on that set, and uses the follow-up majority as an online GRPO signal; the trained policy is then evaluated with a single rollout. We compare online self-debate GRPO with supervised GRPO and four label-free self-improvement methods, and study three alternative debate protocols. In a rollout-matched GPQA study, online self-debate GRPO improves accuracy by 3.39 percentage points over the zero-shot base model.

Key contributions.

- **Method.** We introduce MACA, a reinforcement learning framework that post-trains models on consensus signals derived from multi-agent debate.
- **Insight.** We study consensus signals from multi-agent debate, grounded in peer interaction, as an alternative to single-round majority voting for self-improvement. Iterative experiments show continued gains that diminish over successive training rounds.
- **Empirical validation.** Across multiple model families and reasoning benchmarks, debate-derived post-training improves multi-agent collaboration, single-agent reasoning, self-consistency, and transfer to unseen domains. Online self-debate also supports single-agent self-improvement.

2 Related Work

Existing approaches address sampling inconsistency primarily at inference time. Self-consistency prompting (Wang et al., 2023) samples multiple reasoning paths and selects the majority-voted answer, with extensions for non-verifiable outputs (Chen et al., 2023) and path pruning (Zhu et al., 2024). Self-Refine (Madaan et al., 2023) iteratively revises a response using self-generated feedback. Multi-agent debate (Du et al., 2023; Irving et al., 2018) instead lets several model instances exchange reasoning before producing a final answer. These methods improve inference without updating the underlying policy and require repeated inference-time computation.

Training-time self-improvement methods convert model-generated signals into supervision. TTRL (Zuo et al., 2025) uses the majority answer from independent samples as a pseudo-label for GRPO, while ScPO (Prasad et al., 2025) constructs preference pairs from single-round vote counts. Intuitor (Zhao et al., 2025) uses the policy's token-level certainty as a reward, and Self-Judge (Wu et al., 2026) uses model-based evaluation. We compare these methods with online self-debate GRPO under the same policy-rollout budget.

Prior work has also trained models from multi-agent interactions. Multiagent Finetuning (Subramaniam et al., 2025) applies SFT to reasoning generated through multi-agent interaction, MAGDi (Chen et al., 2024) distills structured interaction graphs into smaller models, and RLDF (Cheng et al., 2024) uses multi-role debate to train a reward model for bias mitigation. MACA studies consensus from same-policy, multi-round debate as a post-training signal with GRPO, DPO, and KTO, and compares these objectives with MV-SFT. Free-MAD (Cui et al., 2025) and Debate or Vote (Choi et al., 2025) study alternatives to standard consensus-based debate; we use adaptations of these protocols in our ablations.

3 Formalizing Self-Consistency

Given a prompt x, an LM with parameters θ defines a distribution $\pi_\theta(y \mid x) = \prod_{t=1}^{|y|} \pi_\theta(y_t \mid x, y_{<t})$ over reasoning trajectories y, from which answers $a = \mathrm{A}(y)$ are extracted. Under temperature sampling, the model samples from a modified distribution $\pi_{\theta,\tau}(y \mid x)$ where token proba-

bilities are adjusted by temperature $\tau > 0$. This induces an answer distribution $P_{\theta,\tau}(a \mid x) = \sum_{y:\mathrm{A}(y)=a} \pi_{\theta,\tau}(y \mid x)$, which gives each answer's probability by summing over all reasoning paths that lead to it. We denote the majority answer as $a^\star_{\theta,\tau}(x) = \arg\max_a P_{\theta,\tau}(a \mid x)$ with majority probability $S^+_{\theta,\tau}(x) = P_{\theta,\tau}(a^\star_{\theta,\tau}(x) \mid x)$. This represents the total probability mass concentrated on the most likely answer, or the model's internal consensus.

Temperature sampling enables exploration of diverse reasoning paths, but reduces the consistency of the final answer. While greedy decoding ($\tau = 0$) trivially approaches perfect consistency, it eliminates exploration and often produces suboptimal solutions (Holtzman et al., 2020). Lower temperatures increase consistency but restrict reasoning diversity. A self-consistent model should maintain high $S^+_{\theta,\tau}(x)$ even at high temperatures, allowing the model to access diverse reasoning trajectories while reliably converging on consistent answers. We measure self-consistency in two ways.

Single-agent sampling consistency. Computing $S^+_{\theta,\tau}(x)$ directly requires summing probabilities over all trajectories that lead to the majority answer, which is intractable. Instead, we estimate it by sampling t independent trajectories with answers $a_1, \ldots, a_t$ and computing:

$$\begin{aligned} s_t^{\theta,\tau}(x) &= \frac{1}{t}\sum_{i=1}^{t} \mathbf{1}[a_i(x) = \hat{a}(x)], \\ \hat{a}(x) &= \text{Majority}\{a_1(x), \ldots, a_t(x)\}. \end{aligned} \tag{1}$$

This measures the fraction of sampled trajectories that agree with the majority answer. As $t \to \infty$, $s_t^{\theta,\tau}(x) \to S^+_{\theta,\tau}(x)$, providing a consistent estimate of the true majority probability. We compute the majority over parseable answers; an unparseable response contributes zero agreement, and a sample set with no parseable answer receives a score of zero.

Multi-agent debate agreement. When M agents produce answers $a_1(x), \ldots, a_M(x)$ through deliberation, we measure the fraction of agents converging on the majority answer:

$$\begin{aligned} d_M^{\theta,\tau}(x) &= \frac{1}{M}\sum_{m=1}^{M} \mathbf{1}[a_m(x) = \hat{a}(x)], \\ \hat{a}(x) &= \text{Majority}\{a_1(x), \ldots, a_M(x)\}. \end{aligned} \tag{2}$$

Higher $d_M^{\theta,\tau}(x)$ indicates stronger consensus among agents within the sampled debate.

4 MACA: Multi-Agent Consensus Alignment

MACA uses M copies of the same model in a multi-round debate. Each agent generates an initial response, sees the other agents' reasoning, and updates its answer over $R - 1$ further rounds. We treat answers that persist through debate as a heuristic signal of stable reasoning.

MACA requires no task-specific ground-truth labels. For each prompt x, the debate produces final responses $\mathcal{Y}(x) = \{y_1, \ldots, y_M\}$ with extracted answers $a_m = \mathcal{A}(y_m)$. The majority consensus $\hat{a}(x) = \text{Majority}\{a_1, \ldots, a_M\}$ partitions $\mathcal{Y}(x)$ into consensus-supporting $\mathcal{G}^+(x) = \{y \in \mathcal{Y}(x) : \mathcal{A}(y) = \hat{a}(x)\}$ and dissenting $\mathcal{G}^-(x) = \{y \in \mathcal{Y}(x) : \mathcal{A}(y) \neq \hat{a}(x)\}$ trajectories. This creates a fixed post-training dataset $\mathcal{D}_{\text{post}} = \{(x, \hat{a}(x), \mathcal{G}^+(x), \mathcal{G}^-(x))\}_{x\in\mathcal{D}}$, where $\mathcal{D}$ is the original prompt set. Debate consensus provides the training signal. We study three RL objectives on these labels—GRPO, DPO, and KTO—and compare them with MV-SFT on consensus-supporting traces. In the schematic objectives below, x denotes the original question together with the anonymized peer responses from the preceding debate round. Alg. 1 in App. C gives the complete iterative debate and post-training loop.

Majority-Vote SFT (MV-SFT) baseline trains the model to mimic consensus-supporting trajectories:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{y^+\in\mathcal{G}^+(x)}[\log \pi_\theta(y^+|x)]. \tag{3}$$

Majority-Vote GRPO (MV-GRPO) uses online sampling with consensus-based rewards. For each prompt x, we sample multiple trajectories from the current policy and assign reward $r_x(y) =$

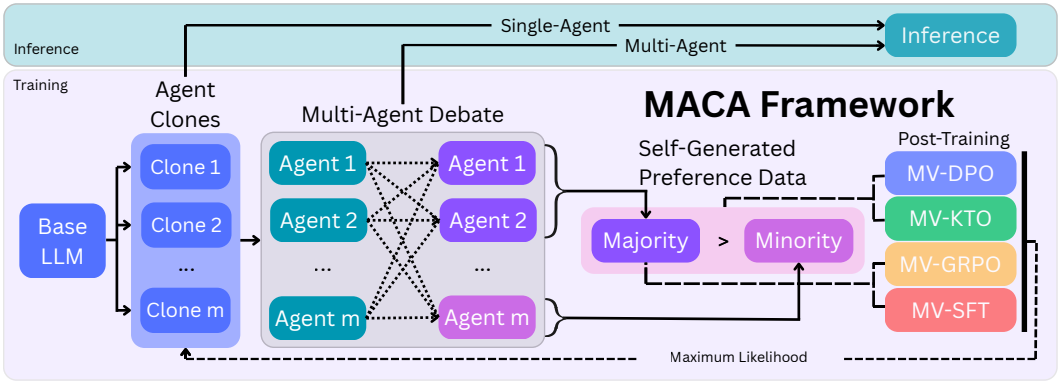


Figure 1: Multi-Agent Consensus Alignment framework: Multiple clones of a base LM debate to generate consensus-supporting and dissenting reasoning trajectories. MACA uses these signals with MV-GRPO, MV-DPO, or MV-KTO; the MV-SFT baseline imitates consensus-supporting traces. Updated agents can be evaluated individually or through multi-agent debate, or used for another training iteration.

$\mathbf{1}[\mathcal{A}(y) = \hat{a}(x)]$ based on whether each sample's answer matches the pre-computed consensus:

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{y\sim\pi_\theta}\left[\tilde{A}_x(y)\sum_t \log \pi_\theta(y_t \mid x, y_{<t})\right] + \lambda\,\text{KL}(\pi_\theta\|\pi_{\text{ref}}). \tag{4}$$

where $\tilde{A}_x(y) = r_x(y) - \bar{r}_x$ is the group-normalized advantage. Variation in agreement rewards provides the within-group contrast used by GRPO.

Majority-Vote DPO (MV-DPO) follows the standard DPO formulation with preference pairs constructed from our pre-generated debate outcomes:

$$\mathcal{L}_{\text{DPO}}(\theta) = -\mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{(y^+,y^-)\in\mathcal{G}^+(x)\times\mathcal{G}^-(x)} \left[\log\sigma\Big(\beta\big(\log\tfrac{\pi_\theta(y^+|x)}{\pi_{\text{ref}}(y^+|x)} - \log\tfrac{\pi_\theta(y^-|x)}{\pi_{\text{ref}}(y^-|x)}\big)\Big)\right]. \tag{5}$$

MV-DPO increases the relative likelihood of consensus-supporting reasoning traces over dissenting traces.

Majority-Vote KTO (MV-KTO) applies KTO's unpaired formulation with debate-derived labels from our fixed dataset, with class-balancing weights λ_+ and λ_-:

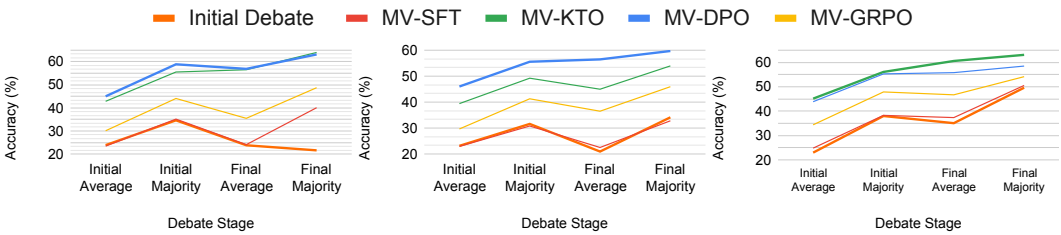
$$\mathcal{L}_{\text{KTO}}(\theta) = -\lambda_+\mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{y^+\in\mathcal{G}^+(x)}\left[\log\sigma\left(\beta\log\tfrac{\pi_\theta(y^+|x)}{\pi_{\text{ref}}(y^+|x)}\right)\right] - \lambda_-\mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{y^-\in\mathcal{G}^-(x)}\left[\log\sigma\left(-\beta\log\tfrac{\pi_\theta(y^-|x)}{\pi_{\text{ref}}(y^-|x)}\right)\right]. \tag{6}$$

KTO's unpaired structure handles imbalanced outcomes where majority trajectories dominate.

The main experiments first collect debate outcomes and fix their consensus labels. MV-SFT, MV-DPO, and MV-KTO train directly on the collected trajectories, while MV-GRPO samples new responses during optimization and rewards agreement with the fixed label. In Sec. 5.3, we study online self-debate GRPO, which regenerates both the responses and consensus label during RL. One shared policy samples four independent initial responses and four follow-up responses conditioned on the anonymized initial set. The follow-up majority becomes the pseudo-label for applying GRPO to the follow-up responses.

Table 1: Multi-agent debate accuracy after debate-derived post-training.

Model	Dataset	Pre / SFT (Baseline)		RL (Our Methods)			Best Δ
		Debate	MV-SFT	MV-GRPO	MV-KTO	MV-DPO	
Qwen2B	MATH	32.40	37.07 ± 3.07	39.00 ± 1.74	**46.47 ± 3.01**	42.60 ± 1.78	↑ 14.07
	GSM8K	49.60	50.53 ± 1.36	54.13 ± 2.02	**63.07 ± 0.64**	58.47 ± 1.62	↑ 13.47
	MathQA	24.20	26.27 ± 0.58	29.93 ± 4.99	**32.60 ± 0.72**	28.33 ± 0.31	↑ 8.40
Llama3B	MATH	37.80	35.33 ± 1.62	48.33 ± 2.19	**52.93 ± 0.99**	51.93 ± 1.67	↑ 15.13
	GSM8K	65.60	64.60 ± 1.59	68.60 ± 1.00	**73.13 ± 0.83**	71.67 ± 3.03	↑ 7.53
	MathQA	21.60	40.07 ± 2.00	48.73 ± 1.60	**64.00 ± 0.53**	63.13 ± 1.89	↑ 42.40
Phi4B	MATH	44.40	45.53 ± 2.53	**49.93 ± 1.33**	45.27 ± 0.70	46.73 ± 1.67	↑ 5.53
	GSM8K	79.60	78.93 ± 0.61	82.67 ± 1.81	82.47 ± 1.14	**84.73 ± 0.31**	↑ 5.13
	MathQA	49.60	50.87 ± 1.62	**63.07 ± 1.21**	51.53 ± 1.14	51.40 ± 1.44	↑ 13.47
Llama8B	MATH	32.80	34.13 ± 0.70	45.93 ± 1.03	53.93 ± 1.80	**59.67 ± 1.33**	↑ 26.87
	GSM8K	74.00	66.27 ± 1.01	81.53 ± 2.81	81.00 ± 1.97	**81.93 ± 1.51**	↑ 7.93
	MathQA	44.60	44.13 ± 1.10	57.27 ± 0.61	62.00 ± 2.03	**69.27 ± 1.55**	↑ 24.67



(a) Llama-3B on MATHQA (b) Llama-8B on MATH (c) Qwen-2B on GSM8K

Figure 2: Performance across stages of multi-agent debate after debate-aware RL. Stages: initial-round average, initial-round majority vote, final-round average, and final-round majority vote.

