MAESTRO: LEARNING TO COLLABORATE VIA CONDITIONAL LISTWISE POLICY OPTIMIZATION FOR MULTI-AGENT LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-agent systems (MAS) built on Large Language Models (LLMs) are being used to approach complex problems and can surpass single model inference. However, their success hinges on navigating a fundamental cognitive tension: the need to balance broad, **divergent** exploration of the solution space with a principled, **convergent** synthesis to the optimal solution. Existing paradigms often struggle to manage this duality, leading to premature consensus, error propagation, and a critical credit assignment problem that fails to distinguish between genuine reasoning and superficially plausible arguments. To resolve this core challenge, we propose the **Multi-Agent Exploration–Synthesis framework Through Role Orchestration (MAESTRO)**, a principled paradigm for collaboration that structurally decouples these cognitive modes. MAESTRO uses a collective of parallel Execution Agents for diverse exploration and a specialized Central Agent for convergent, evaluative synthesis. To operationalize this critical synthesis phase, we introduce **Conditional Listwise Policy Optimization (CLPO)**, a reinforcement learning objective that disentangles signals for strategic decisions and tactical rationales. By combining decision-focused policy gradients with a list-wise ranking loss over justifications, CLPO achieves clean credit assignment and stronger comparative supervision. Experiments on mathematical reasoning and general problem-solving benchmarks demonstrate that MAESTRO, coupled with CLPO, consistently outperforms existing state-of-the-art multi-agent approaches, delivering absolute accuracy gains of 6% on average and up to 10% at best.

1 INTRODUCTION

The rise of large language models (LLMs) have enabled a new type of *multi-agent system* (MAS) (Park et al., 2023; Chen et al., 2023a; Zhu et al., 2025), where multiple model instances collaborate to tackle problems that exceed the capacity of any single model (Zhang et al., 2024a; Qiao et al., 2024; Han et al., 2025). By distributing roles and enabling structured interaction, MASs hold the promise of achieving robustness, creativity, and reliability that emerge from collective intelligence (Cheng et al., 2024; Pezeshkpour et al., 2024). At the heart of any effective collaborative system lies a fundamental cognitive tension. Early work in the psychology of creativity (Runco & Chand, 1995; Brophy, 2001; Zhang et al., 2020) emphasizes that intelligent problem-solving requires a dynamic balance between two seemingly contradictory modes of thought: **Divergent Creativity** and **Convergent Critique**. Guilford's theory of divergent and convergent thinking (Guilford, 1967) formalizes this duality: divergence is the generative process of exploring a wide array of alternative hypotheses, while convergence is the evaluative process of comparing, refining, and synthesizing these options. Without the former, a system risks premature closure; without the latter, it risks incoherence and indecision (Sternberg & Lubart, 1991; Cropley, 2006). Achieving a principled and effective synergy between these two capabilities is the essential challenge for effective LLM agent collaboration.

Despite their diversity, the limitations of existing paradigms point to a set of recurring requirements for advancing multi-agent collaboration. First, an effective system should strike a balance between **divergent exploration and convergent synthesis**, ensuring that creativity is not stifled by premature agreement yet also not lost in unbounded search. Second, it should enable **disentangled credit assignment** across structured outputs (Li et al., 2025; He et al., 2025), so that strategic decisions and

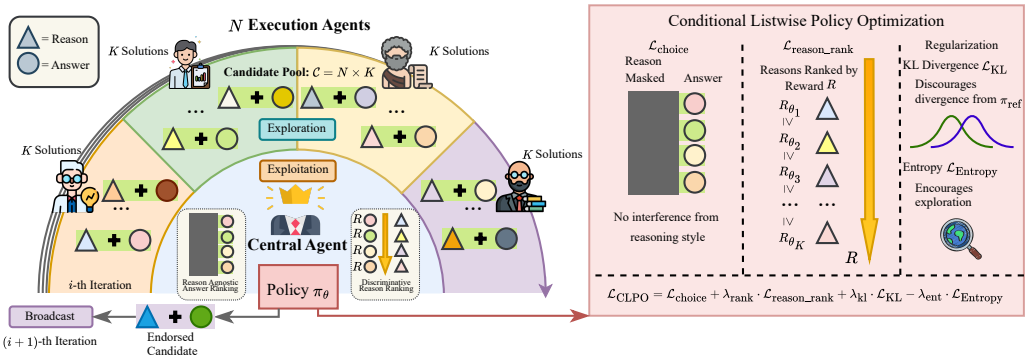

Figure 1: Overview of the **MAESTRO** framework. First, N execution agents each generate K candidate reasoning-answer pairs, forming a broad solution pool. A central agent then governs exploitation by applying discriminative selection over the candidate set. The decision policy π_θ is trained under **Conditional Listwise Policy Optimization (CLPO)**, which integrates a choice-aware objective, a reasoning-rank objective, and regularization terms including KL divergence and entropy. The endorsed candidate is subsequently broadcast for iterative refinement, enabling multi-round improvement within a principled multi-agent collaboration paradigm.

supporting rationales receive distinct and targeted learning signals rather than being conflated into a single monolithic reward. Third, a robust framework requires **transparent and scalable interaction protocols** (Qian et al., 2024; Hu et al., 2024c), where information is propagated in analyzable ways that remain efficient as the number of agents and rounds increases (Yang et al., 2025). Together, these desiderata highlight the limitations of existing approaches and motivate the need for a new paradigm that integrates principled exploration, evaluative precision, and collaborative scalability.

To address these desiderata, we propose the **Multi-Agent Exploration–Synthesis framework Through Role Orchestration (MAESTRO)**, a principled paradigm for multi-agent collaboration (Figure 1). The effectiveness of MAESTRO arises not from any single component, but from the synergistic orchestration of specialized roles. MAESTRO explicitly operationalizes the divergent–convergent duality through a structured role orchestration: (*i*) **Divergence as Collective Exploration**, where multiple Execution Agents generate a broad and diverse candidate pool; (*ii*) **Convergence as List-wise Bayesian Synthesis**, where a Central Agent evaluates these candidates to identify and endorse the most promising solution; and (*iii*) **Broadcast as Public Conditioning**, where the endorsed solution is propagated back to all agents, guiding the next round of exploration. This cycle of divergence, convergence, and broadcast structures collaboration into analyzable and scalable phases. To further optimize the convergence phase, we introduce **Conditional Listwise Policy Optimization (CLPO)**, a reinforcement learning objective that disentangles decision-making from rationale generation. Unlike standard GRPO-style sequence-level training, CLPO allocates learning signal separately to *decisions* (which candidate to endorse) and *reasons* (why this choice is defensible). MAESTRO and CLPO constitute a new paradigm for multi-agent collaboration that integrates cognitive inspiration with principled optimization. The main contributions of this paper are:

- We introduce the Multi-Agent Exploration–Synthesis framework Through Role Orchestration (MAESTRO), a principled paradigm for multi-agent collaboration that explicitly operationalizes the divergent–convergent duality through three coordinated phases.
- We propose Conditional Listwise Policy Optimization (CLPO), an RL objective that decouples signals for *decisions* and *reasons*. CLPO combines group-relative decision optimization with listwise rationale ranking for clean credit assignment and stable convergence.
- Extensive experiments on mathematical and general reasoning benchmarks show that MAESTRO with CLPO achieves significant improvements over state-of-the-art baselines.

2 Related Work

We highlight representative related works in multi-agent LLM collaboration and RL for multi-agent LLMs. For a more in-depth account of related work, see Appendix A.

Multi-Agent LLM Collaboration. Large language model (LLM) based multi-agent systems have been proposed to overcome the inherent limits of single models in context length, sequential reasoning, and skill breadth (Abdelnabi et al., 2023; Wu et al., 2024; Yan et al., 2025; Dai et al., 2025). By coordinating multiple agents, these systems can decompose tasks, critique candidate solutions, and integrate diverse perspectives (Hong et al., 2023; Chen et al., 2023b; Qiao et al., 2024; Pan et al., 2024). One common design follows *prestructured coordination*, where communication topologies and protocols are fixed in advance (Chen et al., 2024; Mukobi et al., 2023; Wang et al., 2023; Abdelnabi et al., 2024). Debate and peer-review frameworks (Du et al., 2023; Chan et al., 2023; Liu et al., 2024) encourage agents to cross-examine one another, while chain or graph structures regulate message flow (Qian et al., 2024; Liu et al., 2023b). These methods reduce hallucinations and improve consistency but often enforce early convergence, limiting exploration and leaving credit assignment opaque (Hu et al., 2024a; Yue et al., 2025). A second line explores *adaptive coordination*, where the collaboration graph is reorganized dynamically during inference. Examples include routing and pruning strategies (Yue et al., 2025; Hu et al., 2024b), as well as workflow and graph-search approaches that optimize interaction structures through reinforcement or evolutionary methods (Zhuge et al., 2024; Zhang et al., 2024c;b).These frameworks improve scalability and efficiency but typically treat feedback as a global property of the entire system, which limits their ability to provide fine-grained credit assignment for individual contributions.

Reinforcement Learning for Multi-Agent LLMs. Reinforcement learning (RL) provides a natural mechanism for improving collaboration in multi-agent LLM systems beyond static prompt design (Madaan et al., 2023; Zelikman et al., 2024; 2022; Zhuang et al., 2024; Zhu et al., 2025). Rather than relying solely on prestructured debate or workflow rules, RL enables agents to adapt interaction patterns from feedback, learning when and how to communicate to achieve stronger group performance (Zhou et al., 2025; Wang et al., 2024; Xu et al., 2025; Wan et al., 2025; Park et al., 2025; Yang & Thomason, 2025). These approaches show that reward-driven updates can uncover strategies for dynamic role assignment, coordination, and decision aggregation. A central challenge in this setting is credit assignment (Liu et al., 2023a; Zhang et al., 2024d;e; Li et al., 2024b). Most existing methods propagate reward at the system level, treating outcomes as global properties of the entire team (Jiang et al., 2025; Lin et al., 2025). This global reward fails to identify the specific contributions of individual agents or to separate the quality of rationales from the correctness of final decisions. Recent efforts attempt to design more targeted objectives (Wei et al., 2025; Alsadat & Xu, 2024), but principled, fine-grained supervision remains limited. Our work focuses specifically on the convergence step: we view it as a structured optimization problem and design an objective that provides more precise credit assignment than existing system-level rewards.

