DO MULTI-AGENT LLMS EXPLORE EACH OTHER?

Anonymous authors
Paper under double-blind review

ABSTRACT

Exploration is essential for reliable autonomy in multi-agent systems, yet it remains unclear whether large language model (LLM) agents can explore effectively when interacting with one another. We show that modern LLM agents often fail to do so, exhibiting myopic and polarized interaction patterns that lead to suboptimal coordination and increased regret. We formalize this challenge as the Multi-Agent Exploration problem, modeling it as a partially observable stochastic game (POSG), in which agents must probe peers to infer their capabilities and identify effective interaction strategies. To address this, we introduce Multi-Agent Contextual Exploration (MACE), a lightweight framework that explicitly promotes exploration through structured peer selection. Across both contextual and parametric diversity settings, exploration behavior and downstream task performance substantially improve when MACE is applied. We further provide theoretical insight that MACE achieves a sublinear expected-regret bound. Overall, our results highlight a fundamental limitation of current LLM agents and underscore the importance of explicitly guided exploration for reliable multi-agent autonomy. Code will be released for reproducible research.

1 INTRODUCTION

Large language models are increasingly deployed not only as isolated assistants, but as autonomous agents embedded in multi-agent systems: communicating, delegating, and making decisions in a decentralized fashion to accomplish complex tasks (Li et al., 2023; Wu et al., 2024; Qian et al., 2024; Hong et al., 2023; Gao et al., 2024; Du et al., 2024). This paradigm is rapidly expanding: open-world platforms now instantiate heterogeneous agent populations that interact autonomously to discuss problems, divide labor, and collectively reason toward shared goals (Piao et al., 2025; Park et al., 2023; Jiang et al., 2026; Zhang et al., 2026). Such systems implicitly rely on the assumption that agents are capable of autonomous decision-making, and that meaningful collective behavior can emerge from agent-to-agent interactions.

But what is required for agents to act autonomously in a reliable manner? A large body of literature in intrinsic motivation and reinforcement learning has demonstrated that *exploration* is foundational to autonomous behavior (Pathak et al., 2017; Burda et al., 2019). Exploration is not merely a mechanism for improving task performance, but the core process by which agents self-generate goals, discover novel states, and acquire reusable strategies without external guidance (Chentanez et al., 2004; Baker et al., 2019; Forestier et al., 2022). The importance of exploration is further amplified in multi-agent settings, where the environment is shaped not only by environmental dynamics, but also by the behaviors and capabilities of other agents, which are often heterogeneous and can only be revealed through direct interaction. Thus, in such settings, agents must proactively explore peers to identify effective collaborators, uncover complementary information, and adapt as the interaction evolves.

Yet despite the centrality of exploration to reliable autonomy, we find that current LLM agents fail to explore effectively, even in the simplest possible settings. We begin with a simple autonomous multi-agent setting—a controlled two-armed bandit experiment, where an LLM agent must repeatedly choose between two peers with unknown success rates and must infer the better one through exploration. Rather than accumulating evidence and converging to the superior peer, Section 2 reveals that LLM agents exhibit premature commitment, locking onto one peer within the first few rounds and persisting regardless of correctness. Critically, this failure is not confined to smaller or less capable models: even capable frontier models like GPT-4 (Achiam et al., 2023) and GPT-5 (OpenAI,

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

2025) exhibit the failure mode, suggesting that insufficient exploration is a fundamental and structural limitation of current LLM agents rather than a capacity deficit.

To characterize this rigorously and capture real-world complexities, we formalize the ***multi-agent exploration problem***, in which agents differ in capabilities or possess distinct contextual information. Multi-agent exploration thus requires actively probing other agents strategically and identifying effective interaction strategies. Unlike standard bandit settings where arms are stationary (Auer et al., 2002), peers here are themselves adaptive agents whose responses evolve as they accumulate their own interaction experience. Moreover, agents operate in a decentralized manner, making decisions based only on partial and dynamically evolving information shaped by ongoing interactions, and exploration failures compound across agents and rounds simultaneously.

To address this, we introduce **Multi-Agent Contextual Exploration (MACE)**, a lightweight solution for the multi-agent exploration problem, promoting exploration through explicit algorithmic guidance. Since solving the multi-agent exploration problem exactly is intractable, the key idea behind MACE is to decompose the joint problem into independent sequential decision-making problems under contextual bandits, where the non-stationarity of multi-agent interactions is encoded into a novel structural feature representation. Specifically, MACE encodes the *relational structure* of each potential interaction into a contextual representation that reflects the utility of querying a given peer in a given context. This allows MACE to distinguish between a peer that is globally under-tested and one that is under-tested in the specific relational context of the current task, providing a finer-grained form of exploration.

Empirically, MACE substantially improves existing baselines across heterogeneous agent settings spanning contextual and capability diversity. Interestingly, in-context exploration—where agents are explicitly prompted to balance exploration and exploitation using the same information available to MACE—often underperforms even random peer selection, confirming that the failure cannot be remedied through prompting alone and that algorithmic structure is necessary. MACE's gains persist into an exploitation phase with frozen parameters and transfer to unseen benchmarks, indicating that the learned interaction strategies capture generalizable structure rather than task-specific artifacts. Theoretically, we analyze peer selection under a realizable linear-bandit model. We show that MACE achieves $O(\sqrt{T}\log T)$ expected cumulative regret. We also construct a stationary instance on which empirical-mean greedy selection incurs $\Omega(T)$ expected regret through premature commitment. Together, these results establish an asymptotically linear expected-regret separation on the constructed instance, illustrating how explicit exploration can prevent persistent peer-selection errors.

We summarize our contributions as follows:

- We formalize the Multi-Agent Exploration problem, motivated by the observation that modern LLM agents often exhibit limited exploration in multi-agent environments. This is fundamental yet underexplored for reliable autonomy in open-world multi-agent settings.
- We introduce Multi-Agent Contextual Exploration (MACE), a lightweight and tractable solution to address the problem via contextual bandit-driven peer selection with explicit exploration incentives.
- We show both empirically and theoretically that explicit exploration significantly improves task performance, and how explicit exploration can prevent persistent peer-selection errors. Our evaluation spans seven models across four major LLM families, involving three benchmarks.

2 Do LLMs Explore in Multi-Agent Environments?

Exploration is a fundamental prerequisite for reliable autonomy of a multi-agent system. An agent must not only act on current beliefs, but also actively probe its environment to discover better strategies and collaborators. In multi-agent LLM systems, this challenge of exploration manifests as deciding which peers to trust, under uncertainty about their capabilities. Despite their strong abilities, it remains unclear whether LLM agents can autonomously balance exploration and exploitation when interacting with other agents. To investigate this, we study a simplified delegation scenario centered on a single agent, isolating the core decision problem of selecting peers with unknown reliability.

Setup. Consider an LLM agent that must repeatedly delegate tasks to one of two peer LLMs, knowing nothing about their abilities upfront. Each delegation reveals a small piece of evidence (*i.e.*, whether the peer succeeds or fails), and the agent must decide how to use that evidence. The agent can keep

testing the peer it knows less about, or commit to the one that has performed well so far. This is the classic exploration-exploitation dilemma in its purest form, and it is precisely the kind of judgment that reliable multi-agent coordination demands.

To study whether LLM agents can exercise this judgment, we design a two-armed bandit setting where an agent must repeatedly delegate arithmetic questions to one of its two peers, A or B. Each peer has a fixed but unknown probability of answering correctly (p_A and p_B respectively). At every round, the agent observes the historical performance and the number of times it delegated to each peer, and is prompted to select an agent. Ideally, the agent should balance exploration (trying less-tested peers) and exploitation (selecting the empirically better one). Specifically, we test settings with probability $(p_A, p_B) = (0.6, 0.5)$, where one peer is better on average, but with a noisy performance gap. This setting requires the agent to perform sufficient exploration before converging to optimal exploitation. Additional experiments with varying p_A values are provided in Appendix A.2.