5 Results and Discussion

We evaluate MACA by post-training four instruction-tuned small LMs: Qwen2.5-1.5B-Instruct (Qwen2B) (Qwen Team, 2024), Llama-3.2-3B-Instruct (Llama3B) (Meta AI, 2024b), Phi-3-mini-128k-instruct (Phi4B) (Microsoft, 2024), and Llama-3.1-8B-Instruct (Llama8B) (Meta AI, 2024a). We train on MATH (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), and MathQA (Amini et al., 2019), with transfer evaluation on SVAMP (Patel et al., 2021), GPQA (Rein et al., 2023), and CommonsenseQA (Talmor et al., 2019). We use 4-bit quantization with QLoRA (Dettmers et al., 2023), limit responses to 256 tokens, and instantiate multi-agent debate with $M = 3$ clones and $R = 2$ rounds at temperature $\tau = 1.0$. Improvements persist when models are tested without token limits and without quantization (App. I.2, App. N). We compare (1) instruction-tuned base models, (2) the MV-SFT baseline, and (3) MACA variants using GRPO, DPO, and KTO. We use 1500/500/500 train/validation/test splits for each training dataset and report results across three seeds. The separate online experiment is described in Sec. 5.3. Debate prompts, training parameters, and other details are provided in App. D.

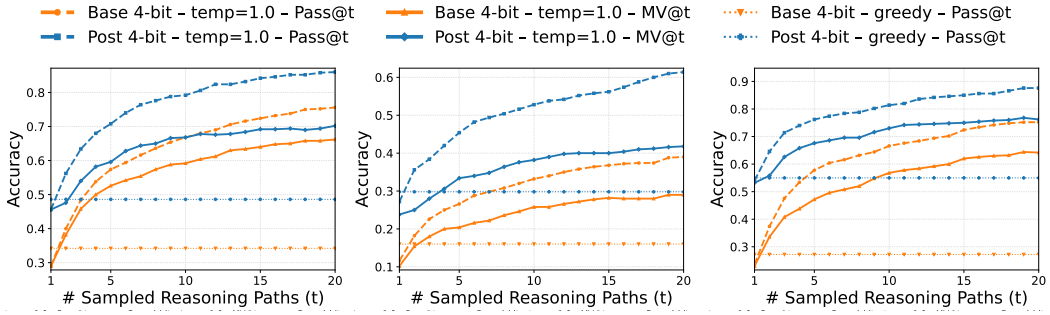
5.1 Multi-Agent Debate Performance

Impact on multi-agent debate setting. We evaluate MACA in the multi-agent setting using final-round majority-vote accuracy (the consensus after debate). Baselines are (1) the base model's debate outcome (Du et al., 2023) and (2) our MV-SFT baseline, which trains on majority traces. Table 1 reports results for scalar-reward MV-GRPO, preference-based MV-DPO and MV-KTO, and imitation-based MV-SFT across model–dataset settings.

Impact on debate dynamics. Fig. 2 summarizes four metrics: initial-round average single-agent accuracy, initial-round ensemble majority vote, final-round average single-agent accuracy conditioned

Table 2: Single-rollout accuracy after multi-agent debate post-training.

Model	Dataset	Base / SFT (Baseline)		RL (Our Methods)			Best Δ
		Base	MV-SFT	MV-GRPO	MV-KTO	MV-DPO	
Qwen2B	MATH	7.67	11.51 ± 0.60	18.09 ± 0.71	20.18 ± 0.67	**23.49 ± 2.30**	↑ 15.82
	GSM8K	23.00	24.84 ± 0.87	34.40 ± 2.08	**45.13 ± 1.80**	43.87 ± 1.92	↑ 22.13
	MathQA	5.00	5.25 ± 0.33	17.27 ± 2.88	**22.16 ± 1.14**	20.91 ± 0.47	↑ 17.16
Llama3B	MATH	27.87	25.89 ± 0.56	35.22 ± 0.44	40.64 ± 1.25	**40.71 ± 0.08**	↑ 12.84
	GSM8K	57.33	55.98 ± 0.68	52.40 ± 2.84	**65.76 ± 1.44**	64.98 ± 1.67	↑ 8.43
	MathQA	23.87	23.44 ± 0.73	30.09 ± 1.98	42.84 ± 0.67	**45.00 ± 2.23**	↑ 21.13
Phi4B	MATH	34.60	34.60 ± 0.82	**37.42 ± 0.16**	33.84 ± 0.78	34.62 ± 1.48	↑ 2.82
	GSM8K	67.27	69.58 ± 0.76	67.13 ± 3.60	75.60 ± 1.80	**76.87 ± 0.36**	↑ 9.60
	MathQA	34.87	34.04 ± 0.58	**45.52 ± 2.19**	33.91 ± 0.16	33.91 ± 0.50	↑ 10.65
Llama8B	MATH	22.93	23.16 ± 0.14	29.66 ± 1.27	39.42 ± 0.44	**46.00 ± 0.35**	↑ 23.07
	GSM8K	57.93	42.09 ± 1.28	62.45 ± 6.01	72.36 ± 1.34	**77.36 ± 0.27**	↑ 19.43
	MathQA	29.67	30.84 ± 0.60	33.07 ± 1.11	38.42 ± 1.22	**51.18 ± 0.24**	↑ 21.51



(a) Llama-3B on MATHQA (b) Qwen-2B on MATH (c) Qwen-2B on GSM8K

Figure 3: Sampling accuracy and self-consistency before and after post-training. Dashed: Pass@t (oracle upper bound), solid: MV@t (majority over t samples), dotted: greedy ($\tau = 0$) accuracy. (Blue): post-trained model. (Orange): base model. Curves computed over 500 prompts.

on peer reasoning, and final-round ensemble majority vote. Post-training produces its largest gains in the final round, where agents use peer context. We also observe that post-training removes the debate-related accuracy drop for Llama-3B on MathQA. App. F reports the corresponding peer-context ablation.

Single-rollout evaluation after multi-agent post-training. We also evaluate one initial response from each policy trained in the preceding multi-agent setup; no online self-debate occurs at evaluation. Table 2 reports results for MV-GRPO, MV-DPO, MV-KTO, and MV-SFT across 12 model–dataset pairs.

5.2 Self-Consistency and Accuracy after Multi-Agent Debate Post-Training

We measure the effect of post-training on sampling consistency $s_t^{\theta,\tau}(x)$, the fraction of sampled trajectories that match the majority answer. As formalized in Sec. 3, $s_t^{\theta,\tau}(x)$ converges to the modal probability $S_{\theta,\tau}^{+}(x)$ as $t \to \infty$. For each model, we sample 20 trajectories on 500 held-out prompts and evaluate $s_t^{\theta,\tau}(x)$ for $t = 1$ to 20 (App. I.1). At $t = 1$, the reported value reflects whether the response is parseable within the token window. MV-DPO and MV-KTO achieve these gains without explicit format rewards (App. L), which can otherwise be spurious (Srivastava et al., 2025). As sample size increases, the metric transitions to agreement across samples. The curves stabilize up to 27.6 percentage points above baseline, showing greater answer concentration under exploratory sampling. These gains persist without token limits (App. I.2).

Table 3: Single-rollout accuracy after post-training across reasoning benchmarks. Models are trained on the datasets in the columns and evaluated on the benchmarks in the rows. Bottom rows are unseen during training: SVAMP (math), GPQA (science), and CSQA (commonsense). All denotes joint training; arrows show absolute gains over the instruction-tuned model, and bold marks the largest gain for each model in each row. The Base column is from this separate stochastic single-policy evaluation; unlike Table 2, it is not averaged across three agents and three seeds.

Test	Qwen2B (post-trained on)					Llama3B (post-trained on)				
	Base	MATH	GSM	MQA	All	Base	MATH	GSM	MQA	All
MATH	10.4	↑ 10.0	↑ 3.8	↑ 10.8	↑ 12.2	32.0	↑ 19.4	↑ 16.4	↑ 18.2	↑ 21.2
GSM	27.0	↑ 20.0	↑ 25.6	↑ 22.6	↑ 27.8	69.6	↑ 6.0	↑ 6.8	↑ 8.4	↑ 10.8
MQA	7.4	↑ 12.6	↑ 17.0	↑ 15.4	↑ 21.4	24.6	↑ 14.0	↑ 13.4	↑ 21.2	↑ 21.6
SVAMP	48.3	↑ 19.0	↑ 18.0	↑ 17.0	↑ 27.7	71.3	↑ 6.0	↑ 6.4	↑ 9.7	↑ 7.1
GPQA	0.5	↑ 6.0	↑ 5.3	↑ 12.8	↑ 16.3	0.7	↑ 5.4	↑ 6.3	↑ 9.8	↑ 10.7
CSQA	3.8	↑ 19.8	↑ 43.0	↑ 54.0	↑ 59.6	53.0	↑ 7.4	↑ 10.6	↑ 11.6	↑ 11.0

Table 4: Single- and multi-agent accuracy with debate-majority-vote (MV) and ground-truth (GT) supervision. The Pre column reports the corresponding pre-training accuracy; bold marks the better supervision source within each objective and evaluation setting.

Llama-8B	Dataset	Pre	SFT		KTO		DPO		GRPO	
			GT	*MV*	*GT*	*MV*	*GT*	*MV*	*GT*	*MV*
Single-Agent	MATH	22.93	**23.73**	23.00	**41.20**	39.40	45.13	**46.40**	29.07	**31.13**
	GSM8K	57.93	40.20	**41.67**	**72.60**	70.87	76.33	**77.67**	61.27	**66.87**
Multi-Agent	MATH	32.80	**34.00**	33.40	**56.00**	55.80	**61.80**	60.80	**48.60**	44.80
	GSM8K	74.00	**65.20**	**65.20**	**81.20**	79.40	81.60	**83.0**	83.20	**84.20**

In multi-agent debate, we evaluate the agreement metric $d_M^{\theta,\tau}(x)$ defined in Sec. 3. Base small LMs initially struggle to reach meaningful consensus: for Qwen-2B on GSM8K, most of the "consensus" comes from random tie-breaking (1/3 agreement) or weak majorities (2/3), with only 13.4% reaching unanimity (Fig. 12, App. O). Post-training reduces non-parseable responses from 13.8% to 0.6% and no-agreement cases from 45.6% to 19.8%, while increasing unanimity from 13.4% to 43.4%. These changes demonstrate stronger within-debate agreement; the corresponding ground-truth accuracy is reported separately in Table 1.

Impact on inference-time sampling. We next examine how gains in self-consistency translate to inference-time performance under various sampling regimes. For each prompt, we draw t trajectories and report two metrics: Pass@t, the fraction of prompts for which at least one of the first t samples is correct (Chen et al., 2021), and MV@t, the fraction for which a majority vote over those t samples matches the ground truth (ties counted as incorrect) (Wang et al., 2023). As shown in Fig. 3, post-training (blue) (i) lifts greedy (τ=0) accuracy, (ii) increases MV@t at each fixed t, and (iii) raises Pass@t, the empirical sampling upper bound, indicating higher achievable accuracy at any given sampling budget. When additional inference compute is available (i.e., $t > 1$), sampling techniques continue to see gains on top of the post-training improvements, showing MACA is complementary to inference-time sampling. The same pattern holds with full-precision, with improvements comparable to their 4-bit counterparts (App. N). Finally, self-consistency gains strongly correlate with accuracy improvements ($r > 0.86$ across all tested inference conditions; see App. I).

Post-training on math datasets transfers to unseen benchmarks. Prior work finds that math training can improve general reasoning (DeepSeek-AI et al., 2026; Akter et al., 2025). Table 3 reports MV-DPO results for models trained on MATH, GSM8K, or MathQA, and on all three combined (All). Training on each dataset improves performance on every evaluated task, including unseen math (SVAMP), science (GPQA), and commonsense reasoning (CSQA). Joint training improves most settings further.

Debate-derived versus ground-truth supervision. Table 4 compares post-training with debate majority-vote labels, derived without task-specific ground-truth answers, to ground-truth labels on

Table 5: Single-agent held-out GPQA accuracy on Qwen3-4B after online RL. Methods are matched by the number of policy rollouts (eight per example). Values are mean ± one standard deviation across three training seeds. The zero-shot base model achieves 56.25%.

Method	Accuracy (%)
Supervised GRPO	55.21 ± 1.19
TTRL	55.99 ± 3.25
ScPO	55.73 ± 4.71
Intuitor	54.69 ± 2.07
Self-Judge	55.21 ± 4.58
MACA (online self-debate GRPO)	$\mathbf{59.64 \pm 2.39}$

Llama-8B. Debate-derived and ground-truth supervision trade wins across objectives and evaluation settings in this table. App. E extends this comparison to other models, and App. J reports the corresponding cross-domain results.

5.3 Online Self-Debate GRPO for Single-Agent Self-Improvement

The preceding experiments use fixed consensus labels. We next generate the consensus signal online during RL. In online self-debate GRPO, agents execute multiple rollouts from one shared policy, not separately parameterized models: the policy debates among its own responses. The resulting consensus supplies the reward signal, and we evaluate the trained policy with a single rollout. We compare this approach with other self-improvement methods matched by the number of training rollouts and examine alternative debate protocols.

Online self-debate as an RL sampling method. Here, *compute-matched* means matched by the number of policy rollouts. We evaluate Qwen3-4B-Instruct-2507 (Qwen3-4B) (Qwen Team, 2025) on GPQA using a 320/128 train/test split and report results across three training seeds. All methods use the same pretrained policy, task prompts, LoRA parameterization, optimization data, and eight policy rollouts per example. The single-round baselines sample eight independent responses. MACA uses one policy to sample four independent initial responses and then four follow-up responses conditioned on the anonymized initial set. Self-Judge additionally uses a frozen model to score its rollouts. In this experiment, unparseable responses do not agree with the pseudo-label, and examples without a usable majority are omitted. Additional training and evaluation details are in App. D.2.

The supervised baseline uses the GRPO objective (Shao et al., 2024), assigning gold-answer correctness rewards to sampled responses and optimizing their within-group relative advantages. Among label-free methods, TTRL (Zuo et al., 2025) uses the single-round majority answer as a pseudo-label for GRPO. ScPO (Prasad et al., 2025) forms a preference pair between a majority response and a dissenting response, then applies vote-margin-weighted DPO with a chosen-response negative log-likelihood term. Intuitor (Zhao et al., 2025) uses token-averaged self-certainty as the reward for GRPO. Our text-only adaptation of Self-Judge (Wu et al., 2026) combines within-group answer frequency with a frozen base model's assessment of plausibility, reasoning quality, and format. Online self-debate GRPO uses the follow-up debate majority as a pseudo-label and applies GRPO to the debate-conditioned responses according to agreement with that label.

