Adaptive Collaboration with Humans: Metacognitive Policy Optimization for Multi-Agent LLMs with Continual Learning

Anonymous authors
Paper under double-blind review

Abstract

While scaling individual Large Language Models (LLMs) has delivered remarkable progress, the next frontier lies in scaling collaboration through multi-agent systems (MAS). However, purely autonomous MAS remain "closed-world" systems, constrained by the static knowledge horizon of pre-trained models. This limitation makes them brittle on tasks requiring knowledge beyond training data, often leading to collective failure under novel challenges. To address this, we propose the **Learning to Intervene via Metacognitive Adaptation (LIMA)** framework, a principled paradigm for human–agent collaboration. LIMA trains agents to learn a metacognitive policy that governs when to solve problems autonomously and when to defer to a human expert. To operationalize this policy, we introduce **Dual-Loop Policy Optimization**, which disentangles immediate decision-making from long-term capability growth. The inner loop applies Group Relative Policy Optimization (GRPO) with a cost-aware reward to optimize deferral decisions, while the outer loop implements continual learning, transforming expert feedback into high-quality supervised signals that strengthen the agent's reasoning ability. Experiments on challenging mathematical and problem-solving benchmarks show that LIMA, equipped with Dual-Loop Policy Optimization, consistently outperforms state-of-the-art MAS, establishing a principled foundation for collaborative and continually improving agentic systems.

1 Introduction

While scaling individual Large Language Models (LLMs) has produced remarkable progress, the next frontier lies in scaling collaboration through *multi-agent systems* (MAS) (Hong et al., 2023; Chen et al., 2023b; Jiang et al., 2023; Ning et al., 2023; Han et al., 2025; Wang et al., 2025a). By coordinating multiple agents to tackle problems beyond the reach of any single model, this paradigm has inspired a wave of innovations from structured debates to dynamic workflow optimization (Zhang et al., 2024a; Qiao et al., 2024; Han et al., 2025). Yet these systems face an inherent ceiling: no matter how sophisticated their interaction protocols, purely autonomous agents remain fundamentally **closed-world**. Their knowledge horizon is bounded by pre-training corpora (Wang et al., 2023b; Srivatsa et al., 2024; Du et al., 2023; Liu et al., 2024). While they can recombine existing information, they cannot generate new knowledge or adapt to unseen contexts. This creates vulnerabilities when tasks demand real-time information, domain-specific expertise, or reasoning patterns absent from training (Zhang et al., 2024d; Chen et al., 2025). In such cases, internal collaboration alone cannot bridge the gap, often leading to collective failure. To break this ceiling and enable open-ended intelligence, a new paradigm is needed. We argue that the most principled path is to integrate **external human expertise**, transforming closed systems into adaptive frameworks capable of continual learning and growth (Sun et al., 2025; Zou et al.).

Within this closed-world paradigm, research has followed two main directions. The first emphasizes optimizing **autonomous collaboration** through increasingly sophisticated interaction protocols. Frameworks based on structured debate (Chan et al., 2023; Liu et al., 2024), topology control (Ong et al., 2024; Chen et al., 2024b), and workflow graph optimization (Zhang et al., 2024b; Li et al., 2025) have demonstrated notable improvements in refining and recombining agents' internal knowledge. However, these methods largely engage in *collective introspection* (Zhang et al., 2024e; Chen et al., 2024a), maximizing the use of existing information without extending beyond the aggregate knowledge boundary. They act as powerful integrators, but not true learners capable of acquiring genuinely new capabilities. Recognizing this intrinsic limitation, a second line of work has sought to incorporate **human expertise** (Takerngsaksiri et al., 2025; Mozannar et al., 2025). Many human-in-the-loop systems (Liu et al., 2023; Pandya et al., 2024) treat humans primarily as passive oracles or supervisors for sub-tasks. This leaves two critical questions unresolved: *when* to defer to the expert, often reduced to heuristics such as low-confidence thresholds rather than learned policies (Kenton et al., 2024; Li et al., 2024b); and *how* to learn from human input, which is typically applied as a one-time fix rather than as a catalyst for long-term capability growth (Mu et al., 2024; Wang et al., 2025b). Importantly, human intervention holds the potential to operate at multiple levels, offering both localized corrections to specific reasoning errors and broader adjustments that reshape the overall collaborative process (Triem & Ding, 2024; Grondin et al., 2025).

This analysis highlights that the key challenge is not whether agents can interact with humans, but whether they can do so intelligently and strategically. Addressing this requires a **metacognitive policy**, a high-level strategy for reasoning about both self-competence and peer competence to guide collaboration. Such a policy must solve two intertwined problems: **when to ask**, which demands moving beyond heuristics to model uncertainty and balance the risk of failure against the cost of intervention; and **how to grow**, which requires mechanisms that turn expert feedback into lasting capability improvements rather than one-time fixes. A paradigm that unifies these elements is essential for building open and continually evolving agentic systems.

To address these challenges, we propose the **Learning to Intervene via Metacognitive Adaptation (LIMA)** framework, a principled paradigm for human–agent collaboration. The key contribution of LIMA is not the mere inclusion of a human in the loop, but the endowment of agents with a sophisticated metacognitive policy that governs when and how to engage with external expertise. LIMA operationalizes this paradigm through three coordinated components: (*i*) **Autonomous Operation**, where agents attempt problem-solving using their evolving capabilities; (*ii*) **Metacognitive Assessment**, where agents evaluate confidence and task difficulty to identify their knowledge boundaries; and (*iii*) **Strategic Deferral**, where human expertise is leveraged as a targeted intervention rather than as a passive oracle. Developing this metacognitive policy requires a dedicated optimization strategy. We therefore introduce **Dual-Loop Policy Optimization (DLPO)**, a reinforcement learning methodology that separates short-term decision-making from long-term capability growth. The inner loop employs Group Relative Policy Optimization (GRPO) with a cost-aware reward to refine the agent's deferral behavior in real time. The outer loop implements continual learning by transforming expert feedback from deferral events into high-quality supervised samples, thereby improving the agent's underlying reasoning ability. Together, LIMA and DLPO move beyond static supervision, enabling agents to learn both *when* to seek guidance and *how* to grow from it.

In summary, the main contributions of this paper are as follows:

- We propose the **Learning to Intervene via Metacognitive Adaptation (LIMA)** framework, a paradigm for human–agent collaboration that equips agents with a metacognitive policy to decide when to strategically defer to human expertise.
- We introduce **Dual-Loop Policy Optimization (DLPO)**, a training methodology that separates short-term deferral decisions from long-term capability growth. The inner loop employs GRPO with a cost-aware reward, while the outer loop leverages expert feedback as supervised signals for continual learning.
- Extensive experiments on mathematical reasoning and general problem-solving benchmarks demonstrate that LIMA with DLPO outperforms both autonomous multi-agent systems, establishing a robust foundation for continually improving agentic collaboration.

2 Related Work

Large language models (LLMs) acting alone are limited by context length, sequential generation, and restricted skill coverage, which constrains their ability to solve complex reasoning tasks (Gabriel et al., 2024; Liang et al., 2023; Xiong et al., 2023; Yin et al., 2023; Zhang et al., 2023). To mitigate these issues, **multi-agent systems** (MAS) have been widely explored, where multiple LLMs are organized into collaborative structures for collective problem solving (Hong et al., 2023; Chen et al., 2023b; Jiang et al., 2023; Qiao et al., 2024; Pan et al., 2024). Early efforts relied on prompt-based paradigms that assign predefined roles or workflows, enabling debate, critique, or corporate-style pipelines (Du et al., 2023; Chan et al., 2023; Wang et al., 2023a; Han et al., 2025). While effective, these designs lack adaptability since their interaction protocols are fixed and cannot evolve through experience. More recent work moves toward structured coordination and adaptive communication. Predefined schemes employ debate or peer-review across chains, trees, or graphs (Liu et al., 2024; Qian et al., 2024), while adaptive methods restructure interactions dynamically via routing, pruning, or workflow search (Zhang et al., 2024b; Zhuge et al., 2024; Yue et al., 2025).

A complementary line of research introduces **human-in-the-loop** collaboration. Humans have been positioned as supervisors, oracles, or evaluators, providing corrections or domain knowledge to strengthen agent performance (Takerngsaksiri et al., 2025; Mozannar et al., 2025; Liu et al., 2023; Pandya et al., 2024). Closely related, Siedler & Gemp (2025) study LLM-mediated guidance in MARL, where an LLM serves as a natural-language controller that interprets and delivers interventions to shape agents' learning trajectories and accelerate training. However, such systems often rely on heuristics (e.g., confidence thresholds) to trigger deferral (Kenton et al., 2024; Li et al., 2024b), and feedback is usually treated as a one-time fix rather than a signal for sustained capability growth (Mu et al., 2024; Wang et al., 2025b). Recent discussions highlight that human involvement can occur at multiple levels, from correcting local reasoning errors to reshaping global collaborative dynamics (Triem & Ding, 2024; Grondin et al., 2025). Together, these directions have advanced the field of MAS, yet challenges remain in moving beyond closed-world recombination toward open and adaptive collaboration. A detailed review of related work is provided in Appendix A.