Behavioral reference. We evaluate three LLM agents on this task: Qwen2.5-7B-Instruct (Yang et al., 2024), GPT-4 (Achiam et al., 2023) and GPT-5 (OpenAI, 2025). In all cases, peer selection is performed purely through in-context information: at each round, the agent is presented with running history and prompted to select a peer. To anchor our interpretation of their behavior, we include a classical exploration algorithm as behavioral reference: Upper Confidence Bound (UCB) (Auer et al., 2002). Unlike the LLM agents, UCB imposes explicit mathematical structure on the selection decision that provably incentivizes exploration. More specifically, UCB maintains an optimistic estimate of each peer's true success rate, and selects the peer with the highest such estimate. This optimism-under-uncertainty principle naturally drives exploration toward under-tested peers while transitioning to exploitation as evidence accumulates. We include UCB as a concrete illustration of what algorithmically guaranteed exploration looks like in this setting, making the absence of exploration in LLM agents directly visible. See Appendix A.1 for further descriptions on the UCB algorithm and other experimental details.

LLM agents exhibit insufficient exploration. Figure 1 shows the distribution of peer A selection frequency across 30 independent runs, where each run consists of $T = 50$ delegation rounds. Optimal behavior corresponds to predominantly selecting peer A while still occasionally exploring peer B. We observe that the behavioral reference UCB (in brown) exhibits such desired pattern: the distribution is concentrated toward higher selection counts, indicating effective exploration followed by consistent exploitation of the better peer.

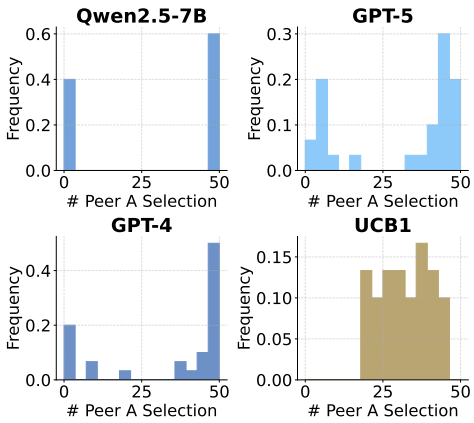


Figure 1: Histogram of the number of peer A selection.

In stark contrast, all LLM agents display highly polarized bimodal distributions, with mass concentrated near both extremes (close to 0 or 50 selections out of 50 rounds). This pattern reveals a characteristic failure mode: rather than gradually accumulating evidence and converging to a reliable preference, LLM agents commit prematurely to one peer within the first few rounds and maintain that commitment for the remainder of the task. Crucially, this early commitment is not always correct—the mass near 0 indicates frequent runs where the agent locked onto the inferior peer and never reconsidered. Overall, LLM agents do not exhibit an innate capability to explore and reliably identify the better option. More detailed analyses are in Appendix A.2.

3 The Multi-Agent Exploration Problem: A Formalization

The observations in Section 2 reveal that modern LLM agents fail to exhibit sufficient exploration in its simplest form. In realistic multi-agent systems, however, this challenge is further amplified. Agents operate in a decentralized manner, making decisions based only on partial and dynamically evolving information shaped by ongoing interactions. The space of possible interaction protocols is vast, the capabilities and context of peers are heterogeneous, and the consequences of poor exploration compound across agents and rounds simultaneously. To capture these complexities, we formalize the problem of ***multi-agent exploration***.

3.1 Problem Formalization

We consider a system of $\mathcal{N} = \{1, ..., N\}$ agents interacting over T timesteps to solve a sequence of tasks. Each agent may possess distinct capabilities, which may arise from differences in model family or available context. And these differences are latent: no agent has direct access to the capabilities of its peers, and they become partially observable through interactions. Each agent i observes its selected peer's response, accumulated over interaction rounds. Multi-agent exploration thus arises as a problem of actively probing other agents to infer their capabilities from partial observation history and identify effective interaction strategies.

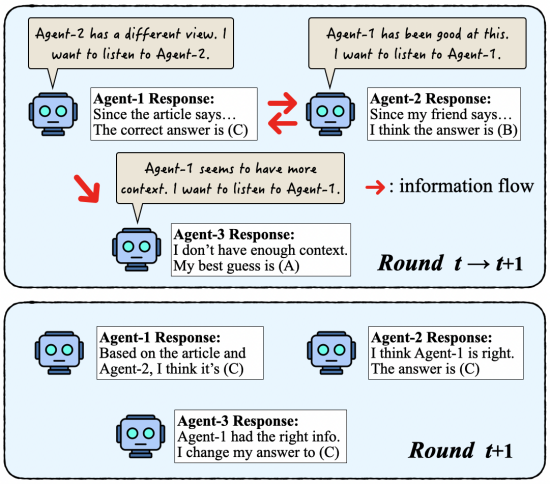


Figure 2: **The Multi-Agent Exploration problem**. At round t, each agent needs to select a peer to interact with, and update its response in the subsequent interaction round.

Formally, we model this process as a Partially Observable Stochastic Game (POSG), defined by the tuple $\mathcal{G} = \langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{P}, \mathcal{Q}, \mathcal{R} \rangle$. Here, latent state $s_t \in \mathcal{S}$ encodes all task-relevant information together with the unknown capabilities of every agent in the system. This encapsulates everything that governs what an effective interaction strategy would look like, yet is not fully observable to anindividual agent. At each round $t \in \{1, \ldots, T\}$, agent $i \in \mathcal{N}$ selects action $a_{i,t} \in \mathcal{N}$, designating the peer to interact with. The joint action $\mathbf{a}_t = (a_{1,t}, \ldots, a_{N,t}) \in \mathcal{A} = \mathcal{N}^N$ captures the system-level interaction structure at round t. The transition function $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ governs how information propagates through the system as a consequence of $\mathbf{a}_t$, inducing a dynamically evolving interaction graph whose topology is itself determined by the agents' decisions. The high-level architecture of the multi-agent exploration problem is illustrated in Figure 2.

Upon taking action $a_{i,t}$, agent i receives an observation $o_{i,t} \in \mathcal{O}_i$ drawn from the observation function $\mathcal{Q} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{O})$, where $\mathcal{O} = \prod_{i=1}^{N} \mathcal{O}_i$.

The observation $o_{i,t}$ consists of the queried peer's response together with the agent's accumulated interaction history. Crucially, agents do not have access to the full state $\mathcal{S}$, and must instead act based on partial observations accumulated over time from its selected peers. That is, before selecting a peer, each agent observes its own current response, local interaction statistics, and summary features computed from all agents' previous-round responses. Only after selection it receives the selected peer's full response. Finally, the reward $r_{i,t} = R_i(s_t, \mathbf{a}_t)$ for agent i, drawn from $\mathcal{R} = \{R_i\}_{i=1}^{N}$ with $R_i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, reflects task performance (e.g., the correctness of the answer, the gain in performance from the round, or the utility of intermediate responses).