Online self-debate GRPO achieves the highest mean accuracy, 3.39 percentage points above the zero-shot base model without gold-answer rewards. App. P reports the broader single-agent comparison across Qwen-2B, Llama-3B, Phi-4B, and Llama-8B on MATH and GSM8K (Table 23).

Robustness to alternative debate protocols. We compare online self-debate GRPO with three protocol ablations. *Initial-only GRPO* applies GRPO only to initial responses, labeled by the final debate decision. *Free-initial GRPO*, motivated by Free-MAD (Cui et al., 2025), uses anti-conformity debate and a trajectory-score decision in place of final-round majority voting, then applies GRPO to the initial responses. *Conformist-initial GRPO*, motivated by Debate or Vote (Choi et al., 2025), keeps responses that match the previous-round majority fixed while dissenting responses reconsider using peer reasoning.

Table 6: Single-agent held-out GPQA accuracy for Qwen3-4B after training with alternative self-debate protocols. Values are mean ± one standard deviation across three training seeds.

Method	Accuracy (%)
MACA (online self-debate GRPO)	$\mathbf{59.64 \pm 2.39}$
Initial-only GRPO	57.03 ± 4.75
Free-initial GRPO	54.69 ± 4.69
Conformist-initial GRPO	58.86 ± 2.39

Online self-debate GRPO outperforms Initial-only GRPO, Free-initial GRPO, and Conformist-initial GRPO (Table 6). The context ablation in App. F (Table 14) shows that conditioning on peer chains of thought improves final-round individual and majority-vote performance across model sizes.

Behavior under incorrect consensus. Across three seeds, the training debate majority was incorrect on $37.9\% \pm 2.3\%$ of examples. On these examples, paired pass@1 accuracy increased from $16.2\% \pm 0.8\%$ before post-training to $18.5\% \pm 2.7\%$ afterward. Corrections ($10.2\% \pm 2.4\%$) occurred more often than regressions ($7.9\% \pm 2.0\%$), although most examples ($73.7\% \pm 2.9\%$) remained incorrect. The mean change was $+2.3 \pm 2.7$ percentage points, compared with $+4.0 \pm 2.9$ points on correct-majority examples. Thus, MACA does not eliminate incorrect consensus, and its gains are smaller when the debate majority is wrong.

6 Conclusion, Limitations, and Future Work

We introduce MACA, a self-supervised post-training framework that uses consensus signals from multi-agent debate to improve reasoning. In the multi-agent setting, we study scalar consensus rewards and preference learning, comparing them with the MV-SFT baseline. Across these experiments, MACA improves collaborative performance (up to +26.87% on MATH), individual reasoning (up to +21.51% on MathQA), and self-consistency (up to +27.6% on GSM8K), while generalizing to unseen benchmarks, including +16.3% on GPQA and +11.6% on CommonsenseQA.

We also study online self-debate GRPO for single-agent self-improvement. In this rollout-matched experiment, held-out GPQA accuracy improves by 3.39 percentage points over the zero-shot base model.

MACA requires sufficient base-model competence to generate meaningful consensus and may amplify biases already present in model outputs. It also does not directly supervise the correctness of intermediate reasoning. Future work could study alternative consensus mechanisms, confidence-weighted voting, heterogeneous agents, and improved use of minority traces.

AI Usage Statement

This work studies how language models can learn from self-generated multi-agent debate. The reasoning traces, debate responses, and consensus labels used for post-training are generated by the models being trained and evaluated. This does not extend to the paper's claims, analyses, or conclusions, for which the authors take responsibility. A frozen language model is also used as a judge in the Self-Judge baseline to score response plausibility, reasoning quality, and format.

References

Syeda Nahida Akter, Shrimai Prabhumoye, Matvei Novikov, Seungju Han, Ying Lin, Evelina Bakhturina, Eric Nyberg, Yejin Choi, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Nemotron-crossthink: Scaling self-learning beyond math reasoning. (arXiv:2504.13941), April 2025. doi: 10.48550/arXiv.2504.13941. URL http://arxiv.org/abs/2504.13941. arXiv:2504.13941.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. 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. 2357–2367, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1245. URL http://aclweb.org/anthology/N19-1245.

Matthew M. Botvinick, Jonathan D. Cohen, and Cameron S. Carter. Conflict monitoring and anterior cingulate cortex: an update. *Trends in Cognitive Sciences*, 8(12):539–546, December 2004. ISSN 13646613. doi: 10.1016/j.tics.2004.10.003. URL https://linkinghub.elsevier.com/retrieve/pii/S1364661304002657.

Justin Chih-Yao Chen, Swarnadeep Saha, Elias Stengel-Eskin, and Mohit Bansal. MAGDi: Structured distillation of multi-agent interaction graphs improves reasoning in smaller language models. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2402.01620.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. (arXiv:2107.03374), 2021. doi: 10.48550/arXiv.2107.03374. URL http://arxiv.org/abs/2107.03374. arXiv:2107.03374.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. Universal self-consistency for large language model generation. (arXiv:2311.17311), November 2023. doi: 10.48550/arXiv.2311.17311. URL http://arxiv.org/abs/2311.17311. arXiv:2311.17311.

Ruoxi Cheng, Haoxuan Ma, Shuirong Cao, Jiaqi Li, Aihua Pei, Zhiqiang Wang, Pengliang Ji, Haoyu Wang, and Jiaqi Huo. Reinforcement learning from multi-role debates as feedback for bias mitigation in LLMs. *arXiv preprint arXiv:2404.10160*, 2024. URL https://arxiv.org/abs/2404.10160.

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. URL https://arxiv.org/abs/2508.17536.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. (arXiv:2110.14168), November 2021. doi: 10.48550/arXiv.2110.14168. URL http://arxiv.org/abs/2110.14168. arXiv:2110.14168.

Yu Cui, Hang Fu, Haibin Zhang, Licheng Wang, and Cong Zuo. Free-MAD: Consensus-free multi-agent debate. *arXiv preprint arXiv:2509.11035*, 2025. URL https://arxiv.org/abs/2509.11035.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang,

Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. (arXiv:2501.12948), January 2026. doi: 10.48550/arXiv.2501.12948. URL http://arxiv.org/abs/2501.12948. arXiv:2501.12948.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. (arXiv:2305.14314), May 2023. doi: 10.48550/arXiv.2305.14314. URL http://arxiv.org/abs/2305.14314. arXiv:2305.14314.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. (arXiv:2305.14325), May 2023. doi: 10.48550/arXiv.2305.14325. URL http://arxiv.org/abs/2305.14325. arXiv:2305.14325.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031, 2021. doi: 10.1162/tacl_a_00410. URL https://aclanthology.org/2021.tacl-1.60/.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. (arXiv:2402.01306), November 2024. doi: 10.48550/arXiv.2402.01306. URL http://arxiv.org/abs/2402.01306. arXiv:2402.01306.

Karl Friston. The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11(2):127–138, February 2010. ISSN 1471-003X, 1471-0048. doi: 10.1038/nrn2787. URL https://www.nature.com/articles/nrn2787.

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:2103.03874), November 2021. doi: 10.48550/arXiv.2103.03874. URL http://arxiv.org/abs/2103.03874. arXiv:2103.03874.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. (arXiv:1904.09751), February 2020. doi: 10.48550/arXiv.1904.09751. URL http://arxiv.org/abs/1904.09751. arXiv:1904.09751.

Geoffrey Irving, Paul Christiano, and Dario Amodei. Ai safety via debate. (arXiv:1805.00899), October 2018. doi: 10.48550/arXiv.1805.00899. URL http://arxiv.org/abs/1805.00899. arXiv:1805.00899.

knoveleng. AMC-23, 2023. URL https://huggingface.co/datasets/knoveleng/AMC-23.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*, 2023. URL https://arxiv.org/abs/2303.17651.

Meta AI. Llama-3.1-8B-Instruct. https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct, 2024a.

Meta AI. Llama-3.2-3B-Instruct. https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct, 2024b.

Microsoft. Phi-3-mini-128k-instruct. https://huggingface.co/microsoft/Phi-3-mini-128k-instruct, 2024.

Earl K. Miller and Jonathan D. Cohen. An integrative theory of prefrontal cortex function. *Annual Review of Neuroscience*, 24(1):167–202, March 2001. ISSN 0147-006X, 1545-4126. doi: 10.1146/annurev.neuro.24.1.167. URL https://www.annualreviews.org/doi/10.1146/annurev.neuro.24.1.167.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are nlp models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2080–2094, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.168. URL https://aclanthology.org/2021.naacl-main.168.

Archiki Prasad, Weizhe Yuan, Richard Yuanzhe Pang, Jing Xu, Maryam Fazel-Zarandi, Mohit Bansal, Sainbayar Sukhbaatar, Jason Weston, and Jane Yu. Self-consistency preference optimization. (arXiv:2411.04109), 2025. doi: 10.48550/arXiv.2411.04109. URL http://arxiv.org/abs/2411.04109. arXiv:2411.04109.

Qwen Team. Qwen2.5-1.5B-Instruct. https://huggingface.co/Qwen/Qwen2.5-1.5B-Instruct, 2024.

Qwen Team. Qwen3-4B-Instruct-2507. https://huggingface.co/Qwen/Qwen3-4B-Instruct-2507, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. (arXiv:2305.18290), 2024. doi: 10.48550/arXiv.2305.18290. URL http://arxiv.org/abs/2305.18290. arXiv:2305.18290.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof qa benchmark. (arXiv:2311.12022), November 2023. doi: 10.48550/arXiv.2311.12022. URL http://arxiv.org/abs/2311.12022. arXiv:2311.12022.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. (arXiv:2402.03300), April 2024. doi: 10.48550/arXiv.2402.03300. URL http://arxiv.org/abs/2402.03300. arXiv:2402.03300.

Amitai Shenhav, Matthew M. Botvinick, and Jonathan D. Cohen. The expected value of control: An integrative theory of anterior cingulate cortex function. *Neuron*, 79(2):217–240, 2013. ISSN 08966273. doi: 10.1016/j.neuron.2013.07.007. URL https://linkinghub.elsevier.com/retrieve/pii/S0896627313006077.

Pragya Srivastava, Harman Singh, Rahul Madhavan, Gandharv Patil, Sravanti Addepalli, Arun Suggala, Rengarajan Aravamudhan, Soumya Sharma, Anirban Laha, Aravindan Raghuveer, Karthikeyan Shanmugam, and Doina Precup. Robust reward modeling via causal rubrics. (arXiv:2506.16507), 2025. doi: 10.48550/arXiv.2506.16507. URL http://arxiv.org/abs/2506.16507. arXiv:2506.16507.

Vighnesh Subramaniam, Yilun Du, Joshua B. Tenenbaum, Antonio Torralba, Shuang Li, and Igor Mordatch. Multiagent finetuning: Self improvement with diverse reasoning chains. (arXiv:2501.05707), March 2025. doi: 10.48550/arXiv.2501.05707. URL http://arxiv.org/abs/2501.05707. arXiv:2501.05707.

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, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL http://aclweb.org/anthology/N19-1421.

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:2203.11171), March 2023. doi: 10.48550/arXiv.2203.11171. URL http://arxiv.org/abs/2203.11171. arXiv:2203.11171.

Zhengxian Wu, Kai Shi, Chuanrui Zhang, Zirui Liao, Jun Yang, Ni Yang, Qiuying Peng, Luyuan Zhang, Hangrui Xu, Tianhuang Su, Zhenyu Yang, Haonan Lu, and Haoqian Wang. When models judge themselves: Unsupervised self-evolution for multimodal reasoning. *arXiv preprint arXiv:2603.21289*, 2026. URL https://arxiv.org/abs/2603.21289.

Shuo Zhang, Yan Tian, Quanying Liu, and Haiyan Wu. The neural correlates of novelty and variability in human decision-making under an active inference framework. *eLife*, 13:RP92892, March 2025. ISSN 2050-084X. doi: 10.7554/eLife.92892.4. URL https://elifesciences.org/articles/92892.

Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. Learning to reason without external rewards. *arXiv preprint arXiv:2505.19590*, 2025. URL https://arxiv.org/abs/2505.19590.

Jiace Zhu, Yuanzhe Huang, Yingtao Shen, Jie Zhao, and An Zou. Path-consistency with prefix enhancement for efficient inference in llms. (arXiv:2409.01281), August 2024. doi: 10.48550/arXiv.2409.01281. URL http://arxiv.org/abs/2409.01281. arXiv:2409.01281.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, Biqing Qi, Youbang Sun, Zhiyuan Ma, Lifan Yuan, Ning Ding, and Bowen Zhou. Ttrl: Test-time reinforcement learning. (arXiv:2504.16084), April 2025. doi: 10.48550/arXiv.2504.16084. URL http://arxiv.org/abs/2504.16084. arXiv:2504.16084.

A EXAMPLES OF POST-TRAINED CONSISTENT REASONING IN SINGLE- AND MULTI- AGENT SETTINGS

This appendix presents examples from the debates, providing a qualitative view of how post-training affects both single-agent reasoning and multi-agent collaboration.

A.1 OVERVIEW OF KEY FINDINGS

In the selected single-agent examples, post-trained responses show clearer step-by-step decomposition, fewer calculation errors, and more concise completion within the token limit.

In the selected multi-agent examples, agents reference and build upon peer solutions, sometimes resolving mistakes through discussion rather than propagating them.

These examples complement the quantitative analyses of reasoning quality, concision, and consensus formation reported elsewhere.

A.2 CASE STUDY 1: SINGLE-AGENT MATHEMATICAL REASONING

A.2.1 PROBLEM: CITIES A AND B RELATIVE SPEED

AMC Problem Statement

Cities A and B are 45 miles apart. Alicia lives in A and Beth lives in B. Alicia bikes towards B at 18 miles per hour. Leaving at the same time, Beth bikes toward A at 12 miles per hour. How many miles from City A will they be when they meet?
Correct Solution: Combined speed = 18 + 12 = 30 mph; Time to meet = 45/30 = 1.5 hours; Distance from A = 18 × 1.5 = 27 miles
Ground Truth: 27

A.2.2 PRE-TRAINING RESPONSE: FUNDAMENTAL CONCEPTUAL CONFUSION

Base Model: Conceptual Breakdown

To solve this problem, we can use the concept of relative motion. Here's a step-by-step breakdown:

- Let's denote the time that Alicia travels as t hours. Since Beth travels in the opposite direction, we can say that the time she travels is also t hours.
- Since they are traveling towards each other, the distance between them will decrease at a rate of $18 + 12 = 30$ miles per hour.
- The initial distance between them is 45 miles. After t hours, the total distance traveled by both Alicia and Beth will be the initial distance plus the distance traveled together, which is $45 + 30t$ miles.
- We can set up an equation using the fact that their combined distance traveled equals the initial distance between them: $45 + 30t = 45$. This is because they meet when the sum of their distances is equal to the initial distance.
- Solving for t, we get $t = 0$ hours. However, this doesn't make sense because it means they meet immediately after starting. Let's try again.
- Since we can't set up a simple equation with this method, let's think about it in terms of their relative speed: 12

[Response cuts off mid-sentence]
Parsed Answer: null

Analysis: Multiple failure modes. The pre-training response reveals several critical reasoning failures: incorrect problem setup, misapplication of distance-time relationships, and ultimately an

illogical result. The model exhausts its token budget exploring unproductive solution paths before reaching a conclusion.

