

# 000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 UNLOCKING THE POWER OF MULTI-AGENT LLM FOR REASONING: FROM LAZY AGENTS TO DELIBERATION

Anonymous authors

Paper under double-blind review

## ABSTRACT

Large Language Models (LLMs) trained with reinforcement learning and verifiable rewards have achieved strong results on complex reasoning tasks. Recent work extends this paradigm to a multi-agent setting, where a meta-thinking agent proposes plans and monitors progress while a reasoning agent executes subtasks through sequential conversational turns. Despite promising performance, we identify a critical limitation: lazy agent behavior, in which one agent dominates while the other contributes little, undermining collaboration and collapsing the setup to an ineffective single agent. In this paper, we first provide a theoretical analysis showing why lazy behavior naturally arises in multi-agent reasoning. We then introduce a stable and efficient method for measuring causal influence, helping mitigate this issue. Finally, as collaboration intensifies, the reasoning agent risks getting lost in multi-turn interactions and trapped by previous noisy responses. To counter this, we propose a verifiable reward mechanism that encourages deliberation by allowing the reasoning agent to discard noisy outputs, consolidate instructions, and restart its reasoning process when necessary. Extensive experiments demonstrate that our framework alleviates lazy agent behavior and unlocks the full potential of multi-agent framework for complex reasoning tasks.

## 1 INTRODUCTION

Recent advances in prompting and training have markedly improved the multi-step reasoning abilities of large language models (LLMs) (Wei et al., 2022; Kojima et al., 2022; Wang et al., 2022; Zhang et al., 2022; Ton et al., 2024; Yeo et al., 2025; Zhu et al., 2025; Chowdhury & Caragea, 2025; Mukherjee et al., 2025; Balcan et al., 2025). Techniques such as chain-of-thought prompting (Wei et al., 2022; Kojima et al., 2022) and structured methods like Tree-of-Thoughts and Graph-of-Thoughts (Yao et al., 2023; Besta et al., 2024) expand the space for deliberation. Building on this, Large Reasoning Models (LRMs) trained with supervised and reinforcement learning using verifiable rewards achieve strong performance on math, code, and planning tasks (Jaech et al., 2024; Guo et al., 2025a; Comanici et al., 2025; Plaat et al., 2024; Huang & Chang, 2022; Zhang et al., 2023b; Li et al., 2025b; Chen et al., 2025). More recently, multi-agent frameworks enable LLMs with specialized roles to collaborate via planning, delegation, and debate, echoing human team dynamics (Li et al., 2023; Wu et al., 2024a; Chen et al., 2023; Du et al., 2023; Yuan & Xie). Likewise, single-agent multi-turn interaction settings have gained attention as another path to enhance reasoning (Wan et al., 2025; Shi et al., 2025; Wei et al., 2025; Zhou et al., 2025; Li et al., 2025c; Lu et al., 2025; Wang et al., 2025; Zhang et al., 2025a; Zeng et al., 2025; Jin et al., 2025).

To support multi-agent and multi-turn reinforcement learning, multi-turn Group Relative Policy Optimization (GRPO) (Wan et al., 2025; Shi et al., 2025; Wei et al., 2025) and its variants (Guo et al., 2025b; Zhang et al., 2025c; Ning et al., 2025; Xue et al., 2025) compute advantages and importance ratios at the turn level, enabling finer-grained optimization and more precise credit assignment. Building on this foundation, ReMA (Wan et al., 2025) introduces a multi-agent LLM reasoning framework with two specialized roles: a *meta-thinking agent*, which decomposes tasks, sets intermediate goals, and adapts based on feedback, and a *reasoning agent*, which performs step-by-step computations and proofs before returning intermediate results. The agents alternate sequentially, but since only a final outcome reward is available, ReMA computes a group advantage following GRPO (Shao et al., 2024) and uniformly assigns this trajectory-level signal to every turn in the rollout.

Despite its effectiveness, we empirically find that ReMA suffers from a critical issue of lazy agents: one of the agents contributes only trivially to the multi-agent system. Although this phenomenon has been widely acknowledged in traditional multi-agent reinforcement learning under sparse-reward settings (Sunehag et al., 2018; Foerster et al., 2018; Castellini et al., 2022; Jaques et al., 2019; Wang et al., 2020; Liu et al., 2023), prior work has primarily focused on scenarios where multiple agents act simultaneously. *In contrast, our findings are surprising because agents in our setting act sequentially.* An early agent’s trivial action not only fails to contribute but also shapes the state for subsequent agents. As later decisions depend on this evolving state, a lazy action can misguide the reasoning trajectory and compound its negative influence. Intuitively, such interdependence across turns should discourage laziness, especially as overall performance improves during training. However, contrary to this expectation, we find that ReMA-trained reasoning agents still adopt shortcut behaviors. As shown in Section 4, our case study reveals that reasoning agents often contribute only trivially, typically by summarizing or copying the meta-thinking agent’s responses without genuine questioning or reflection. As a result, the meta-thinking agent ends up carrying almost the entire reasoning process. Our causal-effect experiments further show that while both agents initially contribute substantially when initialized from the base model, the reasoning agent’s influence diminishes markedly as training progresses, leaving the meta-thinking agent dominant.

The critical issue of lazy agents in multi-agent systems risks collapsing the entire system into a single agent, thereby limiting the potential benefits of collaboration in improving performance. In this paper, we propose Multi-Agent Meta-Reasoning Done Right (**Dr. MAMR**). We begin with a theoretical analysis of multi-turn GRPO to investigate the root cause of lazy agent behavior and identify a key bias in its loss formulation: the normalization term, intended to prevent sequence-level bias toward longer rollouts, inadvertently drives the model to prefer continuations that minimize the number of turns given the same prefix. As a result, agents are implicitly incentivized to complete reasoning with fewer interactions, often bypassing collaborative reflection or correction, and over time, this dynamic gives rise to lazy agents that contribute little to the reasoning process. Our theoretical insight not only explains the emergence of lazy agents but also sheds light on future work in designing objectives for multi-turn reinforcement learning.

While correcting the loss formulation partially mitigates the problem, it does not eliminate it. To further address this issue, we propose measuring the causal influence (Bogdan et al., 2025) of each reasoning step on subsequent process. A challenge arises in online training: the policy generates only a single continuation per step, so the estimated influence reflects just one trajectory. In contrast, considering multiple continuations would show how the step contributes across diverse trajectories, providing a more reliable estimate of its overall contribution and mitigating potential bias introduced by phrasing (Pavlick & Callison-Burch, 2016; McCoy et al., 2019; Merrick & Taly, 2020; Li et al., 2024), but such resampling is computationally prohibitive in online RL. Inspired by Feng et al. (2025); Li et al. (2021), we introduce a Shapley-inspired causal influence method. Instead of evaluating each step in isolation, we group semantically similar steps across rollouts and average their influence scores. This avoids additional sampling and produces robust estimates during training.

As lazy behavior diminishes and agents engage more productively, interaction frequency increases. However, as Laban et al. (2025) show, LLMs in multi-turn settings often overcommit to incomplete early context, making premature assumptions. A similar risk arises here: the meta-thinking agent acts like a user providing incremental instructions, while the reasoning agent may become misled by its own earlier outputs, as confirmed in Sec. 5.3. To overcome this, we propose training the reasoning agent to adaptively discard its prior outputs, re-aggregate instructions, and restart reasoning when needed. To accurately credit such restart behavior, we design a novel verifiable reward mechanism. Building on this, we assign step-level credit by aggregating outcome reward, causal influence, and restart signals. Extensive experiments demonstrate that our method effectively mitigates lazy-agent behavior and unlocks the potential of multi-agent frameworks for complex reasoning.

Overall, our contributions are as follows: **(1)** We identify a critical issue of lazy agents in multi-agent reasoning frameworks and provide a theoretical analysis of multi-turn GRPO to explain its underlying cause. **(2)** We propose a Shapley-inspired method for measuring causal influence at the step level, further mitigating the lazy agent problem. **(3)** As agents engage in more frequent collaboration, we design a novel verifiable reward mechanism for restart behavior, enabling the reasoning agent to recover from noisy intermediate steps and avoid getting lost in prolonged interactions, thereby pushing the performance boundary of multi-agent LLMs on complex reasoning tasks.

108 **2 RELATED WORK**

110 **Multi-Agent RL.** Multi-agent RL (MARL) studies how agents coordinate to maximize collective  
 111 rewards, with credit assignment as a central challenge. Classical approaches include value decom-  
 112 position (Sunehag et al., 2018), counterfactual baselines (Foerster et al., 2018), regression-based  
 113 rewards (Castellini et al., 2022), role-based coordination (Wang et al., 2020), and model-based in-  
 114 fluence estimation (Liu et al., 2023). With LLM agents, MARL has been adapted for multi-turn  
 115 reasoning and dialogue, e.g., turn-level credit assignment (Zeng et al., 2025), critic-driven step-  
 116 wise rewards (Zhou et al., 2025), communication-efficient training (Liao et al., 2025), addressing  
 117 coarse reward traps (Wang et al., 2025), and framing LLM collaboration via MAGRPO (Liu et al.,  
 118 2025a). A persistent issue is lazy agents, motivating causal influence estimation (Bogdan et al.,  
 119 2025; Nguyen et al., 2025; Liu et al., 2024b) to enable finer-grained credit assignment.

120 **LLM reasoning.** Large Language Models (LLMs) excel across diverse NLP tasks (Brown et al.,  
 121 2020; Chowdhery et al., 2023; Du et al., 2022; Dubey et al., 2024; Wenzek et al., 2019). Chain-  
 122 of-thought prompting improves reasoning by eliciting intermediate steps (Wei et al., 2022; Kojima  
 123 et al., 2022; Nye et al., 2021), while extensions like Tree-of-Thoughts and Graph-of-Thoughts en-  
 124 able structured, non-linear reasoning (Yao et al., 2023; Besta et al., 2024). These advances motivate  
 125 Large Reasoning Models (LRMs) (Guo et al., 2025a; Achiam et al., 2023; Grattafiori et al., 2024;  
 126 Xu et al., 2023; Zhou et al., 2022; Wu et al., 2024b; Qi et al., 2024; Chae et al., 2024), which com-  
 127 bine supervised fine-tuning and reinforcement learning to achieve state-of-the-art results on math,  
 128 coding, and planning (Jaech et al., 2024; Guo et al., 2025a; Comanici et al., 2025; Yang et al., 2024a;  
 129 2025; Lightman et al., 2023; Wang et al., 2023). Beyond single-model reasoners, multi-agent frame-  
 130 works leverage role assignment, orchestration, and debate to coordinate specialized LLM agents for  
 131 complex tasks (Li et al., 2023; Wu et al., 2024a; Chen et al., 2023; Du et al., 2023; Yuan & Xie).

132 A detailed review of related work on MARL, hierarchical RL and LLM reasoning is in Appendix A.

133 **3 BACKGROUND**

136 ReMA (Wan et al., 2025) models reasoning as a *multi-turn meta-thinking process* defined as:

$$\mathbf{x} \xrightarrow[\pi_h]{\text{meta-thinking}} \mathbf{m}_1 \xrightarrow[\pi_l]{\text{reasoning}} \mathbf{y}_1 \xrightarrow[\pi_h]{\text{meta-thinking}} \mathbf{m}_2 \xrightarrow[\pi_l]{\text{reasoning}} \mathbf{y}_2 \dots \xrightarrow[\pi_l]{\text{reasoning}} \mathbf{y}_T, \quad (1)$$

139 where  $T$  is the number of turns. The high-level policy  $\pi_h$  (*meta-thinking agent*) generates meta-  
 140 level thoughts  $\mathbf{m}_t$  from the input  $\mathbf{x}$  and history  $\{\mathbf{m}, \mathbf{y}\}_{<t}$ , while the low-level policy  $\pi_l$  (*reasoning  
 141 agent*) produces token-level outputs  $\mathbf{y}_t$  under the guidance of  $\mathbf{m}_t$ . To improve training efficiency,  
 142 both agents share the same model weights  $\theta$  but are distinguished by role-specific system prompts  
 143  $S_h$  and  $S_l$ :  $\pi_h = \pi_\theta(\cdot | S_h, \cdot), \pi_l = \pi_\theta(\cdot | S_l, \cdot)$ .

144 **Multi-turn GRPO** (Wan et al., 2025; Wei et al., 2025; Shi et al., 2025) extends GRPO (Guo et al.,  
 145 2025a) to support end-to-end multi-turn tasks such as mathematical reasoning (Wan et al., 2025)  
 146 and web-based agent decision-making (Wei et al., 2025). A key innovation is the introduction of a  
 147 turn-level importance ratio, enabling fine-grained credit assignment across dialogue turns.

148 Specifically, given the dataset  $\mathcal{D}$  and  $G$  trajectories for each question, the objective is defined as:

$$\begin{aligned} \mathcal{J}(\theta) = & \mathbb{E}_{(\mathbf{x}, \mathbf{y}^*) \sim \mathcal{D}, \{(\mathbf{m}_i, \mathbf{y}_i)\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot | \mathbf{x})} \\ & \left[ \frac{1}{G} \sum_{i=1}^G \frac{1}{T_i} \sum_{t=1}^{T_i} \frac{1}{|\mathbf{y}_{i,t}|} \sum_{j=1}^{|\mathbf{y}_{i,t}|} \left( \min \left( r_{i,t}(\theta) \hat{A}_{i,t,j}, \text{clip}(r_{i,t}(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_{i,t,j} \right) - \beta D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) \right) \right], \end{aligned} \quad (2)$$

154 where  $\mathbf{y}_{i,t,j}$  denotes the  $j$ -th token at turn  $t$  in trajectory  $i$ ,  $\hat{A}_{i,t,j}$  is the token-level advantage and  
 155  $\frac{1}{T_i}$  is a normalization to avoid bias toward rollouts with more turns. The turn-level importance ratio  
 156  $r_{i,t}(\theta)$  is computed as:

$$r_{i,t}(\theta) = \frac{1}{|\mathbf{y}_{i,t}|} \sum_{j=1}^{|\mathbf{y}_{i,t}|} \frac{\pi_\theta(\mathbf{y}_{i,t,j} | \mathbf{x}, \{\mathbf{m}_{i, \cdot}, \mathbf{y}_{i, \cdot}\}_{<t}, \mathbf{m}_{i,t}, \mathbf{y}_{i,t, <j})}{\pi_{\theta_{\text{old}}}(\mathbf{y}_{i,t,j} | \mathbf{x}, \{\mathbf{m}_{i, \cdot}, \mathbf{y}_{i, \cdot}\}_{<t}, \mathbf{m}_{i,t}, \mathbf{y}_{i,t, <j})}, \quad (3)$$

161 which aggregates token-level likelihood ratios within each reasoning turn. Similar variants of Eq. 2  
 162 were also proposed in (Guo et al., 2025b; Zhang et al., 2025c; Ning et al., 2025; Xue et al., 2025).