3.2 Regret

Each agent $i \in \mathcal{N}$ should ideally minimize the cumulative regret over T rounds with respect to the peer with the highest conditional expected reward at each round:

$$\text{Regret}_i := \sum_{t=1}^{T} \left(\max_{a \in [N]} \mu_{i,a}^{(t)} - \mu_{i,a_{i,t}}^{(t)} \right), \tag{1}$$

where $\mu_{i,a_{i,t}}^{(t)}$ is the expected reward for agent i selecting peer $a_{i,t}$ at round t. To characterize the practical effect of insufficient exploration, we additionally consider the expected cumulative reward: $\mathbb{E}\left[\sum_{t=1}^{T} r_{i,t}\right]$, where $r_{i,t}$ denotes the reward obtained at round t. A central challenge in this setting is the exploration-exploitation trade-off under partial observability and agent heterogeneity. Effective policies must balance: (i) *exploration*, *i.e.*, querying diverse peers to estimate their capabilities and reduce uncertainty and regret, and (ii) *exploitation*, *i.e.*, selecting peers believed to yield high rewards. In the next section, we introduce exploration-driven strategies that address this challenge.

4 MULTI-AGENT CONTEXTUAL EXPLORATION (MACE)

While our formalization captures the full structure of the multi-agent exploration problem, solving the finite-horizon POSG problem is NEXP-hard. The core difficulty is that an agent must estimate peer capabilities from sparse, noisy interaction history, while the peers themselves are adaptive (*i.e.*, their responses evolve as they accumulate their own interaction experience). To address the challenge, we introduce **Multi-Agent Contextual Exploration** (**MACE**), a tractable and lightweight solution that explicitly promotes exploration in multi-agent interactions by decomposing the joint problem into independent per-agent decisions. Rather than requiring complex coordination protocols, MACE treats peer selection as a learning problem: each agent adaptively chooses which peer to query based on past observations and contextual signals, with an explicit mechanism to avoid premature commitment.

Contextual bandit formulation. MACE models multi-agent interaction as a contextual multi-armed bandit problem, where each agent independently learns which peer to query at each round. For agent i, selecting a peer $a \in \mathcal{N}$ at round t corresponds to pulling an arm, and the expected reward is modeled as a linear function: $\mu_{i,a,t} \approx \mathbf{x}_{i,a,t}^{\top}\boldsymbol{\theta}_{i,a}$, where $\mathbf{x}_{i,a,t} \in \mathbb{R}^d$ is a context feature vector encoding the interaction between agent i and peer a at timestep t, and $\boldsymbol{\theta}_{i,a} \in \mathbb{R}^d$ is the corresponding weight vector. This reduction to the contextual bandit problem is non-trivial: unlike standard bandit settings where arms are stationary (Auer et al., 2002), the peers here are themselves adaptive agents.

Before each interaction round, a feature-extraction module computes summary features from the agents' previous-round responses. Each selector receives these features for all candidate peers, together with its own response and local interaction statistics. After selecting a peer, the selector receives that peer's full response and uses it to revise its answer. Thus, access to full peer responses is restricted, while the selection policy has access to population-wide response summaries. Both MACE and In-Context Exploration receive the same pre-selection features.

Relational feature design. While scalar performance summaries are simple, they may fail to capture the non-stationary evolution of agent capabilities over time. Therefore, instead of relying solely on scalar summaries, we propose constructing these features to encode the relational structure of the interaction between agent i and peer a. (i) *Response diversity* captures the degree to which peer a's response diverges from agent i's current answer, encoding the potential information gain from the interaction; (ii) *Peer distinctiveness* measures how distant peer a is to others within the agent network; (iii) *Historical performance* provides an exploitation signal via the peer's empirical success rate across prior rounds; and (iv) the *Interaction round* encodes the interaction round within a sample, allowing the agent to modulate exploration intensity over time. We provide detailed descriptions of the features in Appendix B.1, and discuss the learned feature weights in Appendix D.6.

Exploration via optimism. For each agent i and candidate peer $a \in \mathcal{N}$, we maintain a design matrix $\mathbf{A}_{i,a} \in \mathbb{R}^{d\times d}$ and reward vector $\mathbf{b}_{i,a} \in \mathbb{R}^d$, initialized as $\mathbf{A}_{i,a} = \lambda\mathbf{I}$ and $\mathbf{b}_{i,a} = \mathbf{0}$, where $\lambda > 0$ is a regularization parameter. These statistics yield a ridge-regression estimate of the expected reward model weights:

$$\hat{\boldsymbol{\theta}}_{i,a} = \mathbf{A}_{i,a}^{-1}\mathbf{b}_{i,a}. \tag{2}$$

At interaction round t, given contextual features $\mathbf{x}_{i,a,t}$ describing the potential interaction between agent i and peer a, agent i selects a peer according to the LinUCB (Li et al., 2010) rule:

$$a_{i,t} = \arg\max_{a\in[N]} \left[\hat{\boldsymbol{\theta}}_{i,a}^{\top}\mathbf{x}_{i,a,t} + \alpha\sqrt{\mathbf{x}_{i,a,t}^{\top}\mathbf{A}_{i,a}^{-1}\mathbf{x}_{i,a,t}}\right], \tag{3}$$

where the first term estimates the expected reward of querying peer a, and the second term is an uncertainty bonus that encourages exploration, and $\alpha > 0$ is the exploration coefficient (see Appendix D.2 for sensitivity analysis). Critically, because the uncertainty bonus operates over the relational feature space rather than raw selection counts, the agent can distinguish between a peer that is under-tested globally and one that is under-tested in the specific relational context of the current task. This provides a finer-grained form of exploration than scalar UCB affords.

Sufficient statistic updates. After selecting peer $a_{i,t}$, agent i receives reward $r_{i,t}$, defined as the average of the performance metric and the improvement of its response after incorporating information from the selected peer. The sufficient statistics are then updated as

$$\mathbf{A}_{i,a'} \leftarrow \mathbf{A}_{i,a'} + \mathbf{x}_{i,a',t}\mathbf{x}_{i,a',t}^{\top} \;\;,\;\; \mathbf{b}_{i,a'} \leftarrow \mathbf{b}_{i,a'} + r_{i,t}\mathbf{x}_{i,a',t} \qquad (a' := a_{i,t}), \tag{4}$$

after which the parameter estimate $\hat{\boldsymbol{\theta}}_{i,a}$ is recomputed via equation 2. The reward is computed based on both the agent's performance and the magnitude of improvement. Detailed reward function description is in Appendix B.2, and the algorithm pseudocode is provided in Appendix B.3.

5 EXPERIMENTS

5.1 AGENT HETEROGENEITY SETUP

To evaluate exploration in heterogeneous multi-agent environments, we consider two representative sources of heterogeneity commonly encountered in realistic agent ecosystems: *contextual diversity* and *parametric diversity*. These settings capture complementary forms of uncertainty that agents must navigate when selecting collaborators and communication strategies.

Exploration under contextual diversity. This setting models environments in which agents have access to different subsets of the global context, such as distinct retrieved documents, memories, or tools some of which may be incomplete, irrelevant, or misleading. Each agent is unaware of the contextual information available to its peers, and must therefore explore which collaborators have the necessary information to solve a task through interaction. To simulate this setting, we adopt the distractor configuration of HotpotQA (Yang et al., 2018), which contains 10 disjoint contexts: 2 relevant evidence passages and 8 irrelevant distractors. Each evidence passage is then distributed across 10 `Qwen2.5-7B-Instruct` (Yang et al., 2024) agents. For each data sample, we let the agents engage in $R = 5$ rounds of discussions, where each round corresponds to a single timestep.

Exploration under parametric diversity. This setting models environments in which agents differ in intrinsic reasoning ability due to heterogeneous model families or parameter scales, as in realistic open-agent ecosystems. To simulate this setting, we construct a multi-agent system composed of heterogeneous LLMs, comprising 4 model types: {`GPT-5` (OpenAI, 2025), `Qwen2.5-7B-Instruct` (Yang et al., 2024), `Llama3.1-8B-Instruct` (Grattafiori et al., 2024), and `Mistral-7B-v0.3` (Jiang et al., 2023)}. For each sample, agents interact for $R = 3$ rounds on two downstream tasks, Math500 (Lightman et al., 2023) and GPQA (Rein et al., 2024). Note that we use fewer interaction rounds in this setup, as it involves only four agents.