3 METHODOLOGY

We introduce a novel learning paradigm to enhance the collective problem-solving capabilities of multi-agent systems. We design a collaborative process through the lens of a new structural framework, the **Multi-Agent Exploration–Synthesis (MAESTRO)** paradigm, which orchestrates the generation of diverse solutions and the subsequent critical evaluation. At the core of this framework lies our primary algorithmic contribution, **Conditional Listwise Policy Optimization (CLPO)**, a reinforcement learning algorithm specifically designed to train the central decision-making policy. This methodology systematically addresses the challenges of credit assignment and signal poverty inherent in complex, language-based collaborative tasks.

3.1 PRELIMINARIES

We study a round-based collaborative protocol for answering question q. In round t, each of the N execution agents independently samples K candidates conditioned on the current context $s_t^{(i)} := (q, b_{t-1}, z_{t-1}^{(i)})$ for $i \in [N]$, where b_{t-1} denotes the previous public broadcast, and $z_{t-1}^{(i)}$ denotes the i-th agent's private history (state). This yields a total slate $\mathcal{C}_t$ of $N \times K$ candidate responses. The central policy then performs convergence by selecting one candidate in $\mathcal{C}_t$ to endorse and issues a public broadcast b_t that contains the index of the answer and optionally a brief justification. The broadcast b_t then conditions the next round $t + 1$. This process stops after a fixed number of rounds R, or when a stopping rule is met. Supervision is primarily the correctness of the endorsed answer at termination, and an optional term rewards the comparative quality of the justification.

3.2 THE MAESTRO FRAMEWORK: A PARADIGM FOR COLLECTIVE SYNTHESIS

We now formally introduce the Multi-Agent Exploration–Synthesis (MAESTRO) framework. MAESTRO operationalizes the divergent–convergent model of creative problem-solving in a principled manner, by decomposing each round of the collaborative process into two distinct phases, which we view through the lenses of Bayesian inference and information theory.

Phase 1: Divergence as Collective Exploration. The primary objective of the divergence phase is to effectively explore the vast solution space, mirroring the divergent thinking process. This is achieved through a collective of N parallel Execution Agents. Conditioned on the current state $s_t^{(i)}$, each agent is tasked with generating a diverse set of K candidate solutions. Note here that each candidate solution is a complete trajectory, e.g., a full reasoning chain leading to a final answer. Formally, each agent $i \in [N]$ at time t samples from its policy $\pi_{\phi^{(i)}}$ as follows:

$$c_{t,k}^{(i)} \sim \pi_{\phi^{(i)}}(\cdot \mid q, z_{t-1}^{(i)}, b_{t-1}), \quad k \in [K]. \tag{1}$$

This collective effort produces candidate pool $\mathcal{C}_t = \{\{c_{t,k}^{(i)}\}_{k=1}^{K}\}_{i=1}^{N}$. A key metric for this phase is the *coverage probability (p_t)* that the pool contains at least one correct solution:

$$p_t := \Pr\Big(\bigcup_{i=1}^{N}\bigcup_{k=1}^{K} E(c_{t,k}^{(i)}) \,\Big|\, s_t^{(1)}, \ldots, s_t^{(N)}\Big), \tag{2}$$

where $E(c)$ is the event that candidate c is correct. The primary goal of Phase 1 is to increase the expected coverage p_t in a fixed resource budget.

Epsilon-greedy exploration. To prevent over-conditioning during candidate generation, we allocate a small broadcast-agnostic exploration mass using a simple *epsilon-greedy* strategy. Specifically, we sample $\tilde{\pi}_{\phi^{(i)}}(\cdot \mid q, z_{t-1}^{(i)}, b_t) = (1-\varepsilon)\,\pi_{\phi^{(i)}}(\cdot \mid q, z_{t-1}^{(i)}, b_{t-1}) + \varepsilon\,\pi_{\phi^{(i)}}^{\text{base}}(\cdot \mid q)$ with default $\varepsilon = 0.1$, where $\pi_{\phi^{(i)}}$ is defined in (1). This yields a coverage floor: for any subset A of the candidate space, $\tilde{\pi}_{\phi^{(i)}}(A \mid s_t^{(i)}) \geq \varepsilon\,\pi_{\phi^{(i)}}^{\text{base}}(A \mid q)$, so regions reachable by the base policy retain non-zero sampling mass. In practice, we implement the mixture via per-sample random dropout, using the base prompt with probability ε and otherwise conditioning on the broadcast and the agent's private history.

Phase 2: Convergence as List-wise Bayesian Synthesis. Following divergent exploration in Phase 1, the convergence phase is orchestrated by a single Central Agent. Its role is to evaluate and synthesize the collective information in the slate $\mathcal{C}_t$. We view this step as approximating a Bayesian decision over the posterior probabilities for round t:

$$\eta_{t,k}^{(i)} := \Pr\big(E(c_{t,k}^{(i)}) \mid q, \mathcal{C}_t\big), \quad i \in [N],\ k \in [K]. \tag{3}$$

Recall that under a 0–1 loss, the Bayes optimal action selects $(i^\star, k^\star) \in \arg\max_{i,k} \eta_{t,k}^{(i)}$. We therefore train our Central Agent's policy, $\pi_\theta(\cdot \mid q, \mathcal{C}_t)$, to approximate this optimal Bayes decision rule via the CLPO loss (Section 3.3). The success of this phase is measured by the *identification probability (q_t)*, defined as the conditional probability that the policy selects a correct candidate given that the slate $\mathcal{C}_t$ contains at least one correct option. Specifically, let $S_t := \{(i,k) \in [N] \times [K] \mid E(c_{t,k}^{(i)}) \text{ holds}\}$ be the latent set of correct candidates and let $(i_t, k_t) \sim \pi_\theta(\cdot \mid q, \mathcal{C}_t)$ denote the centralized decision. Then the identification probability q_t is defined as:

$$q_t := \Pr\big((i_t, k_t) \in S_t \,\big|\, q, \mathcal{C}_t, \{|S_t| \geq 1\}\big). \tag{4}$$

This metric quantifies the agent's critical evaluation and synthesis capability.

Broadcast as Public Conditioning. After selection in Phase 2, the Central Agent emits a public broadcast b_t, containing the endorsed index and a compact justification. This broadcast b_t conditions the next round $t+1$. We can interpret the flow of information as reducing the Shannon entropy of the ground-truth answer Y with respect to an observer's posterior; by the chain rule for mutual information, we have $H(Y \mid q, b_{1:t}) \leq H(Y \mid q, b_{1:t-1})$ for all t, where $b_{1:t} = (b_1, \ldots, b_t)$.

Overall Dynamics. We summarize the per-round behavior as a coverage–identification factorization conditioned on the public context (q, b_{t-1}). The system first attains *coverage* p_t when the slate contains at least one correct candidate, then achieves *identification* q_t when the central policy selects

a correct candidate. In Appendix B, we show the following cumulative reliability inequality: if we have that both $p_t \geq \underline{p}$ and $q_t \geq \underline{q}$ almost surely for all t, then $\Pr(\text{success within } R \text{ rounds}) \geq 1 - (1 - \underline{pq})^R$. An immediate consequence is the following tail inequality: if $R \geq \frac{1}{\underline{pq}} \log\left(\frac{1}{\delta}\right)$, then the probability of success within the first R rounds is at least $1 - \delta$.

3.3 Conditional Listwise Policy Optimization (CLPO)

Having established the MAESTRO paradigm, the key question becomes how to *optimize* the convergence process so that the Central Agent can reliably approximate the Bayesian decision rule.[1] Conceptually, the two phases of MAESTRO naturally align with the classical exploration–exploitation trade-off: Phase 1 (divergence) expands the hypothesis space through exploration, while Phase 2 (convergence) serves as exploitation, transforming the diverse candidate set into a single endorsed solution with a supporting rationale. This perspective makes the Phase 2 convergence step a natural fit for reinforcement learning (RL) on the objective:

$$\max_{\pi_\theta} \ \mathbb{E}_{(q,\mathcal{C})}\Big[\, r\big(q, \mathcal{C}, \text{Chosen}, \text{Reason}\big)\,\Big], \tag{5}$$

where π_θ denotes the policy of the Central Agent, $\mathcal{C}$ is the candidate pool of responses sub-sampled over all R rounds, and r is our unified reward, which comprises answer correctness and rationale quality assessed via reasoning attributes, as detailed in Appendix C.1. This formulation highlights the two challenge at the heart of convergence: the Central Agent must both *(i)* provide a coherent and discriminative rationale that distinguishes the endorsed solution from its competitors, and also *(ii)* select the correct decision token.

Limitations of Naïve Sequence-Level Optimization. A natural baseline for training (5) is *Group Relative Policy Optimization (GRPO)* (Shao et al., 2024), which contrasts candidates within a group and scales sequence log-probabilities by relative advantage. Although suitable for "pick-one-from-K" settings, applying GRPO to full sequences exposes three core issues. First, it behaves like *pointwise supervision*: updates treat each completion in isolation rather than judging rationales by their strength relative to alternatives. Second, the reward signal is *entangled* across decision and rationale tokens, which obscures credit assignment. Third, this entanglement induces *spurious style effects*, where verbosity or lexical patterns receive undue credit and concise reasoning is penalized.

Conditional Listwise Policy Optimization (CLPO). We propose CLPO, a decoupled training loss that first learns to produce reliable, discriminative rationales and then learns to make a reliable discrete choice. Concretely, CLPO optimizes the rationale span with a conditional listwise ranking objective (Xia et al., 2008) over the entire candidate set. CLPO allocates the reinforcement signal to the decision tokens, using a focused policy-gradient update to sharpen identification without interference from explanation style. By matching each subproblem to the right objective, CLPO resolves credit entanglement, reduces confounding factors from length/style, and stabilizes training.