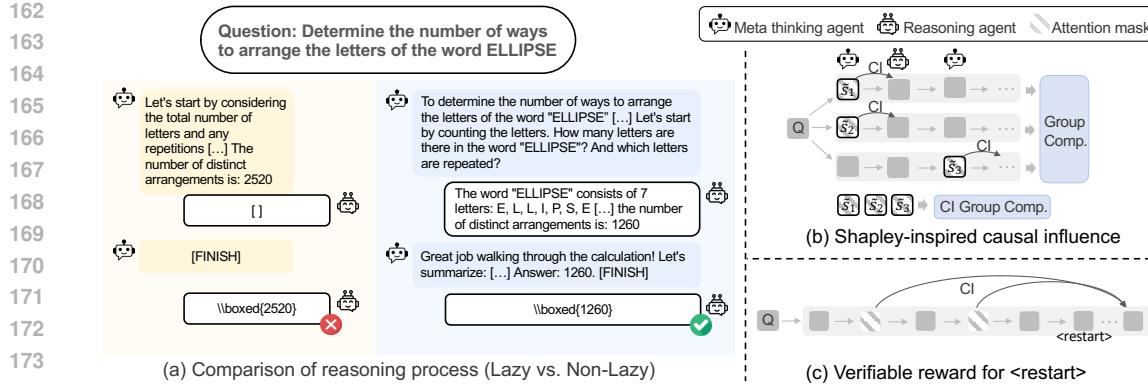


Figure 1: (a) Case study on lazy agents (full process in Appendix D); (b–c) our proposed modules.

#### 4 THE LAZY AGENT ISSUE IN MULTI-AGENT LLM REASONING

In this section, we present empirical evidence of the lazy-agent problem in the multi-agent framework ReMA (Wan et al., 2025). As illustrated in Fig. 1(a), the reasoning agent often outputs blanks at intermediate steps, shifting the burden to the meta-thinking agent and ultimately leading to incorrect answers. By contrast, when both agents actively contribute, collaboration yields correct solutions. To quantify laziness, we measure the causal influence of an agent’s actions by adapting the attention-suppression method from Bogdan et al. (2025). Let  $s_t$  denote an action taken by either agent. We suppress all attention (across layers and heads) to the tokens corresponding to  $s_t$  and define the influence on the subsequent action  $s_{t+1}$  as the KL divergence between the model’s logits with and without suppression. Intuitively, a small divergence indicates that the agent’s step has little impact on subsequent reasoning and thus reflects lazy behavior, whereas a large divergence shows that the step substantially shapes the reasoning process. See Appendix C.1 for experimental details.

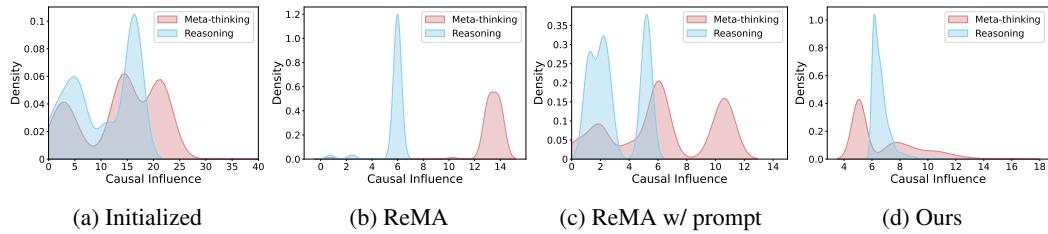


Figure 2: Causal effect comparison. Performance on MATH500 under different configurations: (a) 75.0, (b) 74.4, (c) 75.6, and (d) 78.4.

To examine whether the lazy-agent issue arises from the ReMA framework, we compare causal influence under three settings: (1) untrained agents initialized from the base model, (2) agents trained with ReMA, and (3) agents trained with ReMA but prompted with instructions discouraging trivial responses (details in Appendix C.1). The results are shown in Fig. 2(a–c). (1) Compared to the untrained baseline, the reasoning agent in the standard ReMA setting contributes substantially less than the meta-thinking agent, revealing clear lazy-agent behavior. This imbalance coincides with a performance drop from 75.0 to 74.4 on MATH500 despite training. (2) Adding a prompt to encourage non-trivial responses narrows the causal-effect gap and improves performance from 74.4 to 75.6. However, the reasoning agent still shows weaker influence than in the baseline, and the modest 0.6-point gain suggests reliance on shortcuts rather than meaningful reasoning. In summary, ReMA is prone to producing lazy agents. While prompt engineering can partially mitigate the issue, it does not fully resolve it. This underscores the need for more robust methods, particularly in online reinforcement learning, to ensure balanced agent contributions.

## 216 5 DR. MAMR: MULTI-AGENT META-REASONING DONE RIGHT

### 217 5.1 THEORETICAL ANALYSIS ON THE EMERGENCE OF LAZY AGENT

220 Our preliminary experiments in Sec. 4 reveal the critical issue of lazy agents in advanced multi-  
 221 agent reasoning frameworks. In this section, we provide a theoretical analysis of why such behavior  
 222 emerges, even when the overall system performance appears to improve. Following GRPO (Shao  
 223 et al., 2024) and DAPO (Yu et al., 2025), multi-turn GRPO (Wan et al., 2025; Wei et al., 2025)  
 224 introduces a normalization term to reduce sequence-level bias toward longer rollouts. As shown in  
 225 Eq. 2, the objective includes a factor  $\frac{1}{T_i}$  that averages the turn-level advantages across each trajectory.  
 226 However, we find that this normalization introduces a structural bias: *given the same context, if two*  
 227 *alternative actions produce trajectories with equal final reward but different numbers of turns, the*  
 228 *model favors the action leading to fewer turns.* To illustrate, we consider the reasoning agent as an  
 229 example and present the following theory:

230 **Theorem 1** *Let  $g_t(\tau) = \frac{1}{T(\tau)} Z_t(\tau)$  be the gradient contribution at turn  $t$  for trajectory  $\tau$  with*  
 231  *$Z_t(\tau) \triangleq \frac{1}{|y_t|} \sum_{j=1}^{|y_t|} r_t(\theta) \hat{A}_{t,j} \nabla_\theta \log \pi_\theta(y_{t,j} | x, m_{\leq t}, y_{<t}, y_{t,j})$ . Consider two continuations*  
 232 *from the same prefix: a short trajectory  $\tau^S$  with horizon  $T_S$  and a long trajectory  $\tau^L$  with horizon*  
 233  *$T_L > T_S$ , leading to the same final reward. Define  $\kappa \triangleq \frac{\|Z_t(\tau^L)\|}{\|Z_t(\tau^S)\|}$ . If  $\kappa < \frac{T_L}{T_S}$ , then  $\frac{\|g_t(\tau^S)\|}{\|g_t(\tau^L)\|} > 1$ .*

234 This theorem shows that unless the aggregated contribution  $Z_t(\tau^L)$  is at least  $\frac{T_L}{T_S}$  times larger than  
 235  $Z_t(\tau^S)$ , the gradient update favors the trajectory with fewer turns. Importantly, this holds whether  
 236 the advantages are both positive or both negative: in the latter case, although both trajectories are  
 237 discouraged, the shorter one is penalized less. Consequently, the model is biased toward actions  
 238 that reduce the number of turns, even if longer trajectories are equally rewarding. Empirically, our  
 239 results in Appendix F show that reasoning processes with lazy-agent behavior (e.g., producing empty  
 240 outputs or simply summarizing) consistently involve fewer turns than those without lazy agents at  
 241 the initial training stages, which are critical in shaping policy behavior. *Together, our theorem*  
 242 *and empirical findings explain the emergence of lazy-agent behavior:* the normalization bias steers  
 243 optimization toward trajectories with fewer turns, and reasoning processes exhibiting lazy behavior  
 244 naturally produce shorter trajectories, thereby receiving preferential reinforcement during training.  
 245 We emphasize that our analysis is distinct from Dr.GRPO (Liu et al., 2025b), which examines token-  
 246 level normalization. Their work removes length normalization to prevent the policy from favoring  
 247 shorter correct answers or unnecessarily long incorrect ones. Furthermore, since the number of  
 248 turns  $T$  is far smaller than the number of tokens, the resulting bias in our setting is substantially  
 249 more pronounced.

### 252 5.2 SHAPLEY-INSPIRED CAUSAL INFLUENCE

253 To mitigate the optimization bias in multi-turn GRPO, we first remove the  $\frac{1}{T_i}$  normalization in Eq. 2,  
 254 which alleviates but does not fully eliminate the lazy-agent issue as indicated in ablation study (Sec.  
 255 6.5). Addressing this problem requires measuring the causal influence of each step during online  
 256 training. In practice, however, the policy generates only a single continuation per step, so influence  
 257 must be inferred from one trajectory. This creates two challenges: (1) it offers only a limited view  
 258 of how a step shapes the reasoning process (Xu et al., 2025), and (2) it biases causal influence  
 259 toward specific phrasings rather than underlying ideas (Pavlick & Callison-Burch, 2016; McCoy  
 260 et al., 2019; Merrick & Taly, 2020; Li et al., 2024). Analogous to Shapley values (Li et al., 2021),  
 261 which attribute contributions by averaging marginal effects across all coalitions, step-level causal  
 262 influence in multi-agent RL should reflect average contributions across possible continuations rather  
 263 than a single path. Directly computing such Shapley-style values, however, is infeasible because it  
 264 would require extensive resampling during online RL. To make this tractable, we propose a stable  
 265 Shapley-inspired causal influence measure.

266 We flatten each trajectory into a sequence  $s_{i,1}, s_{i,2}, \dots, s_{i,2T}$ , where  $s_{i,2t-1} = m_{i,t}$  (meta-thinking)  
 267 and  $s_{i,2t} = y_{i,t}$  (reasoning). Each step  $s_{i,t}$  is treated as an *anchor step*, for which we form a group  
 268 of semantically similar steps:

$$269 G_S(s_{i,t}) = \{s_{j,t'} \mid s_{j,t'} \approx s_{i,t}, 1 \leq j \leq N, 1 \leq t' \leq 2T_j\},$$

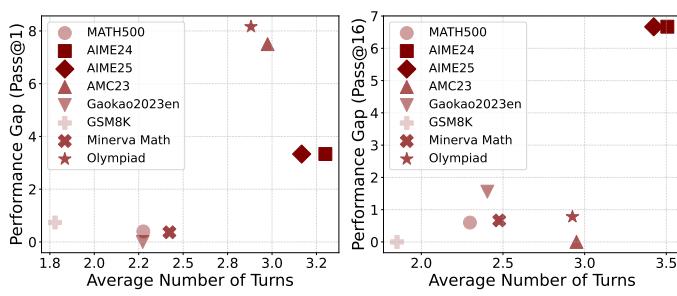


Figure 3: Performance gap between ReMA+ and ReMA across 8 benchmarks. Left: Pass@1. Right: Pass@16. Darker colors represent more difficult benchmarks.

where  $\approx$  denotes semantic similarity. Similarity can be easily measured through semantic distance (see Appendix B), ensuring that steps within the same group express a comparable idea. For each step  $s_{j,t'} \in G_S(s_{i,t})$ , we measure its *one-step causal influence* on the next step. Let  $h_{\leq t'}$  be the local history up to and including step  $t'$ , and  $h_{\leq t'}^{(j)\setminus t'}$  the masked history with  $s_{j,t'}$  removed. We compare the probability of the next output under the full and masked histories:

$$p_{\text{full}}^{(j,t')} = \pi_\theta(s_{j,t'+1} \mid h_{\leq t'}^{(j)}), \quad p_{\text{mask}}^{(j,t')} = \pi_\theta(s_{j,t'+1} \mid h_{\leq t'}^{(j)\setminus t'}), \quad \Delta\ell_{j,t'} \triangleq \log p_{\text{mask}}^{(j,t')} - \log p_{\text{full}}^{(j,t')}.$$
(4)

Finally, the Shapley-inspired causal influence of an anchor step  $s_{i,t}$  is the average across its group:

$$\text{CI}(s_{i,t}) = \frac{1}{|G_S(s_{i,t})|} \sum_{(j,t') : s_{j,t'} \in G_S(s_{i,t})} \Delta\ell_{j,t'}.$$
(5)

Our method ensures reliable causal influence estimation by: (1) averaging the impact of semantically similar steps across rollouts to obtain a stable estimate of an idea’s overall contribution, and (2) aggregating different phrasings of the same idea to reduce wording bias in influence estimates.

### 5.3 REASONING AGENT DELIBERATION FOR MULTI-TURN INTERACTIONS

As each agent contributes more actively, the number of dialogue turns between the meta-thinking and reasoning agents increases. However, prior work shows that longer multi-turn interactions can degrade performance: Laban et al. (2025) compare LLMs in (1) a single-turn setting where the full task is given in one prompt, and (2) a multi-turn setting where the task is decomposed into incremental prompts. They report consistent performance drops in the multi-turn condition, likely because LLMs overcommit to underspecified early context and struggle to recover from initial errors. These findings imply that while multi-agent collaboration can enrich reasoning, it also heightens vulnerability to error propagation when intermediate turns introduce ambiguity. If we view the meta-thinking agent as a user providing step-by-step instructions, then more interactions risk the reasoning agent becoming “lost” in dialogue, as observed by Laban et al. (2025). To mitigate this, we hypothesize that allowing the reasoning agent to discard prior responses, aggregate the meta-thinking prompts, and restart reasoning would be beneficial.