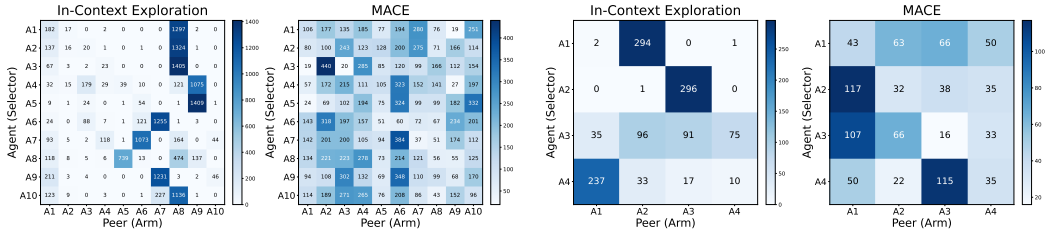
5.2 EXPERIMENTAL DETAILS

Baselines. We compare MACE against three baselines: (1) *In-Context Exploration*: agents make peer selection decisions autonomously using only in-context information. At each round, each agent is presented with the full interaction history, including how many times each peer has been queried and whether their responses were correct, and is explicitly prompted to balance exploration and exploitation. Crucially, this baseline has access to the same information as MACE but relies entirely on the LLM's in-context reasoning to act on it, with no algorithmic structure imposed on the selection decision. (2) *Random*: At each interaction step, agents uniformly sample a peer to interact with at random. (3) *Pre-defined*: The connectivity between agents is pre-defined, where each agent chooses its right-side neighbor as its peer. This shares the spirit of most multi-agent debate architectures with fixed communication structures (Du et al., 2024; Li et al., 2024b).

Evaluation. For contextual diversity, we evaluate answer quality and regret computation on HotpotQA distractor mode (Yang et al., 2018) using Exact Match (EM) and F1, and restrict evaluation to 600 samples labeled as hard. For parametric diversity, we evaluate final-answer accuracy on `Math500` (Lightman et al., 2023) and `GPQA` (Rein et al., 2024). Specifically, we use Level 4-5 problems from Math500 and the Diamond split of GPQA. Furthermore, for each setup, we evaluate performance across two phases. During the *trial-and-error phase*, all methods interact with peers and accumulate observations. During the *exploitation phase*, the linear parameters learned by MACE are frozen and reused without further updates; In-Context Exploration retains its accumulated interaction statistics; and Random and Pre-defined, which maintain no learnable parameters, operate identically to their exploration phase. Each benchmark dataset is split evenly into two halves: the first half is used for the trial-and-error phase, and the second half is reserved for the exploitation phase. Additional experimental details and prompt templates are provided in Appendix C.

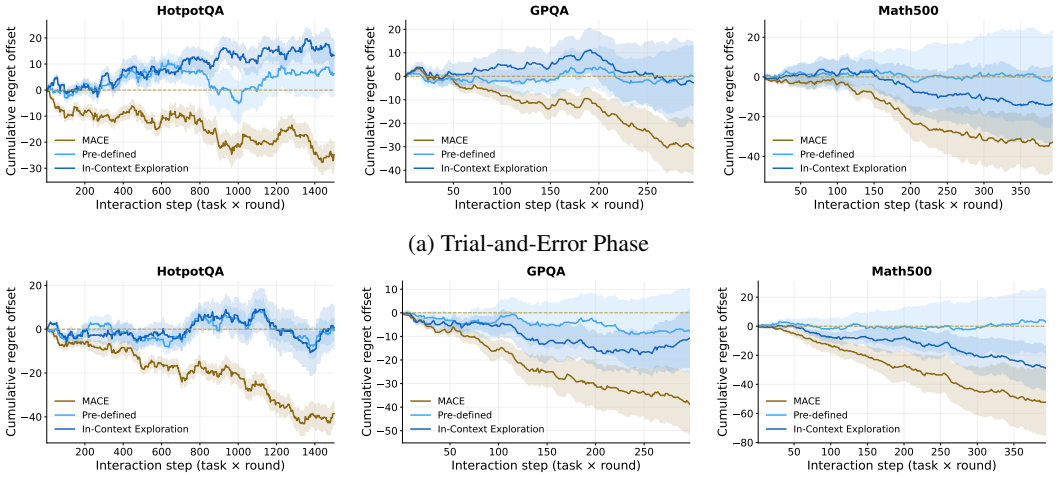
5.3 RESULTS AND DIAGNOSIS

LLM agents exhibit insufficient exploration in multi-agent environments. Figure 3 visualizes peer-selection distributions under exploration settings with both contextual and parametric diversity.



(a) Contextual Diversity (b) Parametric Diversity

Figure 3: **Peer Selection distribution comparison of In-context Exploration and MACE**. (a) HotpotQA benchmark; All 10 agents are Qwen2.5-7B-Instruct, each with heterogeneous context. (b) GPQA benchmark; A1–A4 agents are GPT-5, Qwen2.5-7B-Instruct, Llama3.1-8B-Instruct, and Mistral-7B-v0.3, respectively.



(a) Trial-and-Error Phase

(b) Exploitation Phase

Figure 4: **Cumulative regret offset across tasks and phases.** The cumulative regret offset (Baseline − Random), averaged across agents, is shown over interaction steps (lower is better). The horizontal dashed line at Offset = 0 is the Random baseline, and the shaded areas indicate the standard error across participating agents.

In Figure 3(a), corresponding to contextual diversity, In-Context Exploration produces highly concentrated selection patterns, with agents often locking onto a single peer. This is undesirable because the agent holding the relevant context is randomized across samples and interaction rounds; consistently selecting the same peer therefore signals a clear failure to explore. By contrast, MACE induces substantially more distributed selections, suggesting better coverage of the available peers. A different pattern appears in Figure 3(b), corresponding to parametric diversity. Here, some convergence toward a generally strong agent is expected. However, In-Context Exploration still exhibits unnaturally sharp and poorly calibrated concentration, with agents collapsing onto what appears to be an arbitrary early favorite rather than exploring the full pool. For example, even GPT-5 (A1) selects Qwen-7B (A2) in 294 out of 297 interactions, while almost never querying the other agents. This suggests that LLMs do not reliably identify strong collaborators through systematic exploration, but instead overcommit prematurely to early signals. In contrast, MACE produces more balanced selection patterns, indicating more stable and effective exploration before exploitation.

In-context exploration may underperform random exploration, while explicit guided-exploration yields large gains. In Figure 4, we compare cumulative regret offsets across exploration strategies, defined as the difference in cumulative regret between a given strategy and the random baseline, *i.e.*, $\text{Regret}_i^{\text{Strategy}} - \text{Regret}_i^{\text{Random}}$ (equation 1). The figure reveals a striking result that under *contextual diversity* (top row), In-Context Exploration performs worse than Random peer selection, indicating that autonomous LLM agents not only fail to explore effectively, but can be less reliable than a naive stochastic policy. This suggests that current LLM agents relying on prompting alone produce systematically poor exploration behavior in multi-agent interactions. In contrast, MACE substantially improves performance throughout the exploration phase, demonstrating that even lightweight exploration guidance can dramatically enhance interaction quality. These gains persist into the exploitation phase (bottom row), where the learned policies continue to outperform all baselines after exploration is deliberately disabled. We observe similar trends under parametric diversity setup on both Math500 and GPQA, demonstrating that the benefits of explicitly guided

exploration generalize across heterogeneous model pools and task domains. In Appendix D.1, we also show the task performance for each interaction round. Notably, even the strongest model in the system, GPT-5, benefits from the proposed exploration mechanisms (Table 10–17), suggesting that insufficient exploration is a fundamental limitation of current multi-agent LLMs in general.