Strategic Decision Loss ($\mathcal{L}_{choice}$). The convergence phase ultimately hinges on the central agent's ability to make a precise strategic decision: which candidate to endorse. To ensure a clean credit signal, we allocate the reinforcement gradient exclusively to the decision tokens (the choice and its corresponding answer), conditioned on the rationale context. This disentanglement prevents reasoning length or style from interfering with the discrete choice. Formally, we define:

$$\mathcal{L}_{\text{choice}} = -\mathbb{E}\Big[\sum_{k=1}^{|\mathcal{C}|} A_k \cdot \log \pi_\theta(k \mid q, \mathcal{C})\Big], \tag{6}$$

where the advantage $A_k = r(c_k) - \bar{r}$, and $\bar{r}$ is the average reward within the candidate set. In practice, we mask the rationale tokens and aggregate log-probabilities only over the decision segment.

Tactical Argumentation Loss ($\mathcal{L}_{reason\_rank}$). Agents articulate a justification along with a discrete choice. In our framework, the rationale is generated *before* the final endorsement. We posit that a justification should not only be plausible on its own, but its plausibility should surpass alternatives. To capture this comparative quality, we employ a *Listwise Ranking Loss* (Xia et al., 2008). Formally, let σ be the permutation that sorts the rewards in descending order, $r(c_{\sigma_1}) \geq \cdots \geq r(c_{\sigma_{|\mathcal{C}|}})$. Write

[1] We do not consider optimizing the policies of the execution agents, which may further improve MAESTRO.

Type	Mech	Model	GSM8K	MATH	AIME	AMC	MMLU	HumanEval
SA	Ref	Vanilla	0.7276	0.4285	0.0296	0.0803	0.5799	0.4756
SA	Ref	CoT	0.7422	0.4693	0.0370	0.1165	0.6157	0.5142
SA	Ref	SC	0.8079	0.5128	0.0407	0.1245	0.6830	0.5752
MA	Prog	PHP	0.8001	0.5371	0.0444	0.1566	0.6846	0.5650
MA	Deb	LLM-Debate	0.8352	0.5625	0.0556	0.1928	0.6759	0.5772
MA	Deb	Group-Debate	0.8398	<u>0.5742</u>	0.0519	<u>0.2048</u>	<u>0.6989</u>	0.5793
MA	Dyn	DyLAN	0.8203	0.5532	0.0370	0.1968	0.6685	0.6159
WF	Dyn	GPTSwarm	<u>0.8489</u>	0.5669	<u>0.0578</u>	0.1566	0.6967	0.5955
WF	Dyn	AgentPrune	0.8438	0.5437	0.0481	0.1647	0.6909	0.5711
WF	Dyn	AFlow	0.8375	0.5528	0.0444	0.1205	0.6931	<u>0.6220</u>
WF	E-S	MAESTRO	0.8703	0.5916	0.0556	0.2371	0.7052	0.6267
WF	E-S	w/ SFT	0.8769	0.5983	0.0538	0.2482	0.7085	0.6321
WF	E-S	w/ GRPO	0.8867	0.6129	0.0704	0.2630	0.7168	0.6538
WF	E-S	w/ CLPO	**0.8933**	**0.6285**	**0.0851**	**0.2852**	**0.7238**	**0.6687**

Table 1: Comparison of baseline and proposed methods using the LLaMA-8B backbone. The table organizes models by Type (SA: single-agent, MA: multi-agent, WF: workflow-style framework) and by Mechanism (Reflection, Progressive Prompting, Debate, Dynamic Coordination, and Exploration–Synthesis). Underlined numbers indicate the best-performing baseline on each benchmark.

the justification for the k-th candidate in $\mathcal{C}$ as a token sequence $y_{k,1:L_k}$. We define this loss as:

$$\mathcal{L}_{\text{reason\_rank}} = -\sum_{j=1}^{|\mathcal{C}|} \log \frac{\exp(s_{\sigma_j})}{\sum_{l=j}^{|\mathcal{C}|} \exp(s_{\sigma_l})}, \quad s_k = \frac{1}{L_k} \sum_{\tau=1}^{L_k} \log \pi_\theta\big(y_{k,\tau} \mid y_{k,1:\tau-1},\, q,\, \mathcal{C}\big). \tag{7}$$

The CLPO Objective. The CLPO training objective combines the two losses (6) and (7), with standard regularization terms to ensure stable exploration and prevent catastrophic forgetting. The policy is regularized towards a reference policy π_{ref} (e.g., the initial SFT model) via a KL-divergence term $\mathcal{L}_{\text{KL}} = \mathbb{E}\left[D_{\text{KL}}(\pi_\theta(\cdot \mid q, \mathcal{C}) \,\|\, \pi_{\text{ref}}(\cdot \mid q, \mathcal{C}))\right]$ and an entropy bonus $\mathcal{L}_{\text{Entropy}} = \mathbb{E}\left[H(\pi_\theta(\cdot \mid q, \mathcal{C}))\right]$ that encourages exploration of justifications. The final objective is:

$$\mathcal{L}_{\text{CLPO}} = \mathcal{L}_{\text{choice}} + \lambda_{\text{rank}} \cdot \mathcal{L}_{\text{reason\_rank}} + \lambda_{\text{kl}} \cdot \mathcal{L}_{\text{KL}} - \lambda_{\text{ent}} \cdot \mathcal{L}_{\text{Entropy}}. \tag{8}$$

As we will see shortly, by decoupling the learning objectives for strategic choice and tactical argumentation, CLPO delivers a richer and more stable gradient signal, ensuring clean credit assignment.

4 EXPERIMENTS

Experimental Setup. We evaluate our approach across diverse benchmarks, including mathematical reasoning (GSM8K, MATH, AIME, AMC), factual and analytical reasoning (MMLU), and program synthesis (HumanEval), using Solve Rate, Accuracy, and Pass@1 as evaluation metrics. Baselines span single-agent reasoning methods, peer-interaction frameworks, routing and topology controllers, workflow and graph search approaches, and communication-efficient systems. Unless otherwise noted, experiments use three agents and three communication rounds, with instruction-tuned LLaMA-3B/8B and Qwen-3B/7B models under standard nucleus sampling. All reported results are averaged over three random seeds. See Appendix C.1 for a full account of settings.

4.1 MAIN EXPERIMENTS

Overall Performance. Table 1 shows that MAESTRO consistently surpasses both single-agent and multi-agent baselines across six reasoning benchmarks. On the trainable backbone LLaMA-8B, MAESTRO with CLPO achieves state-of-the-art accuracy, reaching 89.33% on GSM8K and 28.52% on AMC, which corresponds to average gains of 4%–8% over strong baselines such as GPTSwarm, AgentPrune, and Group-Debate. The improvements arise from two complementary effects: parallel exploration increases coverage, while CLPO strengthens the central selector's ability to identify correct solutions. This dual mechanism is especially beneficial on competition-style math tasks

Dataset	Vanilla	CoT	SC	Debate	GPTS	AP	AF	MAESTRO
GSM8K	93.17	93.68	93.32	94.66	94.66	94.89	92.30	**95.60**
MMLU	77.81	78.43	81.05	81.04	82.80	83.02	83.10	**84.09**
HumanEval	85.71	86.69	87.58	84.38	86.28	86.80	90.06	**90.65**

Table 2: Performance comparison on GSM8K, MMLU, and HumanEval using a GPT-4o-mini backbone. MAESTRO consistently achieves the highest accuracy across all benchmarks, outperforming both single-agent methods (Vanilla, CoT, and SC) and existing multi-agent frameworks like Debate, GPTSwarm (GPTS), AgentPrune (AP), and AFlow (AF). The improvements confirm that MAESTRO remains effective even when applied zero-shot to closed-source LLMs.

Model	LLaMA-8B	LLaMA-3B	Qwen-7B	Qwen-3B
Vanilla	0.7276	0.4685	0.9088	0.8337
CoT	0.7422	0.5014	0.9098	0.8456
SC	0.8079	0.5421	0.9295	0.8860
PHP	0.8001	0.6222	0.9330	0.8645
LLM-Debate	0.8352	0.7584	0.9363	0.8714
DyLAN	0.8203	0.7647	0.9315	0.8810
GPTSwarm	0.8489	0.6919	0.9227	0.8678
AgentPrune	0.8438	0.6502	0.9244	0.8643
AFlow	0.8375	0.6837	0.9286	0.8752
MAESTRO w/ CLPO	**0.8933**	**0.8153**	**0.9512**	**0.9083**

Table 3: Performance of collaborative reasoning baselines across four backbone LLMs (LLaMA-8B, LLaMA-3B, Qwen-7B, Qwen-3B) on GSM8K. MAESTRO w/ CLPO consistently achieves the highest accuracy, demonstrating robustness and generality across model architectures.

such as AMC and AIME, where incorrect but fluent candidates often mislead majority-voting or self-consistency. Importantly, the gains are not limited to trainable backbones. As shown in Table 2, even with the closed-source GPT-4o-mini under a prompt-only setting, MAESTRO achieves the best or tied-best results on GSM8K, MMLU, and HumanEval. The consistency across open- and closed-source models indicates that improvements stem from the collaborative orchestration itself rather than parameter updates, establishing MAESTRO as a robust paradigm for multi-agent LLM collaboration.

Cross-Backbone Consistency. To further examine the generality of our optimization strategy, we applied MAESTRO with CLPO across different LLM backbones, including LLaMA-8B, LLaMA-3B, Qwen-7B, and Qwen-3B (Table 3). We observe consistent improvements across all settings. On GSM8K, the accuracy reaches 89.33% with LLaMA-8B, 81.53% with LLaMA-3B, 95.12% with Qwen-7B, and 90.83% with Qwen-3B, establishing clear gains compared to their strongest respective baselines. The effectiveness of CLPO is not confined to a specific model family or size. Instead, as shown in Table 4, the optimization consistently enhances the identification probability q_t, enabling the central synthesis agent to more reliably distinguish correct solutions from plausible distractors. Importantly, this pattern is also reflected on AMC, where the improvements are similarly pronounced, underscoring that the collaborative mechanism combined with CLPO is broadly transferable across architectures. Overall, these findings confirm that MAESTRO with CLPO achieves robust gains across backbones, validating the universality of our collaborative optimization paradigm.