To validate this assumption, we conduct preliminary experiments following the ReMA framework to obtain a meta-thinking agent and a reasoning agent. We then compare the standard ReMA framework with a modified variant, **ReMA+**, where the reasoning agent is guided by a refined system prompt that enables it to adaptively discard its previous outputs when necessary. The full prompt design and additional details are provided in Appendix C.2. In this experiment, we modify the system prompt only at inference time. Effectiveness is evaluated by (i) the performance gap in validation accuracy between ReMA+ and ReMA. Results across eight benchmarks are shown in Fig. 3. From the figure, we observe: (1) Without explicit training for deliberation, ReMA+ consistently matches or outperforms ReMA, with gains of about 8% on AMC23 and Olympiad under Pass@1, and 7% on AIME24 and AIME25 under Pass@16. (2) Even under Pass@16, which gives both frameworks ample chances to succeed, ReMA+ outperforms ReMA on 6 of 8 benchmarks—demonstrating that while LLMs have this capacity, explicit prompting is required for consistent behavior. (3) The per-

324 formance gap widens as benchmark difficulty and the number of turns increase, highlighting the  
 325 value of adaptive deliberation in extended multi-agent interactions.  
 326

327 Building on these observations, we design a method that trains the reasoning agent to adaptively dis-  
 328 card its previous outputs to improve the likelihood of producing a correct final answer. Specifically,  
 329 we introduce a control token, `<restart>`, which instructs the agent to discard prior responses,  
 330 consolidate the instructions, and begin a fresh attempt. To assess the benefit of this mechanism,  
 331 we develop a novel verifiable criterion that measures how discarding history affects the model’s  
 332 probability of generating the correct final output.  
 333

334 **Verifiable Reward for Restart.** Consider the  $i$ -th rollout where the reasoning agent outputs  
 335 `<restart>` at turn  $t$ . In this case, we mask all reasoning-agent outputs that occur strictly be-  
 336 fore  $t$ :  $\mathcal{Y}_{<2t}^{(i)} \triangleq \{s_{j,k} \mid k < 2t, k \bmod 2 = 0\}$ , and define the causal influence of the restart action  
 337 on the final reasoning step  $\mathbf{y}_T^{(i)} = s_{i,2T}$  as

$$\Delta\ell_{i,t} \triangleq \log \pi_\theta(s_{i,2T} \mid h_{\leq 2T}^{(i) \setminus \mathcal{Y}_{<2t}^{(i)}}) - \log \pi_\theta(s_{i,2T} \mid h_{\leq 2T}^{(i)}). \quad (6)$$

338 We define a binary outcome reward  $z_i$  as  $+1$  if the final answer  $\mathbf{y}_T^{(i)}$  is correct, and  $-1$  if it is  
 339 incorrect. The restart reward is then  
 340

$$r_{i,t}^{\text{restart}} = \begin{cases} +1, & \text{if } (z_i = +1 \wedge \Delta\ell_{i,t} > 0) \text{ or } (z_i = -1 \wedge \Delta\ell_{i,t} < 0), \\ -1, & \text{if } (z_i = +1 \wedge \Delta\ell_{i,t} < 0) \text{ or } (z_i = -1 \wedge \Delta\ell_{i,t} > 0), \\ 0, & \text{if } \Delta\ell_{i,t} = 0. \end{cases} \quad (7)$$

341 This reward provides a verifiable signal of whether the restart improves or worsens the model’s  
 342 belief in the final output. If the final answer is correct ( $z_i = +1$ ), the restart is rewarded when  
 343 masking prior reasoning increases confidence ( $\Delta\ell_{i,t} > 0$ ); otherwise it is penalized. The converse  
 344 holds when the final answer is incorrect ( $z_i = -1$ ).  
 345

346 **Aggregated Step-Level Advantage.** Let  $\text{CI}(s_{i,t})$  (Eq. 5) denote the Shapley-inspired causal influ-  
 347 ence for step  $t$  in rollout  $i$ , and let  $r_{i,t}^{\text{restart}}$  be the verifiable restart reward (Eq. 7), defaulting to 0 if  
 348 no `<restart>` is issued. For normalization, each signal  $x$  is first rescaled to  $\tilde{x} \in [-1, 1]$  using  
 349 min–max scaling (details in Appendix B), and then standardized across all rollouts with mean  $\mu_{\tilde{X}}$   
 350 and standard deviation  $\sigma_{\tilde{X}}$  by  $\mathcal{Z}_X(x) \triangleq \frac{\tilde{x} - \mu_{\tilde{X}}}{\sigma_{\tilde{X}}}$ . We apply this procedure to obtain the normalized  
 351 causal signal  $\tilde{C}_{i,t} = \mathcal{Z}_{\{\text{CI}\}}(\text{CI}(s_{i,t}))$  and restart signal  $\tilde{R}_{i,t} = \mathcal{Z}_{\{r_{i,t}^{\text{restart}}\}}(r_{i,t}^{\text{restart}})$ . The overall  
 352 step-level advantage is then defined as a weighted combination:  
 353

$$A_{i,t}^{\text{step}} = \tilde{A}_{i,t} + \alpha \tilde{C}_{i,t} + \beta \tilde{R}_{i,t}, \quad (8)$$

354 where  $\tilde{A}_{i,t}$  is the normalized outcome-based advantage, and  $\alpha, \beta$  are tunable hyperparameters. The  
 355 training objective for Dr. MAMR builds on Eq. 2, removing the  $\frac{1}{T}$  normalization and replacing the  
 356 advantage function with Eq. 8. See implementation details in Appendix B.  
 357

## 364 6 EXPERIMENTS

### 365 6.1 EXPERIMENT SETTINGS

366 **Dataset and Benchmarks:** We conduct experiments on mathematical reasoning by training models  
 367 on DeepScaleR dataset (Luo et al., 2025). The optimized agents are then evaluated across seven  
 368 benchmarks: MATH500 (Lightman et al., 2023), GSM8K (Cobbe et al., 2021), AIME , AMC23  
 369 (Jia LI, 2024) , GaoKao2023En (Zhang et al., 2023a), Minerva Math (Lewkowycz et al., 2022), and  
 370 OlympiadBench (He et al., 2024). See implementation details in Appendix B.  
 371

372 **Baselines and Models:** We compare Dr.MAMR against three baselines: (1) **GRPO** (Guo et al.,  
 373 2025a), where the base model is trained with vanilla GRPO in a single-agent setting; (2) **VRP**  
 374 (**CoT**) (Wan et al., 2025), where the base model is prompted step by step to operate within the  
 375 meta-thinking and reasoning framework; and (3) **ReMA** (Wan et al., 2025), a multi-agent meta-  
 376 reasoning framework trained with multi-turn GRPO. We conduct training and evaluation on the  
 377 Qwen2.5 family, using the 3B, 7B, and 14B Instruct models (Team, 2024).  
 378

378 6.2 RESULTS ON SEVEN BENCHMARKS  
379380 **Question 1:** *How does Dr. MAMR perform in reasoning tasks compared to baselines?*  
381

382 Table 1 reports pass@1 performance across seven benchmarks using 7B and 14B base models. (See  
383 Appendix H.7 for 3B results). From the results, we observe the following: (1) ReMA consistently  
384 underperforms compared to single-agent GRPO, highlighting the severity of the lazy agent issue  
385 and its detrimental effect on multi-agent performance. (2) Dr. MAMR consistently outperforms  
386 single-agent GRPO across all base models. Notably, the performance gain increases with larger  
387 base models that exhibit stronger instruction-following capabilities. This suggests that mitigating  
388 the lazy agent problem enables effective collaboration between agents and leads to better outcomes.  
389 This also points to a promising direction: improving the instruction-following ability of base models,  
390 which supports more effective communication and collaboration between agents, can in turn lead to  
391 better overall system performance. (3) **Our Dr. MAMR elevates the multi-agent system baseline**  
392 **from performing worse than single-agent GRPO to clearly outperforming it**, demonstrating the  
393 potential of multi-agent frameworks in solving complex reasoning tasks when carefully designed.  
394

395 Table 1: Performance on math benchmarks.  
396

397 <b>Model</b>	398 <b>Benchmark</b>	399 <b>GRPO</b>	400 <b>VRP (CoT)</b>	401 <b>ReMA</b>	402 <b>Dr. MAMR (Ours)</b>
403 <b>Qwen2.5</b> -7B -Instruct	MATH500	75.50	75.00	74.40	<b>78.60</b>
	GSM8K	90.50	92.04	90.60	<b>92.12</b>
	AIME24	16.67	6.67	13.33	<b>20.00</b>
	AMC23	55.00	47.50	50.00	<b>62.50</b>
	Gaokao2023en	64.60	56.62	57.92	<b>65.20</b>
	Minerva Math	34.70	35.66	34.93	<b>38.24</b>
	Olympiad Bench	48.60	38.22	42.58	<b>52.34</b>
<b>Average</b>		55.08	50.24	51.97	<b>58.43</b>
405 <b>Qwen2.5</b> -14B -Instruct	MATH500	<b>80.60</b>	78.40	79.20	80.40
	GSM8K	<b>94.50</b>	92.87	93.59	93.69
	AIME24	16.67	10.00	13.33	<b>26.67</b>
	AMC23	60.00	55.00	60.00	<b>67.50</b>
	Gaokao2023en	64.90	66.23	67.53	<b>69.09</b>
	Minerva Math	41.50	38.60	41.91	<b>43.02</b>
	Olympiad Bench	48.20	46.78	45.12	<b>57.03</b>
<b>Average</b>		58.05	55.41	57.24	<b>62.49</b>

413 6.3 TRAINING CURVES  
414415 **Question 2:** *How does the causal influence of agents evolve during training?*  
416

417 In this section, we present a case study on the 7B model, examining how the causal influence of the  
418 meta-thinking agent and the reasoning agent evolves during training under our Dr. MAMR frame-  
419 work, compared to ReMA. We report the results in Fig. 4 (a). From the figure, we observe: (1)  
420 Under ReMA, the reasoning agent’s causal influence initially increases slightly at the beginning of  
421 training but then steadily decreases, eventually approaching zero, while the meta-thinking agent’s  
422 influence grows significantly as it comes to dominate the reasoning process. This indicates that naive  
423 multi-turn GRPO risks collapsing the system into an effective single-agent setup, losing the benefits  
424 of collaboration. (2) In contrast, under Dr. MAMR, the reasoning agent’s influence steadily in-  
425 creases, while the meta-thinking agent also grow consistently, indicating that both agents contribute  
426 meaningfully. This balanced collaboration explains why Dr. MAMR achieves superior performance  
427 across diverse reasoning tasks compared to ReMA.  
428

429 **Question 3:** *How does Dr. MAMR stabilize multi-agent RL compared to the baseline?*  
430

431 Training stability is a key challenge in multi-agent RL. Thus, we compare the training curves and  
432 report mean rewards of Dr. MAMR and ReMA on the training data, with results for the 7B model

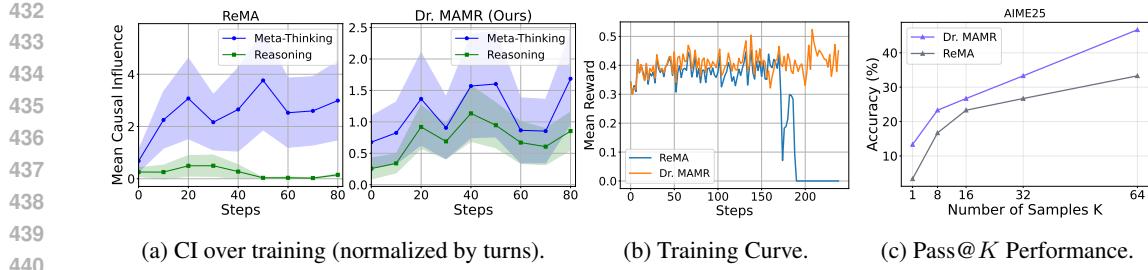


Figure 4: Results on causal influence, training stability, and pass@K.

shown in Fig. 4(b) and additional 3B results provided in Appendix H.8. From the figure, we observe that after 50 steps, Dr. MAMR achieves clearly superior performance. After 150 steps, ReMA collapses with its reward dropping to zero, whereas Dr. MAMR maintains stable training throughout. This demonstrates the benefit of addressing the lazy-agent issue for stabilizing multi-agent RL.

#### 6.4 SCALING ON PASS@K

##### Question 4. How does Dr. MAMR perform when scaling to pass@K?

We examine test-time scaling (Muennighoff et al., 2025; Zhang et al., 2025a) by comparing the pass@K performance of Dr. MAMR and ReMA, which measures whether the correct solution appears within the best result of  $K$  independent attempts. Results on the most challenging benchmark, AIME25, are presented in Fig. 4(c), with additional benchmarks reported in Appendix H.10. The figure shows that **the performance gap between Dr. MAMR and ReMA widens as  $K$  increases, highlighting Dr. MAMR's strength in tackling difficult tasks.**

#### 6.5 ABLATION STUDY

##### Question 5. How does each component contribute to reasoning performance?

We compare the full model against three variants: (1) w/o Normalization Debias (w/o ND), which retains the turn-level normalization from ReMA; (2) w/o Shapley-inspired causal influence (w/o CI); and (3) w/o Restart Behavior (w/o RB). We train these variants and report their benchmark performance in Table 6. From the table, we observe: (1) Dropping either normalization or causal influence causes clear performance drops, showing their complementary role in discouraging shortcuts and promoting balanced contributions. (2) Removing restart behavior also degrades performance across benchmarks, though less severely, highlighting its value in enabling recovery from mistakes and sustaining stable reasoning. We provide a case study on restart behavior in Appendix H.9.

Table 2: Ablation study on the 7B model.

Variant	AIME24	AMC23	Gaokao2023en	Olympiad Bench
Dr.MAMR	20.00	62.50	65.20	52.34
w/o ND	13.33	55.00	63.64	47.85
w/o CI	13.33	52.50	63.38	45.31
w/o RB	16.67	57.50	63.90	50.58

## 7 CONCLUSION

We identify the issue of lazy agents in multi-agent LLM reasoning and trace the issue to the loss structure of multi-turn GRPO. To address it, we introduce a Shapley-inspired causal influence measure and a verifiable reward for restart behavior. Experiments across diverse benchmarks demonstrate that Dr. MAMR effectively mitigates lazy behavior and surpasses strong single- and multi-agent baselines, unlocking the potential of multi-agent frameworks for complex reasoning tasks.

486 ETHICS STATEMENT  
487488 This work does not involve human subjects or the collection of new datasets. Experiments use  
489 established corpora and benchmarks under their licenses.  
490491 REPRODUCIBILITY STATEMENT  
492493 We provided a detailed description implementation details in Appendix B and experimental settings  
494 in Appendix C. Our code is available here: <https://anonymous.4open.science/r/MAMR-D666/>.  
495496 REFERENCES  
497498 Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Ale-  
499 man, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical  
500 report. *arXiv preprint arXiv:2303.08774*, 2023.  
501502 Akhil Bagaria and George Konidaris. Option discovery using deep skill chaining. In *International*  
503 *Conference on Learning Representations*, 2019.504 Maria-Florina Balcan, Avrim Blum, Zhiyuan Li, and Dravyansh Sharma. On learning verifiers for  
505 chain-of-thought reasoning. *arXiv preprint arXiv:2505.22650*, 2025.  
506507 Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawska, Lukas Gian-  
508 inazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczek, et al. Graph of  
509 thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI*  
510 *conference on artificial intelligence*, volume 38, pp. 17682–17690, 2024.511 Paul C Bogdan, Uzay Macar, Neel Nanda, and Arthur Conmy. Thought anchors: Which llm reason-  
512 ing steps matter? *arXiv preprint arXiv:2506.19143*, 2025.  
513514 Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal,  
515 Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are  
516 few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.517 Jacopo Castellini, Sam Devlin, Frans A Oliehoek, and Rahul Savani. Difference rewards policy  
518 gradients. *Neural Computing and Applications*, pp. 1–24, 2022.  
519520 Hyungjoo Chae, Yeonghyeon Kim, Seungone Kim, Kai Tzu-iunn Ong, Beong-woo Kwak,  
521 Moohyeon Kim, Seonghwan Kim, Taeyoon Kwon, Jiwan Chung, Youngjae Yu, et al. Language  
522 models as compilers: Simulating pseudocode execution improves algorithmic reasoning in lan-  
523 guage models. *arXiv preprint arXiv:2404.02575*, 2024.524 Guangyao Chen, Siwei Dong, Yu Shu, Ge Zhang, Jaward Sesay, Börje F Karlsson, Jie Fu, and Yemin  
525 Shi. Autoagents: A framework for automatic agent generation. *arXiv preprint arXiv:2309.17288*,  
526 2023.527 Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu,  
528 Yuhang Zhou, Te Gao, and Wanxiang Che. Towards reasoning era: A survey of long chain-of-  
529 thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*, 2025.  
530531 Jae-Woo Choi, Hyungmin Kim, Hyobin Ong, Youngwoo Yoon, Minsu Jang, Jaehong Kim, et al.  
532 Reactree: Hierarchical task planning with dynamic tree expansion using llm agent nodes. *open-*  
533 *review: https://openreview.net/forum?id=KgKN7F0PyQ*, 2025.534 Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam  
535 Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. PaLM:  
536 Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):  
537 1–113, 2023.538 Jishnu Ray Chowdhury and Cornelia Caragea. Zero-shot verification-guided chain of thoughts.  
539 *arXiv preprint arXiv:2501.13122*, 2025.