Strategies learned by MACE generalize to an unseen task. We evaluate transferability by applying the parameters learned on HotpotQA to a more challenging benchmark, 2WikiMultihopQA (Ho et al., 2020) (details in Appendix C.1). As in the contextual diversity setup, each agent receives partial context, with only a subset having access to the information required to solve the question. All other baselines explore on a 300-sample subset of benchmark, while MACE operates purely in exploitation mode with frozen parameters. Despite never being trained on this benchmark, MACE achieves robust performance that outperforms all baselines in Figure 5. This indicates that the learned interaction strategies capture generalizable structure rather than task-specific artifacts.

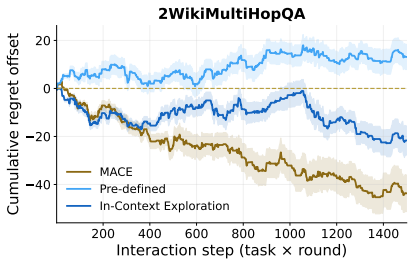


Figure 5: The parameters learned from HotpotQA can be transferred to a different benchmark, 2WikiMultiHopQA (Ho et al., 2020).

The following additional analyses are deferred to the Appendix: ① Interaction round-wise effect on task performance (Appendix D.1), ② Sensitivity analysis on the exploration coefficient α (Appendix D.2), ③ Experiments with a set of stronger frontier models (Appendix D.3), ④ Homogeneous agent setting experiments (Appendix D.4), ⑤ Temporal-Difference algorithm extension of MACE and discussions on potential emergence of cooperative behaviors (Appendix D.5), ⑥ Feature selection analyses (Appendix D.6).

6 Theoretical Analysis

In this section, we provide formal guarantees on the benefit of MACE with explicit exploration under simplified conditions. Theorem 1 establishes a sublinear expected-regret bound for MACE under the stated assumptions. Theorem 2 constructs a stationary instance on which empirical-mean greedy selection incurs linear expected regret through premature commitment. Corollary 1 then establishes an asymptotically linear expected-regret separation on the constructed instance. We specify assumptions for our theorems in Appendix E.1, and all proofs are in Appendix E.

Definition 1 (**Capability Diversity**). *Let each agent $a \in \mathcal{N}$ be associated with a latent capability vector $\mathbf{c}_a \in \mathbb{R}^k$, encoding its strengths across reasoning domains or knowledge areas. The capability diversity of the agent pool is defined as:*

$$\delta := \frac{1}{N}\sum_{a=1}^{N} \|\mathbf{c}_a - \bar{\mathbf{c}}\|_2, \qquad \bar{\mathbf{c}} := \frac{1}{N}\sum_{a=1}^{N} \mathbf{c}_a. \tag{5}$$

The quantity δ measures dispersion in the agents' capability representations. Its relationship to peer-selection rewards depends on how capabilities affect performance on the task distribution. In the stationary construction used below, we substitute scalar capabilities with peer reward means. For instance, with one optimal peer and $N-1$ equally inferior peers separated by a reward gap Δ, this gives $\delta = 2(N-1)\Delta/N^2$.

Theorem 1 (**Expected regret of MACE**). *Under exact linear realizability, boundedness, and noise assumptions (Appendix E.1), let $\lambda \geq 1$ and choose α_T as a simultaneous confidence radius with failure probability $1/T$. For $T \geq 2$, MACE satisfies*

$$\mathbb{E}\big[\mathrm{Regret}_i^{\mathrm{MACE}}\big] \leq 2\alpha_T \sqrt{2TNd\log\left(1+\frac{T}{d\lambda}\right)} + \frac{3}{2}, \tag{6}$$

where $\alpha_T = O(\sqrt{\log T})$, yielding an $O(\sqrt{T}\log T)$ expected-regret bound.

Theorem 2 (**Regret of non-exploring policy**). *For every $N \geq 2$, there exists a stationary N-peer instance with capability diversity $\delta > 0$ on which a greedy agent, after querying each peer once, selects $a_{i,t} \in \arg\max_a \widehat{\mu}_{i,a,t}$ and incurs*

$$\mathbb{E}\Big[\mathrm{Regret}_i^{\text{non-exploring}}\Big] \geq \frac{1}{8}\delta(T-1), \tag{7}$$

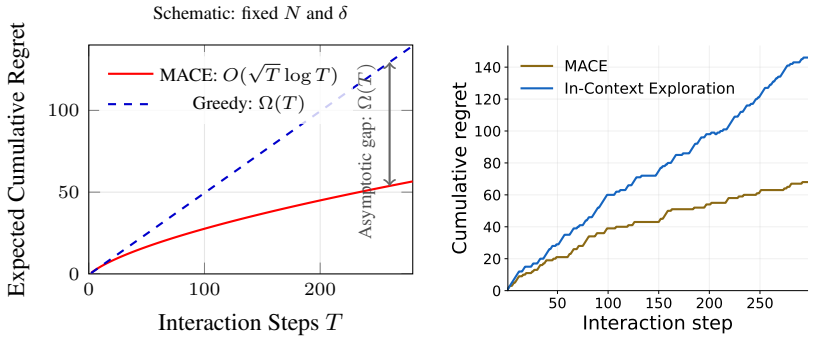


Figure 6: **Illustration of Theorem.** The asymptotic expected-regret bounds (left) align with the empirical behavior of Qwen2.5-7B-Instruct on the GPQA benchmark (right).

for all $T \geq N$, where $\widehat{\mu}_{i,a,t}$ is the empirical reward mean, and scalar capabilities are identified with peer reward means.

Corollary 1 (Benefit of explicit exploration). *On the stationary N-peer instance constructed in Theorem 2, represented as a linear bandit with constant scalar features, MACE with the confidence radius from Theorem 1 satisfies, for $T \geq N$,*

$$\mathbb{E}\left[\mathrm{Regret}_i^{\text{non-exploring}} - \mathrm{Regret}_i^{\text{MACE}}\right] \geq \frac{\delta}{8}(T-1) - 2\alpha_T\sqrt{2TN\log\left(1+\frac{T}{\lambda}\right)} - \frac{3}{2}.$$

For each fixed N, the expected regret gap is $\Omega(T)$ as $T \to \infty$.

Implication: Explicit exploration can prevent persistent peer-selection errors. For every fixed population size $N \geq 2$, our construction exhibits a stationary environment in which greedy selection incurs linear expected regret through premature commitment. Under the stated linear-bandit assumptions, MACE achieves sublinear expected regret, yielding an asymptotically linear separation. This establishes a concrete benefit of explicit exploration.

Empirical illustration. Figure 6 (left) schematically illustrates the separation established by Corollary 1. The right panel shows empirical regret trajectories for `Qwen2.5-7B-Instruct`. The observed advantage of MACE is qualitatively consistent with the benefit of explicit exploration.