4.2 ANALYSIS EXPERIMENTS

Centralized Paradigm Variants: Selection vs. Generation. We now examine two natural centralized paradigms for convergence, namely generation and selection, to clarify why the latter forms the core of MAESTRO (Figure 2). When the central agent directly generates a reasoning trajectory and final answer (CENTRAL-GEN), the accuracy drops substantially. Incorporating self-consistency into generation (CENTRAL-GEN+SC) yields only a marginal gain. In contrast, the selection paradigm (CENTRAL-SELECT, ours) described in Section 3.2 attains the highest accuracy, reaching 0.870 on GSM8K and 0.237 on AMC. Figure 2 (Right) further decomposes the results into coverage and iden-

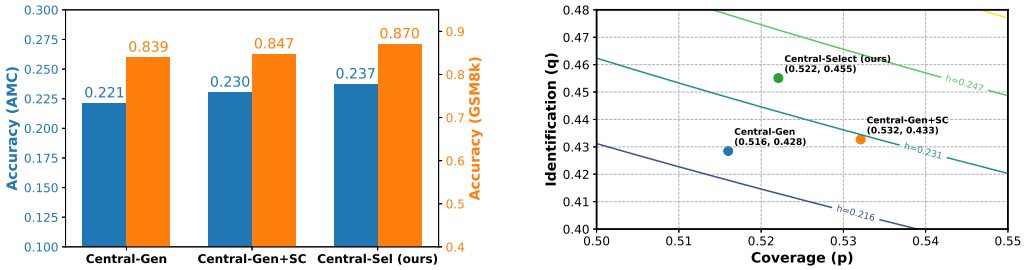

Figure 2: Comparison of collaboration paradigms. **Left:** task accuracy on AMC and GSM8K across different central coordination strategies. **Right:** performance decomposed into coverage and identification rates; central selection transforms diverse reasoning into reliable outcomes.

tification probabilities. The three paradigms exhibit similar coverage, but the identification probability is markedly higher under CENTRAL-SELECT (0.8919 on GSM8K and 0.4551 on AMC), which directly explains its superior end-to-end accuracy. We hypothesize that generation forces the central agent to absorb long and noisy contexts from multiple candidates, often diluting critical distinctions and amplifying misleading patterns. LLMs also tend to prioritize narrative coherence over factual correctness, which makes direct generation vulnerable to self-consistent hallucinations (Farquhar et al., 2024; Banerjee et al., 2025). Self-consistency mitigates randomness but cannot overcome these structural issues. By contrast, the selection paradigm frames convergence as a discriminative comparison among competing candidates, thereby preserving informative differences and reliably elevating correct solutions. Figure 5 (appendix) contains a complementary visualization.

Evidence versus Verdict in Centralized Selection. We examine how different types of candidate information influence the central selector's decisions by comparing three settings: *Reason-only* (only reasoning steps), *Answer-only* (only the final answer), and *Both* (ours, reasoning with the answer). As shown in Figure 3, *Answer-only* yields the weakest performance (GSM8K 0.840, AMC 0.205), while *Reason-only* performs better but remains slightly below the full setting. Since the candidate pool is identical, coverage is unchanged and differences arise from identification capability. The results show that reasoning and outcomes play complementary roles. Without reasoning, the selector lacks evidential structure and often falls back on superficial heuristics. Without final outcomes, it struggles to resolve cases where plausible reasoning paths diverge to different answers. Combining both provides the strongest performance: reasoning paths supply discriminative evidence, while answers anchor the verdict and disambiguate close cases.

Disentangled Optimization Signals in CLPO. To better understand the contribution of each optimization signal in CLPO, we conduct an ablation study by removing either the decision-focused loss $\mathcal{L}_{\text{choice}}$ or the rationale ranking loss $\mathcal{L}_{\text{reason}}$ (Figure 3). Removing $\mathcal{L}_{\text{choice}}$ causes a modest decline, indicating that ranking-based supervision over rationales alone can sustain reasonable convergence. In contrast, removing $\mathcal{L}_{\text{reason}}$ leads to a sharp degradation (AMC 0.261, GSM8K 0.881); without comparative evaluation of explanations, the selector is more easily swayed by persuasive but incorrect candidates. The full CLPO objective achieves the best performance, confirming the necessity of combining both terms. This pattern aligns with our design intuition: $\mathcal{L}_{\text{choice}}$ strengthens decisiveness by refining the probability of endorsing the correct candidate, while $\mathcal{L}_{\text{reason}}$ enforces discriminative evidence quality by forcing correct rationales to outrank distractors. Their joint effect provides clean credit assignment across decisions and justifications, ensuring that convergence is accurate.

4.3 HYPER-PARAMETER ANALYSIS

Scaling Agent Populations and Collaboration Rounds. We study how the size of the agent collective and the number of collaboration rounds influence performance. As shown in Figure 4, increasing the population from two to four steadily improves accuracy (AMC 0.253 $\rightarrow$ 0.3052; GSM8K 0.8693 $\rightarrow$ 0.9037). Broader exploration raises the probability that at least one candidate is correct, and the central selector trained with CLPO can convert this coverage into higher identification accuracy. Beyond four agents, however, gains saturate and slightly decline because redundancy introduces distractors. A similar trend appears when varying the number of collaboration rounds (see Figure 6 in the appendix). Adding one or two rounds improves identification by focusing exploration

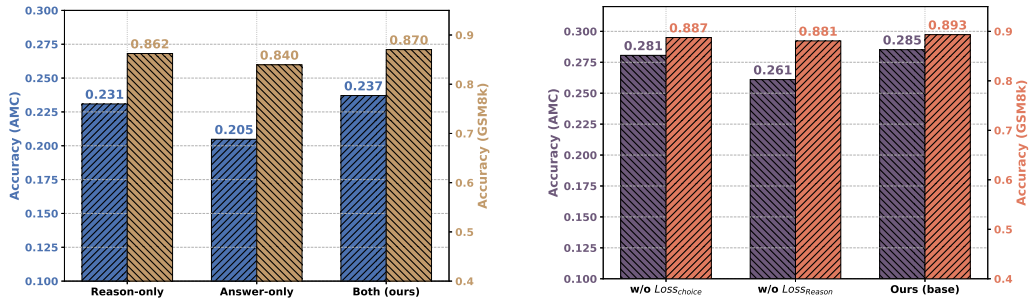

Figure 3: Ablation studies on central selection inputs and CLPO losses. **Left:** Reason-only, Answer-only, and Both settings when passing candidate information to the central selector. **Right:** contributions of loss components studied by removing choice or reasoning supervision.

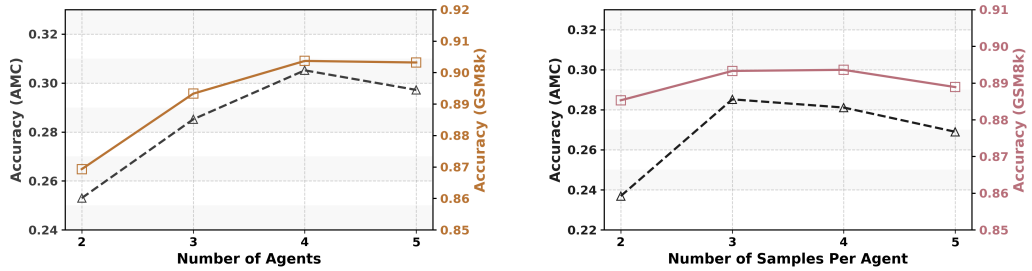

Figure 4: Effect of collaborative scale on reasoning performance. **Left:** accuracy with varying agent numbers. **Right:** impact of sampling multiplicity. In each, the solid line corresponds to GSM8K accuracy (right axis) and the dashed line corresponds to AMC accuracy (left axis).

through broadcasted evidence, while coverage changes little once the initial pool is large. Excessive rounds reduce diversity, amplify early errors through herding, and increase stochastic variance. The results highlight a consistent trade-off: additional agents and rounds enhance coverage and identification up to a point, but beyond that redundancy and bias dominate. Moderate settings of 3–4 agents and 2–3 rounds achieve the best balance. A more detailed analysis is provided in Appendix C.2.

Sampling Depth per Agent: Coverage–Variance Trade-off. We examine how the number of samples per agent (K) affects performance (Figure 4). As K increases from 2 to 3, accuracy improves (AMC $0.2369 \rightarrow 0.2852$; GSM8K $0.8853 \rightarrow 0.8933$). After, gains saturate: GSM8K changes little at $K{=}4$ and declines at $K{=}5$, while AMC peaks at $K{=}3$ before dropping, reflecting the exploration–synthesis decomposition. Larger K initially raises coverage, but multi-sampling from the same policy quickly yields correlated and redundant outputs; deeper sampling inflates within-round variance by drawing repeatedly from one agent rather than diversifying across agents. Once coverage nears saturation, additional samples contribute more noise than signal. These results suggest a practical guideline: allocate budget to enlarging the number of agents to diversify hypotheses, while keeping K modest so that the selector can reliably convert coverage into identification. A more detailed analysis is provided in Appendix C.2.

5 Conclusion

We introduce MAESTRO, a principled framework for multi-agent collaboration that enables both divergent exploration and convergent synthesis. We also present CLPO, an RL method that achieves precise credit assignment through decision-focused optimization and comparative supervision. Together these components yield consistent improvements across diverse reasoning benchmarks and surpass state-of-the-art multi-agent methods. Looking ahead, we plan to investigate unified policy objectives that jointly optimize exploration and synthesis, and continuous learning paradigms that enable multi-agent collectives to refine collaboration dynamics through self-improvement over time.