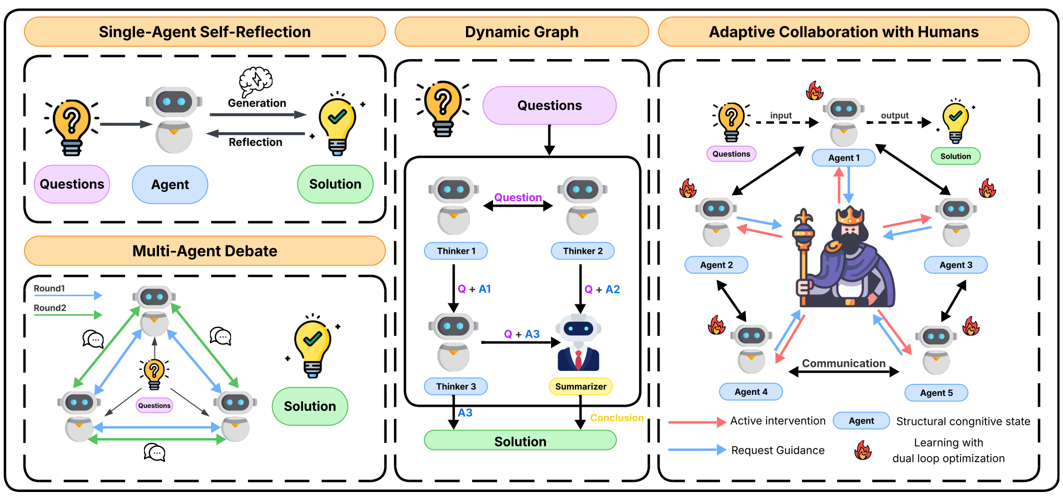


Figure 1: Comparison of collaborative reasoning paradigms. Left: *Single-Agent Self-Reflection*, where an individual agent iteratively improves its reasoning. Middle: *Multi-Agent Debate* and *Dynamic Graph* coordination, where multiple agents interact to refine knowledge integration. Right: *Adaptive Collaboration with Humans*, which augments multi-agent with strategic human guidance and Dual-Loop Policy Optimization, enabling both localized corrections and global improvements.

3 METHODOLOGY

Our methodology builds a multi-agent system designed for adaptive collaboration with a human expert. Figure 1 illustrates this setting in contrast to single-agent and purely multi-agent debate frameworks. At its core is a **metacognitive policy** that enables agents to reason about both their own competence and that of their peers, thereby deciding when to act autonomously and when to defer to external expertise. We formalize this collaborative process as a Metacognitive Markov Decision Process, which provides the foundation for our framework. The framework is then specified through a structured cognitive state space and a functional action space. Finally, we introduce a Dual-Loop Policy Optimization algorithm that combines reinforcement learning to refine the metacognitive policy with continual learning to integrate expert feedback into lasting capability growth.

3.1 PRELIMINARIES: THE METACOGNITIVE MARKOV DECISION PROCESS

We model human–agent collaboration as a **Metacognitive Markov Decision Process (Meta-MDP)**, which formalizes decision-making over high-level cognitive strategies such as autonomous problem-solving or deferral to human expertise. This abstraction provides a principled foundation for defining states, actions, transitions, and rewards in our collaborative framework. The full formalization and detailed design are provided in Appendix B.

3.2 A FRAMEWORK FOR HUMAN-AGENT COLLABORATION

Building on the Meta-MDP, we introduce a framework that operationalizes human–agent interaction through three components: (i) a structured cognitive state space that encodes problem context and metacognitive assessments, (ii) a functional action space representing high-level collaboration strategies, and (iii) an interaction protocol that specifies coordination across rounds. These elements allow agents to reason about tasks while regulating their decision boundaries in a principled way.

3.2.1 STRUCTURED COGNITIVE STATE SPACE

A sophisticated metacognitive policy requires a rich and informative state representation that goes beyond simple dialogue history. We therefore design a **structured cognitive state space** s_t composed of distinct dimensions intended to encode signals about an agent's confidence, its alignment with the group, and heuristic indicators of its current reasoning quality. For each agent i, the local assessment is represented by a feature vector with three components. The first dimension, **Certainty and Confidence ($\mathbf{z}_t^{\text{cert}}$)**, aggregates proxies for the agent's belief in its current solution, such as a *Self-Confidence Score* and a measure of *Solution Uncertainty* (e.g., Shannon entropy over candidate choices). The second dimension, **Social Cohesion and Dissonance ($\mathbf{z}_t^{\text{soc}}$)**, encodes the agent's standing within the group via metrics like *Inter-Agent Agreement* and overall *Answer Diversity*. Finally, the **Argumentative Quality ($\mathbf{z}_t^{\text{arg}}$)** dimension summarizes properties of the agent's generated 'Reason', including features such as *Reasoning Complexity* and *Evidence Grounding*. The full cognitive state s_t for an agent is then the concatenation of these feature vectors with the global problem context $\mathbf{x}_t$:

$$s_t = \text{concat}(\mathbf{x}_t, \mathbf{z}_t^{\text{cert}}, \mathbf{z}_t^{\text{soc}}, \mathbf{z}_t^{\text{arg}}). \tag{1}$$

By structuring the state space in this manner, we provide the policy with a multi-faceted set of task- and interaction-level signals on which to condition its decisions about whether to continue, revise, or defer.

3.2.2 THE STRATEGIC ACTION SPACE

The action space $\mathcal{A}$ in our Meta-MDP is not defined by low-level text generation, but by a discrete set of high-level cognitive strategies. These actions empower the agent to manage its problem-solving process, balancing the exploitation of existing collective knowledge against the exploration of new solutions and the strategic deferral to external expertise. Formally, the action space is defined as $\mathcal{A} = \{a^{\text{eval}}, a^{\text{create}}, a^{\text{defer}}\}$.

Evaluate (a^{eval}): Exploiting Collective Knowledge. This action represents the cognitive stance of convergence and synthesis. When selecting a^{eval}, the agent commits to exploiting the existing knowledge within the multi-agent group. Operationally, it must select and endorse one of the solutions already proposed by its peers in the current round. This action allows the agent to leverage collective intelligence and reinforce high-quality, consensual solutions.

Create (a^{create}): Creative Exploration and Hypothesis Generation. This action embodies the cognitive stance of divergence and exploration. By choosing a^{create}, the agent posits that the current solution pool is insufficient and commits to generating a novel solution sequence ('Choice', 'Reason') from scratch. This action is crucial for breaking cognitive fixation, correcting shared errors within the group, and introducing new, potentially superior reasoning paths.

Defer (a^{defer}): Risk Mitigation and Knowledge Augmentation. This action represents the highest level of metacognitive awareness—the ability to recognize the limits of the system's own capabilities. Selecting a^{defer} signals that the agent assesses the problem's uncertainty or difficulty to be beyond the collective's current ability to solve reliably. Operationally, this triggers a call to the external human expert, whose high-quality demonstration is then used as the round's output. This action serves as both a mechanism for ensuring task success in critical situations and as a conduit for introducing new knowledge into the system via continual learning.

3.2.3 Collaborative Interaction Model

A defining feature of our framework is the integration of human expertise into collective reasoning through a structured, multi-round protocol. At each round t, all N agents receive the shared cognitive state s_t. Each agent i independently samples a metacognitive action $a_{i,t} \sim \pi_\theta(a|s_t)$ and executes it in parallel. The output $y_{i,t}$ depends on the chosen action. When acting autonomously, the agent applies its internal generation process $g_\theta(s_t)$. When deferring, it adopts the authoritative solution $y_{\text{human},t}$ provided by the expert:

$$y_{i,t} = \begin{cases} g_\theta(s_t), & \text{if } a_{i,t} \in \{a^{\text{eval}}, a^{\text{create}}\}, \\ y_{\text{human},t}, & \text{if } a_{i,t} = a^{\text{defer}}. \end{cases} \tag{2}$$

The collection $\{y_{1,t}, \ldots, y_{N,t}\}$ then forms the next state s_{t+1}, ensuring that updates reflect the most reliable signals, whether from autonomous synthesis or human demonstration.

From a learning perspective, the *Defer* action plays a dual role. It serves as **risk mitigation**, ensuring progress under uncertainty by overriding flawed solutions, and as **knowledge augmentation**, injecting expert demonstrations as high-quality samples for continual learning (Section 3.3). Thus, the human collaborator functions not merely as a fallback oracle but as a driver of system improvement.

3.3 Adaptive Policy Optimization with Continual Learning

Mastering the metacognitive challenges described above requires an optimization strategy that balances two competing paths: the high-risk but potentially high-reward route of autonomous problem-solving, and the low-risk but constrained option of deferring to an expert. This trade-off naturally lends itself to reinforcement learning (RL), where the goal is to learn a policy π_θ that maximizes the expected utility of the collaborative process. To address this, we propose a **Dual-Loop Policy Optimization (DLPO)** framework that integrates a reinforcement learning objective for strategic policy optimization with a supervised objective for continual knowledge acquisition.

3.3.1 Inner Loop: Reinforcement Learning for Metacognitive Policy

The inner loop optimizes the agent's high-level policy $\pi_\theta(a|s_t)$ over strategic actions. The key challenge is to provide a learning signal that reflects the trade-off between autonomous success, potential failure, and the cost of expert intervention. This problem is well-suited to **Group Relative Policy Optimization (GRPO)**, which contrasts the relative advantages of actions in each state.