7 Related Works

Recent works have built foundations for multi-agent LLM coordination through frameworks like CAMEL (Li et al., 2023), AutoGen (Wu et al., 2024), ChatDev (Qian et al., 2024), and MetaGPT (Hong et al., 2023), and through debate protocols (Du et al., 2024; Chan et al., 2024; Liu et al., 2024a; Liang et al., 2024; Becker et al., 2025), though most rely on fixed roles or orchestration rules, with a few attempting to grant autonomy (Hou et al., 2025; Dang et al., 2025; Wang et al., 2025; Chen et al., 2023). Separately, studies on single-agent exploration show that LLMs exhibit insufficient exploration in bandits (Krishnamurthy et al., 2024; Nie et al., 2025), underperform humans in open-ended settings (Pan et al., 2025; Zhang et al., 2025), and benefit from explicit algorithmic guidance (Arumugam & Griffiths, 2025), while related work on delegation reveals that verbalized confidence is a poor proxy for partner selection (Tomašev et al., 2026; Buyl et al., 2025; Debnath et al., 2025). Our work *extends these findings to the multi-agent setting, showing that exploration failures compound under agent interaction and proposing MACE to explicitly induce the exploration that current agents lack.* A complete and extended discussion of related works is in Appendix F.

8 Conclusion

In this work, we showed that LLM agents often fail to explore effectively in multi-agent environments, leading to premature commitment, poor peer selection, and higher regret. We formalized this as the Multi-Agent Exploration problem and introduced Multi-Agent Contextual Exploration (MACE), a lightweight solution that explicitly induces exploration through structured peer selection. Across diverse settings, MACE improves both exploration and downstream performance, with gains that transfer to unseen tasks. Our theory establishes that explicit exploration can prevent persistent peer-selection errors, yielding an asymptotically linear expected-regret advantage over greedy selection on a constructed stationary instance. Overall, results imply that reliable multi-agent autonomy may require explicit exploration rather than expecting it to emerge on its own.

AI USE STATEMENT

In this work, we used generative AI tools for polishing the text, to generate scientific plots, and to help prove theorems. We have not used generative AI tools for automating research and experiments. We have reviewed all AI-assisted work. We manually checked the writing faithfully conveys our intended content, and if the plots correctly express values. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies exploration and peer selection in multi-agent LLM systems. All experiments are conducted in simulated multi-agent environments using existing language models and publicly available question-answering benchmarks. The study does not involve human participants, personal data, or deployment in real-world decision-making settings. We use benchmark datasets and model APIs solely for research purposes and in accordance with their respective terms of use.

REPRODUCIBILITY STATEMENT

We provide the formal problem definition, optimization objective, feature construction, reward functions, and complete pseudocode for MACE in the main paper and Appendix B.3. Appendix C documents the benchmark datasets, evaluation splits, agent configurations, backbone models, decoding parameters, interaction rounds, evaluation metrics, computational resources, and complete prompt templates used in our experiments. We additionally report results across both trial-and-error and frozen-parameter exploitation phases, together with baseline comparisons, parameter sensitivity analyses, and round-wise performance. Upon publication, we will also release the implementation needed to reproduce the experiments, including preprocessing and evaluation scripts, configuration files, random seeds, and instructions for constructing the contextual- and parametric-diversity environments.

REFERENCES

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24, 2011.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Saaket Agashe, Yue Fan, Anthony Reyna, and Xin Eric Wang. Llm-coordination: evaluating and analyzing multi-agent coordination abilities in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 8038–8057, 2025.

Alfonso Amayuelas, Jingbo Yang, Saaket Agashe, Ashwin Nagarajan, Antonis Antoniades, Xin Eric Wang, and William Wang. Self-resource allocation in multi-agent llm systems. *arXiv preprint arXiv:2504.02051*, 2025.

Dilip Arumugam and Thomas L Griffiths. Toward efficient exploration by large language model agents. In *The Exploration in AI Today Workshop at ICML*, 2025.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. Emergent tool use from multi-agent autocurricula. In *International conference on learning representations*, 2019.

Jonas Becker, Lars Benedikt Kaesberg, Niklas Bauer, Jan Philip Wahle, Terry Ruas, and Bela Gipp. Mallm: Multi-agent large language models framework. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 418–439, 2025.

Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. In *International Conference on Learning Representations*, 2019.

Maarten Buyl, Yousra Fettach, Guillaume Bied, and Tijl De Bie. Building and measuring trust between large language models. *arXiv preprint arXiv:2508.15858*, 2025.

Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, et al. Why do multi-agent llm systems fail? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025.

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. In *The Twelfth International Conference on Learning Representations*, 2024.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, et al. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *The Twelfth International Conference on Learning Representations*, 2023.

Nuttapong Chentanez, Andrew Barto, and Satinder Singh. Intrinsically motivated reinforcement learning. *Advances in neural information processing systems*, 17, 2004.

Hyeong Kyu Choi and Sharon Li. Modex: Evaluator-free best-of-n selection for open-ended generation. In *ACL*, 2026.

Hyeong Kyu Choi, Jerry Zhu, and Sharon Li. Debate or vote: Which yields better decisions in multi-agent large language models? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Hyeong Kyu Choi, Xiaojin Zhu, and Sharon Li. When identity skews debate: Anonymization for bias-reduced multi-agent reasoning. In *ACL*, 2026.

Yufan Dang, Chen Qian, Xueheng Luo, Jingru Fan, Zihao Xie, Ruijie Shi, Weize Chen, Cheng Yang, Xiaoyin Che, Ye Tian, et al. Multi-agent collaboration via evolving orchestration. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Anushka Debnath, Stephen Cranefield, Emiliano Lorini, and Bastin Tony Roy Savarimuthu. Can llms reason about trust? a pilot study. In *International Workshop on Coordination, Organizations, Institutions, Norms, and Ethics for Governance of Multi-Agent Systems*, pp. 97–114. Springer, 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*, 2024.

Andrew Estornell and Yang Liu. Multi-llm debate: Framework, principals, and interventions. *Advances in Neural Information Processing Systems*, 37:28938–28964, 2024.

Mohamed Amine Ferrag, Norbert Tihanyi, and Merouane Debbah. From llm reasoning to autonomous ai agents: A comprehensive review. *arXiv preprint arXiv:2504.19678*, 2025.

Sébastien Forestier, Rémy Portelas, Yoan Mollard, and Pierre-Yves Oudeyer. Intrinsically motivated goal exploration processes with automatic curriculum learning. *Journal of Machine Learning Research*, 23(152):1–41, 2022.

Dawei Gao, Zitao Li, Xuchen Pan, Weirui Kuang, Zhijian Ma, Bingchen Qian, Fei Wei, Wenhao Zhang, Yuexiang Xie, Daoyuan Chen, et al. Agentscope: A flexible yet robust multi-agent platform. *arXiv preprint arXiv:2402.14034*, 2024.

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

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

Thorsten Händler. Balancing autonomy and alignment: A multi-dimensional taxonomy for autonomous llm-powered multi-agent architectures. *arXiv preprint arXiv:2310.03659*, 2023. doi: 10.48550/arXiv.2310.03659.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020.

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.

Zhipeng Hou, Junyi Tang, and Yipeng Wang. Halo: Hierarchical autonomous logic-oriented orchestration for multi-agent llm systems. *arXiv preprint arXiv:2505.13516*, 2025.

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

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Yukun Jiang, Yage Zhang, Xinyue Shen, Michael Backes, and Yang Zhang. " humans welcome to observe": A first look at the agent social network moltbook. *arXiv preprint arXiv:2602.10127*, 2026.

Lars Benedikt Kaesberg, Jonas Becker, Jan Philip Wahle, Terry Ruas, and Bela Gipp. Voting or consensus? decision-making in multi-agent debate. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 11640–11671, 2025.

Akshay Krishnamurthy, Keegan Harris, Dylan J Foster, Cyril Zhang, and Aleksandrs Slivkins. Can large language models explore in-context? *Advances in Neural Information Processing Systems*, 37:120124–120158, 2024.

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.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pp. 661–670, 2010.