REFERENCES

Sahar Abdelnabi, Amr Gomaa, Sarath Sivaprasad, Lea Schönherr, and Mario Fritz. Llm-deliberation: Evaluating llms with interactive multi-agent negotiation games. *arXiv preprint arXiv:2310.01444*, 2023.

Sahar Abdelnabi, Amr Gomaa, Sarath Sivaprasad, Lea Schönherr, and Mario Fritz. Cooperation, competition, and maliciousness: Llm-stakeholders interactive negotiation. *Advances in Neural Information Processing Systems*, 37:83548–83599, 2024.

Shayan Meshkat Alsadat and Zhe Xu. Multi-agent reinforcement learning in non-cooperative stochastic games using large language models. *IEEE Control Systems Letters*, 2024.

Sourav Banerjee, Ayushi Agarwal, and Saloni Singla. Llms will always hallucinate, and we need to live with this. In *Intelligent Systems Conference*, pp. 624–648. Springer, 2025.

Dennis R Brophy. Comparing the attributes, activities, and performance of divergent, convergent, and combination thinkers. *Creativity research journal*, 13(3-4):439–455, 2001.

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

Guangyao Chen, Siwei Dong, Yu Shu, Ge Zhang, Jaward Sesay, Börje F Karlsson, Jie Fu, and Yemin Shi. Autoagents: A framework for automatic agent generation. *arXiv preprint arXiv:2309.17288*, 2023a.

Junzhe Chen, Xuming Hu, Shuodi Liu, Shiyu Huang, Wei-Wei Tu, Zhaofeng He, and Lijie Wen. Llmarena: Assessing capabilities of large language models in dynamic multi-agent environments. *arXiv preprint arXiv:2402.16499*, 2024.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chen Qian, Chi-Min Chan, Yujia Qin, Yaxi Lu, Ruobing Xie, et al. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. *arXiv preprint arXiv:2308.10848*, 2(4):6, 2023b.

Yuheng Cheng, Ceyao Zhang, Zhengwen Zhang, Xiangrui Meng, Sirui Hong, Wenhao Li, Zihao Wang, Zekai Wang, Feng Yin, Junhua Zhao, et al. Exploring large language model based intelligent agents: Definitions, methods, and prospects. *arXiv preprint arXiv:2401.03428*, 2024.

Arthur Cropley. In praise of convergent thinking. *Creativity research journal*, 18(3):391–404, 2006.

Xiangxiang Dai, Yuejin Xie, Maoli Liu, Xuchuang Wang, Zhuohua Li, Huanyu Wang, and John Lui. Multi-agent conversational online learning for adaptive llm response identification. *arXiv preprint arXiv:2501.01849*, 2025.

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

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

Adrian Garret Gabriel, Alaa Alameer Ahmad, and Shankar Kumar Jeyakumar. Advancing agentic systems: Dynamic task decomposition, tool integration and evaluation using novel metrics and dataset. *arXiv preprint arXiv:2410.22457*, 2024.

Joy Paul Guilford. *The nature of human intelligence.* McGraw-Hill, 1967.

Ai Han, Junxing Hu, Pu Wei, Zhiqian Zhang, Yuhang Guo, Jiawei Lu, and Zicheng Zhang. Joyagents-r1: Joint evolution dynamics for versatile multi-llm agents with reinforcement learning. *arXiv preprint arXiv:2506.19846*, 2025.

Zhitao He, Pengfei Cao, Yubo Chen, Kang Liu, Ruopeng Li, Mengshu Sun, and Jun Zhao. Lego: A multi-agent collaborative framework with role-playing and iterative feedback for causality explanation generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9142–9163, 2023.

Zhitao He, Zijun Liu, Peng Li, May Fung, Ming Yan, Ji Zhang, Fei Huang, and Yang Liu. Enhancing language multi-agent learning with multi-agent credit re-assignment for interactive environment generalization. *arXiv preprint arXiv:2502.14496*, 2025.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. Metagpt: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*, 2023.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. Routerbench: A benchmark for multi-llm routing system. *arXiv preprint arXiv:2403.12031*, 2024a.

Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. *arXiv preprint arXiv:2408.08435*, 2024b.

Yuwei Hu, Runlin Lei, Xinyi Huang, Zhewei Wei, and Yongchao Liu. Scalable and accurate graph reasoning with llm-based multi-agents. *arXiv preprint arXiv:2410.05130*, 2024c.

Yoichi Ishibashi and Yoshimasa Nishimura. Self-organized agents: A llm multi-agent framework toward ultra large-scale code generation and optimization. *arXiv preprint arXiv:2404.02183*, 2024.

Ziqi Jia, Junjie Li, Xiaoyang Qu, and Jianzong Wang. Enhancing multi-agent systems via reinforcement learning with llm-based planner and graph-based policy. *arXiv preprint arXiv:2503.10049*, 2025.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. *arXiv preprint arXiv:2306.02561*, 2023.

Zhouyang Jiang, Bin Zhang, Airong Wei, and Zhiwei Xu. Qllm: Do we really need a mixing network for credit assignment in multi-agent reinforcement learning? *arXiv preprint arXiv:2504.12961*, 2025.

Dapeng Li, Hang Dong, Lu Wang, Bo Qiao, Si Qin, Qingwei Lin, Dongmei Zhang, Qi Zhang, Zhiwei Xu, Bin Zhang, et al. Verco: Learning coordinated verbal communication for multi-agent reinforcement learning. *arXiv preprint arXiv:2404.17780*, 2024a.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36:51991–52008, 2023.

Huao Li, Hossein Nourkhiz Mahjoub, Behdad Chalaki, Vaishnav Tadiparthi, Kwonjoon Lee, Ehsan Moradi Pari, Charles Lewis, and Katia Sycara. Language grounded multi-agent reinforcement learning with human-interpretable communication. *Advances in Neural Information Processing Systems*, 37:87908–87933, 2024b.

Wenhao Li, Dan Qiao, Baoxiang Wang, Xiangfeng Wang, Wei Yin, Hao Shen, Bo Jin, and Hongyuan Zha. Multi-agent credit assignment with pretrained language models. In *International Conference on Artificial Intelligence and Statistics*, pp. 1945–1953. PMLR, 2025.

Yuxi Li. Reinforcement learning applications. *arXiv preprint arXiv:1908.06973*, 2019.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*, 2023.

Muhan Lin, Shuyang Shi, Yue Guo, Vaishnav Tadiparthi, Behdad Chalaki, Ehsan Moradi Pari, Simon Stepputtis, Woojun Kim, Joseph Campbell, and Katia Sycara. Speaking the language of teamwork: Llm-guided credit assignment in multi-agent reinforcement learning. *arXiv preprint arXiv:2502.03723*, 2025.

Enze Liu, Gautam Akiwate, Mattijs Jonker, Ariana Mirian, Grant Ho, Geoffrey M Voelker, and Stefan Savage. Forward pass: On the security implications of email forwarding mechanism and policy. In *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*, pp. 373–391. IEEE, 2023a.

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

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. Dynamic llm-agent network: An llm-agent collaboration framework with agent team optimization. *arXiv preprint arXiv:2310.02170*, 2023b.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594, 2023.

Zhao Mandi, Shreeya Jain, and Shuran Song. Roco: Dialectic multi-robot collaboration with large language models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 286–299. IEEE, 2024.

Gabriel Mukobi, Hannah Erlebach, Niklas Lauffer, Lewis Hammond, Alan Chan, and Jesse Clifton. Welfare diplomacy: Benchmarking language model cooperation. *arXiv preprint arXiv:2310.08901*, 2023.

Xuefei Ning, Zinan Lin, Zixuan Zhou, Zifu Wang, Huazhong Yang, and Yu Wang. Skeleton-of-thought: Prompting llms for efficient parallel generation. *arXiv preprint arXiv:2307.15337*, 2023.

Bo Pan, Jiaying Lu, Ke Wang, Li Zheng, Zhen Wen, Yingchaojie Feng, Minfeng Zhu, and Wei Chen. Agentcoord: Visually exploring coordination strategy for llm-based multi-agent collaboration. *arXiv preprint arXiv:2404.11943*, 2024.

Chanwoo Park, Seungju Han, Xingzhi Guo, Asuman Ozdaglar, Kaiqing Zhang, and Joo-Kyung Kim. Maporl: Multi-agent post-co-training for collaborative large language models with reinforcement learning. *arXiv preprint arXiv:2502.18439*, 2025.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.

Pouya Pezeshkpour, Eser Kandogan, Nikita Bhutani, Sajjadur Rahman, Tom Mitchell, and Estevam Hruschka. Reasoning capacity in multi-agent systems: Limitations, challenges and human-centered solutions. *arXiv preprint arXiv:2402.01108*, 2024.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, et al. Chatdev: Communicative agents for software development. *arXiv preprint arXiv:2307.07924*, 2023.

Chen Qian, Zihao Xie, Yifei Wang, Wei Liu, Kunlun Zhu, Hanchen Xia, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, et al. Scaling large language model-based multi-agent collaboration. *arXiv preprint arXiv:2406.07155*, 2024.

Shuofei Qiao, Ningyu Zhang, Runnan Fang, Yujie Luo, Wangchunshu Zhou, Yuchen Eleanor Jiang, Chengfei Lv, and Huajun Chen. Autoact: Automatic agent learning from scratch for qa via self-planning. *arXiv preprint arXiv:2401.05268*, 2024.

Mark A Runco and Ivonne Chand. Cognition and creativity. *Educational psychology review*, 7(3): 243–267, 1995.

Yu Shang, Yu Li, Keyu Zhao, Likai Ma, Jiahe Liu, Fengli Xu, and Yong Li. Agentsquare: Automatic llm agent search in modular design space. *arXiv preprint arXiv:2410.06153*, 2024.

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

Robert J Sternberg and Todd I Lubart. An investment theory of creativity and its development. *Human development*, 34(1):1–31, 1991.