Reward Formulation. We design a reward function $R(s_t, a_t)$ that incorporates the **Cost of Inquiry**. Autonomous actions receive a binary ground-truth reward $R_{\text{gt}} \in \{+1, -1\}$, while the 'Defer' action yields a discounted reward reflecting expert reliability and intervention cost:

$$R(s_t, a_k) = \begin{cases} R_{\text{gt}}(y_k), & a_k \in \{a^{\text{eval}}, a^{\text{create}}\}, \\ R_{\text{human}} - C, & a_k = a^{\text{defer}}, \end{cases} \tag{3}$$

where R_{human} accounts for expert accuracy and C is a tunable penalty.

GRPO Objective. Given the reward vector $\mathbf{R}_t = [R(s_t, a_1), \ldots, R(s_t, a_K)]$, advantages are computed by centering rewards:

$$A(s_t, a_k) = R(s_t, a_k) - \tfrac{1}{K} \sum_{j=1}^{K} R(s_t, a_j). \tag{4}$$

The policy gradient objective is:

$$\mathcal{L}_{\text{PG}}(\theta) = -\mathbb{E}_{s_t, a_t \sim \pi_\theta} \big[A(s_t, a_t) \log \pi_\theta(a_t|s_t) \big]. \tag{5}$$

Two regularizers ensure stability: a KL-penalty constrains deviation from the reference policy π_{ref}, and an entropy bonus promotes exploration. The final inner-loop loss is:

$$\mathcal{L}_{\text{Inner}} = \mathcal{L}_{\text{PG}} + \beta_{\text{kl}} \mathcal{L}_{\text{KL}} - \beta_{\text{ent}} \mathcal{L}_{\text{Entropy}}. \tag{6}$$

3.3.2 OUTER LOOP: CONTINUAL LEARNING FROM EXPERT FEEDBACK

While the inner loop optimizes how the agent *uses* its current abilities, a truly adaptive system must also *expand* them. Reinforcement learning alone cannot overcome the knowledge ceiling of the base LLM, as it improves decision policies without introducing fundamentally new skills. To break this ceiling, we introduce an outer optimization loop for **Continual Learning from Expert Demonstrations**.

This loop is activated by the 'Defer' action, which indicates that the agent has identified a knowledge gap. When deferring, the agent receives a high-quality demonstration $y_{\text{human}} = (t_1, \dots, t_L)$ from the expert, which is converted into a supervised fine-tuning (SFT) sample. The training objective is to maximize the likelihood of this sequence, conditioned on the state s_t, by minimizing the cross-entropy loss:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\sum_{i=1}^{L} \log \pi_\theta(t_i \mid s_t, t_{1:i-1}). \tag{7}$$

In this design, the inner RL loop determines *when* to defer, while the outer loop teaches *what* to learn from expert input. Together, they establish an apprentice–mentor dynamic: the agent strategically invokes human guidance and systematically assimilates it into lasting capability growth.

3.3.3 THE FINAL DUAL-LOOP POLICY OPTIMIZATION OBJECTIVE

The inner and outer loops are optimized jointly to train a single agent that is both strategically adept and continually improving. The final training objective is a principled combination of the reinforcement learning signal from the inner loop and the conditional supervised signal from the outer loop. The total loss, $\mathcal{L}_{\text{total}}$, is computed over a batch of experiences:

$$\mathcal{L}_{\text{total}}(\theta) = \mathbb{E}_{(s_t, a_t)} \left[\mathcal{L}_{\text{Inner}}(\theta) + \lambda_{\text{sft}} \cdot \mathbb{I}(a_t = a^{\text{defer}}) \cdot \mathcal{L}_{\text{SFT}}(\theta) \right], \tag{8}$$

where $\mathcal{L}_{\text{Inner}}(\theta)$ is the full GRPO objective, λ_{sft} is a hyperparameter balancing the two learning signals, and $\mathbb{I}(\cdot)$ is the indicator function that ensures the SFT loss is only applied when the 'Defer' action is taken.

4 EXPERIMENTS

Experimental Setup. We evaluate our method on a broad suite of benchmarks, including general language understanding (*MMLU*), program synthesis (*HumanEval*), and quantitative mathematics (*GSM8K*, *MATH*, *AIME*, *AMC*). Following related works (Liu et al., 2023; Pandya et al., 2024), we employ **GPT-4o-mini** as a proxy human expert, leveraging its strong reasoning capability to simulate human interventions. Detailed experimental and training settings are provided in Appendix C.1.

Overall Performance. Table 1 shows that our **LIMA** framework establishes a new state-of-the-art, consistently surpassing strong autonomous multi-agent baselines across all six reasoning benchmarks. These baselines, including debate-style (*e.g.*, LLM-Debate), topology-based (*e.g.*, DyLAN), and graph-optimization (*e.g.*, GPTSwarm, AFLOW) methods, remain confined to "closed-world" collaboration, where performance is capped by the agents' internal knowledge and often falters on problems requiring non-obvious reasoning paths. In contrast, LIMA introduces an "open-world" dynamic by enabling agents to strategically access external expertise, directly addressing this knowledge ceiling. On the Llama3-8B backbone, our trained agent achieves average gains of 7%–12% over the strongest autonomous baselines, with the largest improvements on competition-style math tasks such as AIME, where cascade failures from flawed premises are common. By learning a metacognitive policy to defer under high uncertainty, LIMA avoids these pitfalls and effectively leverages superior guidance. These results confirm that performance gains stem not from complex interaction alone, but from principled integration of external knowledge and the agent's learned ability to decide when to invoke it.

Model Scalability and Generality. We test whether the proposed framework transfers across heterogeneous backbones and model sizes by evaluating Qwen2.5-7B, Qwen2.5-3B, LLaMA3-8B, and LLaMA3-3B on GSM8K. Table 2 shows large variation in autonomous baselines: larger models

Model	GSM8K	AMC	AIME	MATH	HumanEval	MMLU
Vanilla	72.76 (+0.00)	8.03 (+0.00)	2.96 (+0.00)	42.85 (+0.00)	47.56 (+0.00)	57.99 (+0.00)
CoT	74.22 (+1.46)	11.65 (+3.62)	3.70 (+0.74)	46.93 (+4.08)	51.42 (+3.86)	61.57 (+3.58)
SC	80.79 (+8.03)	12.45 (+4.42)	4.07 (+1.11)	51.28 (+8.43)	57.52 (+9.96)	68.30 (+10.31)
PHP	80.01 (+7.25)	15.66 (+7.63)	4.44 (+1.48)	53.71 (+10.86)	56.50 (+8.94)	68.46 (+10.47)
Debate	83.52 (+10.76)	19.28 (+11.25)	5.56 (+2.60)	56.25 (+13.40)	57.72 (+10.16)	67.59 (+9.60)
G-Debate	83.98 (+11.22)	<u>20.48</u> (+12.45)	5.19 (+2.23)	<u>57.42</u> (+14.57)	57.93 (+10.37)	<u>69.89</u> (+11.90)
DyLAN	82.03 (+9.27)	19.68 (+11.65)	3.70 (+0.74)	55.32 (+12.47)	61.59 (+14.03)	66.85 (+8.86)
G-Swarm	<u>84.89</u> (+12.13)	15.66 (+7.63)	<u>5.78</u> (+2.82)	56.69 (+13.84)	59.55 (+11.99)	69.67 (+11.68)
A-Prune	84.38 (+11.62)	16.47 (+8.44)	4.81 (+1.85)	54.37 (+11.52)	57.11 (+9.55)	69.09 (+11.10)
AFlow	83.75 (+10.99)	12.05 (+4.02)	4.44 (+1.48)	55.28 (+12.43)	<u>62.20</u> (+14.64)	69.31 (+11.32)
LIMA	88.23 (+15.47)	27.32 (+19.29)	8.12 (+5.16)	62.52 (+19.67)	65.78 (+18.22)	71.35 (+13.36)
w/ DLPO	**91.25** (+18.49)	**30.30** (+22.27)	**9.30** (+6.34)	**65.46** (+22.61)	**67.82** (+20.26)	**73.58** (+15.59)

Table 1: Comparison of baseline and proposed methods using the LLaMA3-8B backbone. All values are percentages (the percent sign is omitted in the table). Values in parentheses denote absolute differences relative to the *Vanilla* baseline (first row). Underlined numbers indicate the best-performing baseline on each benchmark. The additional experiments on scalability, cost, stronger proxy experts, and performance gain analyses are provided in Appendix D.