A.2.3 Post-Training Response: Correct Systematic Solution

Post-Training: Correct and Concise

To solve this problem, we can follow these steps:

- We know the distance between Cities A and B is 45 miles.
- Alicia and Beth are biking towards each other at a combined speed of 18 + 12 = 30 miles per hour.
- Since they are biking towards each other, we can use the concept of relative speed to find the time it takes for them to meet.
- We can calculate the time using the formula: time = distance / speed. Using this formula, the time they take to meet is 45 miles / 30 miles per hour, which equals 1.5 hours.
- Now that we have the time, we can find the distance covered by Alicia, since we know her speed is 18 miles per hour. Distance covered by Alicia is 18 × 1.5, which equals 27 miles.

Therefore, Alicia and Beth will meet 27 miles from City A.
$\boxed{27}$

Analysis: Systematic improvement. In this example, the post-trained response applies the correct concept, proceeds logically, and completes within the token limit without an explicit format reward.

A.3 Case Study 2: Multi-Agent Mathematical Reasoning

A.3.1 Problem: Road Trip Gas Mileage Calculation

Problem Statement

Sophia is thinking of taking a road trip in her car, and would like to know how far she can drive on a single tank of gas. She has traveled 100 miles since last filling her tank, and she needed to put in 4 gallons of gas to fill it up again. The owner's manual for her car says that her tank holds 12 gallons of gas. How many miles can Sophia drive on a single tank of gas?
Correct Solution: 100 ÷ 4 = 25 mpg; 25 × 12 = 300 miles
Ground Truth: 300

A.4 Complete Multi-Agent Transcripts: Pre-Training Failure (Iteration 0)

A.4.1 Round 1: Initial Agent Responses

User Prompt

Solve the following math problem. Sophia is thinking of taking a road trip in her car, and would like to know how far she can drive on a single tank of gas. She has traveled 100 miles since last filling her tank, and she needed to put in 4 gallons of gas to fill it up again. The owner's manual for her car says that her tank holds 12 gallons of gas. How many miles can Sophia drive on a single tank of gas?
Provide a bullet point summary of your step-by-step reasoning. Your final answer should be a single numerical number, in the form \boxed{answer}, at the end of your response.

Agent 1: Mathematical Error

To solve this problem, I need to determine Sophia's car's fuel efficiency and then calculate how far she can drive on a full tank.
Step 1: Determine fuel efficiency from the given information.

- Sophia traveled 100 miles and used 4 gallons of gas
- Fuel efficiency = 100 miles ÷ 4 gallons = 25 miles per gallon

Step 5: Convert the fuel efficiency from mpg to gallons per mile (GPM):

- Since 1 mpg means traveling 1 mile with each gallon, Sophia will use 4 gallons × (1 mile/mpg) = 4 miles per tank.

Step 6: Calculate total distance for full tank:

- Total distance = 4 miles per tank × 12 gallons = 48 miles

$\boxed{48}$

Agent 2: Incomplete Reasoning

If the distance covered in one tank of fuel is 100 miles, then the Miles Per Gallon (MPG) is calculated as follows...
[Response appears incomplete and truncated mid-calculation]
After reviewing the problem again, I need to be more careful with my calculations.
Let me reconsider: Sophia used 4 gallons to travel some distance, and her tank holds 12 gallons total.
I'm getting confused by the different approaches. Let me try again: 100 miles with 4 gallons means...
Final answer: $\boxed{48}$

Agent 3: Correct but Overruled

Let me solve this step by step:
Step 1: Calculate fuel efficiency from given data:

- Distance traveled: 100 miles
- Gas used: 4 gallons
- Fuel efficiency: 100 ÷ 4 = 25 miles per gallon

Step 2: Calculate range on full tank:

- Full tank capacity: 12 gallons
- Total range: 25 mpg × 12 gallons = 300 miles

Therefore, Sophia can drive 300 miles on a single tank of gas.
$\boxed{300}$

A.4.2 Round 2: Failed Collaborative Discussion

Agent 1: Doubling Down on Error

After reviewing all approaches, I maintain my calculation:
Based on the calculations, I'm confident that 48 miles is correct because we need to convert the efficiency properly.
$\boxed{48}$

Pre-Training Result: Consensus selected 48 miles despite Agent 3's correct reasoning.

A.4.3 Complete Multi-Agent Transcripts: Post-Training Success (Iteration 1)

A.4.4 Round 1: Perfect Agent Responses

Agent 1: Complete Correction

To solve this problem, I need to determine Sophia's fuel efficiency and calculate her range on a full tank.
Step 1: Calculate Sophia's car's fuel efficiency:

- Sophia traveled 100 miles since last fill-up
- She used 4 gallons of gas
- Fuel efficiency = 100 miles ÷ 4 gallons = 25 miles per gallon

Step 2: Calculate range on full tank:

- Tank capacity: 12 gallons
- Range = 25 mpg × 12 gallons = 300 miles

Therefore, Sophia can drive 300 miles on a single tank of gas.
$\boxed{300}$

Unanimous Agreement

Agent 1: "I see all agents used the same systematic approach. The calculation is clear: 25 mpg × 12 gallons = 300 miles."
Agent 2: "Perfect agreement on methodology. The fuel efficiency calculation is the key step, then applying it to full tank capacity."
Agent 3: "Unanimous consensus on 300 miles. All reasoning aligns perfectly."

Post-Training Result: Perfect consensus on 300 miles with all agents showing correct reasoning.

A.4.5 Analysis: Training Effects on Multi-Agent Collaboration

Before training in this example, two agents converge on the same calculation error and overrule the correct agent. After training, all three agents apply the correct method and reach unanimous agreement on the correct answer.

B Grounding Behaviors in Base-Model Multi-Agent Debates

Having provided examples of the impact of post-training on single- and multi-agent reasoning, we now highlight representative examples from some initial debates among base models demonstrating spontaneous grounding behaviors that emerge during multi-agent reasoning. Note that each debate consists of an initial round (three agents provide independent solutions) and a final round (agents update after seeing peers' responses).

B.1 Example 1: Airport Timing Problem

Problem

Amber and Josh are flying to France for vacation. Their flight leaves at 8:00 pm and as an international flight, it's recommended that they check in 2 hours early. It will take 45 minutes to drive to the airport and another 15 minutes to park their vehicle and make their way to the terminal. What is the latest time they can leave their house to make it to the airport in time?

The correct solution requires working backwards from the 8:00 PM flight time: they must arrive by 6:00 PM (2 hours early), and with 1 hour total travel time, they must leave by 5:00 PM.

Initial round:

Agent 1: Directional Error

- Flight leaves at 8:00 pm, need to check in 2 hours early = 6:00 pm arrival
- Calculate travel time: 45 + 15 = 60 minutes = 1 hour
- Add travel time to departure time: 6:00 pm + 1 hour = 7:00 pm

Error: Adds travel time instead of subtracting it
Answer: 7:00 pm

Agent 2: Fundamental Misunderstanding

- Find total travel time: 45 + 15 = 60 minutes = 1 hour
- Add travel time to flight departure time: 8:00 pm + 1 hour = 9:00 pm

Completely ignores 2-hour check-in requirement and adds travel time instead of subtracting it
Answer: 9:00 pm

Agent 3: Correct Solution

- Flight at 8:00 pm requires arrival by 6:00 pm (2 hours early)
- Travel time: 45 min driving + 15 min parking = 60 minutes
- Must leave by: 6:00 pm - 1 hour = 5:00 pm

Answer: 5:00 pm

Final round (after seeing peers):

Agent 1: Adopts Worse Error

"After reviewing other solutions, I update my approach:

- Departure time from house: 8:00 pm
- Total travel time: 45 + 15 = 60 minutes = 1 hour
- Add travel time to departure: 8:00 pm + 1 hour = 9:00 pm"

Abandons partial understanding to adopt Agent 2's incorrect reasoning as well
Answer: 9:00 pm

Agent 2: Persists with Error

Maintains original incorrect answer of 9:00 pm

Agent 3: Explicitly Identifies Error Pattern

"The other agents seem to be adding time to the flight departure rather than working backwards from the required arrival time. The correct approach:

- Required arrival: 6:00 pm (2 hours before 8:00 pm flight)
- Subtract travel time: 6:00 pm - 1 hour = 5:00 pm departure"

Recognizes error in other agents' responses and upholds its own correct understanding
Answer: 5:00 pm

Key observation. Agent 1 recognizes the 6:00 PM arrival requirement but then adopts Agent 2's error. Agent 3 identifies the common error of adding time to the flight departure and maintains the correct backward calculation. This example shows how statistical aggregation can propagate a shared mistake.

B.2 Example 2: Self-Correction in Mathematical Reasoning

Problem

A store offers a 20% discount on all items. If Sarah buys a jacket originally priced at $80 and pants originally priced at $50, how much does she pay in total after the discount?

The correct solution: Total original price = $130, discount = 20% of $130 = $26, final price = $104.

Initial round:

Agent 1: Calculation Error

Original total: $80 + $50 = $130
Discount: 20% = 0.2 × $130 = $26
Final price: $130 + $26 = $156
Error: Adds discount instead of subtracting
Answer: $156

Agent 2: Correct Approach

Total: $130
After 20% discount: $130 × 0.8 = $104
Answer: $104

Agent 3: Misapplies Discount

Jacket discount: 20% of $80 = $16, price = $64
Pants discount: 20% of $50 = $10
Total: $64 + $10 = $74
Error: Adds discounted jacket price to pants discount amount
Answer: $74

Final round:

Agent 1: Self-Correction

"I realize I added the discount instead of subtracting. Looking at Agent 2's approach of multiplying by 0.8 is cleaner: $130 × 0.8 = $104"
Recognizes own error and adopts correct method
Answer: $104

Agent 3: Fixes Calculation

"Jacket after discount: $80 × 0.8 = $64
Pants after discount: $50 × 0.8 = $40
Total: $64 + $40 = $104"
Answer: $104

Key observation. Agent 1 explicitly acknowledges their error ("I added the discount instead of subtracting") and adopts Agent 2's more efficient method. Agent 3 corrects their calculation error after seeing consistent peer answers.

B.3 Example 3: Geometric Problem - Method Simplification

Problem

A rectangle has a perimeter of 24 units and an area of 32 square units. What are the dimensions of the rectangle?

The correct solution: If dimensions are length l and width w, then $2(l + w) = 24$ so $l + w = 12$, and $lw = 32$. Solving this system gives dimensions 8 and 4.

Initial round:

Agent 1: Algebraic Approach

Let length = l, width = w
From perimeter: $2l + 2w = 24$, so $l + w = 12$
From area: $lw = 32$
Setting up quadratic: $w(12 - w) = 32$
$12w - w^2 = 32$
$w^2 - 12w + 32 = 0$
Stops without solving the quadratic

Agent 2: Direct Factoring

$l + w = 12$ and $lw = 32$
Looking for factors of 32 that sum to 12: 8 and 4
Dimensions: 8 × 4
Answer: 8 units by 4 units

Agent 3: Quadratic Solution

From $w^2 - 12w + 32 = 0$
Using quadratic formula or factoring: $(w - 8)(w - 4) = 0$
So $w = 8$ or $w = 4$
Dimensions: 8 × 4
Answer: 8 units by 4 units

Final round:

Agent 1: Adopts Simpler Method

"Agent 2's approach of finding factors of 32 that sum to 12 is much more direct than solving the quadratic. Since 8 × 4 = 32 and 8 + 4 = 12, the dimensions are 8 by 4."
Recognizes and adopts more efficient solution method
Answer: 8 units by 4 units

Key observation. Agent 1 explicitly acknowledges that Agent 2's factor-finding approach is "much more direct" than the algebraic method and adopts the more efficient solution.

B.4 Summary

These examples illustrate observable grounding behaviors in multi-agent debates:

Error propagation: Agents sometimes abandon partially correct reasoning to adopt peer errors (Example 1, Agent 1).

Explicit error recognition: Agents identify specific errors in their own reasoning ("I added the discount instead of subtracting") and in peers' approaches ("adding time to flight departure").

Method adoption: Agents recognize more efficient solution methods from peers and explicitly acknowledge this ("Agent 2's approach... is much more direct").

Self-correction through peer comparison: Seeing consistent peer answers prompts agents to revisit and correct their calculations.

These interactions provide training information beyond final-answer majority labels: the debate traces contain peer critiques, revisions, and adopted solution strategies. MACA uses these traces and their consensus labels for post-training.