Mirac Suzgun and Adam Tauman Kalai. Meta-prompting: Enhancing language models with task-agnostic scaffolding. *arXiv preprint arXiv:2401.12954*, 2024.

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

Qwen Team. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

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

Shenzhi Wang, Chang Liu, Zilong Zheng, Siyuan Qi, Shuo Chen, Qisen Yang, Andrew Zhao, Chaofei Wang, Shiji Song, and Gao Huang. Avalon's game of thoughts: Battle against deception through recursive contemplation. *arXiv preprint arXiv:2310.01320*, 2023.

Xuejiao Wang, Guoqing Zhi, Zhihao Tang, Hao Jin, Qianyue Zhang, and Nan Li. Self-aware intelligent medical rescue unmanned team via large language model and multi-agent reinforcement learning. In *Proceedings of the 2024 International Symposium on AI and Cybersecurity*, pp. 119–124, 2024.

Yuan Wei, Xiaohan Shan, and Jianmin Li. Lero: Llm-driven evolutionary framework with hybrid rewards and enhanced observation for multi-agent reinforcement learning. *arXiv preprint arXiv:2503.21807*, 2025.

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

Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th international conference on Machine learning*, pp. 1192–1199, 2008.

Kai Xiong, Xiao Ding, Yixin Cao, Ting Liu, and Bing Qin. Examining inter-consistency of large language models collaboration: An in-depth analysis via debate. *arXiv preprint arXiv:2305.11595*, 2023.

Yanggang Xu, Weijie Hong, Jirong Zha, Geng Chen, Jianfeng Zheng, Chen-Chun Hsia, and Xinlei Chen. Scalable uav multi-hop networking via multi-agent reinforcement learning with large language models. *arXiv preprint arXiv:2505.08448*, 2025.

Bingyu Yan, Zhibo Zhou, Litian Zhang, Lian Zhang, Ziyi Zhou, Dezhuang Miao, Zhoujun Li, Chaozhuo Li, and Xiaoming Zhang. Beyond self-talk: A communication-centric survey of llm-based multi-agent systems. *arXiv preprint arXiv:2502.14321*, 2025.

Wei Yang and Jesse Thomason. Learning to deliberate: Meta-policy collaboration for agentic llms with multi-agent reinforcement learning. *arXiv preprint arXiv:2509.03817*, 2025.

Yingxuan Yang, Huacan Chai, Shuai Shao, Yuanyi Song, Siyuan Qi, Renting Rui, and Weinan Zhang. Agentnet: Decentralized evolutionary coordination for llm-based multi-agent systems. *arXiv preprint arXiv:2504.00587*, 2025.

Zhangyue Yin, Qiushi Sun, Cheng Chang, Qipeng Guo, Junqi Dai, Xuanjing Huang, and Xipeng Qiu. Exchange-of-thought: Enhancing large language model capabilities through cross-model communication. *arXiv preprint arXiv:2312.01823*, 2023.

Yanwei Yue, Guibin Zhang, Boyang Liu, Guancheng Wan, Kun Wang, Dawei Cheng, and Yiyan Qi. Masrouter: Learning to route llms for multi-agent systems. *arXiv preprint arXiv:2502.11133*, 2025.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D Goodman. Quiet-star: Language models can teach themselves to think before speaking. *arXiv preprint arXiv:2403.09629*, 2024.

Ceyao Zhang, Kaijie Yang, Siyi Hu, Zihao Wang, Guanghe Li, Yihang Sun, Cheng Zhang, Zhaowei Zhang, Anji Liu, Song-Chun Zhu, et al. Proagent: building proactive cooperative agents with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17591–17599, 2024a.

Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. Cut the crap: An economical communication pipeline for llm-based multi-agent systems. *arXiv preprint arXiv:2410.02506*, 2024b.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, et al. Aflow: Automating agentic workflow generation. *arXiv preprint arXiv:2410.10762*, 2024c.

Jintian Zhang, Xin Xu, Ningyu Zhang, Ruibo Liu, Bryan Hooi, and Shumin Deng. Exploring collaboration mechanisms for llm agents: A social psychology view. *arXiv preprint arXiv:2310.02124*, 2023.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of reinforcement learning and control*, pp. 321–384, 2021.

Natalia Zhang, Xinqi Wang, Qiwen Cui, Runlong Zhou, Sham M. Kakade, and Simon Shaolei Du. Multi-agent reinforcement learning from human feedback: Data coverage and algorithmic techniques. *arXiv preprint arXiv:2406.11896*, 2024d.

Weitao Zhang, Zsuzsika Sjoerds, and Bernhard Hommel. Metacontrol of human creativity: The neurocognitive mechanisms of convergent and divergent thinking. *NeuroImage*, 210:116572, 2020.

Yang Zhang, Shixin Yang, Chenjia Bai, Fei Wu, Xiu Li, Zhen Wang, and Xuelong Li. Towards efficient llm grounding for embodied multi-agent collaboration. *arXiv preprint arXiv:2405.14314*, 2024e.

Heng Zhou, Hejia Geng, Xiangyuan Xue, Li Kang, Yiran Qin, Zhiyong Wang, Zhenfei Yin, and Lei Bai. Reso: A reward-driven self-organizing llm-based multi-agent system for reasoning tasks. *arXiv preprint arXiv:2503.02390*, 2025.

Guobin Zhu, Rui Zhou, Wenkang Ji, and Shiyu Zhao. Lamarl: Llm-aided multi-agent reinforcement learning for cooperative policy generation. *IEEE Robotics and Automation Letters*, 2025.

Yuan Zhuang, Yi Shen, Zhili Zhang, Yuxiao Chen, and Fei Miao. Yolo-marl: You only llm once for multi-agent reinforcement learning. *arXiv preprint arXiv:2410.03997*, 2024.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. Gptswarm: Language agents as optimizable graphs. In *Forty-first International Conference on Machine Learning*, 2024.

A A Comprehensive Review of Related Work

A.1 Multi-Agent LLM Collaboration

Single LLM agents, despite their impressive individual capabilities, face fundamental limitations in context length, sequential generation, and breadth of expertise. These constraints hinder performance on tasks that demand parallel information processing, complementary skill sets, and the synthesis of diverse perspectives (Gabriel et al., 2024; Liang et al., 2023; Xiong et al., 2023; Yin et al., 2023; Zhang et al., 2023). To overcome these bottlenecks, researchers have increasingly turned to multi-agent systems (MAS), where collectives of LLM-powered agents coordinate to realize forms of collective intelligence in domains such as software engineering, complex planning, and scientific discovery (Hong et al., 2023; Chen et al., 2023b; Jiang et al., 2023; Ning et al., 2023; Qiao et al., 2024; Pan et al., 2024; Suzgun & Kalai, 2024; Chen et al., 2023a; Ishibashi & Nishimura, 2024).

Early approaches largely follow a *prompt-based paradigm*, where roles, protocols, and workflows are specified by hand. Debate-style and critique frameworks (Du et al., 2023; Chan et al., 2023; Chen et al., 2024; Mukobi et al., 2023; Wang et al., 2023; Abdelnabi et al., 2024) as well as corporate-style pipelines such as MetaGPT (Hong et al., 2023; Qian et al., 2023) exemplify this direction. These systems demonstrate the promise of structured collaboration but remain brittle because their strategies are statically prescribed and cannot adapt or learn from experience (Jiang et al., 2023; Liang et al., 2023; He et al., 2023).

Beyond static prompting, recent work introduces more principled coordination schemes. *Prestructured paradigms* adopt fixed interaction topologies, such as chains, trees, or graphs, to organize communication and enforce critique (Du et al., 2023; Liu et al., 2024; Qian et al., 2024). In parallel, *self-organizing paradigms* dynamically adapt collaboration graphs during inference using search, pruning, or routing methods, as seen in DyLAN, MasRouter, GPTSwarm, and AFLOW (Liu et al., 2023b; Hu et al., 2024b; Shang et al., 2024; Zhang et al., 2024b; Zhuge et al., 2024; Zhang et al., 2024c; Hu et al., 2024a; Yue et al., 2025). These frameworks improve efficiency and flexibility, yet they often reduce coordination to architectural wiring and lack mechanisms for fine-grained credit assignment.

Complementary efforts focus on *role specialization* and *organizational analogies*, where agents are differentiated as planners, solvers, or verifiers, or even structured as corporate roles such as CEO and engineer (Hong et al., 2023; Li et al., 2023; Mandi et al., 2024; Talebirad & Nadiri, 2023; Du et al., 2023; Chen et al., 2023b). Communication protocols vary between centralized, decentralized, and hierarchical settings, as well as synchronous versus asynchronous exchanges (Jiang et al., 2023; Ning et al., 2023; Pan et al., 2024; Liang et al., 2023; Zhang et al., 2023; Du et al., 2023; Chan et al., 2023; Chen et al., 2024). These design choices trade off scalability, robustness, and overhead but leave unresolved the fundamental question of how to separate decision-making from justification in a principled manner.

Overall, existing MAS paradigms illuminate diverse strategies for orchestrating collaboration, but they remain limited in their ability to balance broad exploration with reliable convergence and to assign credit cleanly across agents and rationales.

A.2 Reinforcement Learning for Multi-Agent LLMs.

A central trend in multi-agent LLM research is to move beyond static prompt engineering toward learning from interaction. Early work explored supervised fine-tuning (SFT) on expert demonstrations, which injects cooperative behaviors by imitation but is limited in adaptability to unseen coordination settings (Madaan et al., 2023; Zelikman et al., 2024). In contrast, reinforcement learning (RL) supplies a reward-driven mechanism that allows agents to refine strategies from experience and discover emergent collaboration patterns (Zhu et al., 2025; Zhuang et al., 2024). In practice, SFT often initializes base policies, while multi-agent reinforcement learning (MARL) further tailors them under task feedback (Zhu et al., 2025; Li, 2019; Zhang et al., 2021).