540 Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser,  
 541 Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to  
 542 solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

543 Gheorghe Comanici, Eric Bieber, Mike Schaeckermann, Ice Pasupat, Noveen Sachdeva, Inderjit  
 544 Dhillon, Marcel Blstein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the  
 545 frontier with advanced reasoning, multimodality, long context, and next generation agentic capa-  
 546 bilities. *arXiv preprint arXiv:2507.06261*, 2025.

547 Cristina Cornelio, Flavio Petruzzellis, and Pietro Lio. Hierarchical planning for complex tasks with  
 548 knowledge graph-rag and symbolic verification. *arXiv preprint arXiv:2504.04578*, 2025.

549 Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim  
 550 Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. GLaM: Efficient scaling of language  
 551 models with mixture-of-experts. In *International Conference on Machine Learning*, pp. 5547–  
 552 5569. PMLR, 2022.

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

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

559 Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm  
 560 agent training. *arXiv preprint arXiv:2505.10978*, 2025.

561 Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson.  
 562 Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial  
 563 intelligence*, volume 32, 2018.

564 Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In  
 565 *International Conference on Machine Learning*, pp. 10835–10866. PMLR, 2023.

566 Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad  
 567 Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd  
 568 of models. *arXiv preprint arXiv:2407.21783*, 2024.

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

572 Yiran Guo, Lijie Xu, Jie Liu, Dan Ye, and Shuang Qiu. Segment policy optimization: Ef-  
 573 fective segment-level credit assignment in rl for large language models. *arXiv preprint  
 574 arXiv:2505.23564*, 2025b.

575 Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu,  
 576 Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for  
 577 promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint  
 578 arXiv:2402.14008*, 2024.

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

582 Jiaheng Hu, Zizhao Wang, Peter Stone, and Roberto Martín-Martín. Disentangled unsupervised  
 583 skill discovery for efficient hierarchical reinforcement learning. *Advances in Neural Information  
 584 Processing Systems*, 37:76529–76552, 2024.

585 Jen-tse Huang, Jiaxu Zhou, Tailin Jin, Xuhui Zhou, Zixi Chen, Wenxuan Wang, Youliang Yuan,  
 586 Michael R Lyu, and Maarten Sap. On the resilience of llm-based multi-agent collaboration with  
 587 faulty agents. *arXiv preprint arXiv:2408.00989*, 2024.

594 Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey.  
 595 *arXiv preprint arXiv:2212.10403*, 2022.  
 596

597 Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Os-  
 598 trow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint*  
 599 *arXiv:2410.21276*, 2024.

600 Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec  
 601 Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv*  
 602 *preprint arXiv:2412.16720*, 2024.  
 603

604 Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, DJ Strouse,  
 605 Joel Z Leibo, and Nando De Freitas. Social influence as intrinsic motivation for multi-agent deep  
 606 reinforcement learning. In *International conference on machine learning*, pp. 3040–3049. PMLR,  
 607 2019.

608 Edward Beeching Jia Li. Numinamath. [<https://github.com/project-numina/aimo-progress-prize>] ([https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina\\_dataset.pdf](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf)), 2024.  
 609  
 610  
 611

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

615 Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large  
 616 language models are zero-shot reasoners. *Advances in neural information processing systems*,  
 617 35:22199–22213, 2022.  
 618

619 Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. Llms get lost in multi-turn  
 620 conversation. *arXiv preprint arXiv:2505.06120*, 2025.  
 621

622 Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ra-  
 623 masesh, Ambrose Sloane, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative  
 624 reasoning problems with language models. *Advances in Neural Information Processing Systems*,  
 625 35:3843–3857, 2022.

626 Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbulin, and Bernard Ghanem. Camel: Com-  
 627 municative agents for “mind” exploration of large language model society. *Advances in Neural*  
 628 *Information Processing Systems*, 36:51991–52008, 2023.  
 629

630 Jiahui Li, Kun Kuang, Baoxiang Wang, Furui Liu, Long Chen, Fei Wu, and Jun Xiao. Shapley  
 631 counterfactual credits for multi-agent reinforcement learning. In *Proceedings of the 27th ACM*  
 632 *SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 934–942, 2021.

633 Meng Li, Hengyang Sun, Yanjun Huang, and Hong Chen. Shapley value: from cooperative game to  
 634 explainable artificial intelligence. *Autonomous Intelligent Systems*, 4(1):2, 2024.  
 635

636 Xiaomin Li, Mingye Gao, Zhiwei Zhang, Jingxuan Fan, and Weiyu Li. Ruleadapter: Dynamic rules  
 637 for training safety reward models in RLHF. In *Forty-second International Conference on Machine*  
 638 *Learning*, 2025a. URL <https://openreview.net/forum?id=tcMWVgYf8f>.

639  
 640 Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and  
 641 Zhicheng Dou. Webthinker: Empowering large reasoning models with deep research capability.  
 642 *arXiv preprint arXiv:2504.21776*, 2025b.  
 643

644 Yubo Li, Xiaobin Shen, Xinyu Yao, Xueying Ding, Yidi Miao, Ramayya Krishnan, and Rema Pad-  
 645 man. Beyond single-turn: A survey on multi-turn interactions with large language models. *arXiv*  
 646 *preprint arXiv:2504.04717*, 2025c.  
 647

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

648 Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan  
 649 Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth*  
 650 *International Conference on Learning Representations*, 2023.

651

652 Yu-An Lin, Chen-Tao Lee, Chih-Han Yang, Guan-Ting Liu, and Shao-Hua Sun. Hierarchical pro-  
 653 grammatic option framework. *Advances in Neural Information Processing Systems*, 37:126677–  
 654 126724, 2024.

655 Boyin Liu, Zhiqiang Pu, Yi Pan, Jianqiang Yi, Yanyan Liang, and Du Zhang. Lazy agents: A  
 656 new perspective on solving sparse reward problem in multi-agent reinforcement learning. In  
 657 *International Conference on Machine Learning*, pp. 21937–21950. PMLR, 2023.

658

659 Dingbang Liu, Shohei Kato, Wen Gu, Fenghui Ren, Jun Yan, and Guoxin Su. Integrating suboptimal  
 660 human knowledge with hierarchical reinforcement learning for large-scale multiagent systems.  
 661 *Advances in Neural Information Processing Systems*, 37:102744–102767, 2024a.

662

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

665

666 Xiaoyu Liu, Paiheng Xu, Junda Wu, Jiaxin Yuan, Yifan Yang, Yuhang Zhou, Fuxiao Liu, Tianrui  
 667 Guan, Haoliang Wang, Tong Yu, et al. Large language models and causal inference in collabora-  
 668 tion: A survey. *arXiv preprint arXiv:2403.09606*, 2024b.

669

670 Yuchi Liu, Jaskirat Singh, Gaowen Liu, Ali Payani, and Liang Zheng. Towards hierarchical multi-  
 671 agent workflows for zero-shot prompt optimization. *arXiv preprint arXiv:2405.20252*, 2024c.

672

673 Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee,  
 674 and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint*  
 675 *arXiv:2503.20783*, 2025b.

676

677 Rui Lu, Zhenyu Hou, Zihan Wang, Hanchen Zhang, Xiao Liu, Yujiang Li, Shi Feng, Jie Tang, and  
 678 Yuxiao Dong. Deepdive: Advancing deep search agents with knowledge graphs and multi-turn  
 679 rl. *arXiv preprint arXiv:2509.10446*, 2025.

680

681 Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Tang, Manan Roongta, Colin  
 682 Cai, Jeffrey Luo, Tianjun Zhang, Erran Li, Raluca Ada Popa, and Ion Stoica. Deep-  
 683 scaler: Surpassing o1-preview with a 1.5b model by scaling rl. <https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2>, 2025. Notion Blog.

684

685 R Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic  
 686 heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*, 2019.

687

688 Luke Merrick and Ankur Taly. The explanation game: Explaining machine learning models using  
 689 shapley values. In *International Cross-Domain Conference for Machine Learning and Knowledge  
 690 Extraction*, pp. 17–38. Springer, 2020.

691

692 Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke  
 693 Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time  
 694 scaling. *arXiv preprint arXiv:2501.19393*, 2025.

695

696 Sagnik Mukherjee, Abhinav Chinta, Takyong Kim, Tarun Anoop Sharma, and Dilek Hakkani-Tür.  
 697 Premise-augmented reasoning chains improve error identification in math reasoning with llms.  
 698 *arXiv preprint arXiv:2502.02362*, 2025.

699

700 Minh Hoang Nguyen, Van Dai Do, Dung Nguyen, Thin Nguyen, and Hung Le. Causalplan: Em-  
 701 powering efficient lilm multi-agent collaboration through causality-driven planning. *arXiv preprint*  
 702 *arXiv:2508.13721*, 2025.

703

704 Yansong Ning, Wei Li, Jun Fang, Naiqiang Tan, and Hao Liu. Not all thoughts are generated equal:  
 705 Efficient lilm reasoning via multi-turn reinforcement learning. *arXiv preprint arXiv:2505.11827*,  
 706 2025.

702 Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David  
 703 Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. Show your work:  
 704 Scratchpads for intermediate computation with language models. 2021.

705 Ellie Pavlick and Chris Callison-Burch. Simple ppdb: A paraphrase database for simplification. In  
 706 *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume*  
 707 *2: Short Papers)*, pp. 143–148, 2016.

708 Aske Plaat, Annie Wong, Suzan Verberne, Joost Broekens, Niki van Stein, and Thomas Bäck. Rea-  
 709 soning with large language models, a survey. *CoRR*, 2024.

710 Zhenting Qi, Mingyuan Ma, Jiahang Xu, Li Lyra Zhang, Fan Yang, and Mao Yang. Mutual reason-  
 711 ing makes smaller llms stronger problem-solvers. *arXiv preprint arXiv:2408.06195*, 2024.

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

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

718 Yucheng Shi, Wenhao Yu, Zaitang Li, Yonglin Wang, Hongming Zhang, Ninghao Liu, Haitao Mi,  
 719 and Dong Yu. Mobilegui-rl: Advancing mobile gui agent through reinforcement learning in online  
 720 environment. *arXiv preprint arXiv:2507.05720*, 2025.

721 Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max  
 722 Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition  
 723 networks for cooperative multi-agent learning based on team reward. In *Proceedings of the 17th*  
 724 *International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2085–2087, 2018.

725 Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL <https://qwenlm.github.io/blog/qwen2.5/>.

726 Jean-Francois Ton, Muhammad Faaiz Taufiq, and Yang Liu. Understanding chain-of-thought in llms  
 727 through information theory. *arXiv preprint arXiv:2411.11984*, 2024.

728 Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David  
 729 Silver, and Koray Kavukcuoglu. Feudal networks for hierarchical reinforcement learning. In  
 730 *International conference on machine learning*, pp. 3540–3549. PMLR, 2017.

731 Ziyu Wan, Yunxiang Li, Xiaoyu Wen, Yan Song, Hanjing Wang, Linyi Yang, Mark Schmidt, Jun  
 732 Wang, Weinan Zhang, Shuyue Hu, et al. Rema: Learning to meta-think for llms with multi-agent  
 733 reinforcement learning. *arXiv preprint arXiv:2503.09501*, 2025.

734 Peiyi Wang, Lei Li, Zhihong Shao, RX Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang  
 735 Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *arXiv*  
 736 *preprint arXiv:2312.08935*, 2023.

737 Tonghan Wang, Heng Dong, Victor Lesser, and Chongjie Zhang. Roma: Multi-agent reinforcement  
 738 learning with emergent roles. In *International Conference on Machine Learning*, pp. 9876–9886.  
 739 PMLR, 2020.

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

743 Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin,  
 744 Kefan Yu, Minh Nhat Nguyen, Licheng Liu, et al. Ragen: Understanding self-evolution in llm  
 745 agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*, 2025.

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

756 Zhepei Wei, Wenlin Yao, Yao Liu, Weizhi Zhang, Qin Lu, Liang Qiu, Changlong Yu, Puyang Xu,  
 757 Chao Zhang, Bing Yin, et al. Webagent-r1: Training web agents via end-to-end multi-turn rein-  
 758 force learning. *arXiv preprint arXiv:2505.16421*, 2025.

759

760 Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán,  
 761 Armand Joulin, and Edouard Grave. CCNet: Extracting high quality monolingual datasets from  
 762 web crawl data. *arXiv preprint arXiv:1911.00359*, 2019.

763

764 Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun  
 765 Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-  
 766 agent conversations. In *First Conference on Language Modeling*, 2024a.

767

768 Siwei Wu, Zhongyuan Peng, Xinrun Du, Tuney Zheng, Minghao Liu, Jialong Wu, Jiachen Ma, Yizhi  
 769 Li, Jian Yang, Wangchunshu Zhou, et al. A comparative study on reasoning patterns of openai’s  
 770 o1 model. *arXiv preprint arXiv:2410.13639*, 2024b.

771

772 Xiaohan Xu, Chongyang Tao, Tao Shen, Can Xu, Hongbo Xu, Guodong Long, and Jian-guang Lou.  
 773 Re-reading improves reasoning in language models. 2023.

774

775 Yixuan Even Xu, Yash Savani, Fei Fang, and Zico Kolter. Not all rollouts are useful: Down-sampling  
 776 rollouts in llm reinforcement learning. *arXiv preprint arXiv:2504.13818*, 2025.

777

778 Zhenghai Xue, Longtao Zheng, Qian Liu, Yingru Li, Xiaosen Zheng, Zejun Ma, and Bo An. Sim-  
 779 plefir: End-to-end reinforcement learning for multi-turn tool-integrated reasoning. *arXiv preprint  
 780 arXiv:2509.02479*, 2025.

781

782 Kazeto Yamamoto, Takashi Onishi, and Yoshimasa Tsuruoka. Hierarchical reinforcement learning  
 783 with abductive planning. *arXiv preprint arXiv:1806.10792*, 2018.