Algorithm 1 Multi-Agent Consensus Alignment Via Debate-RL

input: agents $\{\pi_{\theta_m}\}_{m=1}^{M}$, rounds R, iters L, batch size B, method in MV-SFT (baseline), MV-GRPO, MV-DPO, or MV-KTO
for $\ell = 1, \ldots, L$ **do**
  sample batch $\{x^{(\ell,i)}\}_{i=1}^{B}$ of prompts from $\mathcal{D}$
  for $i = 1, \ldots, B$ **do** ▷ Generate debate trajectories
    $y_{m,1}^{(\ell,i)} \sim \pi_{\theta_m}(\cdot \mid x^{(\ell,i)}) \quad \forall m$
    for $r = 2, \ldots, R$ **do** ▷ Condition on peers' previous round outputs
      $x_{m,r}^{(\ell,i)} = [x^{(\ell,i)}; \{y_{j,r-1}^{(\ell,i)}\}_{j \neq m}]$
      $y_{m,r}^{(\ell,i)} \sim \pi_{\theta_m}(\cdot \mid x_{m,r}^{(\ell,i)}) \quad \forall m$
    end for
    Parse final answers $a_m^{(\ell,i)} = \mathcal{A}(y_{m,R}^{(\ell,i)}) \quad \forall m$
    Aggregate consensus $\hat{a}(x^{(\ell,i)}) = \text{Majority}\{a_1^{(\ell,i)}, \ldots, a_M^{(\ell,i)}\}$
    Extract final contexts $x_m^{(\ell,i)} \leftarrow x_{m,R}^{(\ell,i)}$ for all m
    Partition responses by consensus alignment:
      $\mathcal{G}_+^{(\ell,i)} = \{y_{m,R}^{(\ell,i)} : a_m^{(\ell,i)} = \hat{a}(x^{(\ell,i)})\}$
      $\mathcal{G}_-^{(\ell,i)} = \{y_{m,R}^{(\ell,i)} : a_m^{(\ell,i)} \neq \hat{a}(x^{(\ell,i)})\}$
  end for
  build dataset: $\mathcal{D}_{\text{post}} = \{(x^{(\ell,i)}, \hat{a}(x^{(\ell,i)}), \mathcal{G}_+^{(\ell,i)}, \mathcal{G}_-^{(\ell,i)})\}_{i=1}^{B}$
  training data by method:
    MV-SFT: for each m, use $\{(x_m^{(\ell,i)}, y_{m,R}^{(\ell,i)}) : y_{m,R}^{(\ell,i)} \in \mathcal{G}_+^{(\ell,i)}\}_i$.
    MV-GRPO: for each m, store $\{(x_m^{(\ell,i)}, \hat{a}(x^{(\ell,i)}))\}_i$ for reward computation on new samples.
    MV-DPO/MV-KTO: pool $\{(x_m^{(\ell,i)}, y_{m,R}^{(\ell,i)})\}_{i,m}$ partitioned by $\mathcal{G}^+, \mathcal{G}^-$ across agents.
  update policies: apply majority-vote objective to refine each π_{θ_m}
end for

C MACA Algorithm: Additional Implementation Details

C.1 Agent Heterogeneity and Specialization

In our current experiments, we use homogeneous agents (clones) that update independently over a single training iteration, enabling evaluation of both individual and averaged performance to measure the benefits of divergence. We also explore the impact of training over multiple such iterations of debate and post-training in App. G. For multi-iteration runs, agent checkpoints can be managed in two ways: agents may either remain synchronized by resetting checkpoints to the best performing one after every iteration, or diverge by updating independently from distinct trajectory data, allowing specialization and diversity through differential learning. Additionally, the MACA framework also supports heterogeneous agents, allowing different language models to be independently optimized through this framework, where agents represent distinct models or architectures from the outset, each bringing inherently different capabilities or perspectives.

D Experimental Setup

D.1 Training Parameters

Table 7 reports the hyperparameter search ranges for the main multi-agent experiments. We searched learning rates from 10^{-7} to 10^{-5} and selected 10^{-5} for the reported runs. For MV-KTO and MV-DPO, we searched $\beta \in [0.1, 0.3]$ and used $\beta = 0.1$ in the reported runs. Selected LoRA rank/alpha values were 64–128 for MV-DPO and MV-KTO, 32–64 for MV-GRPO, and 128 for the MV-SFT baseline. Tables 7–10 provide the remaining setup.

GSM8K, MATH, and MathQA used 1500/500/500 train/valid/test splits. SVAMP, GPQA, CSQA, and AMC were used only for testing, with 300, 448, 500, and 40 test examples, respectively.

Table 7: Hyperparameter search ranges for the main multi-agent experiments.

Parameter	**SFT**	**GRPO**	**DPO**	**KTO**
Learning rate	10^{-7}–10^{-5}	10^{-7}–10^{-5}	10^{-7}–10^{-5}	10^{-7}–10^{-5}
Weight decay	1×10^{-2}	1×10^{-2}	1×10^{-2}	1×10^{-2}
Batch size	1–8	8	1–8	1–8
Epochs	1–3	1–3	1–3	1–3
Gradient accumulation steps	2–4	2–4	2–4	2–4
Optimizer	AdamW	AdamW	AdamW	AdamW
LoRA rank (r)	8–128	8–64	8–128	8–128
LoRA alpha	8–128	8–64	8–128	8–128
LoRA dropout	0.1	0.1	0.1	0.1
LoRA target modules	q_proj, k_proj, v_proj, o_proj			
Entropy coefficient	-	0.01	-	-
Beta search range	-	-	0.1–0.3	0.1–0.3
Number of generations per step	-	8	-	-
Maximum sequence length	Model-dependent (2048–4096)			
Mixed precision	bf16			
Quantization	4-bit (BitsAndBytesConfig)			

Table 8: Generation parameters.

Parameter	**Value**
Temperature	1.0
Top-p sampling	0.9
Maximum new tokens	256
Do sample	True
Pad token ID	EOS token ID

D.2 Online Self-Debate Experiment

The Qwen3-4B experiment uses a fixed 320/128 GPQA train/test split shared by all methods and three training seeds. Each method trains for one epoch with LoRA rank 32, LoRA alpha 64, and learning rate 10^{-5}. Training uses eight policy rollouts per prompt: the single-round baselines draw eight independent responses, while online self-debate draws four initial responses followed by four responses conditioned on the anonymized initial set. Rollouts use temperature 1.0 with thinking disabled. The maximum prompt, response, and total context lengths are 24,576, 8,192, and 32,768 tokens, respectively.

Evaluation uses one rollout per held-out prompt with temperature 0.7, top-p 0.8, top-k 20, an 11,264-token response limit, and a 12,288-token total context limit. Answers are extracted as choices A–D. Unparseable responses receive no agreement reward, and training examples without a usable majority are omitted. Self-Judge uses additional frozen-model scoring calls, which are not counted as policy rollouts.

D.3 Datasets

We evaluated model mathematical reasoning using seven publicly available datasets spanning a range of difficulty and subject areas: MATH (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), MathQA (Amini et al., 2019), SVAMP (Patel et al., 2021), GPQA (Rein et al., 2023), AMC 23 knoveleng (2023), and CommonsenseQA/CSQA (Talmor et al., 2019).

Table 9: Multi-agent debate parameters.

Parameter	**Value**
Number of agents	3
Debate rounds	2
Batch debate size	8–24
Use majority vote	True
Include debate context	True
Use async debate	True

Table 10: Model and dataset configuration.

Parameter	Value
Base models	Qwen2B, Llama3B, Phi4B, Llama8B
Training datasets	GSM8K, MATH, MathQA
Test datasets	GSM8K, MATH, MathQA, SVAMP, GPQA, CSQA, AMC
Training size	1500 samples
Test size	500 samples
Validation size	500 samples
Use quantization	4-bit BitsAndBytesConfig
Use adapter mode	True (LoRA)
Use scheduler	True

MATH (Hendrycks et al., 2021): The MATH dataset contains 12,500 high school mathematics problems from algebra, geometry, combinatorics, and number theory, each requiring multi-step reasoning and precise mathematical solutions.

GSM8K (Cobbe et al., 2021): GSM8K is composed of 8,500 grade-school-level word problems emphasizing arithmetic and logical reasoning, with step-by-step annotated solutions for each problem.

MathQA (Amini et al., 2019): MathQA features over 37,000 question-answer pairs based on quantitative reasoning, requiring models to convert natural language problems into mathematical expressions and perform multi-step computation.

SVAMP (Patel et al., 2021): SVAMP consists of carefully reworded arithmetic word problems designed to probe model robustness and prevent reliance on annotation artifacts, focusing on genuine multi-step arithmetic reasoning.

GPQA (Rein et al., 2023): GPQA provides 448 expert-curated, multiple-choice questions covering graduate-level biology, physics, and chemistry, emphasizing deep, multi-stage reasoning and robust factual understanding.

AMC 23 (knoveleng, 2023): This dataset includes recent problems drawn from the 2023 American Mathematics Competitions (AMC), spanning algebra, combinatorics, geometry, and number theory, and is useful for assessing model performance on expert-constructed math tasks

CommonsenseQA (Talmor et al., 2019): CommonsenseQA is a challenging multiple-choice question-answering dataset requiring models to apply commonsense reasoning over diverse everyday scenarios. The dataset comprises 12,247 questions, each designed to probe deeper, non-trivial conceptual knowledge beyond factual recall, making it a benchmark for evaluating commonsense understanding in language models.

Together, these datasets cover mathematical, scientific, and commonsense reasoning.

D.4 Multi-Agent Debate Infrastructure

Our multi-agent debate system addresses the computational challenges of training and deploying multiple agents efficiently through specialized infrastructure for quantized training isolation and scalable debate inference.

D.4.1 Quantized MARL Training

Current post-training libraries lack support for multi-gpu distributed training with 4-bit quantization and PEFT adapters. We implement training isolation where each agent trains on a single GPU with explicit resource assignment, using gradient checkpointing and paged optimizers (paged_adamw_8bit) for memory efficiency. This enables stable multi-agent RL with QLoRA while preserving quantization benefits.

D.4.2 Scalable Debate Inference Framework

The inference system supports multi-agent debates through adapter hot-swapping and dynamic resource management. A centralized manager shares quantized base models across agents and caches lightweight LoRA adapters, avoiding full model reloads when switching agents. A scheduler distributes debate workloads across available GPUs by round.

D.5 Debate Prompt Templates

This appendix details the prompts used in our multi-agent debate system.

D.5.1 Initial Question Prompt

Used in the first round of debate.

For GPQA, MathQA, CommonsenseQA

Initial Prompt: Multiple Choice

```
Answer the following multiple choice question as accurately
   as possible.
{question}

Provide a bullet point summary of your step-by-step reasoning
   .
Your final answer should be a single choice letter in the
   form
\boxed{answer}, at the end of your response.
```

For GSM8K, SVAMP, AMC, MATH

Initial Prompt: Math Problem

```
Solve the following math problem. {question}

Provide a bullet point summary of your step-by-step reasoning
   .
Your final answer should be a single numerical number, in the
    form
\boxed{answer}, at the end of your response.
```

Table 11: Single-agent accuracy with debate-majority-vote-derived (DMV) and ground-truth (GT) supervision. Bold indicates the higher score in each pair.

Model	Dataset	Debate	SFT		KTO		DPO		GRPO	
			GT	*DMV*	*GT*	*DMV*	*GT*	*DMV*	*GT*	*DMV*
Qwen2B	MATH	7.67	12.13	**12.20**	12.67	**19.67**	17.13	**23.00**	**21.33**	17.33
	GSM8K	23.00	24.20	**24.80**	45.40	**47.20**	**46.47**	44.20	**39.73**	32.07
Llama3B	MATH	27.87	**26.80**	25.60	12.53	**39.20**	38.40	**40.67**	**36.80**	35.00
	GSM8K	57.33	54.13	**56.73**	27.27	**66.27**	56.87	**66.73**	**54.93**	52.20
Phi4B	MATH	34.60	33.07	**35.20**	**33.73**	33.00	**37.00**	36.33	**37.80**	37.33
	GSM8K	67.27	**71.67**	70.20	**76.13**	75.47	75.80	**76.60**	**74.80**	70.40
Llama8B	MATH	22.93	**23.73**	23.00	**41.20**	39.40	45.13	**46.40**	29.07	**31.13**
	GSM8K	57.93	40.20	**41.67**	**72.60**	70.87	76.33	**77.67**	61.27	**66.87**

D.5.2 Debate Phase Prompt

Used in subsequent rounds of debate to incorporate peer context.

Debate Phase Prompt

```
Here are solutions from other agents:
{agent_responses}

Using each response as additional advice, give an updated
   bullet point summary of your step-by-step reasoning to
   the question: {question}

Make sure your final answer is in the form \boxed{answer}, at
    the end of your response.
```

E Ablation: Ground Truth vs Debate Majority Vote

Tables 11 and 12 report ablations comparing post-training on oracle ground-truth (GT) labels versus (DMV) labels across MATH and GSM8K benchmarks.

In the single-agent setting (Table 11), DMV and GT supervision trade wins across models and objectives. The results show that debate-derived labels can be competitive with task-specific ground-truth labels in the evaluated settings, although neither supervision source dominates consistently.

In the multi-agent setting (Table 12), DMV and GT supervision again trade wins across models and objectives. The comparison shows that debate-derived supervision can match or exceed GT in some settings, while GT remains stronger in others.

Together, these results motivate debate-derived majority-vote supervision as a label-free alternative worth studying when task-specific ground-truth answers are unavailable. Its relative performance depends on the model, objective, dataset, and evaluation setting.

F Ablation: Effect of Peer Context in Multi-Agent Debate

Tables 13 and 14 examine whether conditioning on peer responses during training improves debate performance.

Training with peer context improves multi-agent debate performance in most evaluated settings (Table 14). Initial-round performance improves in 5 of 8 cases, final-round individual accuracy in 7 of 8, and final-round majority-vote accuracy in 6 of 8. The largest differences include L8B-MATH

Table 12: Multi-agent accuracy with debate-majority-vote-derived (*DMV*) and ground-truth (*GT*) supervision. Bold indicates the higher score in each pair.

Model	Dataset	Init	SFT		KTO		DPO		GRPO	
			GT	*DMV*	*GT*	*DMV*	*GT*	*DMV*	*GT*	*DMV*
Qwen2B	MATH	32.4	38.6	**40.6**	40.4	**46.2**	39.4	**41.2**	**43.6**	40.2
	GSM8K	49.6	50.0	**51.6**	**66.2**	63.8	**61.6**	57.0	**60.4**	51.8
Llama3B	MATH	37.8	**35.6**	33.6	15.8	**51.8**	50.8	**51.4**	49.2	**50.8**
	GSM8K	65.6	**65.8**	64.0	29.4	**73.8**	67.0	**73.8**	64.4	**69.6**
Phi4B	MATH	44.4	44.0	**46.0**	**46.8**	44.6	**50.2**	48.6	**51.4**	50.8
	GSM8K	79.6	**79.8**	78.4	**82.6**	81.2	83.8	**84.4**	81.6	**83.4**
Llama8B	MATH	32.8	**34.0**	33.4	**56.0**	55.8	**61.8**	60.8	**48.6**	44.8
	GSM8K	74.0	**65.2**	**65.2**	**81.2**	79.4	81.6	**83.0**	83.2	**84.2**

Table 13: Multi-agent comparison of Context vs No Context runs across Initial Debate, SFT, KTO, DPO, and GRPO. Bold indicates which setting performed better in each pair.

Model	Dataset	Debate	SFT		KTO		DPO		GRPO	
			Context	*NoCtx*	*Context*	*NoCtx*	*Context*	*NoCtx*	*Context*	*NoCtx*
Qwen2B	MATH	32.4	**40.6**	38.0	46.2	**46.6**	41.2	**43.4**	**40.2**	39.8
	GSM8K	49.6	**51.6**	**51.6**	**63.8**	59.2	57.0	**57.8**	51.8	**57.2**
Llama3B	MATH	37.8	33.6	**37.4**	**51.8**	27.0	**51.4**	51.0	**50.8**	47.0
	GSM8K	65.6	64.0	**65.4**	**73.8**	59.0	**73.8**	70.6	**69.6**	50.0
Phi4B	MATH	44.4	**46.0**	45.0	44.6	**46.0**	**48.6**	48.4	50.8	**52.8**
	GSM8K	79.6	78.4	**80.0**	81.2	**82.0**	**84.4**	81.8	83.4	**82.0**
Llama8B	MATH	32.8	**33.4**	31.4	**55.8**	49.2	**60.8**	57.2	**44.8**	44.6
	GSM8K	74.0	65.2	**67.4**	79.4	**80.6**	**83.0**	81.6	**84.2**	71.0

final-round individual accuracy (57.80 versus 51.47), L3B-MATH final-round individual accuracy (45.80 versus 41.13), and L8B-MATH majority-vote accuracy (60.80 versus 57.20). These results associate peer-context training with higher final-round accuracy in the evaluated configurations.