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

Yunxuan Li, Yibing Du, Jiageng Zhang, Le Hou, Peter Grabowski, Yeqing Li, and Eugene Ie. Improving multi-agent debate with sparse communication topology. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7281–7294, 2024b.

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

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 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*, 2024a.

Wei Liu, Chenxi Wang, Yifei Wang, Zihao Xie, Rennai Qiu, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, and Chen Qian. Autonomous agents for collaborative task under information asymmetry. *Advances in Neural Information Processing Systems*, 37:2734–2765, 2024b.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. A dynamic llm-powered agent network for task-oriented agent collaboration. In *First Conference on Language Modeling*, 2024c.

Allen Nie, Yi Su, Bo Chang, Jonathan Lee, Ed H Chi, Quoc V Le, and Minmin Chen. Evolve: Evaluating and optimizing llms forin-context exploration. In *The Exploration in AI Today Workshop at ICML*, 2025.

OpenAI. Gpt-5. https://openai.com/, 2025.

Lan Pan, Hanbo Xie, and Robert Wilson. Large language models think too fast to explore effectively. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Jaden Park, Jungtaek Kim, Jongwon Jeong, Robert D. Nowak, Kangwook Lee, and Yong Jae Lee. Exploration and exploitation errors are measurable for language model agents. *arXiv preprint arXiv:2604.13151*, 2026. URL https://arxiv.org/abs/2604.13151.

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.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pp. 2778–2787. PMLR, 2017.

Jinghua Piao, Yuwei Yan, Jun Zhang, Nian Li, Junbo Yan, Xiaochong Lan, Zhihong Lu, Zhiheng Zheng, Jing Yi Wang, Di Zhou, et al. Agentsociety: Large-scale simulation of llm-driven generative agents advances understanding of human behaviors and society. *arXiv preprint arXiv:2502.08691*, 2025.

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. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 15174–15186, 2024.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First conference on language modeling*, 2024.

Manish Sanwal and Ishan Deva. An autonomous multi-agent llm framework for agile software development. *International Journal of Trend in Scientific Research and Development (IJTSRD)*, 8 (5):892–898, 2024. doi: 10.5281/zenodo.14179129.

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

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3 (1):9–44, 1988.

Nenad Tomašev, Matija Franklin, and Simon Osindero. Intelligent ai delegation. *arXiv preprint arXiv:2602.11865*, 2026.

Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D Nguyen. Multi-agent collaboration mechanisms: A survey of llms. *arXiv preprint arXiv:2501.06322*, 2025.

Qian Wang, Tianyu Wang, Zhenheng Tang, Qinbin Li, Nuo Chen, Jingsheng Liang, and Bingsheng He. Megaagent: A large-scale autonomous llm-based multi-agent system without predefined sops. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 4998–5036, 2025.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023.

Zhaotian Weng, Antonis Antoniades, Deepak Nathani, Zhen Zhang, Xiao Pu, and Xin Eric Wang. Group-evolving agents: Open-ended self-improvement via experience sharing. *arXiv preprint arXiv:2602.04837*, 2026.

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.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2369–2380, 2018.

Yunbei Zhang, Kai Mei, Ming Liu, Janet Wang, Dimitris N Metaxas, Xiao Wang, Jihun Hamm, and Yingqiang Ge. Agents in the wild: Safety, society, and the illusion of sociality on moltbook. *arXiv preprint arXiv:2602.13284*, 2026.

Ziyuan Zhang, Darcy Wang, Ningyuan Chen, Rodrigo Mansur, and Vahid Sarhangian. Comparing exploration-exploitation strategies of llms and humans: Insights from standard multi-armed bandit experiments. *arXiv preprint arXiv:2505.09901*, 2025.

Appendix

Table of Contents

A Appendix for the Delegation Experiment **16**
- A.1 Experimental Setup . . . 16
- A.2 Histograms for All p_A Values . . . 16
- A.3 Further Analysis: Explicit Policies and Computational Support . . . 18

B Appendix for Multi-Agent Contextual Exploration (MACE) **19**
- B.1 Feature Design . . . 19
- B.2 Reward Function . . . 20
- B.3 Algorithm . . . 21

C Appendix for the Main Experiments **22**
- C.1 Dataset Details . . . 22
- C.2 Empirical Regret Computation . . . 23
- C.3 Backbone Model Details . . . 23
- C.4 Resources . . . 23
- C.5 Prompt Templates . . . 23

D Further Experiments and Analyses **25**
- D.1 Round-wise Performance . . . 25
- D.2 Sensitivity Analysis of Exploration Coefficient α . . . 35
- D.3 Experiments on Stronger LLMs . . . 35
- D.4 Fully Homogeneous Agent Setup . . . 36
- D.5 MACE-TD: A Temporal-Difference Extension . . . 37
- D.6 Feature Analyses . . . 40

E Theoretical Assumptions and Proofs **42**
- E.1 Assumptions . . . 42
- E.2 Proof of Theorem 1 . . . 42
- E.3 Proof of Theorem 2 . . . 44
- E.4 Discussion . . . 45

F Appendix for Related Works **45**
- F.1 Autonomous Multi-Agent LLM Systems . . . 45
- F.2 LLM Exploration Capabilities . . . 46

G Broader Impact **47**

A Appendix for the Delegation Experiment

A.1 Experimental Setup

Specifications for the Delegated Tasks. Questions are randomly generated very simple arithmetic expressions involving addition, subtraction, and multiplication over integers in $[1, 100]$. Some example questions are: "What is 10 + 32?", "What is 40 - 21", or "What is 24×3?". When delegated a question, the chosen peer returns the correct answer with probability p_i, and an incorrect answer otherwise.

LLM Prompt Template. For all LLM-based delegation agents, we use the prompt shown below. At each round, the model is given the current arithmetic question together with the historical performance statistics of Peer A and Peer B, including the number of prior delegations, number of correct responses, and empirical success rate. The prompt explicitly reminds the model to balance exploitation (favoring peers with stronger observed accuracy) and exploration (testing peers with fewer prior trials). The model is then asked to output a final delegation decision in a constrained XML-style format, either `<choice>Peer A</choice>` or `<choice>Peer B</choice>`.

Runtime Setting. We use temperature 1.2 for Qwen2.5-7B-Instruct to allow various decision trajectories, and use the default setting for GPT-4 and GPT-5. The maximum number of tokens is set to 2048 for all settings.

Delegation Task Prompt Template

You are coordinating a team of two peers—Peer A and Peer B—who answer arithmetic questions. Your sole responsibility is to decide which peer to delegate each question to, based on their past performance. You cannot answer the question yourself; you must pick a peer.

Reason carefully: a peer with a higher historical success rate is probably more reliable, but also consider how many times you have tried each peer—a peer with very few attempts may be underexplored.
Always end your response with your choice in exactly this format: <choice>Peer A</choice> or <choice>Peer B</choice>

Question to delegate: `<question>`

Peer performance history:
Peer A: asked `<N>` times, correct `<n>` times (success rate: `<K>`%)
Peer B: asked `<N>` times, correct `<n>` times (success rate: `<K>`%)

Which peer do you want to delegate this question to?
State your final answer as: <choice>Peer A</choice> or <choice>Peer B</choice>

UCB1 Algorithm Details. As a principled exploration baseline, we compare against the UCB1 algorithm (Auer et al., 2002). Let $n_i(t)$ denote the number of times peer $i \in \{A, B\}$ has been selected up to round t, and let $\hat{\mu}_i(t)$ be its empirical success rate. At each round, UCB1 selects the peer maximizing

$$\hat{\mu}_i(t) + c\sqrt{\frac{\log t}{n_i(t)}},$$

where the first term encourages exploitation of high-performing peers and the second term encourages exploration of under-sampled peers. In our experiments, we use an exploration coefficient of $c = 1.0$. Each peer is selected once initially to avoid undefined confidence bonuses. This baseline provides a statistically grounded reference for adaptive exploration under uncertainty.