784

785 Tom Yan and Zachary Lipton. A theoretical case-study of scalable oversight in hierarchical rein-  
 786 force learning. *Advances in Neural Information Processing Systems*, 37:27295–27339, 2024.

787

788 An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jian-  
 789 hong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical  
 790 expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024a.

791

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

795

796 Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. Regularizing hidden states  
 797 enables learning generalizable reward model for llms. *Advances in Neural Information Processing  
 798 Systems*, 37:62279–62309, 2024b.

799

800 Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik  
 801 Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Ad-  
 802 vances in neural information processing systems*, 36:11809–11822, 2023.

803

804 Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. Limo: Less is more  
 805 for reasoning. *arXiv preprint arXiv:2502.03387*, 2025.

806

807 Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-  
 808 of-thought reasoning in llms. *arXiv preprint arXiv:2502.03373*, 2025.

809

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

813

814 Yangyang Yu, Zhiyuan Yao, Haohang Li, Zhiyang Deng, Yuechen Jiang, Yupeng Cao, Zhi Chen,  
 815 Jordan Suchow, Zhenyu Cui, Rong Liu, et al. Fincon: A synthesized llm multi-agent system with  
 816 conceptual verbal reinforcement for enhanced financial decision making. *Advances in Neural  
 817 Information Processing Systems*, 37:137010–137045, 2024.

810 Yurun Yuan and Tengyang Xie. Reinforce llm reasoning through multi-agent reflection. In *Forty-*  
 811 *second International Conference on Machine Learning*.  
 812

813 Siliang Zeng, Quan Wei, William Brown, Oana Frunza, Yuriy Nevmyvaka, and Mingyi Hong. Re-  
 814 *inforcing multi-turn reasoning in llm agents via turn-level credit assignment. arXiv preprint*  
 815 *arXiv:2505.11821*, 2025.

816 Chen Zhang, Xinyi Dai, Yaxiong Wu, Qu Yang, Yasheng Wang, Ruiming Tang, and Yong  
 817 Liu. A survey on multi-turn interaction capabilities of large language models. *arXiv preprint*  
 818 *arXiv:2501.09959*, 2025a.

819 Guibin Zhang, Muxin Fu, Guancheng Wan, Miao Yu, Kun Wang, and Shuicheng Yan. G-memory:  
 820 Tracing hierarchical memory for multi-agent systems. *arXiv preprint arXiv:2506.07398*, 2025b.

821 Kaiyi Zhang, Ang Lv, Jinpeng Li, Yongbo Wang, Feng Wang, Haoyuan Hu, and Rui Yan.  
 822 *Stephint: Multi-level stepwise hints enhance reinforcement learning to reason. arXiv preprint*  
 823 *arXiv:2507.02841*, 2025c.

824 Xiaotian Zhang, Chunyang Li, Yi Zong, Zhengyu Ying, Liang He, and Xipeng Qiu. Evaluating the  
 825 performance of large language models on gaokao benchmark. *arXiv preprint arXiv:2305.12474*,  
 826 2023a.

827 Yifan Zhang, Jingqin Yang, Yang Yuan, and Andrew Chi-Chih Yao. Cumulative reasoning with  
 828 large language models. *arXiv preprint arXiv:2308.04371*, 2023b.

829 Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Arik. Chain of agents:  
 830 Large language models collaborating on long-context tasks. *Advances in Neural Information*  
 831 *Processing Systems*, 37:132208–132237, 2024.

832 Zhiwei Zhang, Hui Liu, Xiaomin Li, Zhenwei Dai, Jingying Zeng, Fali Wang, Minhua Lin, Ramraj  
 833 Chandraevan, Zhen Li, Chen Luo, et al. Bradley-terry and multi-objective reward modeling are  
 834 complementary. *arXiv preprint arXiv:2507.07375*, 2025d.

835 Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in  
 836 large language models. *arXiv preprint arXiv:2210.03493*, 2022.

837 Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuur-  
 838 mans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex  
 839 reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

840 Yifei Zhou, Song Jiang, Yuandong Tian, Jason Weston, Sergey Levine, Sainbayar Sukhbaatar, and  
 841 Xian Li. Sweet-rl: Training multi-turn llm agents on collaborative reasoning tasks. *arXiv preprint*  
 842 *arXiv:2503.15478*, 2025.

843 Dawei Zhu, Xiyu Wei, Guangxiang Zhao, Wenhao Wu, Haosheng Zou, Junfeng Ran, Xun Wang,  
 844 Lin Sun, Xiangzheng Zhang, and Sujian Li. Chain-of-thought matters: improving long-context  
 845 language models with reasoning path supervision. *arXiv preprint arXiv:2502.20790*, 2025.

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864 **A DETAILED RELATED WORKS**  
865866 **A.1 MULTI-AGENT RL**  
867

868 Multi-agent RL addresses how multiple agents coordinate in a shared environment to maximize  
869 collective performance, with a central challenge being credit assignment: determining each agent’s  
870 contribution to the overall reward. Classical solutions include value decomposition (VDN; Sunehag  
871 et al., 2018), counterfactual baselines (COMA; Foerster et al., 2018), regression-based reward func-  
872 tions with default-action substitution (Dr.Reinforce; Castellini et al., 2022), social influence estima-  
873 tion via KL divergence (Jaques et al., 2019), role-based coordination through role networks (Wang  
874 et al., 2020), and model-based transition prediction to measure influence (Liu et al., 2023). With the  
875 advent of large language model (LLM) agents, MARL techniques have been extended to multi-turn  
876 reasoning and cooperative dialogue: turn-level credit assignment to reduce misattributed rewards  
877 (Zeng et al., 2025), SWEET-RL for critic-driven step-wise rewards (Zhou et al., 2025), MARFT for  
878 alleviating agent inactivity and communication inefficiency (Liao et al., 2025), RAGEN for address-  
879 ing the “Echo Trap” caused by coarse reward signals (Wang et al., 2025), and MAGRPO for framing  
880 LLM collaboration as cooperative MARL with tailored reward design (Liu et al., 2025a). A persis-  
881 tent challenge across both classical and LLM-based MARL is the emergence of lazy agents that  
882 contribute little while relying on others. Recent work therefore explores causal influence estimation  
883 (Bogdan et al., 2025; Nguyen et al., 2025; Liu et al., 2024b), introducing methods such as black-  
884 box resampling and attention suppression to quantify how an agent’s utterance shapes subsequent  
885 decisions, thereby enabling finer-grained credit assignment and mitigating agent inactivity.

886 **A.2 HIERARCHICAL RL**  
887

888 Hierarchical multi-agent systems coordinate cooperation by assigning high-level controllers to de-  
889 compose tasks for lower-level workers, a design shown to improve scalability, robustness, and long-  
890 horizon reasoning. (Yu et al., 2024) introduce a manager–analyst paradigm for structured decomposi-  
891 tion, while Chain-of-Agents (CoA) leverages chained communication to approximate hierarchical  
892 coordination with greater flexibility (Zhang et al., 2024). Empirical studies further demonstrate that  
893 boss–worker hierarchies outperform flat or linear structures under failure conditions (Huang et al.,  
894 2024). Memory-oriented approaches, exemplified by Tracing Hierarchical Memory for Multi-Agent  
895 Systems, highlight how layered storage and retrieval mechanisms enable adaptive long-horizon col-  
896 laboration (Zhang et al., 2025b). On the planning side, Hierarchical Planning for Complex Tasks  
897 with Knowledge-Graph RAG and Symbolic Verification integrates structured decomposition with  
898 formal verification (Cornelio et al., 2025). More explicit architectures, such as HMAW and ReAc-  
899 Tree, explore CEO–Manager–Worker hierarchies or adaptive tree structures for general task allo-  
900 cation (Liu et al., 2024c; Choi et al., 2025). Hierarchical reinforcement learning (HRL) organizes  
901 control as high-level planning over subgoals with low-level execution, enabling agents to operate  
902 over long horizons via temporal abstraction and reusable skills. Early neural HRL emphasized  
903 top-down goal setting—e.g., high-level goal embeddings steering a low-level policy (Vezhnevets  
904 et al., 2017) and quickly expanded toward unsupervised skill discovery to populate the low-level  
905 option set (Bagaria & Konidaris, 2019). Beyond purely reactive control, symbolic reasoning has  
906 been fused with HRL to support plan construction and revision (Yamamoto et al., 2018). Recent  
907 advances improve the reliability and interpretability of options themselves: programmatic, human-  
908 readable sub-policies selected by the high-level planner enhance generalization to longer tasks (Lin  
909 et al., 2024), while theoretically grounded supervision clarifies how limited human feedback can be  
910 efficiently allocated across hierarchical levels in goal-conditioned settings (Yan & Lipton, 2024).  
911 Complementary lines strengthen the low level: disentangling unsupervised skill discovery yields  
912 cleaner building blocks (Hu et al., 2024), and integrating imperfect expert priors improves multi-  
913 agent coordination under hierarchical control (Liu et al., 2024a).

914 **A.3 LLM REASONING**  
915

916 Large Language Models (LLMs) have demonstrated strong performance across a wide range of nat-  
917 ural language tasks (Brown et al., 2020; Chowdhery et al., 2023; Du et al., 2022; Dubey et al., 2024;  
918 Wenzek et al., 2019). Early research found that prompting models to reason step by step, an approach  
919 known as chain-of-thought (CoT) prompting, can significantly improve performance on arithmetic,  
920 commonsense, and symbolic reasoning tasks by eliciting intermediate reasoning steps (Wei et al.,

2022; Kojima et al., 2022; Nye et al., 2021). Building on this, researchers have explored non-linear reasoning structures. For example, Tree-of-Thoughts (ToT) organizes candidate reasoning paths into a search tree with lookahead capabilities, while Graph-of-Thoughts (GoT) generalizes reasoning to arbitrary graphs of “thought” nodes and edges, expanding the space for structured deliberation (Yao et al., 2023; Besta et al., 2024). These advances have inspired the development of Large Reasoning Models (LRMs), models explicitly trained for multi-step reasoning (Guo et al., 2025a; Achiam et al., 2023; Grattafiori et al., 2024; Xu et al., 2023; Zhou et al., 2022; Wu et al., 2024b; Qi et al., 2024; Chae et al., 2024). Typically, LRMs undergo supervised fine-tuning followed by a reinforcement learning stage, and have achieved state-of-the-art results on challenging tasks such as math, coding, and task planning (Jaech et al., 2024; Guo et al., 2025a; Comanici et al., 2025; Yang et al., 2024a; 2025; Lightman et al., 2023; Wang et al., 2023). The success of strong single-model reasoners has also spurred multi-agent approaches, where complex tasks are decomposed and coordinated among specialized LLM agents via role assignment, orchestration, and debate, mirroring human teamwork (Li et al., 2023; Wu et al., 2024a; Chen et al., 2023; Du et al., 2023; Yuan & Xie).

## B IMPLEMENTATION DETAILS

### B.1 TRAINING OBJECTIVE OF DR. MAMR

The training objective for Dr. MAMR builds on Eq. 2 with two key modifications: (i) we remove the  $\frac{1}{T_i}$  normalization over the number of turns, and (ii) we replace the step-level advantage with the weighted formulation in Eq. 8. The resulting objective is defined as follows:

$$\mathcal{J}_{\text{Dr. MAMR}}(\theta) = \mathbb{E}_{(\mathbf{x}, \mathbf{y}^*) \sim \mathcal{D}, \{(\mathbf{m}_i, \mathbf{y}_i)\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot | \mathbf{x})} \left[ \frac{1}{G} \sum_{i=1}^G \sum_{t=1}^{T_i} \frac{1}{|\mathbf{y}_{i,t}|} \sum_{j=1}^{|\mathbf{y}_{i,t}|} \left( \min(r_{i,t}(\theta) A_{i,t}^{\text{step}}, \text{clip}(r_{i,t}(\theta), 1 - \epsilon, 1 + \epsilon) A_{i,t}^{\text{step}}) - \beta D_{\text{KL}}(\pi_{\theta} \| \pi_{\text{ref}}) \right) \right], \quad (9)$$

where the step-level advantage is given by Eq. 8:

$$A_{i,t}^{\text{step}} = \tilde{A}_{i,t} + \alpha \tilde{C}_{i,t} + \beta \tilde{R}_{i,t}.$$

### B.2 TRAINING ON DEEPSALER

We conduct all experiments using the Verl RL framework (Sheng et al., 2024). Given the substantial computational cost, we fix the hyperparameters at  $\alpha = \beta = 0.1$  across all experiments, as this setting provides stable performance. We use bfloat16 precision for training, with a batch size of 128 and 128 sampled rollouts per training step.

### B.3 SHAPLEY-INSPIRED CAUSAL INFLUENCE

To group semantically similar steps for causal influence estimation, we use Qwen2.5-0.5B<sup>1</sup> as the embedding model. Each step  $s_{i,t}$  in a trajectory is encoded into a dense vector representation, and semantic similarity between steps is measured using cosine similarity. For each anchor step, we form a group  $G_S(s_{i,t})$  by including all steps whose embeddings have cosine similarity of at least 0.9 with the anchor. This threshold ensures that only highly similar steps—those expressing essentially the same idea, regardless of minor wording differences—are grouped together.

### B.4 COLD START FOR META-THINK AND REASONING

For the experiments in Table 1, we adopt RL-from-base since the Qwen2.5B-Instruct family (Team, 2024) demonstrates strong instruction-following capability. Moreover, as shown in ReMA (Wan et al., 2025), the performance gap between RL-from-base and RL-from-SFT is marginal.