Model	Qwen2.5-7B	Qwen2.5-3B	LLaMA3-8B	LLaMA3-3B
Vanilla	90.88 (+0.00)	83.37 (+0.00)	72.76 (+0.00)	46.85 (+0.00)
CoT	90.98 (+0.10)	84.56 (+1.19)	74.22 (+1.46)	50.14 (+3.29)
SC	92.95 (+2.07)	88.60 (+5.23)	80.79 (+8.03)	54.21 (+7.36)
PHP	93.30 (+2.42)	86.45 (+3.08)	80.01 (+7.25)	62.22 (+15.37)
LLM-Debate	<u>93.63</u> (+2.75)	87.14 (+3.77)	83.52 (+10.76)	75.84 (+28.99)
DyLAN	93.15 (+2.27)	<u>88.10</u> (+4.73)	82.03 (+9.27)	<u>76.47</u> (+29.62)
GPTSwarm	92.27 (+1.39)	86.78 (+3.41)	<u>84.89</u> (+12.13)	69.19 (+22.34)
AgentPrune	92.44 (+1.56)	86.43 (+3.06)	84.38 (+11.62)	65.02 (+18.17)
AFlow	92.86 (+1.98)	87.52 (+4.15)	83.75 (+10.99)	68.37 (+21.52)
LIMA	**96.38** (+5.50)	**93.51** (+10.14)	**91.25** (+18.49)	**81.35** (+34.50)

Table 2: Performance of baselines across four LLM backbones on GSM8K. All values are percentages (percent sign omitted). Parentheses show absolute differences (percentage points) relative to the *Vanilla* row for each backbone. LIMA refers to the agent trained with the proposed DLPO.

are stronger than their 3B counterparts, and Qwen2.5 backbones start from higher scores than LLaMA3. Despite these differences, **LIMA** improves every backbone. Relative to the strongest non-LIMA baseline in each column, the absolute gains are 6.36 percentage points on LLaMA3-8B over GPTSwarm (91.25 vs. 84.89), 4.88 points on LLaMA3-3B over DyLAN (81.35 vs. 76.47), 2.75 points on Qwen2.5-7B over LLM-Debate (96.38 vs. <u>93.63</u>), and 5.41 points on Qwen2.5-3B over DyLAN (93.51 vs. <u>88.10</u>). The improvements are particularly pronounced for smaller models, where collaboration quality compensates for limited capacity. LIMA also surpasses several stronger-size baselines; for example, on Qwen2.5-3B it exceeds multiple 7B methods. These observations indicate that the benefit is largely orthogonal to the model's intrinsic knowledge. Rather than injecting task facts, LIMA provides a transferable collaboration and selection policy that reliably raises performance across families and sizes, with the largest payoffs where models start weaker.

Proactive Human Intervention for Local Reasoning Correction. A central motivation for our study is that humans should not only respond passively when multi-agent systems request assistance, but also possess the ability to intervene proactively. As discussed earlier, the role of humans in collaborative systems extends beyond providing final solutions; equally important is their capacity to identify and correct local reasoning flaws during the interaction process. Leveraging our interactive collaboration architecture, we therefore investigate the effect of active human intervention. In particular, we contrast two settings: replacing human feedback with GPT-4o-mini as a proxy expert, and incorporating real human experts directly. As shown in Table 3 (a), both variants improve performance compared with pure LIMA, and real human intervention yields the strongest gains.

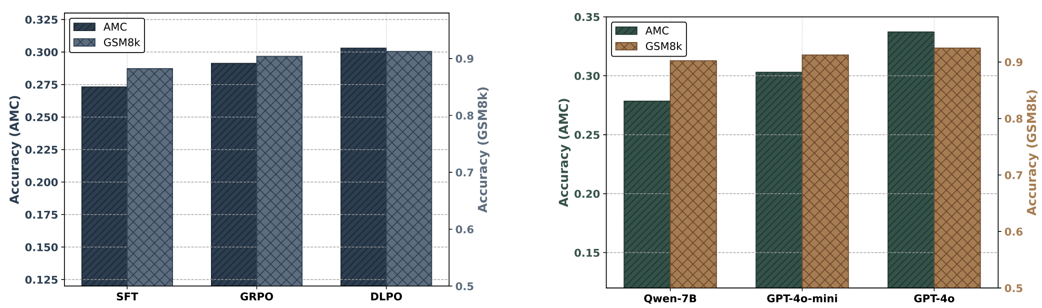
Method	GSM8K	AMC	HE
LIMA			
w/ GPT-Intervene	0.9475	0.3162	0.7354
w/ Human-Intervene	0.9617	0.3359	0.7231

(a) Mode A: Active human intervention during agent reasoning. HE denotes HumanEval dataset.

Method	GSM8K	AMC	HE
LIMA			
w/ GPT-Help	0.9236	0.3030	0.6835
w/ Human-Help	0.9317	0.2875	0.6717

(b) Mode B: Human assistance only when the system requests guidance.HE denotes HumanEval dataset.

Table 3: Human involvement under two modes. Left: Mode A evaluates active human intervention during agent reasoning. Right: Mode B evaluates responses when agents explicitly request guidance. Results are reported as Solve Rate (%) on GSM8K (with subset sampling) and AMC.



(a) Effect of training paradigms.

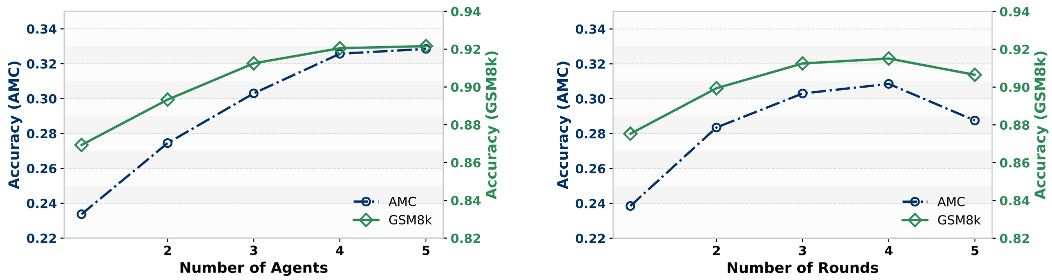
(b) Effect of replacing human collaborators

Figure 2: Ablation studies on training paradigms and external collaborators. Subfigure (a) compares different optimization strategies, while (b) evaluates the effect of substituting human expertise with LLMs of varying strengths.

This indicates that humans are especially adept at detecting local inconsistencies and steering the reasoning trajectory before errors accumulate. Furthermore, both GPT-based and human-based intervention outperform purely request-driven help, suggesting that proactive intervention and passive assistance should be integrated to fully exploit the benefits of human involvement.

On-Demand Human Assistance versus Artificial Expertise. To further examine the effect of different types of intelligence on multi-agent systems, we conduct a controlled comparison between human experts and GPT-4o-mini when agents explicitly request guidance. Results in Table 3 (b) show that on GSM8K subset, which involves relatively straightforward grade-school problems, human experts achieve higher solve rates, effectively solving most queries. However, on AMC, which contains competition-level mathematical problems, human performance does not surpass GPT-4o-mini, reflecting the limits of individual knowledge in specialized domains. These findings highlight the extensibility of our framework: it can flexibly incorporate either human collaborators or artificial experts, adapting to the strengths of each.

Ablation Studies on Learning Paradigms and Expert Substitutes. We perform two ablation studies to assess the contributions of our framework. The first examines three training paradigms: SFT-only, GRPO-only, and the complete DLPO method. SFT steadily expands the agents' knowledge by assimilating expert demonstrations. GRPO strengthens decision-making by teaching the agents to balance the risks of autonomous attempts against the costs of expert deferrals. When combined, DLPO achieves the most consistent and robust improvements, as it unifies continual knowledge acquisition with metacognitive policy optimization Figure 2(a). The second ablation study evaluates different surrogate experts, including Qwen-7B, GPT-4o-mini, and GPT-4o. As shown in Figure 2 (b), stronger experts provide more reliable interventions and higher overall performance, yet even weaker substitutes contribute meaningfully. These results highlight the robustness of the framework to imperfect guidance and its extensibility to diverse sources of external expertise. A more detailed analysis is provided in Appendix C.2.



(a) Accuracy as a function of the number of agents. (b) Accuracy as a function of the number of rounds.

Figure 3: Effect of scaling collaborative configurations. (a) shows how increasing the number of agents impacts accuracy on AMC and GSM8K, while (b) analyzes how varying the number of interaction rounds influences performance. Together, the results highlight the trade-offs between broader exploration through more agents and deeper refinement through additional rounds.

The Value of Collective Exploration. To examine how the size of the collective influences performance, we evaluate the LIMA framework with varying numbers of autonomous agents ($N \in \{1, \ldots, 5\}$) on AMC and GSM8K. As illustrated in Figure 3 (a), accuracy improves consistently as the agent count increases, surpassing 0.92 on GSM8K with four agents. This pattern validates a central principle of our design: **collective exploration**. Increasing the number of parallel agents broadens the search space, producing more diverse candidate solutions and raising the likelihood of finding a correct reasoning path, especially on challenging tasks. However, the gains diminish as the number of agents grows beyond four, with the curve plateauing by five. Importantly, the results confirm that the learned metacognitive policy scales effectively: rather than being overwhelmed by a larger pool of outputs, the system successfully synthesizes them into progressively stronger decisions.

Optimal Depth of Iterative Collaboration. To assess the role of iterative refinement, we evaluate the LIMA framework with varying numbers of collaborative rounds ($R \in \{1, \ldots, 5\}$). As shown in Figure 3 (b), the effect of additional rounds is distinctly non-monotonic. On both AMC and GSM8K, performance improves steadily at first, reaching its peak in the fourth round. For instance, on the challenging AMC benchmark, accuracy rises from a single-round baseline of about 0.24 to nearly 0.31 after four rounds of interaction. These gains highlight the benefit of multi-round collaboration, where agents leverage peer feedback to correct early errors and converge on stronger solutions. However, extending the process beyond this optimal depth results in diminishing and eventually negative returns, with accuracy declining in the fifth round. This pattern suggests that excessive interaction introduces failure modes such as **error amplification**, where minor mistakes propagate and intensify, or **cognitive fixation**, where agents collectively reinforce a flawed line of reasoning. The findings underscore a key design trade-off: iterative refinement is valuable, but not universally beneficial. Effective systems therefore require mechanisms to identify when additional collaboration is productive and when it risks entrenching errors.