G Iterative Improvement

Table 15 examines whether iterative training beyond the first iteration yields continued improvements. While It-1 produces the substantial gains reported in our main results, iterations 2 and 3 demonstrate continued modest improvements: in 23 of 24 evaluation settings, either It-2 or It-3 achieves the best performance. For example, Phi-4B on MATH majority vote increases from 55.00 (It-1) to 57.40 (It-3), and Llama-8B on GSM8K improves from 82.80 to 85.60.

These gains show clear diminishing returns compared to the It-0→It-1 jump, with typical improvements of 1-3 percentage points between iterations. Performance occasionally dips between adjacent iterations (e.g., L3B-GSM8K majority vote: 72.00→74.60→70.80) but generally trends upward. This pattern suggests that iterative training continues to extract useful signal from debate-generated data, though with decreasing marginal benefit after the initial iteration.

H DPO Pair Selection Strategy Analysis

We compared two strategies for creating preference pairs from multi-agent debate data: majority/minority partitioning versus confidence-based selection using model log-probabilities. Analysis covered 2,226 agent responses from 742 problems across multiple models and datasets. In this analysis, majority/minority partitioning produces larger separations on every reported metric. Majority responses achieve 68.0% accuracy versus 28.1% for minority responses (39.8 percentage point gap), while

Table 14: Effect of peer-context post-training on final-round ensemble reasoning with MV-DPO.

	Initial Round Avg		Final Round Avg		Final Round MV	
Model-Data	NoCtx	Context	NoCtx	Context	NoCtx	Context
Qwen2B-MATH	22.33	**23.00**	37.13	**38.33**	**43.40**	41.20
Qwen2B-GSM8K	**46.53**	44.20	52.53	**54.73**	**57.80**	57.00
Llama3B-MATH	40.27	**40.67**	41.13	**45.80**	51.00	**51.40**
Llama3B-GSM8K	65.87	**66.73**	67.87	**70.60**	70.60	**73.80**
Phi4B-MATH	**39.47**	36.33	**39.53**	34.73	48.40	**48.60**
Phi4B-GSM8K	74.67	**76.60**	74.60	**77.80**	81.80	**84.40**
Llama8B-MATH	**49.87**	46.40	51.47	**57.80**	57.20	**60.80**
Llama8B-GSM8K	75.80	**77.67**	78.00	**81.20**	81.60	**83.00**

Table 15: Iterative alternation between debate generation and post-training across four iterations shows substantial initial gains (It-0 to It-1) followed by diminishing returns. Model abbreviations: Q2B=Qwen-2B, L3B=Llama-3B, L8B=Llama-8B, P4B=Phi-4B.

	Initial Round Avg				Final Round Avg				Final Round MV			
Model-Data	It-0	It-1	It-2	It-3	It-0	It-1	It-2	It-3	It-0	It-1	It-2	It-3
Q2B-MATH	7.67	17.40	18.00	**19.33**	21.47	43.13	40.73	**43.93**	32.40	47.40	41.20	**48.60**
Q2B-GSM8K	23.00	44.20	44.67	**44.73**	35.07	58.67	**59.93**	59.47	49.60	60.80	62.20	**62.60**
L3B-MATH	27.87	40.93	39.33	**41.93**	24.27	48.27	46.93	**48.73**	37.80	**55.00**	54.00	53.60
L3B-GSM8K	57.33	64.60	**67.07**	65.20	49.20	68.73	**71.13**	69.07	65.60	72.00	**74.60**	70.80
P4B-MATH	34.60	43.27	43.67	**43.80**	34.37	48.00	48.93	**50.20**	44.40	55.00	55.80	**57.40**
P4B-GSM8K	67.27	75.73	**76.20**	75.47	68.53	77.00	**79.20**	77.60	79.60	81.40	**84.40**	83.00
L8B-MATH	22.93	44.53	44.67	**44.87**	22.53	55.73	**57.60**	56.07	32.80	58.20	**60.40**	59.80
L8B-GSM8K	57.93	77.00	77.87	**78.80**	56.53	80.80	80.73	**82.93**	74.00	82.80	82.60	**85.60**

high-confidence responses achieve 51.5% accuracy versus 33.2% for low-confidence responses (18.3 percentage point gap). The majority strategy yields an effect size of Cohen's d = 1.832 compared with 0.281 for confidence-based selection, a 6.5× difference in discriminative power. Majority voting also provides usable preference signals in 70.5% of examples versus 35.2% for confidence-based selection. These results support majority/minority partitioning for DPO pair construction in the evaluated mathematical-reasoning data.

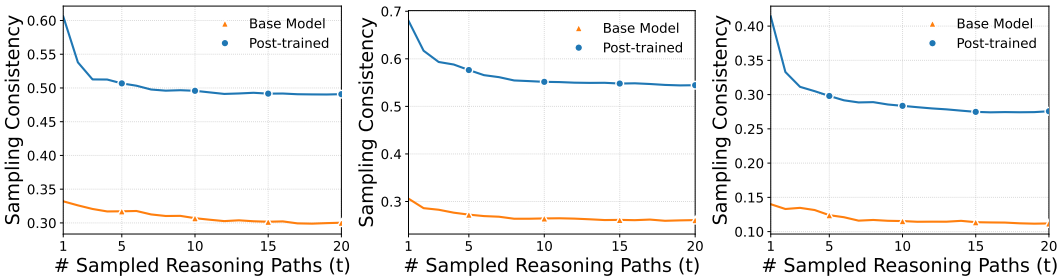
I Self-Consistency and Accuracy Correlation Analysis

We analyze the correlation between self-consistency and accuracy improvements across three experimental conditions to assess the robustness of our findings under different token generation limits and quantization settings. We test three conditions: (1) Token Capped + Quantized (256 tokens, 4-bit), our standard experimental setup matching computational constraints; (2) Token Uncapped + Non-Quantized (2048 tokens with no observed truncation, full precision), representing maximum generation quality without computational constraints; and (3) Token Uncapped + Quantized (2048 tokens with no observed truncation, 4-bit), a balanced approach removing token truncation while maintaining efficiency.

All conditions show strong positive correlations ($r > 0.86$) between self-consistency and accuracy. We therefore use self-consistency as a secondary diagnostic that tracks accuracy in these experiments, rather than as a substitute for ground-truth evaluation. The capped mean is $r = 0.975$, compared with 0.915 for uncapped full precision and 0.920 for uncapped quantized inference. In uncapped settings, post-trained models generate 22–36% shorter responses while remaining more accurate. Across these conditions, self-consistency gains range from 6.9 to 27.6 percentage points and accuracy gains from 6.3 to 24.7 percentage points.

Table 16: Self-consistency vs accuracy correlation across experimental conditions. "Capped" refers to 256 token limit, which is used throughout the work due to computational constraints in multi-agent RL settings, and "Uncapped" refers to 2048 token limit where no response truncation occurred.

Condition	Model-Dataset	Self-Consistency (%)		Accuracy (%)		Correlation
		Base	Post	Base	Post	(r)
Capped +Quant	Llama3B-MathQA	31.2	**49.4**	29.1	**45.8**	0.954
	Qwen2B-Math	12.0	**27.8**	11.6	**26.8**	0.989
	Qwen2B-GSM8K	26.4	**54.0**	25.9	**52.7**	0.982
Uncapped +Full	Llama3B-MathQA	49.5	**68.5**	45.1	**60.9**	0.869
	Qwen2B-Math	50.4	**57.2**	48.2	**54.4**	0.944
	Qwen2B-GSM8K	51.4	**76.4**	50.8	**75.5**	0.933
Uncapped +Quant	Llama3B-MathQA	49.8	**67.3**	45.8	**60.8**	0.899
	Qwen2B-Math	42.8	**49.7**	38.9	**45.2**	0.935
	Qwen2B-GSM8K	50.6	**73.5**	48.9	**71.3**	0.927



(a) Llama-3B on MATHQA (b) Qwen-2B on GSM8K (c) Qwen-2B on MATH

Figure 4: Consistency before and after MACA post-training with token constraints (maximum of 256 tokens). Pre-trained models (Orange) show low sampling consistency across sampled trajectories. Post-training with MACA (Blue) substantially improves sampling consistency. Averaged over 500 test prompts with 20 trajectories each.

I.1 Self-Consistency Curves

Figure 4 shows sampling consistency before and after MACA post-training. Pre-trained models show low sampling consistency across sampled trajectories, while post-training substantially improves consistency, with curves stabilizing up to 27.6 percentage points above baseline.

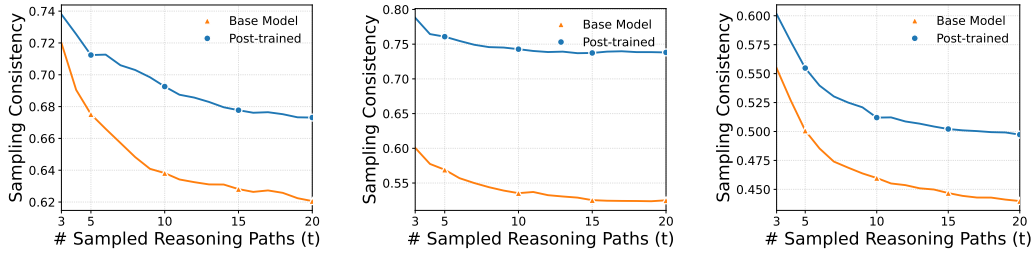
I.2 Self-Consistency Improvements Without Token Constraints

Figure 5 shows self-consistency curves for the aforementioned model–dataset pairs without token constraints. Improvements remain in every evaluated configuration, showing that they are not solely an artifact of truncation. The effect sizes are smaller than in the main results; these models were trained on 256-token responses and evaluated here with longer responses.

J Impact of MACA on General Reasoning

We compare debate-derived majority-vote supervision with ground-truth supervision on training domains and unseen reasoning domains. Tables 17 and 18 report these comparisons across mathematical training domains and out-of-distribution tasks.

Table 17 shows that debate-derived and ground-truth supervision trade wins across training domains. Table 18 reports results on unseen mathematical word problems (SVAMP), science reasoning (GPQA),



(a) Llama-3B on MathQA (b) Qwen-2B on GSM8K (c) Qwen-2B on MATH

Figure 5: Consistency before and after MACA post-training without token constraints. Models trained with 256-token debates still show gains when tested with full-length responses, though with reduced effect sizes due to the weaker training signal compared to testing conditions. Colors: Blue: post-trained model, Orange: base model.

Table 17: Within-distribution comparison of debate-derived majority-vote and ground-truth supervision. Models are trained on MATH, GSM8K, and MathQA. Bold indicates the higher score in each model–dataset pair.

Model	Method	MATH		GSM8K		MathQA	
		Debate-MV	GT	Debate-MV	GT	Debate-MV	GT
Qwen2B	Base	10.4	10.4	27.0	27.0	7.4	7.4
	SFT	**10.8**	10.4	25.6	**26.4**	8.2	**8.8**
	GRPO	19.4	**21.0**	45.2	**48.6**	18.6	**19.6**
	KTO	22.6	**23.2**	**54.8**	54.6	**28.8**	28.6
	DPO	**24.8**	24.2	51.4	**52.0**	**24.2**	24.0
Llama3B	Base	32.0	32.0	69.6	69.6	24.6	24.6
	SFT	**33.2**	32.4	64.2	64.2	**26.4**	25.2
	GRPO	45.8	**46.4**	**75.8**	74.8	**36.2**	31.8
	KTO	**48.0**	47.8	76.0	**76.8**	**41.4**	40.6
	DPO	53.2	**53.6**	**80.4**	77.8	**46.2**	45.4

and commonsense reasoning (CSQA), where both supervision approaches produce cross-domain gains in the evaluated settings.

These results show that debate-generated consensus can provide a useful training signal without task-specific answer labels. This makes it a candidate for settings where expert labels are unavailable or costly.

K Post-Training Method Impact on Log-Probability Distribution

Figure 6 shows how different post-training methods affected log-probability distributions for Qwen-2B on MATH. Preference-based methods (DPO, KTO) increased the density of majority distributions, with KTO showing particularly pronounced effects: higher peaks and tighter tails indicating more concentrated probability mass around consensus responses. SFT left the majority distribution largely unchanged while substantially reducing minority distribution density. GRPO created the most dramatic separation between distributions, shifting both leftward (lower log-probabilities overall) but with the minority distribution shifting much more substantially than the majority. While these different patterns (probability concentration for preference methods versus selective penalization for GRPO) all corresponded with performance improvements, further research is needed to better understand the relationship between these specific distributional changes and the impact on consistency.

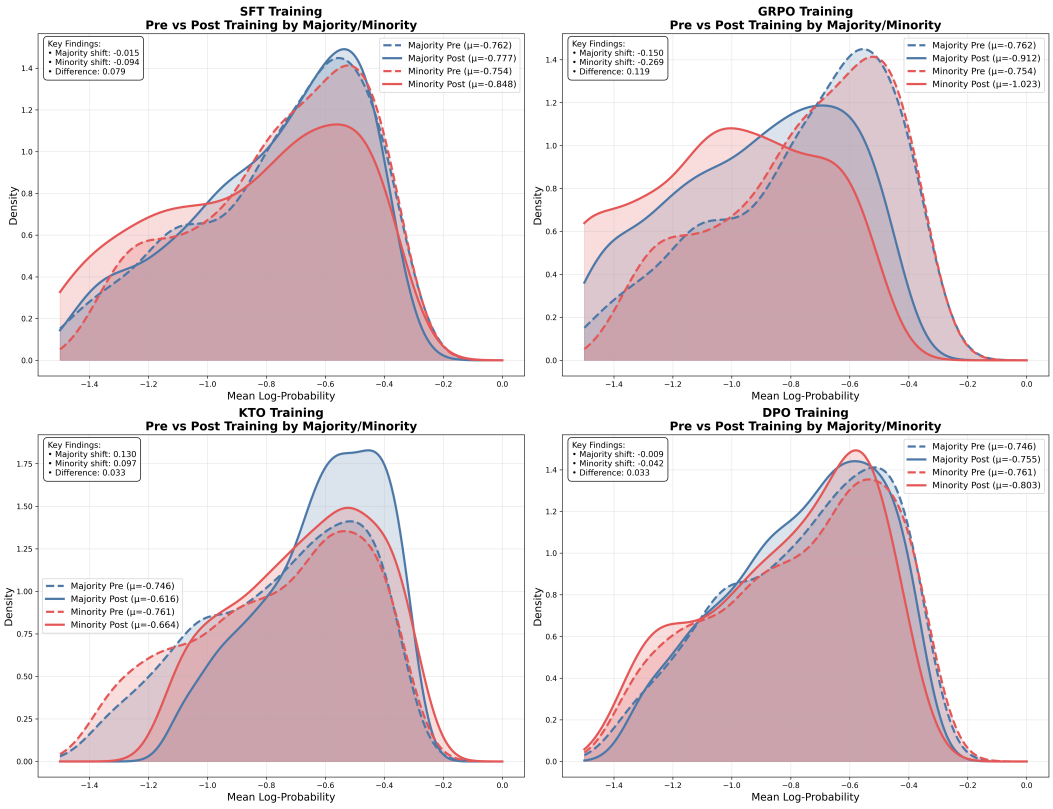


Figure 6: Log probability distributions for majority and minority answers before and after post-training (Qwen2B, Math).