<sup>1</sup><https://huggingface.co/Qwen/Qwen2.5-0.5B>

972 B.5 COLD START FOR RESTART BEHAVIOR  
973974  
975 However, it is difficult for the base model to exhibit restart behavior directly. To address this, we  
976 collect expert data and apply supervised fine-tuning (SFT) to the base model. Specifically, building  
977 on the meta-thinking and reasoning datasets collected by ReMA (Wan et al., 2025; Ye et al.,  
978 2025), we use GPT-4o (Hurst et al., 2024) to adversarially insert a few steps of noisy reasoning,  
979 followed by a restart, in order to simulate scenarios where the reasoning agent becomes lost in the  
980 conversation and the restart behavior enables recovery. For supervised fine-tuning, we use the Lla-  
981 maFactory codebase, training the model for 3 epochs with a learning rate of 1e-5, a cosine learning  
982 rate scheduler, and a batch size of 8. We employ DeepSpeed ZeRO-2 for distributed training.  
983  
984985 C MORE EXPERIMENTAL DETAILS  
986987  
988 C.1 PRELIMINARY EXPERIMENTS ON CAUSAL INFLUENCE  
989990 C.1.1 EXPERIMENTAL SETUP  
991992  
993 We follow the setting in (Wan et al., 2025) and train models on 7.5k training samples in MATH  
994 (Hendrycks et al., 2021) and test on datasets: GSM8K (Cobbe et al., 2021), AIME24<sup>2</sup>, AMC23<sup>3</sup>,  
995 GaoKao2023En (Zhang et al., 2023a), Minerva Math (Lewkowycz et al., 2022), and Olympiad  
996 Bench (He et al., 2024). We generate reasoning processes on the evaluation benchmark and subse-  
997 quently measure the causal influence of each response following the descriptin in Sec. 4.  
998  
9991000 C.1.2 PROMPTS  
10011002  
1003 System Prompts for Meta-Think and Reasoning Agents of ReMA  
1004

## 1005 META-THINK AGENT SYSTEM PROMPT

1006 You are a meta-think agent that represents human high-level thinking processes. When solving a  
1007 question, you will have a discussion with a human. Each time, think about what to do next. For  
1008 example:1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025

- Exploring multiple angles and approaches
- Breaking down the solution into clear steps
- Continuously reflecting on intermediate results honestly and adapting your strategy as you progress
- Backtracking when necessary
- Requesting exploration of multiple solutions individually
- Finally, confirm the answer with the tag [FINISH]

## REASONING AGENT SYSTEM PROMPT

Please reason step by step following the given instruction. When asked to finalize your answer, put  
your answer within \boxed{ }.<sup>2</sup><https://huggingface.co/datasets/AI-MO/aimo-validation-aime><sup>3</sup><https://huggingface.co/datasets/AI-MO/aimo-validation-amc>

1026  
1027**Refined System Prompts for Meta-Think and Reasoning Agents**1028  
1029  
1030  
1031**META-THINK AGENT SYSTEM PROMPT**

You are a meta-think agent that represents human high-level think process. When solving a question, you will have a discussion with the human, and each time you will think about what to do next. For example:

1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039

- Exploring multiple angles and approaches
- Breaking down the solution into clear steps
- Continuously reflecting on intermediate results honestly and adapting your strategy as you progress
- Backtracking when necessary
- Requesting exploration of multiple solutions individually
- Finally, confirm the answer with the tag [FINISH].

1040  
1041

**Please do not focus on completing the task by calculating the final answer;** that step will be handled by a separate reasoning agent.

1042  
1043  
1044**REASONING AGENT SYSTEM PROMPT**

You are a reasoning agent that follows structured problem-solving instructions step by step. Your goals are:

1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054

- Follow the given instruction precisely.
- Reason step by step toward a solution.
- Avoid producing empty or blank outputs at any step.
- If uncertain, provide your best reasoning and partial answer rather than outputting nothing.
- Always provide a meaningful and non-empty response, even during intermediate steps.
- When you receive the signal [FINISH], finalize your answer and place it within \boxed{ }.
- If unable to finalize, explain why and still output your best available answer within \boxed{ }.

1055  
1056

**Remember: You must never produce trivial outputs.**

1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067**C.2 PRELIMINARY EXPERIMENTS ON HOW DELIBERATION HELPS MULTI-TURN REASONING**1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076

In this section, we provide additional details about the preliminary experiments referenced in Sec. 5.3, which demonstrate that deliberation can enhance performance in multi-turn reasoning tasks.

1077  
1078  
1079

We adopt the experimental setup outlined in Sec. C.1.1 to train both the meta-thinking agent and the reasoning agent using the original ReMA framework (Wan et al., 2025). To evaluate the impact of deliberation, we introduce a variant of ReMA at inference time by modifying the system prompt of the reasoning agent as follows:

1080  
 1081     **System Prompt for Reasoning Agent of ReMA+**  
 1082     Please reason step by step. When asked to finalize your answer, place your result within  
 1083     \boxed{ } .  
 1084  
 1085     **Deliberation Instruction**  
 1086     If your previous reasoning is based on assumptions that are contradicted or made incomplete  
 1087     by new instructions, you must explicitly output:  
 1088     <restart>  
 1089     Reason: <One-sentence justification>  
 1090  
 1091     **Rethinking Rules for New Information**  
 1092     1. Upon receiving new information, re-evaluate all prior assumptions, steps, and con-  
 1093        clusions.  
 1094     2. If the new information invalidates or alters your earlier reasoning, clearly acknowl-  
 1095        edge this and issue a <restart>.  
 1096     3. Do not simply add new information to prior reasoning — verify that the existing  
 1097        logic still holds.  
 1098     4. Ask clarifying questions only if essential information is missing and further rea-  
 1099        soning is blocked.  
 1100     5. If you revise your reasoning due to new information, explicitly state:  
 1101        "Given this new information, I need to revise my  
 1102        approach."  
 1103  
 1104  
 1105  
 1106

## D CASE STUDY ON LAZY BEHAVIOR

1107  
 1108  
 1109  
 1110     **Case Study**  
 1111  
 1112     [Question]  
 1113     Determine the number of ways to arrange the letters of the word 'ELLIPSE'. **[Groundtruth: 1260]**  
 1114

1115     **ReMA**  
 1116  
 1117     **Meta-thinking:**  
 1118     Let's start by considering the total number of letters and any repetitions. The word ELLIPSE  
 1119     has 7 letters, with L repeated twice. To find the number of distinct arrangements, we use the  
 formula for permutations of a multiset:

$$\frac{n!}{n_1! \cdot n_2! \cdot \dots \cdot n_k!}$$

1120     For ELLIPSE: -  $n = 7$  - L appears 2 times - E, I, P, S each appear once  
 1121  
 1122

$$\frac{7!}{2!} = \frac{5040}{2} = 2520$$

1123  
 1124     **Reasoning:**  
 1125

□

1126  
 1127     **Meta-thinking:**  
 1128

1129     [FINISH]  
 1130

1131     **Reasoning:**  
 1132

2520  
 1133

1134  
1135

Dr. MAMR

1136

**Meta-thinking:**

1137

To determine the number of ways to arrange the letters of the word "ELLIPSE", we need to consider the total number of letters and any repetitions of letters.

1138

1. First, let's count the total number of letters and identify any repeated letters.

1139

2. Then, we'll use the formula for permutations of a multiset.

1140

Let's start by counting the letters. How many letters are there in the word "ELLIPSE"? And which letters are repeated?

1141

**Reasoning:**

1142

The word "ELLIPSE" consists of 7 letters: E, L, L, I, P, S, E. We can see that the letter 'E' is repeated twice, and the letter 'L' is also repeated twice. The other letters, I, P, and S, each appear once. To determine the number of distinct arrangements of these letters, we use the formula for permutations of a multiset:

1143

$$\frac{n!}{n_1! \cdot n_2! \cdot \dots \cdot n_k!}$$

1144

For the word "ELLIPSE": -  $n = 7$  - 'E' occurs 2 times, 'L' occurs 2 times, others occur once

1145

$$\frac{7!}{2! \cdot 2! \cdot 1! \cdot 1! \cdot 1!} = \frac{5040}{4} = 1260$$

1260

1146

**Meta-thinking:**

1147

Great job walking through the calculation! Let's summarize:

1148

- 7 letters in total
- 'E' and 'L' both appear twice
- Used multiset formula correctly
- Answer: 1260

1149

1150

1151

1152

1153

1154

1155

1156

## E PROOFS

1157

Following Eq.2, ReMA's training objective (per batch) is:

1158

1159

1160

1161

1162

1163

1164

1165

1166

1167

$$\mathcal{J}(\theta) = \mathbb{E} \left[ \frac{1}{G} \sum_{i=1}^G \frac{1}{T_i} \sum_{t=1}^{T_i} \frac{1}{|y_{i,t}|} \sum_{j=1}^{|y_{i,t}|} (\min\{u_{i,t,j}, v_{i,t,j}\} - \beta D_{\text{KL}}(\pi_\theta \parallel \pi_{\text{ref}})) \right], \quad (10)$$

where

$$u_{i,t,j} = r_{i,t}(\theta) \hat{A}_{i,t,j}, \quad (11)$$

$$v_{i,t,j} = \text{clip}(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon) \hat{A}_{i,t,j}. \quad (12)$$

Turn-level importance ratio:

1171

1172

1173

1174

1175

1176

1177

1178

1179

1180

$$r_{i,t}(\theta) = \frac{1}{|y_{i,t}|} \sum_{j'=1}^{|y_{i,t}|} \underbrace{\frac{\pi_\theta(y_{i,t,j'} \mid c_{i,t,j'})}{\pi_{\theta_{\text{old}}}(y_{i,t,j'} \mid c_{i,t,j'})}}_{=: r_{i,t,j'}(\theta)}, \quad (13)$$

1181

where the context is:

1182

1183

1184

$$c_{i,t,j'} := (x_i, \{m_{i,\cdot}, y_{i,\cdot}\}_{<t}, m_{i,t}, y_{i,t,<j'}).$$

We derive the turn- $t$  contribution for a fixed trajectory  $i$ , i.e.,

1185

1186

1187

$$L_{i,t}(\theta) := \frac{1}{T_i} \cdot \frac{1}{|y_{i,t}|} \sum_{j=1}^{|y_{i,t}|} \min\{u_{i,t,j}, v_{i,t,j}\} - \frac{\beta}{T_i} \cdot \frac{1}{|y_{i,t}|} \sum_{j=1}^{|y_{i,t}|} D_{\text{KL}}(\pi_\theta(\cdot \mid c_{i,t,j}) \parallel \pi_{\text{ref}}(\cdot \mid c_{i,t,j})). \quad (14)$$

1188 Without considering the KL divergence and clipping,  
 1189

$$1190 \quad L_{i,t}(\theta) = \frac{1}{T_i} \cdot \frac{1}{|y_{i,t}|} \sum_j r_{i,t}(\theta) \hat{A}_{i,t,j} = \frac{1}{T_i} \cdot \bar{A}_{i,t} \cdot r_{i,t}(\theta), \quad (15)$$

1192 where  
 1193

$$1194 \quad \bar{A}_{i,t} := \frac{1}{|y_{i,t}|} \sum_{j=1}^{|y_{i,t}|} \hat{A}_{i,t,j}. \quad (16)$$

1197 Compute the gradient:  
 1198

$$\begin{aligned} 1199 \quad \nabla_{\theta} r_{i,t}(\theta) &= \nabla_{\theta} \left( \frac{1}{|y_{i,t}|} \sum_{j'=1}^{|y_{i,t}|} r_{i,t,j'}(\theta) \right) \\ 1200 &= \frac{1}{|y_{i,t}|} \sum_{j'=1}^{|y_{i,t}|} \nabla_{\theta} \left( \frac{\pi_{\theta}(y_{i,t,j'} \mid c_{i,t,j'})}{\pi_{\theta_{\text{old}}}(y_{i,t,j'} \mid c_{i,t,j'})} \right) \\ 1201 &= \frac{1}{|y_{i,t}|} \sum_{j'=1}^{|y_{i,t}|} \frac{1}{\pi_{\theta_{\text{old}}}(y_{i,t,j'} \mid c_{i,t,j'})} \nabla_{\theta} \pi_{\theta}(y_{i,t,j'} \mid c_{i,t,j'}) \\ 1202 &= \frac{1}{|y_{i,t}|} \sum_{j'=1}^{|y_{i,t}|} \frac{\pi_{\theta}(y_{i,t,j'} \mid c_{i,t,j'})}{\pi_{\theta_{\text{old}}}(y_{i,t,j'} \mid c_{i,t,j'})} \nabla_{\theta} \log \pi_{\theta}(y_{i,t,j'} \mid c_{i,t,j'}) \\ 1203 &= \frac{1}{|y_{i,t}|} \sum_{j'=1}^{|y_{i,t}|} r_{i,t,j'}(\theta) \nabla_{\theta} \log \pi_{\theta}(y_{i,t,j'} \mid c_{i,t,j'}). \end{aligned} \quad (17)$$

1217 Therefore, the exact per-turn gradient (no clipping, no KL) is:  
 1218

$$1219 \quad \nabla_{\theta} L_{i,t}(\theta) = \frac{1}{T_i} \cdot \bar{A}_{i,t} \cdot \left( \frac{1}{|y_{i,t}|} \sum_{j'=1}^{|y_{i,t}|} r_{i,t,j'}(\theta) \nabla_{\theta} \log \pi_{\theta}(y_{i,t,j'} \mid c_{i,t,j'}) \right) \quad (18)$$

1223 We define the aggregated turn- $t$  stochastic contribution as  
 1224

$$1225 \quad Z_t(\tau) \triangleq \frac{1}{|y_t|} \sum_{j=1}^{|y_t|} r_t(\theta) \hat{A}_{t,j} \nabla_{\theta} \log \pi_{\theta}(y_{t,j} \mid x, m_{\leq t}, y_{<t}, y_{t,<j}). \quad (19)$$

1228 Then the ReMA gradient contribution at turn  $t$  is  
 1229

$$1230 \quad g_t(\tau) = \frac{1}{T(\tau)} Z_t(\tau). \quad (20)$$

1232 Let  
 1233

$$1234 \quad \kappa \triangleq \frac{\|Z_t(\tau^L)\|}{\|Z_t(\tau^S)\|}.$$

1235 Then the relative gradient magnitude satisfies  
 1236

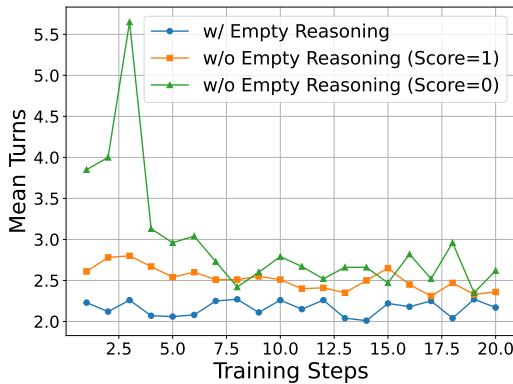
$$1237 \quad \frac{\|g_t(\tau^S)\|}{\|g_t(\tau^L)\|} = \frac{T_L}{T_S} \cdot \frac{1}{\kappa}.$$

1239 In particular, if  $\kappa < \frac{T_L}{T_S}$ , then  
 1240

$$1241 \quad \frac{\|g_t(\tau^S)\|}{\|g_t(\tau^L)\|} > 1.$$

## 1242 F PRELIMINARY EXPERIMENTS FOR THEORETICAL ANALYSIS

1244 We present preliminary results on the mean number of turns in the reasoning process for cases with  
 1245 empty outputs and trivially copy the other’s response (i.e., reasoning process exhibiting lazy-agent  
 1246 behavior) and those without empty outputs (i.e., reasoning process without lazy agents). We report  
 1247 results from the first 20 training steps, as this initial stage is critical in shaping the agent’s behavior.  
 1248 As shown in Fig. 5, the number of turns for reasoning process exhibiting lazy-agent behavior is  
 1249 consistently smaller than that of non-lazy agents.