5 Conclusion

In this paper, we presented the **LIMA** framework, which equips multi-agent systems with a metacognitive policy for deciding when to act autonomously and when to defer to human expertise. Through our **Dual-Loop Policy Optimization** strategy, combining GRPO for risk-aware decision-making with continual learning from expert demonstrations, agents achieve both short-term adaptability and long-term growth. Experiments across diverse reasoning benchmarks show that LIMA consistently outperforms autonomous and multi-agent baselines. Additional studies with human experts highlight the unique role of proactive intervention in correcting local reasoning errors and strengthening collaboration. In future work, we plan to explore fully dynamic collaboration paradigms and endow multi-agent systems with stronger evolutionary capabilities, moving toward open-ended and adaptive intelligence.

REFERENCES

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.

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.

Jiaju Chen, Yuxuan Lu, Xiaojie Wang, Huimin Zeng, Jing Huang, Jiri Gesi, Ying Xu, Bingsheng Yao, and Dakuo Wang. Multi-agent-as-judge: Aligning llm-agent-based automated evaluation with multi-dimensional human evaluation. *arXiv preprint arXiv:2507.21028*, 2025.

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*, 2024a.

Shuhao Chen, Weisen Jiang, Baijiong Lin, James Kwok, and Yu Zhang. Routerdc: Query-based router by dual contrastive learning for assembling large language models. *Advances in Neural Information Processing Systems*, 37:66305–66328, 2024b.

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.

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, Jauhri, et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024.

Xueyang Feng, Zhi-Yuan Chen, Yujia Qin, Yankai Lin, Xu Chen, Zhiyuan Liu, and Ji-Rong Wen. Large language model-based human-agent collaboration for complex task solving. *arXiv preprint arXiv:2402.12914*, 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.

Suzie Grondin, Arthur Charpentier, and Philipp Ratz. Beyond human intervention: Algorithmic collusion through multi-agent learning strategies. *arXiv preprint arXiv:2501.16935*, 2025.

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.

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.

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

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.

Zachary Kenton, Noah Siegel, János Kramár, Jonah Brown-Cohen, Samuel Albanie, Jannis Bulian, Rishabh Agarwal, David Lindner, Yunhao Tang, Noah Goodman, et al. On scalable oversight with weak llms judging strong llms. *Advances in Neural Information Processing Systems*, 37: 75229–75276, 2024.

Boyi Li, Zhonghan Zhao, Der-Horng Lee, and Gaoang Wang. Adaptive graph pruning for multi-agent communication. *arXiv preprint arXiv:2506.02951*, 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.

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.

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.

Jijia Liu, Chao Yu, Jiaxuan Gao, Yuqing Xie, Qingmin Liao, Yi Wu, and Yu Wang. Llm-powered hierarchical language agent for real-time human-ai coordination. *arXiv preprint arXiv:2312.15224*, 2023.

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.

Jianqiao Lu, Wanjun Zhong, Wenyong Huang, Yufei Wang, Fei Mi, Baojun Wang, Weichao Wang, Lifeng Shang, and Qun Liu. Self: Language-driven self-evolution for large language model. 2023.

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.

Hussein Mozannar, Gagan Bansal, Cheng Tan, Adam Fourney, Victor Dibia, Jingya Chen, Jack Gerrits, Tyler Payne, Matheus Kunzler Maldaner, Madeleine Grunde-McLaughlin, et al. Magentic-ui: Towards human-in-the-loop agentic systems. *arXiv preprint arXiv:2507.22358*, 2025.

Chunjiang Mu, Hao Guo, Yang Chen, Chen Shen, Die Hu, Shuyue Hu, and Zhen Wang. Multi-agent, human–agent and beyond: a survey on cooperation in social dilemmas. *Neurocomputing*, 610:128514, 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.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, M Waleed Kadous, and Ion Stoica. Routellm: Learning to route llms with preference data. *arXiv preprint arXiv:2406.18665*, 2024.

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.

Ravi Pandya, Michelle Zhao, Changliu Liu, Reid Simmons, and Henny Admoni. Multi-agent strategy explanations for human-robot collaboration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 17351–17357. IEEE, 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.

Feng Peiyuan, Yichen He, Guanhua Huang, Yuan Lin, Hanchong Zhang, Yuchen Zhang, and Hang Li. Agile: A novel reinforcement learning framework of llm agents. *Advances in Neural Information Processing Systems*, 37:5244–5284, 2024.

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.

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.

Philipp D Siedler and Ian Gemp. Llm-mediated guidance of marl systems. *arXiv preprint arXiv:2503.13553*, 2025.

KV Srivatsa, Kaushal Kumar Maurya, and Ekaterina Kochmar. Harnessing the power of multiple minds: Lessons learned from llm routing. *arXiv preprint arXiv:2405.00467*, 2024.

Chuanneng Sun, Songjun Huang, and Dario Pompili. Llm-based multi-agent decision-making: Challenges and future directions. *IEEE Robotics and Automation Letters*, 2025.

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

Wannita Takerngsaksiri, Jirat Pasuksmit, Patanamon Thongtanunam, Chakkrit Tantithamthavorn, Ruixiong Zhang, Fan Jiang, Jing Li, Evan Cook, Kun Chen, and Ming Wu. Human-in-the-loop software development agents. In *2025 IEEE/ACM 47th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pp. 342–352. IEEE, 2025.

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.

Haley Triem and Ying Ding. "tipping the balance": Human intervention in large language model multi-agent debate. *Proceedings of the Association for Information Science and Technology*, 61 (1):361–373, 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*, 2023a.

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.

Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. Unleashing the emergent cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration. *arXiv preprint arXiv:2307.05300*, 2023b.

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

Ziyan Wang, Zhicheng Zhang, Fei Fang, and Yali Du. M3hf: Multi-agent reinforcement learning from multi-phase human feedback of mixed quality. *arXiv preprint arXiv:2503.02077*, 2025b.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

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.

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.

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.

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.

Xuan Zhang, Yang Deng, Zifeng Ren, See-Kiong Ng, and Tat-Seng Chua. Ask-before-plan: Proactive language agents for real-world planning. *arXiv preprint arXiv:2406.12639*, 2024d.

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.

Henry Peng Zou, Wei-Chieh Huang, Yaozu Wu, Yankai Chen, Chunyu Miao, Hoang Nguyen, Yue Zhou, Weizhi Zhang, Liancheng Fang, Langzhou He, et al. Llm-based human-agent collaboration and interaction systems: A survey.

A Related Work

A.1 Collaboration Paradigms in Multi-Agent LLM Systems.

Early research has shown that single LLM agents face inherent limitations in context length, sequential generation, and breadth of skills, which restrict their ability to solve complex tasks requiring diverse perspectives or parallel reasoning (Gabriel et al., 2024; Liang et al., 2023; Xiong et al., 2023; Yin et al., 2023; Zhang et al., 2023). To overcome these bottlenecks, recent work has explored **multi-agent systems** (MAS), where multiple LLMs are orchestrated to realize collective intelligence across domains such as software engineering, planning, and problem solving (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).

Most existing frameworks rely on prompt-based paradigms that predefine roles, communication protocols, or workflow structures. These designs enable debate, critique, and corporate-style pipelines, achieving notable gains in coordination efficiency (Du et al., 2023; Chan et al., 2023; Chen et al., 2024a; Mukobi et al., 2023; Wang et al., 2023a; Abdelnabi et al., 2024; Han et al., 2025). However, because they are hand-crafted and do not adapt through experience, such systems remain fundamentally **closed-world**: they can only recombine existing knowledge rather than acquire genuinely new capabilities (Wang et al., 2023b; Liu et al., 2024; Chen et al., 2024b).

Beyond fixed prompts, two further directions have emerged. **Prestructured coordination** employs fixed debate or peer-review topologies, such as chains, trees, or graphs, to refine reasoning (Du et al., 2023; Liu et al., 2024; Qian et al., 2024). In contrast, **self-organizing approaches** adapt the interaction graph dynamically through search, pruning, routing, or evolutionary mechanisms (Hu et al., 2024; Shang et al., 2024; Zhang et al., 2024b; Zhuge et al., 2024; Zhang et al., 2024c; Yue et al., 2025). These advances highlight the importance of who communicates and when, yet they primarily optimize internal coordination and leave unaddressed the problem of learning from external expertise.

In contrast, our work introduces a centralized and iterative collaboration framework that explicitly incorporates **human expertise** as an open-world resource. Rather than treating human feedback as a passive oracle or one-time correction (Takerngsaksiri et al., 2025; Mozannar et al., 2025; Liu et al., 2023; Pandya et al., 2024), we propose to endow agents with a metacognitive policy that governs both the timing of deferral and the assimilation of human guidance into lasting improvements. This approach differs fundamentally from prior debate, routing, and workflow-search systems, which often lack principled credit assignment between reasoning and decision outcomes (Chan et al., 2023; Talebirad & Nadiri, 2023; Wei et al., 2025). By integrating external knowledge with learned metacognitive adaptation, our framework moves beyond static collaboration to establish a pathway toward adaptive and continually improving multi-agent intelligence.