Table 18: Cross-domain comparison of debate-derived majority-vote and ground-truth supervision. Models are trained on MATH, GSM8K, or MathQA and evaluated on unseen reasoning tasks. Bold indicates the higher score in each model–dataset pair.

Model	Method	SVAMP		GPQA		CSQA	
		Debate-MV	GT	Debate-MV	GT	Debate-MV	GT
Qwen2B	Base	48.30	48.30	0.45	0.45	3.80	3.80
	SFT	**53.30**	53.00	**17.90**	0.89	16.80	**18.80**
	GRPO	**60.30**	58.33	**8.70**	7.81	**10.80**	9.60
	KTO	76.00	**76.33**	16.70	**16.96**	**63.40**	60.80
	DPO	**65.00**	64.67	19.64	**20.98**	**62.2**	60.80
Llama3B	Base	71.30	71.30	0.67	0.67	53.00	53.00
	SFT	68.33	**72.0**	**2.23**	2.23	57.40	**57.80**
	GRPO	75.00	**79.33**	**6.92**	5.13	**63.20**	59.40
	KTO	76.70	**78.67**	8.93	**9.82**	**62.20**	61.80
	DPO	78.40	**80.67**	**11.40**	11.60	**64.00**	62.40

L Preference Learning as an Implicit Format Reward

This section investigates how preference learning through consensus signals acts as an implicit format reward without explicit formatting supervision. We examine the baseline capability requirements for consensus formation, how token constraints interact with the results, and how much of the measured accuracy gain remains after accounting for completion and formatting effects.

L.1 Baseline Capability Requirements for Consensus Formation

Our experiments show that MACA requires sufficient baseline capability to produce some correct responses. Under a strict 256-token limit, Qwen2B produces no correct AMC responses, so its consensus labels cannot identify the ground-truth answer. Increasing the token limit to 512 permits some correct responses, though inefficient reasoning can still cause truncation.

Table 19: Impact of token limits and baseline capability on AMC performance. The table reports accuracy for base and post-trained models ("All") under different token constraints. When no sampled response is correct (Qwen2B at 256 tokens), consensus cannot provide a correct training target.

	max_new_tokens = 256		max_new_tokens = 512	
	Base	All	Base	All
Qwen2B	0.0	0.0	5.0	**12.5**
Llama3B	7.5	**10.0**	10.0	**20.0**

MACA requires the base model to produce some correct responses from which a correct consensus target can form. Table 19 shows no gain when Qwen2B produces no correct AMC responses under the 256-token limit, while gains appear once the longer limit permits correct responses.

L.2 Token Efficiency and Reasoning Quality Improvements

Post-training simultaneously addresses token efficiency and reasoning quality. Models trained with debate under constrained token limits generate more parseable answers within the budget while achieving higher accuracy on those answers. To verify that improvements extend beyond mere format optimization, we evaluated models post-trained with 256-token debates using 512-token test budgets.

Table 20 shows that performance gains persist under increased token limits, so they are not explained solely by the original generation limit. We also analyze the sequence lengths of the trajectories generated for App. I, finding that uncapped post-trained models generate responses that are 22–36% shorter than their base models. Together, these results show accuracy and concision gains across the evaluated token budgets.

Table 20: Improvements from post-training with max_new_tokens = 256 translate when tested with larger token counts (512). “All" represents a model post-trained on Math, MathQA, and GSM8K; its row reports improvement deltas over the base model performance.

Model	Trained on	Tested on (max_new_tokens = 512)					
		MATH	GSM8K	MathQA	SVAMP	AMC	GPQA
Qwen2B	Base	32.8	47.4	18.8	53.7	7.5	12.5
	All	↑ 15.6	↑ 24.6	↑ 22.2	↑ 20.0	↑ 2.5	↑ 11.2
Llama3B	Base	60.8	76.8	43.0	78.0	10.0	9.2
	All	↑ 6.4	↓ 0.2	↑ 11.2	↑ 3.0	↑ 10.0	↑ 14.5

Table 21: Decomposition of performance improvements into completion gains and residual gains. “From Completion" estimates the accuracy recovered by allowing truncated base-model responses to complete. “Beyond Completion" is the remaining accuracy difference and may include other effects. Percentages show each component’s share of the total gain.

Model	Dataset	Before	After	Total Gain	From Completion	Beyond Completion
Qwen2B	CSQA	3.8	63.4	↑ 59.6	↑ 18.6 (31%)	↑ 41.0 (69%)
	GPQA	0.4	16.7	↑ 16.3	↑ 0.7 (4%)	↑ 15.6 (96%)
	GSM8K	24.6	54.8	↑ 30.2	↑ 0.0 (0%)	↑ 30.2 (100%)
	MATH	10.4	22.6	↑ 12.2	↑ 0.0 (0%)	↑ 12.2 (100%)
	MathQA	7.4	28.8	↑ 21.4	↑ 2.0 (9%)	↑ 19.4 (91%)
	SVAMP	48.3	76.0	↑ 27.7	↑ 0.4 (1%)	↑ 27.3 (99%)
Llama3B	CSQA	53.0	64.0	↑ 11.0	↑ 1.2 (11%)	↑ 9.8 (89%)
	GPQA	0.7	11.4	↑ 10.7	↑ 0.9 (8%)	↑ 9.8 (92%)
	GSM8K	69.6	80.4	↑ 10.8	↑ 0.2 (2%)	↑ 10.6 (98%)
	MATH	41.2	53.2	↑ 12.0	↑ 0.2 (2%)	↑ 11.8 (98%)
	MathQA	29.2	46.2	↑ 17.0	↓ 0.2 (-1%)	↑ 17.2 (101%)
	SVAMP	71.3	75.7	↑ 4.3	↑ 0.0 (0%)	↑ 4.3 (100%)

L.3 Decomposing the Sources of Improvement

To examine whether the gains stem from improved completion alone, we decompose the accuracy difference into the gain from avoiding truncation and the remaining gain.

Across these settings, 69–100% of the measured improvement remains after subtracting the estimated contribution from completed responses. Base models exhibit high truncation rates in some settings (e.g., 74.8% for Qwen2B on CSQA and 82.8% on MATH), which post-training reduces. The remaining gain shows that reduced truncation alone does not explain the observed accuracy differences.

L.4 Formatting Improvements and Remaining Reasoning Gains

To further isolate reasoning improvements from formatting effects, we analyze how post-training affects answer formatting compliance. Table 22 shows the percentage of responses with correct reasoning but incorrect formatting, and calculates reasoning gains that persist beyond all formatting improvements. Post-training dramatically reduces formatting losses (e.g., from 38.8% to 0.6% on CSQA for Qwen2B). However, the “Reasoning Gain Beyond Formatting" column reveals that substantial improvements remain even after perfect formatting is assumed. For instance, Qwen2B’s 59.6 percentage point improvement on CSQA includes 38.2 points from better formatting; the remaining 21.4 points represent accuracy improvements distinct from formatting.

Table 22: Analysis of formatting improvements versus reasoning gains. “Accuracy Lost to Format Errors" shows the percentage of responses with correct reasoning but incorrect formatting (e.g., writing “The answer is A" instead of `\boxed{A}`). “Reasoning Gain Beyond Formatting" shows improvements that persist even after accounting for all formatting fixes, calculated as: Total Gain - (Format Loss Before - Format Loss After).

Model	Dataset	Accuracy Lost to Format Errors (%)		Reasoning Gain Beyond Formatting (%)
		Before Training	After Training	
Qwen2B	CSQA	38.8	0.6	↑ 21.4
	GPQA	6.2	1.6	↑ 11.7
	MathQA	5.8	0.6	↑ 16.2
Llama3B	CSQA	7.0	1.4	↑ 5.4
	GPQA	7.6	3.8	↑ 6.9
	MathQA	5.8	2.8	↑ 14.0

L.5 Parser Implementation Details

To quantify formatting improvements versus reasoning gains, we employed two parsers with different strictness levels. Our standard parser requires answers in `\boxed{}` format and returns None for any deviation. The relaxed parser, used for impact analysis in Table 22, accepts common natural language patterns like “The answer is A" or “Answer: 42" by searching the entire response for valid answer formats.

This relaxed parser first attempts the strict extraction, then falls back to regex patterns that capture answers expressed naturally in text. For multiple choice, it accepts patterns like “[Tt]he answer is ([A-E])" or “Answer: ([A-E])". For numerical answers, it extracts from patterns like “= 42" at line endings or “The final answer is 42". When multiple patterns match, it takes the last occurrence, mimicking how humans identify the final answer in a reasoning chain.

The gap between strict and relaxed parser accuracies defines the “Accuracy Lost to Format Errors" in Table 22. Strict formatting measures whether outputs follow the requested machine-readable format, while the relaxed parser captures additional human-interpretable answers. Accuracy gains remain under the relaxed parser, so improved strict-format compliance does not fully explain the observed differences.

M Training Curves

The training dynamics of our post-training methods provide insight into how models learn to refine responses towards multi-agent consensus preferences. Figures 7a–8b show example training curves across key metrics during post-training with MV-DPO, MV-KTO, MV-GRPO, and MV-SFT.

For the preference objectives shown, reward margins between chosen (consensus) and rejected (non-consensus) responses increase during training. In MV-DPO and MV-KTO, margins start near zero and grow as the models increasingly favor majority-labeled outputs. MV-KTO shows a similar pattern without requiring strict paired comparisons.

In the displayed runs, training accuracy increases and losses decrease smoothly without visible instability or collapse.

Log probabilities of rejected responses decline across post-training methods, reflecting the models' increasing tendency to assign lower likelihood to outputs outside the consensus. This is most pronounced in MV-DPO and MV-GRPO, where rejected rewards fall more steeply, delineating a clear separation between preferred and discouraged responses. MV-SFT, operating via imitation learning, exhibits strong improvements in token accuracy and loss while reinforcing consensus-aligned responses effectively.

Overall, the asymmetric reward trajectories—chosen-response rewards remaining relatively stable or increasing slightly while rejected-response rewards decline—indicate that post-training primarily discourages minority-labeled outputs.

Table 23: Single-agent comparison of RL with debate and RL with single-round majority vote. Methods are TTRL (single-round R0 MV-GRPO), ScPO (single-round R0 MV-DPO), and MV-DPO (MACA: debate final-round R1 MV-DPO with debate context after collaborative exchange). Accuracy is averaged over three seeds, with standard deviation reported alongside the instruction-tuned base model.

	Qwen-2B	Llama-3B	Phi-4B	Llama-8B
		MATH		
Base	9.4	40.2	39.8	36.0
TTRL	**27.4 ± 2.9**	45.5 ± 5.7	45.9 ± 2.1	43.5 ± 0.2
ScPO	11.7 ± 1.1	43.6 ± 0.2	39.9 ± 0.5	39.7 ± 0.6
MV-DPO	26.1 ± 0.4	**52.7 ± 0.7**	**46.7 ± 0.2**	**53.1 ± 0.8**
		GSM8K		
Base	26.4	72.8	76.4	67.8
TTRL	**50.9 ± 1.6**	50.5 ± 8.3	81.2 ± 1.6	54.3 ± 5.7
ScPO	42.3 ± 3.4	74.4 ± 0.8	82.3 ± 0.5	80.7 ± 1.2
MV-DPO	50.2 ± 0.9	**76.5 ± 0.3**	**83.7 ± 0.3**	**86.0 ± 0.2**

These curves show that post-training increasingly favors responses aligned with multi-agent consensus.

N Post-Training Self-Consistency Improvements Translate from 4-Bit Quantized Model to Full Model

As shown in Figures 9-11, post-training improvements on 4-bit quantized models transfer to full-precision models, and the improvement margin persists with larger numbers of trajectories sampled ($t = 50$).

O Impact of Post-training on Debate Agreement Rates

Figure 12 compares agreement distributions for the base model (Iteration 0) and post-trained model (Iteration 1).

Full final-round consensus increases from 13.4% of examples before training to 43.4% after MV-DPO post-training. The distribution shifts from 1/3 agreement toward 2/3 and 3/3 agreement.

Unparseable final-round responses also drop from 13.8% to 0.6% without an explicit format reward. This is consistent with an implicit preference for responses that complete within the token limit and follow the requested answer format.

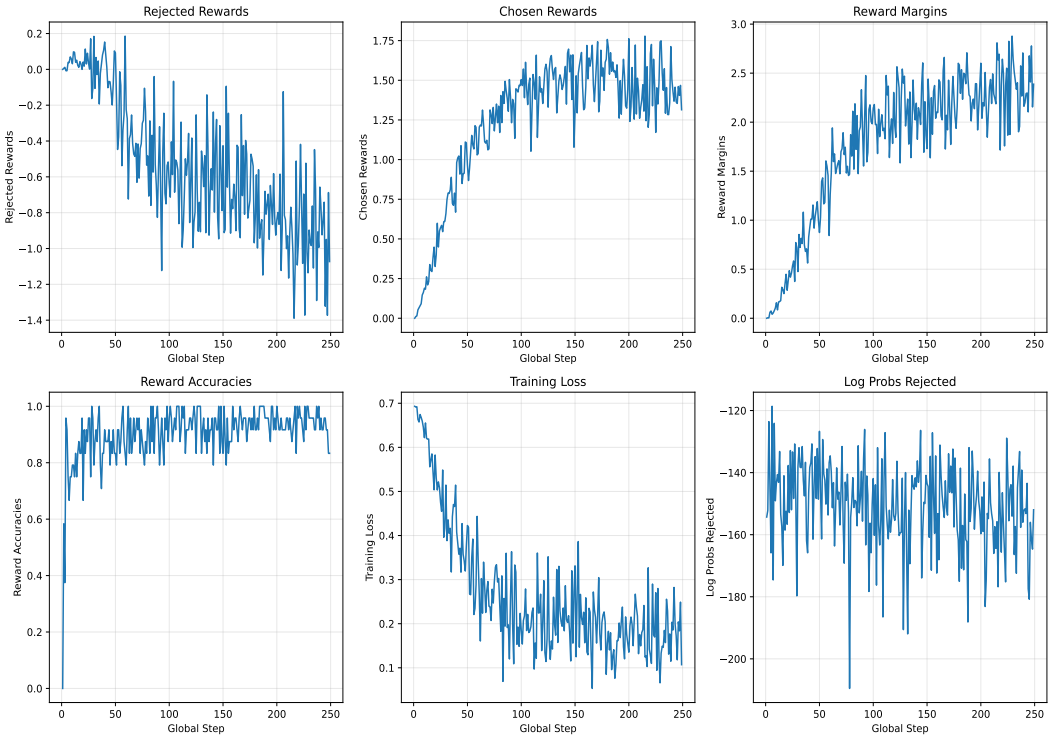
P Comparison of RL on Debate and Single-Round Majority Vote in the Single-Agent Setting

Table 23 reports the broader single-agent comparison across four model sizes and two benchmarks, including standard deviation over three seeds. These results complement the single-agent online-RL experiment in Table 5. We compare MACA using final-round majority vote and debate context with single-round majority-vote RL using DPO (akin to ScPO (Prasad et al., 2025)) and GRPO (akin to TTRL (Zuo et al., 2025)).