1264 Figure 5: Mean number of turns comparing reasoning processes w/ and w/o lazy-agent behavior.

## 1267 G PROCESS REWARD FAILS TO MITIGATE THE LAZY AGENT ISSUE

1269 In this section, we present experimental results showing that using a process reward model to as-  
 1270 sign credit at each turn fails to mitigate the lazy agent issue in ReMA. We follow the experimental  
 1271 setup described in Section C.1, and adopt Qwen/Qwen2.5-Math-PRM-7B<sup>4</sup> as the process re-  
 1272 ward model (PRM). We revise Eq. 3 as follows:

$$1273 \quad r_{i,t}(\theta) = r_{i,t}(\theta) + a_{i,t}, \quad (21)$$

1275 where  $a_{i,t}$  denotes the process reward for the  $t$ -th turn in the  $i$ -th rollout, provided by the PRM.

1277 The training curve of this approach is shown in Fig. 6. As observed, the model collapses rapidly  
 1278 after only 30 training steps. We attribute this to reward hacking, a well-known failure mode in RLHF  
 1279 settings, as documented in prior work (Zhang et al., 2025d; Gao et al., 2023; Yang et al., 2024b; Li  
 1280 et al., 2025a).

## 1281 H ADDITIONAL EXPERIMENT RESULTS

### 1284 H.1 RESULTS ON ADDITIONAL MODEL FAMILY

1285 In this section, following the experimental setup described in Sec. 6.1, we conduct experiments  
 1286 using two base models: Meta-Llama-3-8B-Instruct<sup>5</sup> and Llama-3.1-8B-Instruct<sup>6</sup>. We compare our  
 1287 proposed Dr. MAMR framework against ReMA and the single-agent GRPO baseline. The results  
 1288 are summarized in Table 3.

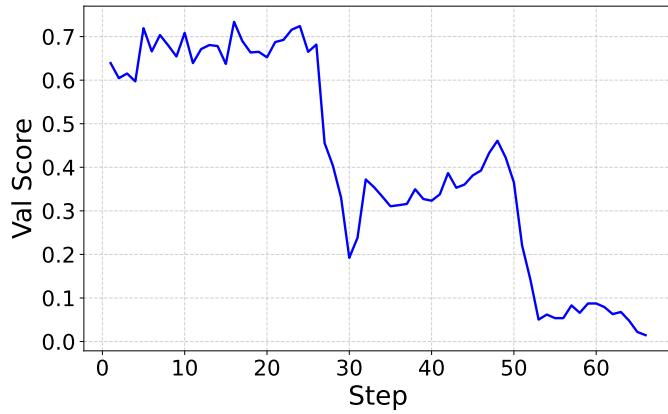
1289 Across both Llama base models—which overall exhibit lower raw performance than the Qwen2.5  
 1290 series—we consistently observe that Dr. MAMR yields substantial improvements over both ReMA  
 1291 and GRPO. This demonstrates that the benefits of our multi-agent meta-reasoning framework persist  
 1292 even with weaker base models, highlighting its robustness and general applicability.

1294 <sup>4</sup>Qwen/Qwen2.5-Math-PRM-7B

1295 <sup>5</sup>meta-llama/Meta-Llama-3-8B-Instruct

1296 <sup>6</sup>meta-llama/Meta-Llama-3.1-8B-Instruct

1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309



1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349

Figure 6: Training curve of ReMA with process reward assigned for each turn.

Table 3: Performance on math reasoning benchmarks (Llama family).

Model	Benchmark	GRPO	ReMA	Dr. MAMR
Llama3-8B-Instruct	MATH500	40.4	33.80	43.20
	GSM8K	82.79	79.38	83.47
	AIME24	0.00	0.00	3.33
	AMC23	25.00	22.50	32.50
	Gaokao2023en	29.87	28.57	32.49
	Minerva Math	17.28	13.97	22.06
	Olympiad Bench	10.52	8.89	16.30
<b>Average</b>		29.4	26.73	33.34
Llama3.1-8B-Instruct	MATH500	55.20	53.20	58.60
	GSM8K	88.32	87.26	90.22
	AIME24	10.00	6.67	13.33
	AMC23	27.50	20.00	30.50
	Gaokao2023en	44.94	37.14	48.95
	Minerva Math	32.35	28.31	36.77
	Olympiad Bench	16.44	19.56	22.67
<b>Average</b>		39.25	36.97	43.01

## H.2 HYPERPARAMETER ANALYSIS

In this section, we analyze the sensitivity of Dr. MAMR to its two hyperparameters: the weight for the causal-influence reward ( $\alpha$ ) and the weight for the restart-behavior reward ( $\beta$ ). Following the experimental setup in Sec. 6.1, we conduct two sets of experiments: (1) Fix  $\alpha = 0.1$  and vary  $\beta \in \{0.1, 0.3, 0.5\}$ . (2) Fix  $\beta = 0.1$  and vary  $\alpha \in \{0.1, 0.3, 0.5\}$ . The results are shown in Tables 4 and 5.

**(1) Effect of varying  $\beta$  (restart-behavior reward).** When  $\beta$  increases from 0.1 to 0.3, Dr. MAMR maintains very stable average performance (58.43  $\rightarrow$  58.16). Even when  $\beta$  is further increased to 0.5, where performance drops to 53.15, Dr. MAMR still outperforms ReMA (51.97) by a clear margin.

*Interpretation.*  $\beta$  affects only the reasoning agent’s restart behavior. A moderate value of  $\beta$  (0.1 to 0.3) guides this behavior without disrupting the main optimization process. However, an excessively large  $\beta$  risks over-encouraging or over-penalizing restarts, which discourages beneficial exploratory steps and ultimately leads to degraded performance.

**(2) Effect of varying  $\alpha$  (causal-influence reward).** Increasing  $\alpha$  from 0.1 to 0.3 yields only a mild change in performance (58.43 to 56.30). Even at  $\alpha = 0.5$ , where the score is 52.77, Dr. MAMR

1350 maintains performance above the ReMA baseline (51.97), indicating that the method remains robust  
 1351 across a broad range of  $\alpha$  values.  
 1352

1353 *Interpretation.*  $\alpha$  encourages each agent to take actions that can influence the other’s internal rea-  
 1354 soning trajectory. When  $\alpha$  becomes large, this incentive may occasionally place more emphasis on  
 1355 cross-agent influence than on improving final reasoning quality. As a result, the overall performance  
 1356 tends to vary more noticeably with respect to  $\alpha$  than to  $\beta$ .  
 1357

1358 **(3) Overall robustness.** Across all combinations of  $\alpha$  and  $\beta$ , mild hyperparameter choices (e.g.,  
 1359  $\alpha = 0.1$  with  $\beta = 0.3$ , or  $\beta = 0.1$  with  $\alpha = 0.3$ ) achieve performance comparable to the best  
 1360 setting of  $\alpha = 0.1, \beta = 0.1$ . Even when the hyperparameters are set to larger values such as 0.5,  
 1361 Dr. MAMR still surpasses ReMA, as Dr. MAMR addresses the fundamental lazy-agent issue. Over-  
 1362 all, these results show that Dr. MAMR provides robust improvements without requiring extensive  
 1363 hyperparameter tuning.  
 1364

Table 4: Hyperparameter analysis with  $\alpha = 0.1$ .

Model	Benchmark	ReMA	Dr. MAMR ( $\beta = 0.1$ )	( $\beta = 0.3$ )	( $\beta = 0.5$ )
Qwen2.5 -7B -Instruct	MATH500	74.40	78.60	78.80	77.20
	GSM8K	90.60	92.12	91.59	86.05
	AIME24	13.33	20.00	20.00	16.67
	AMC23	50.00	62.50	60.00	52.50
	Gaokao2023en	57.92	65.20	66.50	63.12
	Minerva Math	34.93	38.24	38.60	34.56
	Olympiad Bench	42.58	52.34	51.60	45.04
	Average	51.97	58.43	58.16	53.15

Table 5: Hyperparameter analysis with  $\beta = 0.1$ .

Model	Benchmark	ReMA	Dr. MAMR ( $\alpha = 0.1$ )	( $\alpha = 0.3$ )	( $\alpha = 0.5$ )
Qwen2.5 -7B -Instruct	MATH500	74.40	78.60	77.80	76.80
	GSM8K	90.60	92.12	91.18	88.10
	AIME24	13.33	20.00	16.67	13.33
	AMC23	50.00	62.50	57.50	52.50
	Gaokao2023en	57.92	65.20	64.16	62.08
	Minerva Math	34.93	38.24	36.77	33.82
	Olympiad Bench	42.58	52.34	50.00	42.77
	Average	51.97	58.43	56.30	52.77

### H.3 MORE ABLATION STUDY

1392 In this section, we provide an extended ablation study to evaluate the contribution of each key  
 1393 component in Dr. MAMR. Table 6 summarizes the results for the 7B model across several math-  
 1394 reasoning benchmarks. These ablations are designed specifically to address the reviewer’s questions  
 1395 regarding (1) the effectiveness of the proposed Shapley-inspired grouping mechanism, (2) the choice  
 1396 of the cosine-similarity threshold for grouping, and (3) the robustness of the causal-influence (CI)  
 1397 computation to different embedding models.  
 1398

#### (1) Effectiveness of Shapley-inspired grouping:

1399 A core part of Dr. MAMR is the Shapley-inspired semantic grouping, which clusters intermediate  
 1400 reasoning segments into conceptually meaningful groups before computing causal influence. To  
 1401 assess its importance, we report a w/o Shapley in CI variant, where grouping is removed and CI is  
 1402 computed directly on every segment independently. As shown in Table 6, removing Shapley group-  
 1403 ing worsens performance across all benchmarks (e.g., AIME24: 20.00  $\rightarrow$  16.67, AMC23: 62.50  
 $\rightarrow$  60.00). This confirms that grouping reduces noise in the CI estimates by aggregating semanti-

1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
cally redundant reasoning segments, and that the Shapley mechanism meaningfully contributes to the stability and effectiveness of the causal-influence reward.

(2) Varying the semantic-similarity threshold:

Dr. MAMR uses a cosine-similarity threshold of 0.9 to decide whether two reasoning segments belong to the same group. To evaluate sensitivity to this design choice, we vary the threshold to 0.95 and 0.8. Results in Table 6 show: (1) 0.95 (more restrictive grouping): performance remains similar or slightly improved on some datasets, indicating that Dr. MAMR is stable when grouping is more conservative. (2) 0.8 (more aggressive grouping): performance drops more noticeably, suggesting that overly broad grouping merges semantically distinct segments, which weakens CI's ability to identify the truly influential steps.

Overall, Dr. MAMR maintains strong performance for a wide range of thresholds, demonstrating robustness, but extremely permissive grouping can blur important distinctions in the reasoning trajectory.

(3) Effect of embedding model used for grouping:

By default, semantic grouping uses Qwen2.5-0.5B embeddings. To test robustness to this choice, we replace the embeddings with: (1) Qwen2-0.5B, a smaller instruction-tuned model (2) Qwen3-Embedding-0.6B, a specialized embedding model

As shown in Table 6, the method remains stable across embedding variants. Notably, Qwen3-Embedding-0.6B improves grouping quality (e.g., Gaokao2023en: 65.20 → 66.49, Olympiad: 52.34 → 54.52), likely because its representations better capture semantic similarity. These results indicate that CI is robust to embedding model choice, and improvements can be obtained with stronger embedding encoders.

Table 6: Ablation study on the 7B model.

Variant	AIME24	AMC23	Gaokao2023en	Olympiad Bench
Dr.MAMR	20.00	62.50	65.20	52.34
w/o ND	13.33	55.00	63.64	47.85
w/o CI	13.33	52.50	63.38	45.31
w/o RB	16.67	57.50	63.90	50.58
w/o Shapley in CI	16.67	60.00	64.42	49.22
w/ Qwen2-0.5B for CI	16.67	60.00	63.12	51.56
w/ Qwen3-Embedding-0.6B for CI	20.00	62.50	66.49	54.52
w/ threshold 0.95 for CI	20.00	60.00	65.71	52.73
w/ threshold 0.8 for CI	16.67	57.50	62.86	50.20

Table 7: Comparison to stronger single-agent RL baseline.

Model	Benchmark	GRPO	DAPO	Dr. GRPO	GSPO	Dr. MAMR
Qwen2.5 -7B -Instruct	MATH500	75.50	75.80	75.60	76.20	78.60
	GSM8K	90.50	93.26	91.13	91.05	92.12
	AIME24	16.67	13.33	16.67	16.67	20.00
	AMC23	55.00	60.00	52.50	57.50	62.50
	Gaokao2023en	64.60	63.64	64.42	65.71	65.20
	Minerva Math	34.70	35.60	35.30	37.50	38.24
	Olympiad Bench	48.60	49.61	47.07	50.78	52.34
		Average	55.08	55.89	54.67	56.49
						58.43

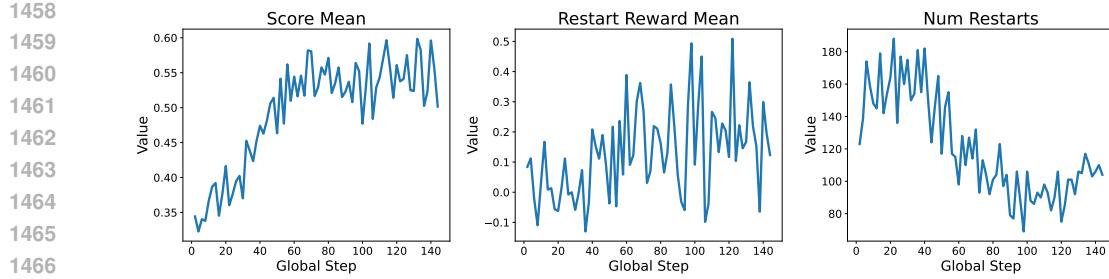


Figure 7: Training dynamics of restart behavior.

#### H.4 TRAINING DYNAMICS OF RESTART BEHAVIOR

In this section, we provide a detailed analysis of the training dynamics associated with the restart behavior in Dr. MAMR. Following the experimental setting described in Sec. 6.1, we set  $\alpha = \beta = 0.1$ , use 8 rollouts per prompt, and adopt a batch size of 128. At each global step, we track three quantities: (1) the mean reward obtained on the training batch, (2) the total number of restart actions triggered by the reasoning agent, and (3) the average reward assigned to those restart actions. The results are visualized in Fig. 7.

From the figure, several important patterns emerge. First, as training progresses, the overall mean reward of Dr. MAMR increases steadily, and the mean reward associated specifically with restart actions exhibits a similar upward trend. This indicates that the restart reward signal is informative: it provides consistent and meaningful feedback to the reasoning agent, rather than introducing noise or instability. The agent learns to exploit the restart mechanism in a way that aligns with improved task performance.

Second, although the number of restart actions experiences a mild increase during the earliest phase of training—an expected effect due to the restart reward signal not yet being fully calibrated—this trend reverses quickly. After approximately 40 training steps, the number of restart actions begins to decline and eventually stabilizes around step 100. While the restart frequency does decrease, the reduction is only partial, indicating that the agent is learning to avoid unnecessary or redundant restarts. Importantly, this stabilization occurs despite the fact that the reward for restart actions continues to increase. In other words, the positive restart reward does not cause the agent to enter a degenerate mode where restarts are over-triggered simply to accumulate reward.