A.2 Multi-Agent Reinforcement Learning for LLMs.

A growing line of work seeks to move beyond static prompt engineering and endow multi-agent LLM systems with adaptive learning capabilities. Early efforts rely on supervised fine-tuning (SFT) to inject collaborative patterns by imitating expert demonstrations or curated trajectories (Lu et al., 2023; Madaan et al., 2023; Zelikman et al., 2022; Wei et al., 2021). While effective for seeding cooperative behaviors, SFT remains limited by its offline nature and cannot adapt to novel contexts. Reinforcement learning (RL) has thus emerged as a natural complement, enabling agents to refine policies through trial-and-error interaction and reward-driven adaptation (Zhu et al., 2025; Zhuang et al., 2024). In practice, SFT often serves as initialization, with RL providing fine-grained policy improvement under feedback (Zhu et al., 2025; Li, 2019; Zhang et al., 2021).

Recent research highlights three major directions. First, compiling language into structured controllers—such as graphs, code, or plans—allows RL to optimize execution policies over symbolic abstractions rather than raw text (Zhuang et al., 2024; Jia et al., 2025; Zhu et al., 2025). Second, online collaboration is adapted through RL-based task decomposition, communication routing, and role assignment, which allow dynamic coordination beyond static protocols (Zhou et al., 2025; Wang et al., 2024; Xu et al., 2025; Li et al., 2024a). Third, several studies explore learning reasoning policies directly in language space using GRPO or PPO-style updates, often integrating tools or human input when beneficial (Wan et al., 2025; Park et al., 2025; Han et al., 2025; Peiyuan et al., 2024; Feng et al., 2024). These approaches underscore the importance of credit assignment and reward shaping for aligning emergent behaviors in high-dimensional language action spaces (Wei et al., 2025; Jiang et al., 2025; Alsadat & Xu, 2024; Lin et al., 2025).

Our work is closely aligned with this trajectory but emphasizes a key gap: most existing RL approaches focus on optimizing intra-agent or inter-agent coordination while leaving the system's knowledge boundary fixed. In contrast, we introduce a dual-loop perspective where RL is responsible for learning a metacognitive deferral policy, and expert demonstrations triggered by deferral events fuel continual learning. This integration addresses both immediate decision-making and long-term capability growth, providing a principled path toward genuinely adaptive multi-agent systems.

B Methodology

B.1 Preliminaries: The Metacognitive Markov Decision Process

We formalize the dynamics of human–agent collaboration as a **Meta-Cognitive Markov Decision Process (Meta-MDP)**, defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$. A Meta-MDP provides a principled framework for sequential decision-making where actions correspond to high-level cognitive strategies. Formally, a Meta-MDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$. At each round t of the multi-agent collaboration, the process unfolds as follows: The state $s_t \in \mathcal{S}$ is a **structured cognitive state representation**, which encapsulates not only the external problem context but also the agent's internal assessment of its own and its peers' current understanding, as we will detail in Section 3.2. Based on this rich state, the agent selects a metacognitive action a_t from a discrete action space $\mathcal{A}$, which includes functional strategies such as solving the problem autonomously or deferring to the expert. The system then transitions to a new state s_{t+1} according to the transition function $P(s_{t+1}|s_t, a_t)$. A reward $R(s_t, a_t)$ is issued, designed to incentivize both task success and the efficient utilization of the expert resource. The overarching goal is to learn an optimal metacognitive policy $\pi^*(a_t|s_t)$ that maximizes the expected cumulative reward, thereby training an agent that can rationally balance autonomous problem-solving with strategic reliance on human guidance.

C Experiment

C.1 Experimental Settings

Benchmarks and Evaluation. To evaluate our framework, we conduct experiments on a broad collection of benchmarks that test complementary aspects of reasoning ability. These tasks span three domains: general knowledge and analytical reasoning, program synthesis, and mathematical problem solving.

For general knowledge, we use the MMLU benchmark, which includes 57 subject areas in a multiple-choice format; performance is measured by classification Accuracy. For program synthesis, we adopt HumanEval, where models generate code solutions from natural-language specifications; following convention, we report Pass@1, the proportion of single attempts that succeed on all hidden tests.

For quantitative reasoning, we consider four math-focused datasets with concise numerical answers: GSM8K (grade-school arithmetic word problems), MATH (competition-level problems covering algebra, geometry, combinatorics, and number theory), AIME (short-form olympiad-style tasks), and AMC (large-scale contest problems). Performance on these datasets is reported as Solve Rate, defined by exact match against each dataset's normalized reference solution.

All evaluations are conducted on the official datasets with standard prompting protocols. We exclude external tools and retrieval, ensuring that improvements stem from our collaboration framework rather than auxiliary resources. In single-agent settings, inference is run deterministically. For multi-agent experiments involving stochastic sampling, we fix random seeds, repeat runs, and report averaged results. Confidence intervals are included in the appendix. This setup isolates the contribution of our proposed method and allows for fair comparison against existing approaches.

Baselines. To ensure a fair and comprehensive comparison, we evaluate our framework against three broad families of baseline methods that represent the dominant paradigms in collaborative reasoning with LLMs:

1. **Single-Agent Solvers.** These methods rely on a single model instance without peer interaction. They capture the performance limits of prompting alone. Examples include direct decoding under a standard prompt (*Vanilla*), reasoning traces generated by *Chain-of-Thought (CoT)* prompting, and multi-sample aggregation methods such as *Self-Consistency (SC)*. Self-reflection strategies (e.g., *Reflection*, *RASC*) are also included, where the model internally revises its outputs without external assistance.
2. **Interactive Multi-Agent Deliberation.** This class introduces explicit communication among multiple agents. Agents generate, critique, and refine one another's proposals. Approaches such as *LLM-Debate* implement structured argue–respond cycles, while pairwise or pooled critique frameworks (e.g., *PHP*) simulate peer-review processes. These baselines assess whether systematic interaction alone, without external expertise, can reduce errors and improve reasoning robustness.
3. **System-Level Coordination Frameworks.** Some approaches treat collaboration as an optimization problem over computational graphs. Adaptive topology and routing methods (e.g., *DyLAN*, *MasRouter*) dynamically determine communication patterns, while workflow- and search-based systems (e.g., *GPTSwarm*, *AFLOW*) orchestrate reusable reasoning modules. Communication-pruning strategies such as *AgentPrune* improve scalability by filtering redundant interactions. These baselines highlight efficiency and coordination at scale.

For all baselines, we control the backbone model, prompting setup, and generation budget (number of agents, rounds, and outputs). When multiple candidates are produced, we apply the baseline's canonical reduction method (e.g., majority vote). No retrieval augmentation or external tools are used. This categorization clarifies whether improvements arise from stronger *single-agent reasoning*, richer *peer verification*, or more effective *coordination*, providing a clear context for evaluating our method.

Implementation Details. Our experiments employ three agents engaged in collaborative reasoning over three successive rounds. Each agent is drawn from instruction-tuned open models, specifically **Qwen2.5-7B-Instruct**, **Qwen2.5-3B-Instruct** (Team, 2024), **Llama-3.1-8B-Instruct**, and **Llama-3.2-3B-Instruct** (Dubey et al., 2024). All models are fine-tuned with a parameter-efficient LoRA configuration, using a rank of 16. To ensure efficient execution, we rely on the HuggingFace Transformers framework, enabling both 8-bit quantization and key–value caching for reduced memory usage and faster inference. Decoding follows a nucleus sampling scheme with $p = 0.95$, a temperature of 0.7, and a maximum generation length of 512 tokens. For tasks requiring reproducibility, such as pairwise evaluation, we reduce the temperature to 0.3. Each experimental setup is repeated with three independent random seeds, and results are averaged to control for variance. For optimization, we use Adam with an initial learning rate of 5×10^{-5} and apply a cosine decay schedule. Training incorporates an entropy regularization term of 0.01 to encourage exploration, and a KL penalty of 1.0 to anchor the learned policy to the supervised initialization. Training runs for three epochs maximum with a global batch size of 256, distributed across four NVIDIA A100 GPUs (80 GB each).

Definition of Human Expert. In principle, the human collaborator in our framework refers to a real person who can provide external knowledge and corrective interventions. However, following prior studies that approximate human input with advanced language models (Liu et al., 2023; Pandya et al., 2024), we also adopt intelligent LLMs as practical substitutes. In most experiments, we use **GPT-4o-mini** as the default proxy for the human expert, striking a balance between cost and effectiveness. To more rigorously assess the framework's native design for human–agent collaboration, we additionally conduct experiments with two complementary settings: (i) *proactive intervention*, where the human actively identifies local reasoning errors, and (ii) *passive assistance*, where the human responds only when explicitly queried. For comparison with real experts, we further involve several PhD students in computer science with extensive research experience and specialized knowledge relevant to the benchmark datasets. This setup enables us to disentangle the influence of simulated versus real human input on the multi-agent system's performance.

C.2 Experimental Results

Ablation Study on Learning Paradigms. We compare three training settings: SFT-only, GRPO-only, and the full DLPO method that integrates GRPO with continual learning. As shown in Figure 2(a), the results reveal several important patterns. Pure SFT improves the baseline by continuously assimilating expert demonstrations, which allows the agents to reduce recurring reasoning mistakes and improve performance on the evaluated tasks. GRPO on its own strengthens the decision policy by teaching agents to weigh the trade-off between autonomous attempts and costly deferrals. While both settings are beneficial in isolation, their gains are limited when applied separately. The combined DLPO method achieves the strongest and most stable improvement, demonstrating that continual acquisition of expert knowledge and the optimization of metacognitive decision-making reinforce one another.