Recent efforts fall into three complementary directions. First, some approaches compile language into structured controllers before learning, such as translating dialogue into plans, graphs, or code, which grounds RL optimization in compact symbolic spaces (Zhuang et al., 2024; Jia et al., 2025). Second, others focus on adaptive collaboration online, dynamically refining task decomposition,

agent assignment, or communication routing through RL signals (Zhou et al., 2025; Wang et al., 2024; Xu et al., 2025; Li et al., 2024a). Third, direct policy optimization for reasoning behaviors has gained traction, with GRPO- and PPO-style updates applied to cooperative justification and answer selection, often combined with tool use or human feedback (Wan et al., 2025; Park et al., 2025; Han et al., 2025). Across these directions, RL provides the flexibility to align multi-agent dynamics with task objectives rather than relying solely on fixed prompts or wiring rules.

At the same time, this line of work highlights several core challenges. A prominent difficulty is credit assignment: linguistic outputs entangle the correctness of discrete decisions with the plausibility of accompanying rationales, making it unclear what aspect of behavior is being rewarded (Wei et al., 2025; Jiang et al., 2025). Another challenge is efficient exploration in vast language action spaces, where agents may generate superficially diverse but semantically redundant outputs (Liu et al., 2023a; Zhang et al., 2024d). Finally, there is the issue of alignment of emergent behaviors, since collaboration can amplify biases or drift without proper reward shaping (Alsadat & Xu, 2024; Lin et al., 2025).

Our work follows this trajectory while placing a sharper emphasis on the convergence step of collaboration. Rather than treating group outcomes as a monolithic reward signal, we recast convergence as a structured optimization problem that separates the supervision of rationales from decision signals. This perspective motivates the design of a new RL objective that provides comparative supervision across rationales while preserving clean decision gradients, complementing existing GRPO-style multi-agent optimization.

B Derivation of the Cumulative Reliability Inequality

In this section we provide a derivation of the cumulative reliability inequality from Section 3.2.

We start by defining the history (i.e., filtration) $\mathcal{F}_t := (q, \theta_{1:t}, \zeta_{1:t})$, with $\mathcal{F}_0 := q$, where $\theta_{1:t}$ denotes the randomness of the execution agents for the first t rounds, and $\zeta_{1:t}$ denotes the randomness of the central agent for the first t rounds. Let $\mathsf{Cand}_t := \{\exists\,(i,k) \in [N] \times [K] \text{ s.t. } E(c_{t,k}^{(i)}) \text{ holds}\}$ denote the event the round t slate $\mathcal{C}_t$ contains at least one correct candidate. Let $(i_t, k_t) \sim \pi_\theta(\cdot \mid q, \mathcal{C}_t)$ be the central decision made at time t. Then we have that, assuming $p_t \geq \underline{p}$ and $q_t \geq \underline{q}$ for non-random $\underline{p}, \underline{q}$ almost surely for all t:

$$
\begin{aligned}
h_t &:= \Pr(\text{Success}_t \mid \mathcal{F}_{t-1}) \\
&= \mathbb{E}_{\mathcal{C}_t|\mathcal{F}_{t-1}}[\Pr(\text{Success}_t \mid \mathcal{F}_{t-1}, \mathcal{C}_t)] \\
&\overset{(a)}{=} \mathbb{E}_{\mathcal{C}_t|\mathcal{F}_{t-1}}[\mathbf{1}\{\mathsf{Cand}_t\} \Pr((i_t, k_t) \in S_t \mid q, \mathcal{C}_t, \{|S_t| \geq 1\})] \\
&\overset{(b)}{=} \mathbb{E}_{\mathcal{C}_t|\mathcal{F}_{t-1}}[\mathbf{1}\{\mathsf{Cand}_t\} q_t] \\
&\overset{(c)}{\geq} \mathbb{E}_{\mathcal{C}_t|\mathcal{F}_{t-1}}[\mathbf{1}\{\mathsf{Cand}_t\}] \underline{q} \\
&= \Pr(\mathsf{Cand}_t \mid \mathcal{F}_{t-1}) \underline{q} \\
&\overset{(d)}{=} \Pr(\mathsf{Cand}_t \mid q, s_t^{(1:N)}) \underline{q} \\
&\overset{(e)}{=} p_t \underline{q} \\
&\overset{(f)}{\geq} \underline{p}\underline{q},
\end{aligned}
$$

where in (a) we used the fact that the decision (i_t, k_t) is generated conditioned only on $(q, \mathcal{C}_t)$, (b) is the definition of q_t from (4), (c) uses our lower bound assumption on q_t, (d) uses the fact that the candidate decisions $\mathcal{C}_t$ are generated conditioned on $(q, z_{t-1}^{(1:N)}, b_{t-1})$, which is contained within $\mathcal{F}_{t-1}$, (e) is the definition of p_t from (2), and (f) uses our lower bound assumption on p_t.

Now, let $X_t := \mathbf{1}\{\text{Success}_t\}$. By definition we have that X_t is $\mathcal{F}_t$-measurable. Hence by repeated applications of the tower-property of conditional expectations,

$$\begin{aligned}
\Pr(\text{Fail all } R \text{ rounds}) &= \mathbb{E}\left[\prod_{t=1}^{R}(1-X_t)\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\prod_{t=1}^{R}(1-X_t) \mid \mathcal{F}_{R-1}\right]\right] \\
&= \mathbb{E}\left[\prod_{t=1}^{R-1}(1-X_t)\mathbb{E}\left[1-X_R \mid \mathcal{F}_{R-1}\right]\right] \\
&= \mathbb{E}\left[\prod_{t=1}^{R-1}(1-X_t)(1-h_R)\right] \\
&\overset{(a)}{\leq} \mathbb{E}\left[\prod_{t=1}^{R-1}(1-X_t)\right](1-\underline{pq}) \leq \cdots \leq (1-\underline{pq})^R,
\end{aligned}$$

where (a) follows from above where we established $h_t \geq \underline{pq}$.

C Experiment

C.1 Experimental Settings

Datasets & Benchmarks. We evaluate the framework on three task families designed to stress complementary aspects of collective reasoning, namely precise numeric inference, broad factual and analytical judgment, and executable synthesis. This spectrum assesses both the "diverge" capacity (hypothesis coverage) and the "converge" capacity (principled selection).

Mathematical reasoning. We use **GSM8K**, **MATH**, **AIME**, and **AMC**. GSM8K comprises grade-school word problems with single numeric targets; MATH covers competition-level problems across algebra, number theory, geometry, and combinatorics; AIME consists of short-answer olympiad items with integer solutions; AMC includes large-scale contest questions (we report on the standard subset with unambiguous numeric targets). Performance is measured by *Solve Rate*, the proportion of items whose predicted answer exactly matches the ground truth under benchmark normalization rules.

General reasoning. We use **MMLU**, spanning 57 subjects from STEM to humanities under a four-choice multiple-choice format. Performance is reported as *Accuracy*, i.e., the fraction of correctly selected options, under the benchmark's standard few-shot setting.

Code generation. We use **HumanEval**, where models synthesize functions from natural-language specifications. Performance is reported as *Pass@1*, the percentage of prompts for which the single generated solution passes all hidden unit tests.

Unless otherwise noted, we follow official splits and prompting guidelines, do not use external tools or retrieval, and keep evaluation deterministic for single predictions. When stochastic sampling is required (e.g., for self-consistency or multi-agent generation), we fix seeds and average over repeated runs; confidence intervals are reported in the appendix. This protocol ensures comparability with prior work while isolating the contribution of the collaboration paradigm and training objective.

Baselines. We compare against collaborative LLM methods organized by their underlying *collaboration mechanism*, rather than model brand. (i) *Single-agent reasoning*: **Vanilla** (direct decoding), **CoT** (chain-of-thought prompting), and **SC** (self-consistency with majority vote). (ii) *Peer interaction*: **LLM-Debate** (multi-round argumentation with shared transcripts), **GroupDebate** (multi-agent debate with voting-based aggregation) and **PHP** (pairwise critique without a global selector). (iii) *Routing/topology control*: **DyLAN** (layered agent network with pruning and early-stop consensus). (iv) *Workflow/graph search*: **GPTSwarm** (optimization of reasoning graphs over multiple prompting strategies) and **AFLOW** (Monte-Carlo search over reusable operators). (v) *Communication efficiency*: **AgentPrune** (sparse message passing to reduce cost while maintaining accuracy).

For all baselines we use the same base models, adopt each method's official prompts and stopping criteria, and match collaboration budgets (rounds, agents, and generations). When methods output multiple candidates, we apply their canonical aggregation (e.g., majority vote or ranker). This taxonomy clarifies whether improvements come from stronger generation (divergence), more reliable selection (convergence), or better workflow, providing a diagnostic comparison to our exploration–synthesis paradigm.

Prompt Templates. To make our experimental setup transparent and reproducible, we explicitly document the instruction prompts used by different agents in our framework. These templates capture the roles and responsibilities of both reasoning agents and the center arbiter, highlighting how they collaborate through structured interaction. For clarity, we present concrete examples in the domain of mathematical reasoning problems, which serve as a representative case for illustrating the prompt design.