This observation is precisely the behavior we aim to achieve: the reasoning agent is not “hacked” into producing restart tokens blindly. Instead, it learns when a restart is genuinely beneficial based on the current prompt and the intermediate reasoning trajectory.

Together, these findings provide compelling evidence that the restart mechanism in Dr. MAMR is both stable and well-behaved. Even when rewarded positively, the model does not collapse into overuse of restart behavior. Instead, it balances restart decisions in a principled manner, demonstrating robustness and reliability in practical training conditions.

#### H.5 RESULTS ON CODE BENCHMARK

To evaluate the generalization capability of Dr. MAMR beyond math reasoning, we conduct experiments on code generation benchmarks. We use the DeepCoder dataset<sup>7</sup> for training and adopt DeepSeek-R1-Distill-Qwen-1.5B as the base model. We compare Dr. MAMR against two baselines: the single-agent GRPO and the multi-agent ReMA, evaluating performance on three widely used code reasoning benchmarks: LiveCodeBench, Codeforces, and HumanEval+. For each benchmark, we report pass@ $k$  metrics at  $k \in \{1, 2, 4, 8\}$ .

As shown in Tables 10, 8, and 9, Dr. MAMR consistently outperforms both GRPO and ReMA across all benchmarks and almost all pass@ $k$  settings. On LiveCodeBench, Dr. MAMR achieves a substantial improvement in pass@1 (19.35%) compared to GRPO (15.77%) and ReMA (15.05%).

<sup>7</sup>agentica-org/DeepCoder-Preview-Dataset

1512 Similar gains are observed on HumanEval+ (61.35% vs. 58.28%/57.67%). These results demon-  
 1513 strate that Dr. MAMR not only excels in math reasoning but also generalizes effectively to complex  
 1514 code-based reasoning tasks, validating its robustness across diverse domains.

1515 We further visualize the training dynamics of Dr. MAMR compared to ReMA in Figure 8. The  
 1516 plotted curve reports the mean reward throughout the RL training process. We observe that while  
 1517 ReMA’s performance quickly plateaus and deteriorates—ultimately collapsing to near-zero values—  
 1518 Dr. MAMR demonstrates sustained and progressively improving performance. This indicates  
 1519 that Dr. MAMR enables stable optimization in long-horizon multi-agent reasoning tasks.

1520  
1521  
1522  
1523  
1524 Table 8: Comparison on the Codeforces benchmark (pass@k).

Metric	GRPO	ReMA	Dr. MAMR
pass@1	5.88	4.66	6.22
pass@2	8.33	7.35	8.33
pass@4	11.03	11.27	12.25
pass@8	15.44	14.95	15.93

1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540 Table 9: Comparison on the HumanEval+ benchmark (pass@k).

Metric	GRPO	ReMA	Dr. MAMR
pass@1	58.28	57.67	61.35
pass@2	69.33	70.55	71.78
pass@4	77.30	76.69	79.14
pass@8	83.44	82.82	84.66

1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556 Table 10: Comparison on the LiveCodeBench benchmark (pass@k)

Metric	GRPO	ReMA	Dr. MAMR
pass@1	15.77	15.05	19.35
pass@2	19.35	17.56	22.58
pass@4	25.45	22.22	26.88
pass@8	29.39	24.37	32.26

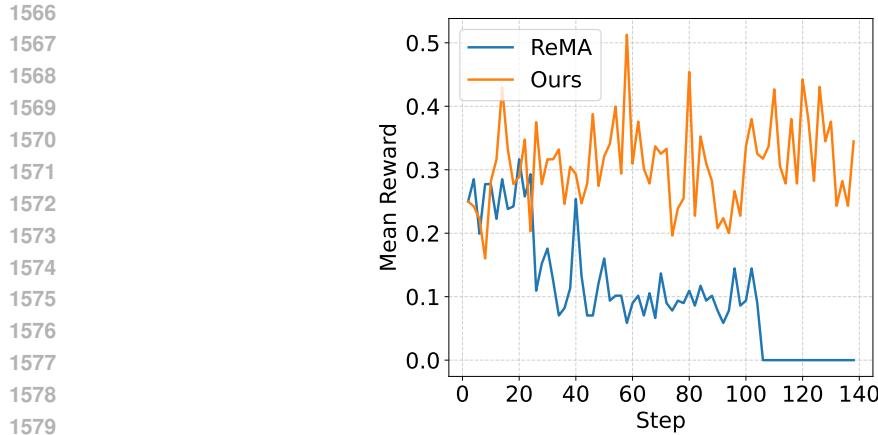


Figure 8: Training curves comparison on code dataset.

### System Prompts for Meta-Think and Reasoning Agents (Coding)

#### META-THINK AGENT SYSTEM PROMPT

You are a meta-think agent that represents human high-level thought process when solving coding problems. You will have a discussion with the reasoning agent. Each time you think about what to do next, consider:

- Analyzing the problem requirements and constraints carefully
- Breaking down the solution into clear algorithmic steps
- Identifying key data structures and algorithms needed
- Considering edge cases and potential bugs
- Requesting implementation of specific functions or code blocks
- Reviewing intermediate code and suggesting improvements
- Backtracking when the approach doesn't work
- Finally confirm when the solution is complete with the tag [FINISH]

#### REASONING AGENT SYSTEM PROMPT

You are a reasoning agent that implements code solutions step by step following the meta-think agent's instructions. When writing code:

- Implement the solution clearly and correctly
- Include proper error handling for edge cases
- Use appropriate data structures and algorithms
- Write clean, readable code with proper formatting
- When asked to finalize your answer, put your complete code solution within a markdown code block using triple backticks (```python ... ```)
- Make sure your final code is complete and can be executed directly

## H.6 TRAINING TIME COMPARISON

We report the per-step training time of Dr. MAMR compared to GRPO, with results shown in Figure 9. As illustrated in the figure, the training time of Dr. MAMR is generally comparable to that of the single-agent GRPO baseline.

## H.7 RESULTS ACROSS BENCHMARKS ON 3B MODEL

In this section, we present additional pass@1 performance results on the 3B model, as shown in Table 11. From the table, we observe that our Dr. MAMR consistently outperforms both single-agent

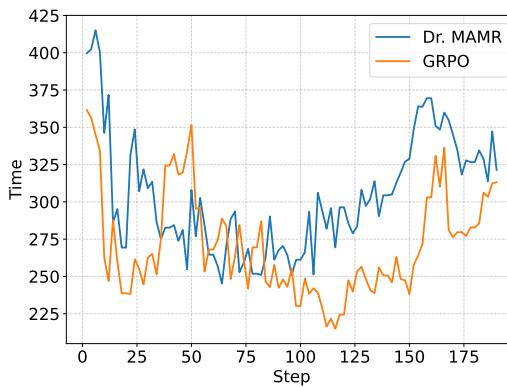


Figure 9: Per-step training time comparison.

GRPO and ReMA. However, the performance gains are less pronounced on the 7B and 14B models. We attribute this to their weaker instruction-following capability, which limits the performance upper bound of the multi-agent system.

Table 11: Performance on math benchmarks with 3B base model.

Model	Benchmark	GRPO	VRP (CoT)	ReMA	Dr. MAMR (Ours)
Qwen2.5 -3B -Instruct	MATH500	65.60	65.20	62.60	66.20
	GSM8K	85.30	72.02	83.17	85.37
	AIME24	13.33	3.33	3.33	16.67
	AMC23	40.00	20.00	42.50	50.00
	Gaokao2023en	54.30	30.91	52.73	55.33
	Minerva Math	31.20	16.91	26.47	32.35
	Olympiad Bench	30.20	6.07	27.56	30.57
<b>Average</b>		45.70	30.63	42.62	48.07

## H.8 TRAINING CURVE COMPARISON ON THE 3B MODEL

To further examine the training stability of smaller base models, we provide additional experiments on the 3B model. In this setting, we use MATH as the training dataset to rule out the possibility that collapse is solely caused by overly difficult data. The training curves reporting mean reward are shown in Fig. 10.

From the figure, we observe that under the ReMA framework, the 3B model collapses rapidly after only 20 training steps. In contrast, our Dr. MAMR framework maintains stable progress throughout training. This highlights how critical the lazy-agent issue becomes when the base model is relatively weak. Importantly, even with a less capable model, Dr. MAMR is still able to ensure stable training, underscoring its robustness.

## H.9 CASE STUDY ON RESTART BEHAVIOR

In this section, we present a case study on restart behavior, illustrating how restarts help the reasoning agent recover from its earlier mistakes. The full reasoning process is shown below. This case study demonstrates how restart behavior enables the agent to abandon an incomplete reasoning trajectory and reinitiate a more systematic approach. By explicitly reconsidering the intersection points, the agent successfully recovered the correct quadrilateral and computed the ground-truth area of  $4\sqrt{3}$ . Without such a restart, the agent would likely have produced an incorrect or incomplete solution.

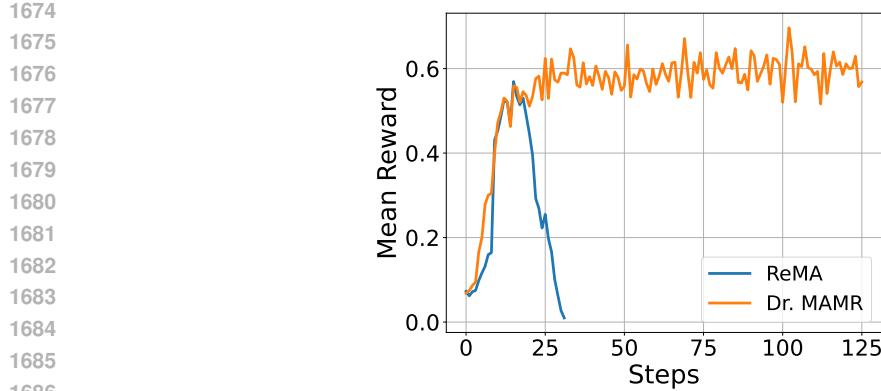


Figure 10: Training curves of 3B model trained with ReMA and Dr. MAMR.

## Case study on restart behavior

[Question] Find the area of the quadrilateral formed by the intersection points of the parabolas  $y^2 = 3x$ ,  $y^2 = -3x$ , and the circle  $x^2 + y^2 = 4$ . [Groundtruth:  $4\sqrt{3}$ ]

**Meta-thinking:**

To find the area of the quadrilateral formed by the points of intersection between the parabolas and the circle, begin by determining the intersection points. Substitute  $y^2 = 3x$  and  $y^2 = -3x$  into  $x^2 + y^2 = 4$  and work through the feasible solutions.

**Reasoning:**

For  $y^2 = 3x$ :  $x^2 + 3x = 4 \Rightarrow (x+4)(x-1) = 0$ , so  $x = 1$  or  $-4$ . But  $x = -4$  gives  $y^2 = -12$ . Thus  $(1, \pm\sqrt{3})$ .

For  $y^2 = -3x$ :  $x^2 - 3x = 4 \Rightarrow (x-4)(x+1) = 0$ , so  $x = 4$  or  $-1$ . But  $x = 4$  gives  $y^2 = -12$ . Thus  $(-1, \pm\sqrt{3})$ .

**Meta-thinking:**

The feasible intersections suggest four symmetric points on the circle:  $(1, \pm\sqrt{3})$  and  $(-1, \pm\sqrt{3})$ . It's natural to check the configuration these points form and then evaluate the area accordingly.

**Reasoning:**

<restart> Decision: Rethink from scratch. The feasible intersections are: - From  $y^2 = 3x$ :  $(1, \pm\sqrt{3})$  - From  $y^2 = -3x$ :  $(-1, \pm\sqrt{3})$

Thus, four points of intersection:  $A = (1, \sqrt{3})$ ,  $B = (1, -\sqrt{3})$ ,  $C = (-1, \sqrt{3})$ ,  $D = (-1, -\sqrt{3})$ .

**Meta-thinking:**

These points form a rectangle. - Width:  $|1 - (-1)| = 2$  - Height:  $|\sqrt{3} - (-\sqrt{3})| = 2\sqrt{3}$

Area:

$$\text{Area} = 2 \times 2\sqrt{3} = 4\sqrt{3}$$

[FINISH]

**Reasoning:**

$$4\sqrt{3}$$

## H.10 SCALING

In this section, we present the full results of test-time scaling on pass@K performance when comparing Dr. MAMR with ReMA, as shown in Fig. H.10. We observe that on harder benchmarks such as AIME24 and OlympiadBench, the performance gap between Dr. MAMR and ReMA widens as  $K$  increases, while on relatively easier benchmarks, Dr. MAMR consistently achieves better performance. These results highlight the strong capability of Dr. MAMR in handling a wide range of reasoning tasks.

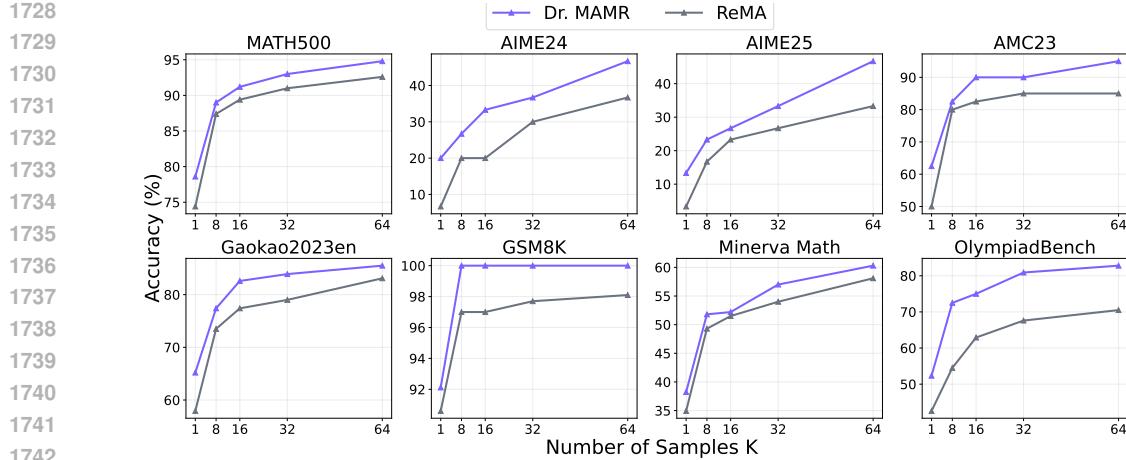


Figure 11: Pass@K performance.

## I THE USE OF LARGE LANGUAGE MODELS

Large Language Models (LLMs) were used to assist in writing and polishing this manuscript. Specifically, an LLM was employed to improve clarity, refine language, check grammar, and enhance overall readability.

The LLM was not involved in ideation, research design, data analysis, or the development of scientific content. All research concepts, methods, and analyses were independently conceived and conducted by the authors.

The authors take full responsibility for the manuscript's content, including any text refined with LLM assistance. All LLM-generated content adheres to ethical standards and does not constitute plagiarism or scientific misconduct.

1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781