Ablation Study on Human Substitutes. We further examine how the system behaves when the human collaborator is replaced by different surrogate experts of varying strength, specifically Qwen-7B, GPT-4o-mini, and GPT-4o. Figure 2(b) presents the comparison, which highlights two insights. Stronger models provide more consistent and higher-quality interventions, naturally leading to better overall performance. At the same time, even weaker surrogates still contribute meaningfully, showing that the system is robust to imperfect guidance and capable of integrating diverse forms of external input. Importantly, the results validate the extensibility of our design: the collaborative loop does not rely on a particular expert, but instead offers a general mechanism for incorporating any external intelligence, whether it is another LLM or a human expert.

The Value of Collective Exploration. To examine how the size of the collective influences performance, we evaluate the LIMA framework with varying numbers of autonomous agents ($N \in 1, \ldots, 5$) on AMC and GSM8K. As illustrated in Figure 3 (a), performance improves consistently as the agent count increases. On AMC, scaling from a single agent to three agents yields a notable boost in accuracy, and a similar upward trend is observed on GSM8K, where the system surpasses 0.92 accuracy with four agents. This pattern validates a central principle of our design: **collective exploration**. Increasing the number of parallel agents broadens the search space, producing more diverse candidate solutions and raising the likelihood of finding a correct reasoning path, especially on challenging tasks where solutions are non-trivial. However, the gains diminish as the number of agents grows beyond four, with the curve beginning to plateau by five agents. This indicates a trade-off between the marginal benefit of additional perspectives and the computational overhead they incur. Importantly, the results confirm that the learned metacognitive policy scales effectively: rather than being overwhelmed by a larger pool of outputs, the system successfully synthesizes them into progressively stronger decisions.

Optimal Depth of Iterative Collaboration. To assess the role of iterative refinement, we evaluate the LIMA framework with varying numbers of collaborative rounds ($R \in 1, \ldots, 5$). As shown in Figure 3 (b), the effect of additional rounds is distinctly non-monotonic. On both AMC and GSM8K, performance improves steadily at first, reaching its peak in the fourth round. For instance, on the challenging AMC benchmark, accuracy rises from a single-round baseline of about 0.24 to nearly 0.31 after four rounds of interaction. These gains highlight the benefit of multi-round collaboration, where agents leverage peer feedback to correct early errors and converge on stronger solutions. However, extending the process beyond this optimal depth results in diminishing and eventually negative returns, with accuracy declining in the fifth round. This pattern suggests that excessive interaction introduces failure modes such as **error amplification**, where minor mistakes propagate and intensify, or **cognitive fixation**, where agents collectively reinforce a flawed line of reasoning. The findings underscore a key design trade-off: iterative refinement is valuable, but not universally beneficial. Effective systems therefore require mechanisms to identify when additional collaboration is productive and when it risks entrenching errors.

Table 4: Accuracy (%) on five benchmarks. The top block studies the effect of external experts and the DLPO meta-policy on top of the LIMA collaboration architecture. The bottom block compares DLPO against naive defer strategies, all with access to the same human expert.

Method	**GSM8K**	**AMC**	**MATH**	**HumanEval**	**MMLU**
Effect of external experts and DLPO					
LIMA (No Defer)	85.45	19.38	57.17	58.36	68.93
LIMA (Self Defer)	85.92	20.25	57.69	59.28	69.34
LIMA (Human Defer)	88.23	27.32	62.52	65.78	71.35
LIMA + DLPO (Self Defer)	86.71	24.52	59.28	61.54	69.91
LIMA + DLPO (Human Defer)	**91.25**	**30.30**	**65.46**	**67.82**	**73.58**
Learned vs. naive defer policies with human expert					
Random Defer (Human)	86.53	25.52	59.46	61.75	70.12
Uniform Defer (Human, Budget K)	87.82	26.87	62.81	65.35	70.59
Always Defer (Human)	94.38	40.27	68.85	84.32	81.45
LIMA + DLPO (Human Defer)	91.25	30.30	65.46	67.82	73.58

D Additional Experiments

D.1 Performance Gain from External

Experimental setup. We evaluate the impact of external high level feedback and the DLPO meta-policy on top of the LIMA collaboration architecture. The variant *LIMA (No Defer)* disables any external expert and runs the base multi agent system alone. *LIMA (Self Defer)* augments this system with a self expert: when the controller chooses to defer, it calls the same LIMA backbone to produce an additional candidate solution without introducing new external knowledge. *LIMA (Human Defer)* instead routes these defer actions to a pool of human domain experts, who provide full solutions and reasoning traces that act as high level feedback. The variants with "+ DLPO" train a metacognitive policy with GRPO to choose between create, evaluate and defer actions. *LIMA + DLPO (Self Defer)* uses the self expert as the teacher, while *LIMA + DLPO (Human Defer)* combines the learned policy with real human experts under a fixed consultation budget and also uses their demonstrations for continual SFT updates. We report accuracy on GSM8K, AMC, MATH, HumanEval and MMLU.

To study whether performance gains come from the meta-policy or from the presence of a strong external expert, we fix a human expert and compare DLPO against naive defer strategies under the same setting. *Random Defer (Human)* chooses to consult the human expert for each instance with a fixed probability, which yields an expected number of human calls close to the DLPO budget. *Uniform Defer (Human, Budget K)* selects a fixed number of instances to defer, matching the total number of human consultations used by DLPO, and distributes these calls uniformly across the evaluation set. *Always Defer (Human)* forwards every instance directly to the human expert and uses the human answer as the final prediction. This variant ignores any cost of inquiry and serves as an upper bound that corresponds to full manual solving rather than a realistic deployment.

Experimental analysis. The first block shows that external feedback and the DLPO meta-policy contribute in complementary ways. Moving from *LIMA (No Defer)* to *LIMA (Self Defer)* yields only mild gains, which indicates that extra rollouts from the same backbone provide limited benefit when no new knowledge is introduced. Replacing the self expert with human experts in *LIMA (Human Defer)* produces consistent improvements on all five benchmarks, especially on AMC, MATH and HumanEval, which confirms that high level human feedback is valuable even under a simple defer rule. Adding DLPO on top of self defer already brings a clear boost over *LIMA (Self Defer)*, most notably on AMC and MATH, which shows that a learned metacognitive policy helps the system decide when additional reasoning is worthwhile even without external knowledge. The full variant *LIMA + DLPO (Human Defer)* achieves the best accuracy among all cost aware methods and improves over *LIMA (Human Defer)* by several points on every benchmark. This pattern indicates that the performance gains are not only due to stronger feedback, but also due to the way DLPO selects when to defer and how it converts expert interactions into long term improvements.

Table 5: Accuracy and token usage on GSM8K and AMC as a function of the number of agents. We report accuracy together with average input, output, and total tokens per instance under a fixed backbone and decoding configuration.

# Agents	GSM8K	AMC	Avg. Input Tokens	Avg. Output Tokens	Avg. Total Tokens
1	0.8693	0.2337	1670	591	2261
2	0.8933	0.2745	3675	1273	4948
3	0.9125	0.3030	7910	2032	9942
4	0.9205	0.3257	13489	2782	16271
5	0.9216	0.3285	23316	3526	26842
6	0.9271	0.3123	29464	4280	33744
8	0.9405	0.3574	38584	5574	44158
10	0.9378	0.3528	50772	7184	57956

The second block isolates the effect of the learned defer policy under access to the same human expert. Both *Random Defer (Human)* and *Uniform Defer (Human, Budget K)* benefit from human feedback, yet they remain below *LIMA + DLPO (Human Defer)* on all benchmarks. The gap is especially visible on the more challenging datasets, where DLPO improves over random defer by roughly five points on AMC and six points on MATH and still outperforms uniform defer by a meaningful margin. This shows that selective, state dependent defer decisions are more effective than using the same human budget in a task agnostic way. The *Always Defer (Human)* variant achieves the highest raw scores but does so by fully offloading the task to humans and ignoring any cost of inquiry, which makes it an unrealistic reference point. Our method approaches this upper bound while using a limited number of consultations, which supports the claim that the framework improves performance not only by adding external feedback but by learning how to use that feedback efficiently.

D.2 Scaling to larger agent populations

Experimental setup. Most prior multi-agent work on math, reasoning, and code benchmarks evaluates relatively small teams, typically with at most five agents. Following this convention, our original submission focused on configurations up to four agents. In response to the review, we extend the study to larger populations and jointly measure accuracy and inference cost. Using the same backbone, decoding configuration, and meta-policy, we vary the number of agents on GSM8K and AMC from 1 up to 10. For each configuration we report task accuracy together with the average input, output, and total tokens per instance, as summarized in Table 5.

Experimental analysis. The results reveal a clear pattern. From 1 to 4 agents, accuracy improves sharply. For example, GSM8K rises from 0.8693 (1 agent) to 0.9205 (4 agents), and AMC from 0.2337 to 0.3257, while the average total token usage grows from 2,261 to 16,271 per instance. Beyond 4 agents, the gains become much smaller. Increasing the team size from 4 to 6 agents raises GSM8K only from 0.9205 to 0.9271 and leaves AMC at a similar level (0.3257 versus 0.3123), yet the total tokens increase from 16,271 to 33,744. With 8 and 10 agents, GSM8K and AMC continue to improve slightly (for example GSM8K 0.9405 at 8 agents and 0.9378 at 10 agents, AMC 0.3574 and 0.3528), but the total cost grows to 44,158 and 57,956 tokens per instance.