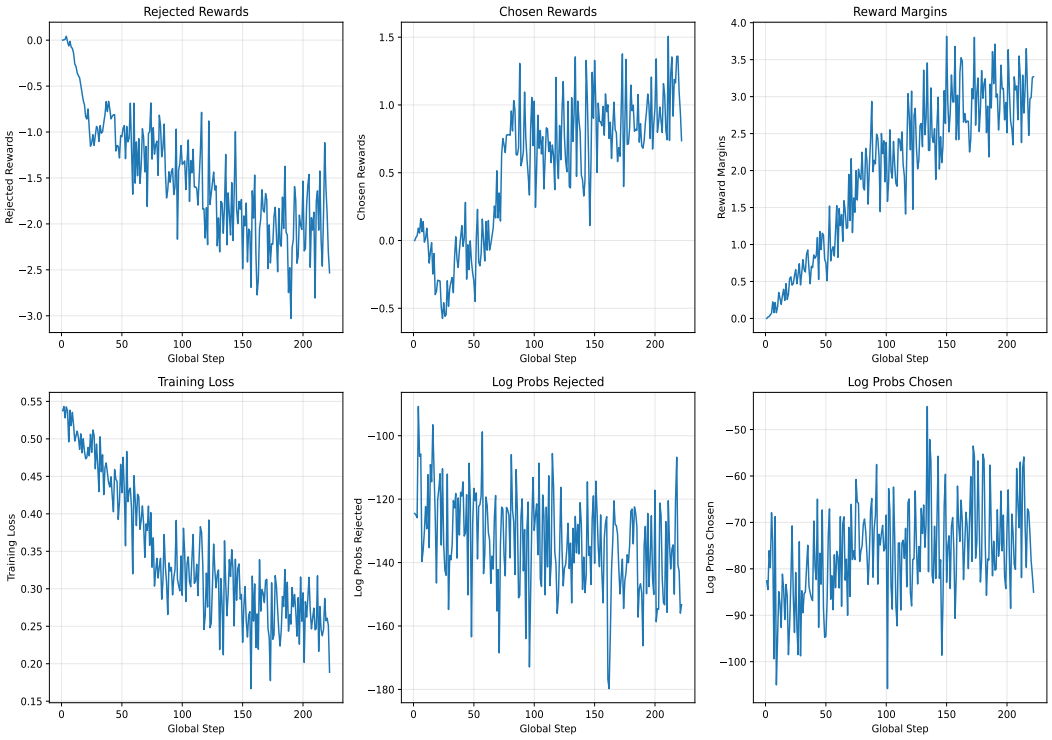
Q Efficiency

Here we compare the compute–performance tradeoffs of MACA's MV-DPO with debate context against other single-agent benchmark methods. ScPO requires 0.075–0.58 GPU hours with -0.6% to +13.6% accuracy changes, TTRL requires 2.2–7.7 GPU hours with -26.4% to +18% changes,

and MACA (MV-DPO with debate) requires 0.73–1.68 GPU hours with positive changes in the evaluated settings. Figure 13 reports these performance changes alongside GPU throughput and training runtime.

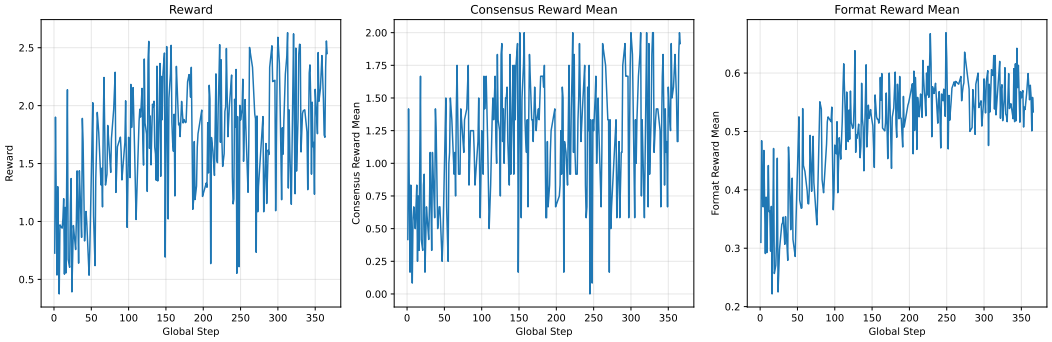


(a) MV-DPO training curves

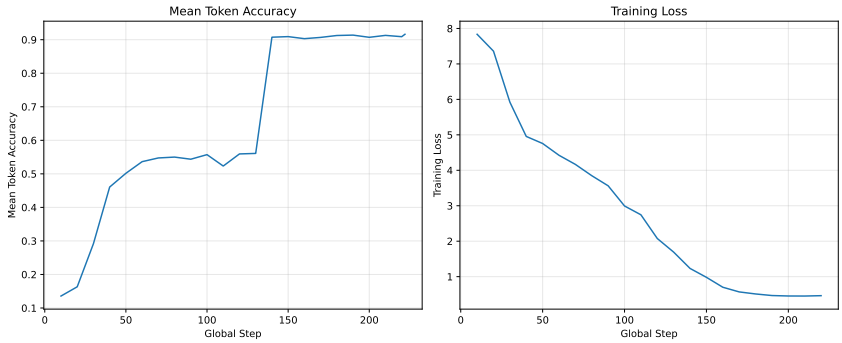


(b) MV-KTO training curves

Figure 7: Training curves for MV-DPO and MV-KTO. (a) MV-DPO: increasing reward margin between majority (chosen) and minority (rejected) responses, with declining rejected log probabilities. (b) MV-KTO: similar reward margin increase using unpaired examples, with rejected log probabilities decreasing and chosen increasing.



(a) MV-GRPO training curves



(b) MV-SFT training curves

Figure 8: Training curves for MV-GRPO and MV-SFT. (a) MV-GRPO: consensus and format rewards both increase. (b) MV-SFT: token accuracy increases while loss decreases.

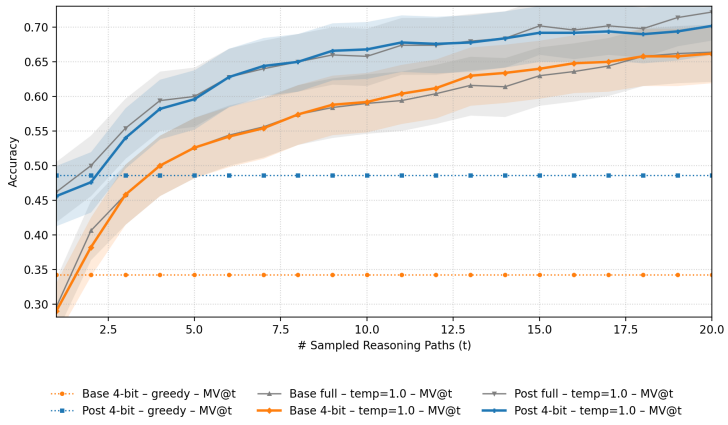


Figure 9: Llama-3B on MathQA (MV@t, t=20). Blue: post-trained 4-bit model, Orange: base 4-bit model, Grey: full-precision model.

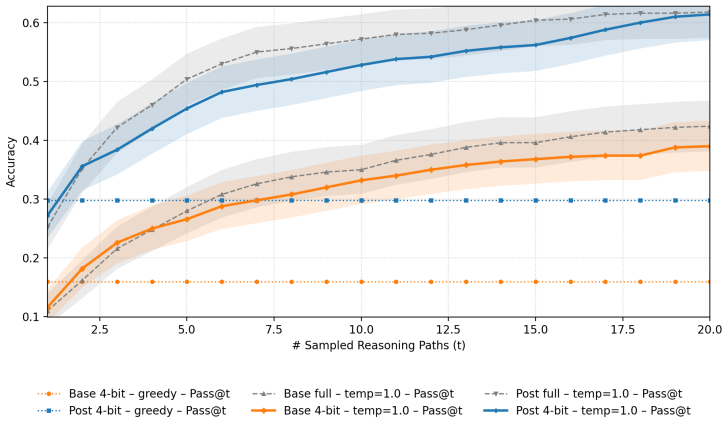


Figure 10: Qwen-2B on MATH (Pass@t, t=20). Blue: post-trained 4-bit model, Orange: base 4-bit model, Grey: full-precision model.

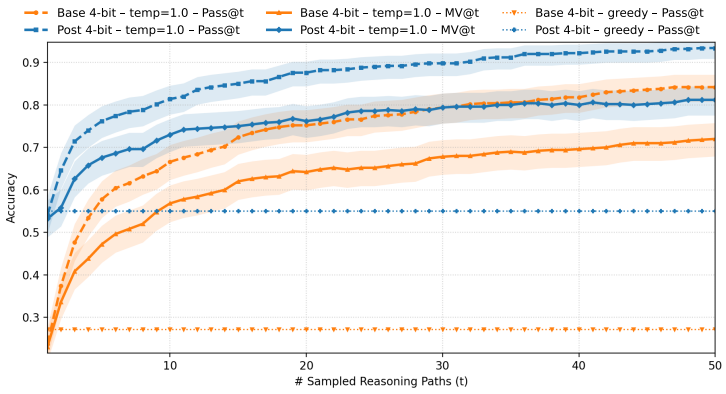


Figure 11: Qwen-2B on GSM8K (Pass@t, MV@t; t=50). Blue: post-trained 4-bit model, Orange: base 4-bit model.

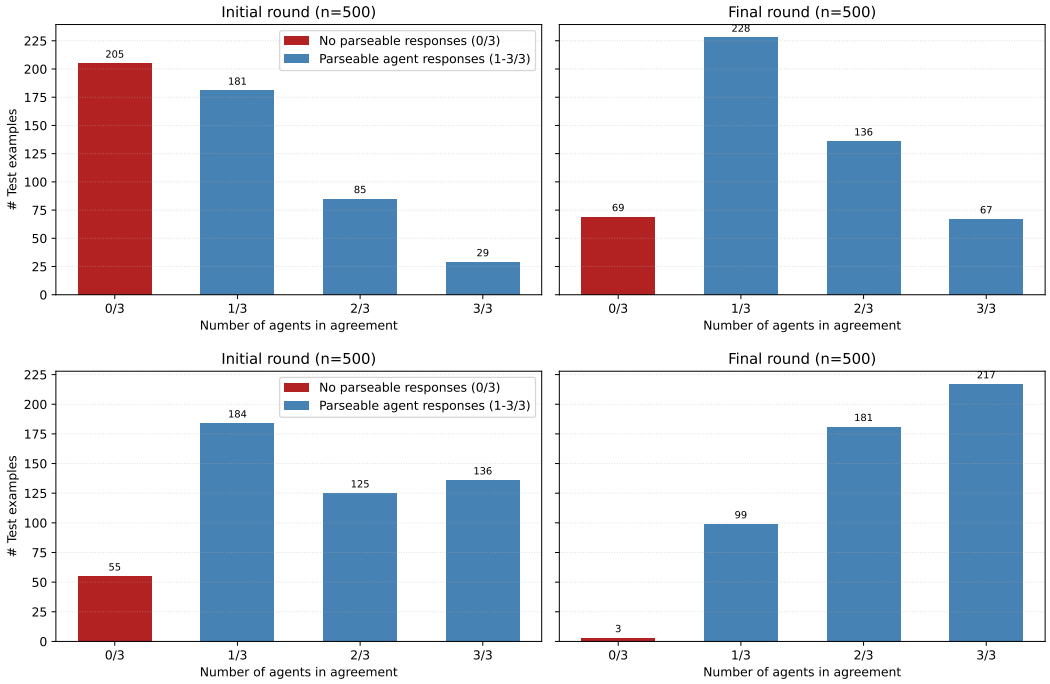


Figure 12: Answer completeness and agent agreement for Qwen2B on GSM8K. Top: base-model debate; bottom: post-trained debate; left: initial round; right: final round.

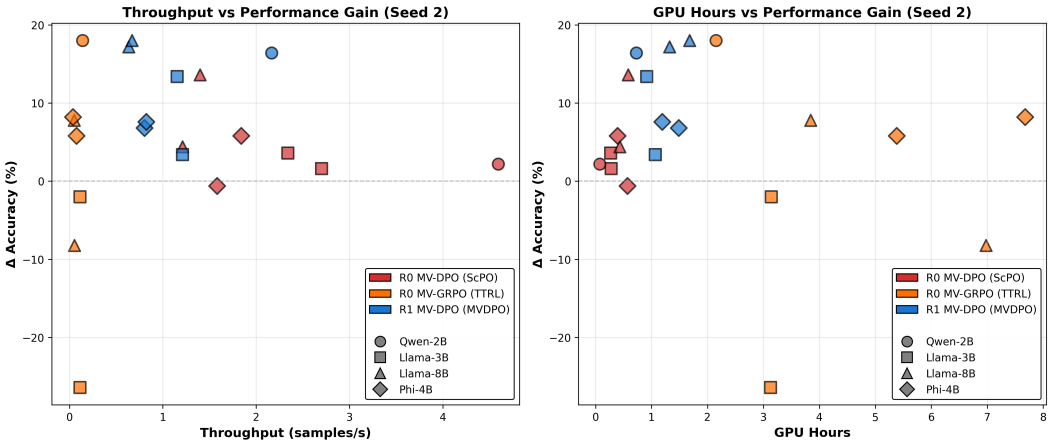


Figure 13: GPU throughput and training runtime compared with performance lift for single-agent post-training methods