Execution Agent Prompt (Initial Round)

You are Reasoning Agent #{agent_id}. Your task is to carefully solve the given math problem step by step. Clearly show your reasoning process, making sure that each transformation is logically valid. Avoid skipping important intermediate steps.
At the end of your reasoning, provide the final numeric answer in the exact format: `\boxed{...}`.
Problem: {{math_question}}

Execution Agent Prompt (Interactive Round)

You are Reasoning Agent #{agent_id}. You previously proposed multiple solutions and now also receive the Center Arbiter's synthesis.
Re-evaluate the problem carefully, considering both your earlier solutions and the Arbiter's feedback. Generate refined solutions that correct any mistakes if needed, ensuring logical consistency.
Each output must end with the final numeric answer in the exact format: `\boxed{...}`.
Problem: {{math_question}}
Your Previous Solutions: {...}
Center Arbiter's Feedback: {...}

Central Agent Prompt

You are the Center Arbiter, responsible for evaluating candidate solutions proposed by agents. Carefully read the original problem and all candidate solutions. Compare their reasoning, detect mistakes if present, and identify the most reliable candidate.
Then, following the strict format below, provide a short justification, the chosen candidate index, and the final numeric answer in `\boxed{...}`.
Problem: {{math_question}}
Candidates:
- Candidate 1: {...}
- Candidate 2: {...}
- Candidate 3: {...}

STRICT OUTPUT FORMAT:
Reason: {detailed justification}
Chosen: {candidate_id}
Final: `\boxed{...}`

Implementation Details. Our experiments are conducted with a compact configuration where three agents interact across three communication rounds for each query. The agents are instantiated from widely used instruction-tuned models including **Llama-3.1-8B-Instruct**, **Llama-3.2-3B-**

Model	LLaMA-8B (GSM8K)			Qwen-7B (GSM8K)		
	Coverage	Identification	ACC	Coverage	Identification	ACC
MAESTRO	0.9757	0.8919	0.8702	0.9886	0.9519	0.9410
w/ CLPO	0.9773	0.9141	0.8933	0.9901	0.9607	0.9512

Table 4: Comparison of coverage, identification, and accuracy (ACC) on GSM8K under two backbones.

Instruct (Dubey et al., 2024), and **Qwen2.5-7B-Instruct** as well as **Qwen2.5-3B-Instruct** (Team, 2024). All models are accessed through the HuggingFace Transformers library with 8-bit quantization to reduce GPU memory usage. We enable KV caching throughout the experiments to improve generation efficiency. We adopt a unified generation setup across all experiments. Unless otherwise noted, nucleus sampling with $p = 0.95$ is used and the maximum output length is set to 512 tokens. The default temperature is 0.7, which balances diversity and stability. For tasks requiring deterministic evaluation, such as pairwise preference comparisons or revision prompts, we reduce the temperature to 0.3. The central agent is always assigned a temperature of 0.0 to enforce deterministic decisions and avoid stochastic drift. To ensure comparability across methods, all models share the same decoding settings and random seeds are fixed. This setup follows common practice in LLM evaluation and ensures that performance differences stem from the collaboration paradigm rather than decoding hyperparameters. During both supervised fine-tuning (SFT) and policy optimization, we adopt parameter-efficient fine-tuning using LoRA. Unless otherwise noted, the LoRA rank is set to 16, with scaling factor $\alpha = 32$ and dropout 0.05. Only LoRA parameters, LayerNorm statistics, and bias terms are updated, while all other weights remain frozen. Training uses Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$) with an initial learning rate of 5×10^{-5}, decayed following a cosine schedule. Gradient norms are clipped at 1.0 to stabilize optimization. We train with a global batch size of 256 distributed across four A100 GPUs (80GB each), using mixed precision (bfloat16) for efficiency. The rank-loss coefficient is tuned over $\{0.1, 0.5, 0.8, 1.0\}$. To encourage exploration we add an entropy bonus of 0.01, while maintaining consistency with the SFT reference policy via a KL regularization weight of 0.1. Each run proceeds for three epochs, and results are averaged over three random seeds (25, 42, and 99) to ensure robustness and mitigate variance.

Unified Reward. We define unified reward that combines answer correctness and reason quality. For each candidate c_i, the correctness score is $\mathrm{acc}_i \in [0, 1]$, defined as $\mathrm{acc}_i = \mathbf{1}[\mathrm{answer}_i = \hat{a}]$ for objective/numeric tasks (with an optional tolerance) or as a test-pass rate for programming tasks. Reason quality is computed from the candidate's rationale (and its own final answer/interface statement) via a unified set of binary attributes: structure/readability (stepwise clarity, explicit intermediate quantities), soundness to own answer (derivation strictly leads to its own final answer without contradictions), constraint/format adherence (range, integrality, lowest terms, function signature, etc.), premise/evidence alignment (key facts in the rationale match the prompt or given materials), error diagnosis/refutation (identifies common failure modes or flaws in competing candidates and explains their mechanism), and executability/safety (for implementation tasks, the rationale is consistent with runnability/safety without unjustified risky operations). Each attribute is recognized jointly by programmatic checks (rules, AST/signature validation, equation re-evaluation, range/unit checks, keyword/pattern matching) and a GPT judge that performs semantic recognition over the question and candidate and outputs $\{\text{Yes}, \text{No}\}$. Denoting the decision on attribute k as $d_k(c_i) \in \{1, 0\}$, we average only determined attributes to obtain a single rationale score $s_i = \frac{1}{|\mathcal{V}_i|} \sum_{k \in \mathcal{V}_i} d_k(c_i), \mathcal{V}_i = \{\, k \mid d_k(c_i) \in \{0, 1\} \,\}$. The training reward is a simple weighted fusion, $R_i = w_c\, \mathrm{acc}_i + w_r\, s_i, \quad w_c, w_r \geq 0, \;\; w_c + w_r = 1$, with the practical choice $w_c \geq w_r$ to discourage "fluent but wrong" rationales. Considering the trade-off between prioritizing correctness and still learning discriminative rationales, we adopt the simple default weights $w_c = 0.6$ and $w_r = 0.4$. And we employ gpt-4o as the semantic judge for attribute recognition.

C.2 Experimental Results.

Number of Rounds: balancing evidence aggregation and bias amplification. Figure 6 shows that increasing the number of collaboration rounds improves performance at first, then saturates and may decline. GSM8K peaks around two rounds and AMC around three rounds. This pat-

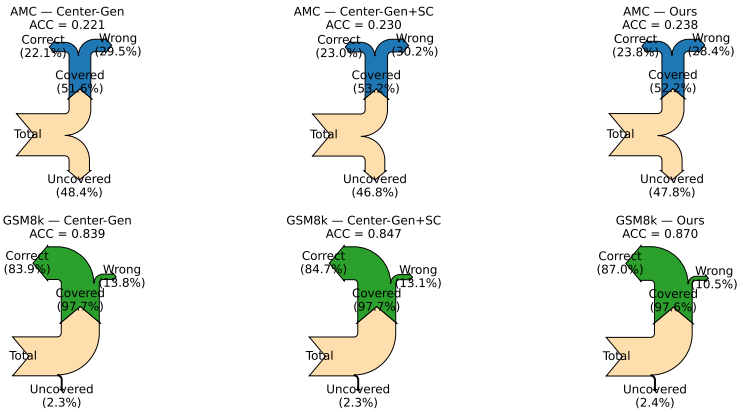

Figure 5: Sankey diagram illustrating performance on AMC and GSM8K under different central coordination strategies. Each flow decomposes accuracy into coverage and identification outcomes, showing how centralized selection more effectively converts diverse reasoning into correct solutions.

tern matches our coverage–identification decomposition. The first additional round injects public evidence through broadcast, which focuses subsequent exploration and lifts the identification probability q_t because the central policy compares candidates under a clearer hypothesis space. Coverage p_t changes little once the initial candidate pool is large, so early gains are mainly due to improved identification. Beyond the peak, returns diminish and can turn negative. Repeated conditioning on previous broadcasts reduces effective diversity and increases redundancy, which lowers the probability that new rounds add genuinely novel evidence. If an early broadcast is confidently wrong, later rounds tend to herd toward the same error, creating bias amplification that hurts q_t. More rounds also introduce additional stochastic variance while consuming budget, which further limits net gains. Overall, a small number of rounds is most effective: two rounds on GSM8K and two to three rounds on AMC strike a good balance by converting collective coverage into reliable identification without over-conditioning the agents.

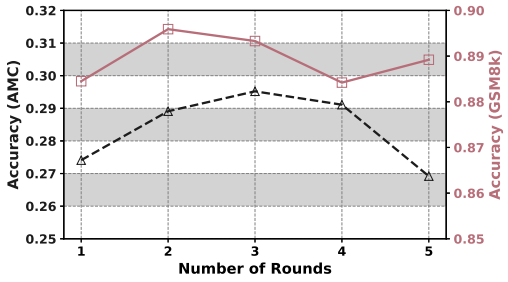

Figure 6: Effect of collaboration rounds on performance. Accuracy is reported for AMC and GSM8K. Performance improves with additional rounds up to a moderate level, then saturates or declines, highlighting the trade-off between evidence aggregation and bias amplification.

Sampling Depth per Agent: Coverage–Variance Trade-off. We analyze how the number of samples drawn by each agent (K) affects collaborative performance. As shown in Figure 4 (b), increasing K from 2 to 3 improves accuracy on both AMC (from 0.2369 to 0.2852) and GSM8K (from 0.8853 to 0.8933). Beyond this point the gains saturate: GSM8K changes marginally at K=4 (0.8936) and declines at K=5 (0.8889), while AMC peaks at K=3 and then drops to 0.2811 and 0.2690. This pattern reflects our exploration–synthesis decomposition. Increasing K initially raises the chance that at least one candidate is correct, which improves coverage p_t. However, multi-sampling from the *same* agent policy quickly becomes correlated and redundant, so the marginal gain in coverage diminishes. At the same time the candidate set grows and introduces more plausible distractors, which elevates the burden on the central selector and can depress identification q_t. In

practice, deeper per-agent sampling also inflates within-round variance because it relies on stochastic decoding from a single policy instance rather than diversifying across agents. Consequently, once coverage is near saturation, additional K contributes more noise than signal and identification becomes the limiting factor. Taken together with the agent-scaling results, these observations suggest a practical guideline: for a fixed budget, allocate capacity to increasing the *number of agents* to diversify hypotheses, and keep K modest (three to four at most) so that the central synthesis can reliably convert collective coverage into higher identification.

D Declaration on the Use of Large Language Models

In the preparation of this work, the authors used GPT-5 and GPT-4o for two specific purposes. First, GPT-5 was employed to polish the writing, improve clarity, and ensure grammatical correctness throughout the manuscript. Second, GPT-4o was used during dataset construction to assist in evaluating the quality of reasoning annotations. After using these tools, the authors reviewed and edited all content as needed and take full responsibility for the final version of the publication.