These extended results clarify that our earlier focus on four agents was not arbitrary. The case of four agents lies near a practical sweet spot where the marginal accuracy gain per additional agent begins to saturate, while the communication and inference cost continues to grow rapidly. In practice this suggests that collaborative configurations should be chosen as a task dependent trade off between accuracy and computational cost. We will add this scaling analysis and a performance–cost plot to the revised version to justify our emphasis on small to medium team sizes in the main experiments.

D.3 Approximate inter agent consistency via partial sampling

Experimental setup. You also raise a valuable concern about the complexity of full inter agent consistency checks. In the original design, we compute pairwise consistency features between

Table 6: Comparison between full pairwise consistency and partial sampling consistency for larger agent teams. We report accuracy on GSM8K and AMC and the average total tokens per instance for each variant.

#Agents	GSM8K (Full)	GSM8K (Partial)	AMC (Full)	AMC (Partial)	Avg. (Full)	Avg. (Partial)
5	0.9216	0.9187	0.3285	0.3217	26842	15315
6	0.9271	0.9156	0.3123	0.3025	33744	17861
8	0.9405	0.9332	0.3574	0.3516	44158	22758
10	0.9378	0.9283	0.3528	0.3362	57956	26394

agents, which has quadratic cost in the number of agents. We agree that this becomes prohibitive for very large populations and we appreciate you highlighting this point.

To address this, we conduct an additional ablation inspired by communication efficient multi agent methods. We test a simple approximate scheme in which each agent only compares its outputs with a sampled subset of peers. In this *partial sampling* variant, each agent communicates with at most $K = 3$ other agents, which reduces the complexity from $O(n^2)$ pairwise checks to approximately $O(nK)$. We evaluate this variant on GSM8K and AMC with 5, 6, 8, and 10 agents, and we report accuracy together with the average total tokens per instance. The results are summarized in Table 6.

Experimental analysis. On both GSM8K and AMC, partial sampling remains very close to full pairwise consistency in terms of accuracy. For example, with 8 agents, GSM8K changes from 0.9405 (full) to 0.9332 (partial), and AMC from 0.3574 to 0.3516. With 10 agents, GSM8K changes from 0.9378 to 0.9283 and AMC from 0.3528 to 0.3362. In most cases the differences are within about one absolute point. In contrast, the cost reduction is substantial. At 8 agents, the average total tokens decrease from 44,158 (full) to 22,758 (partial), and at 10 agents from 57,956 to 26,394, which corresponds to roughly a 40–50% reduction in communication and inference cost.

These results show that the metacognitive policy and consistency features do not require dense all to all communication. A simple sampled consistency scheme preserves most of the performance while significantly lowering cost and scaling more gracefully with the number of agents. Our framework is also orthogonal to existing communication optimization techniques, such as methods that learn sparse interaction graphs or prune redundant agents. LIMA can be combined with such approaches so that the meta policy operates on a communication topology that is already optimized for larger populations.

D.4 Sequential multi task learning and forgetting

Experimental setup. To study long term capability growth and potential catastrophic forgetting, we follow a sequential two task protocol. We treat GSM8K as Task A and AMC as Task B. Starting from the same base LIMA backbone and meta policy, we evaluate three stages:

1. **Base.** No DLPO training. We evaluate the initial model on GSM8K and AMC and obtain accuracies A_0 and B_0.
2. **After Task A.** We run the full DLPO procedure with expert feedback only on GSM8K. We collect defer demonstrations, update the base model with outer loop SFT, and then evaluate the updated model on both tasks, yielding A_1 and B_1.
3. **After Task A $\rightarrow$ B.** Starting from the GSM8K trained model, we run DLPO with expert feedback on AMC. During outer loop SFT we replay demonstrations from both GSM8K and AMC, so the model sees a mixture of Task A and Task B examples. We then evaluate again on both tasks, obtaining A_2 and B_2.

In addition to DLPO with replay, we include a *naive sequential SFT* baseline that uses the same expert demonstrations but does not use DLPO or replay. This baseline first fine tunes on GSM8K demonstrations only, then fine tunes on AMC demonstrations only. It corresponds to the standard sequential fine tuning setting that is known to induce catastrophic forgetting. All reported values are accuracies in percent.

Table 7: Sequential learning across GSM8K (Task A) and AMC (Task B). We report accuracy (%) for DLPO with replay and a naive sequential SFT baseline at three stages: Base (no training), After Task A (trained on GSM8K only), and After Task A→B (sequentially trained on GSM8K then AMC).

Method	Stage	GSM8K Accuracy (A)	AMC Accuracy (B)
DLPO (ours)	Base	88.23	27.32
DLPO (ours)	After Task A	91.45	26.78
DLPO (ours)	After Task A→B	90.97	32.58
Naive sequential SFT	Base	88.23	27.32
Naive sequential SFT	After Task A	90.47	22.65
Naive sequential SFT	After Task A→B	88.71	29.36

Experimental analysis. The results in Table 7 show clear differences between DLPO with replay and naive sequential SFT. For DLPO, the base model starts at 88.23 on GSM8K and 27.32 on AMC. After DLPO on GSM8K, GSM8K improves to 91.45, while AMC remains roughly similar at 26.78. After further DLPO on AMC with replay, AMC increases to 32.58 and GSM8K remains high at 90.97. The forgetting on GSM8K is modest: the drop from 91.45 to 90.97 is 0.48 points, while the gain over the base model is still more than 2.7 points.

By contrast, the naive sequential SFT baseline exhibits much stronger forgetting. Using the same GSM8K demonstrations, sequential SFT improves GSM8K from 88.23 to 90.47, but reduces AMC from 27.32 to 22.65. After fine tuning only on AMC demonstrations, AMC rises to 29.36 while GSM8K drops to 88.71. The drop from 90.47 to 88.71 corresponds to about 1.8 points of forgetting, which removes most of the earlier gain on GSM8K.

Overall, these results indicate that the DLPO outer loop, when combined with replay of earlier defer demonstrations, can support sequential learning with limited forgetting: the model gains new capability on AMC while retaining most of the improvements on GSM8K. In contrast, naive sequential SFT with the same expert data shows significantly stronger degradation on GSM8K after training on AMC. This provides empirical evidence that the framework is not only an offline performance booster on fixed benchmarks, but also a reasonable starting point for long term capability growth across a sequence of related reasoning tasks. In the revised version, we soften the language around "long term capability growth" to clarify that the current evidence is for multi task, sequential improvements in a two task setting, and that extending DLPO to longer task sequences and richer continual learning benchmarks is an important direction for future work.

D.5 Effect of Different LLM Experts for the Human Proxy

Experimental setup. In the main experiments, we used GPT-4o-mini as the default proxy for the "human expert" channel. To examine how the framework behaves with experts of different strengths, we keep the backbone model, collaboration architecture, and DLPO configuration fixed, and vary only the model used as the external expert. We compare three choices: a LLaMA3 model (weaker LLM expert), GPT-4o-mini, and GPT-4o. We report accuracy on GSM8K, AMC, MATH, HumanEval, and MMLU.

Experimental analysis. Table 8 shows two consistent patterns. First, as the expert model becomes stronger (from LLaMA3 to GPT-4o-mini to GPT-4o), performance improves monotonically on all five benchmarks. For example, AMC increases from 24.52 to 30.30 to 37.25, and HumanEval from 61.54 to 67.82 to 74.61. This confirms that the framework scales smoothly with expert capability and that our main results with GPT-4o-mini are conservative relative to what GPT-4o can achieve. Second, the qualitative behavior of the method is stable across all expert choices: in each case, adding an external expert on top of LIMA yields clear gains, and the DLPO policy continues to extract additional benefit under a cost-aware defer scheme. In practice, the choice between GPT-4o-mini and GPT-4o is therefore an application-level trade-off between accuracy and inference cost rather than a limitation of the framework itself.

Table 8: Accuracy (%) with different LLM experts used as the human proxy. We keep the backbone and collaboration architecture fixed and vary only the expert model.

Human Proxy	GSM8K	AMC	MATH	HumanEval	MMLU
LLaMA3-8B	86.71	24.52	59.28	61.54	69.91
GPT-4o-mini	91.25	30.30	65.46	67.82	73.58
GPT-4o	93.58	37.25	68.37	74.61	75.42

E Declaration on the Use of Large Language Models

In preparing this work, we made use of several large language models for different purposes. First, **GPT-4o-mini** and **GPT-4o** were integrated directly into our experimental framework, where they served as proxies for human experts in the human-in-the-loop setting. This design choice follows prior research and allowed us to evaluate the framework under controlled and repeatable conditions while balancing cost and effectiveness. Second, **GPT-5** was employed to assist with improving the clarity, organization, and readability of the manuscript. The model helped refine phrasing and grammar, but all conceptual contributions, methodological design, and experimental analysis were developed by the authors. All content was carefully reviewed, edited, and validated by the authors, who take full responsibility for the accuracy and integrity of the final publication.