A.2 Histograms for All p_A Values

Here, we present the complete set of histogram comparisons for the delegation experiment described in Section 2, shown in Figure 7. Across all settings, the contrast with the UCB baseline is striking.

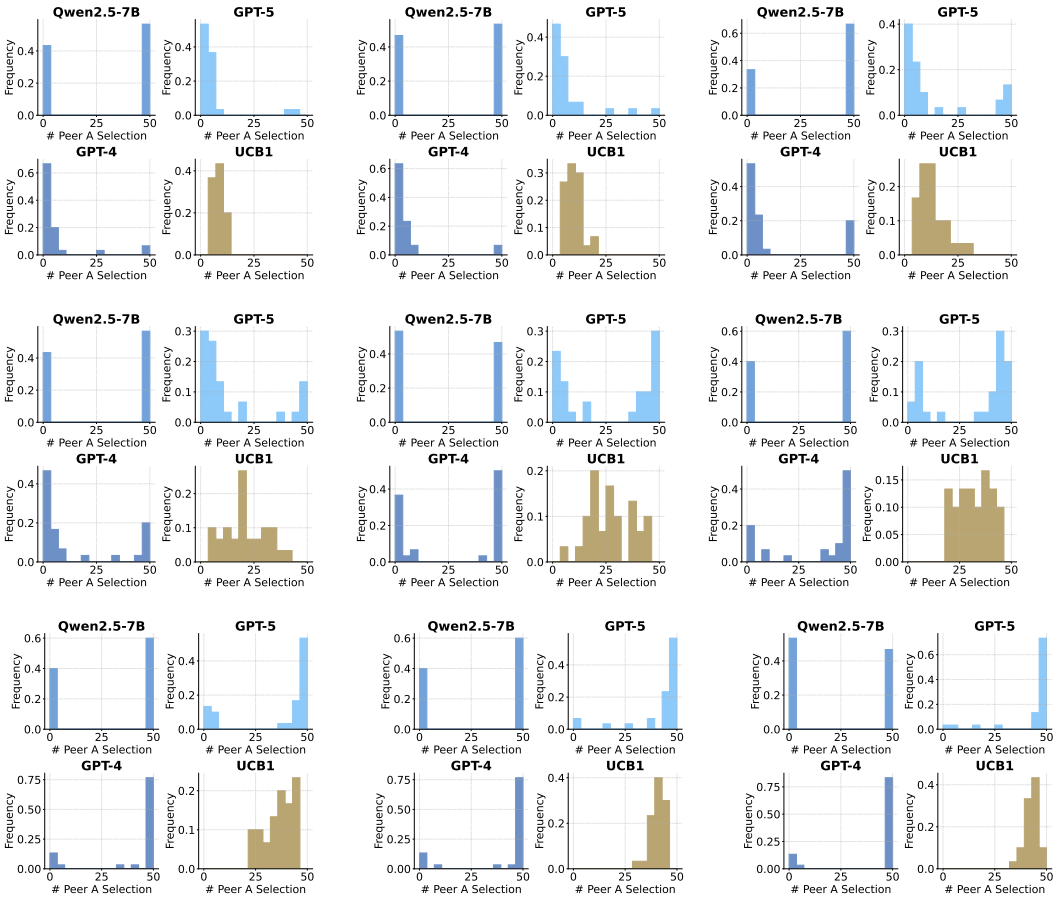


Figure 7: **Histogram comparison across various p_A values**. From left to right and from top to bottom, $p_A = [0.1, 0.2, \ldots, 0.9]$ while p_B is fixed to 0.5.

Whereas UCB exhibits smooth unimodal distributions centered near the optimal allocation implied by each p_A, all three LLMs instead display highly polarized behaviors, with selections sharply concentrated near the extremes of either 0 or 50. It is also noteworthy that the LLMs exhibit polarized behaviors even when $p_A = p_B = 0.5$. These results indicate that, rather than gradually balancing exploration and exploitation, the LLMs often commit early to one peer and repeatedly reinforce that initial choice.

Among the tested models, Qwen2.5-7B-Instruct demonstrates the most surprising failures of calibration. Even in settings where peer A is clearly inferior—for example when $p_A = 0.1$ and $p_B = 0.5$—the model still frequently allocates nearly all selections to peer A. Such behavior suggests severe overreaction to noisy early outcomes and an inability to recover through continued exploration. GPT-5 exhibits comparatively stronger exploratory tendencies than the other two LLMs, producing somewhat broader distributions in several settings. Nevertheless, its behavior remains far from principled exploration strategies such as UCB, and it still often collapses into near-deterministic decisions with insufficient trial-and-error search. Overall, these results reinforce that current LLM agents struggle to sustain statistically grounded exploration, even in simple delegation environments.

CDF of the Peer A Selection Ratios. For clearer comparison, we plot in Figure 8 the cumulative distribution function (CDF) of the Peer A selection rate, aggregated across different p_A settings $(0.1, 0.2, \ldots, 0.9)$ and 30 independent runs. The CDF provides a compact summary of how frequently each method allocates its selections to Peer A across the full sweep of environments.

Consistent with the histogram results in Figure 7, all LLM agents exhibit sharp jumps in the CDF near selection rates close to 0 and close to 1. These abrupt increases indicate that the models

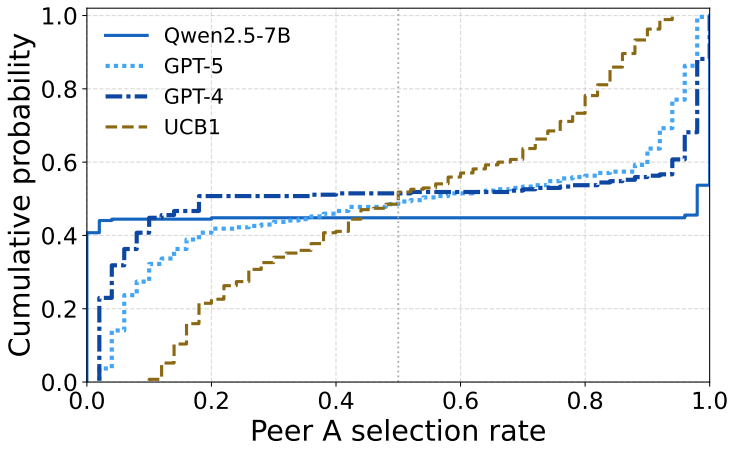


Figure 8: CDF plot of the Peer A selection ratio

overwhelmingly favor near-extreme behaviors: they tend to allocate almost all selections either to Peer B or to Peer A, with relatively little mass in the intermediate region. In other words, the LLMs rarely maintain balanced exploratory strategies and instead collapse to near-deterministic commitments. By contrast, UCB displays a much smoother and more gradual increase over the entire range, with little probability mass at the extremes. This pattern reflects the adaptive exploration behavior of principled bandit algorithms, which continuously trade off exploration and exploitation rather than prematurely locking onto a single option.

A.3 Further Analysis: Explicit Policies and Computational Support

To distinguish difficulty in spontaneously adopting an exploration strategy from difficulty in executing a prescribed policy, we further try augmenting the two-arm delegation experiment with three controls:

- **LLM + UCB Formula:** The complete UCB selection formula is provided in the prompt, and the LLM computes the scores and selects a peer.
- **LLM + UCB Calculator:** An external calculator computes the UCB scores, and the LLM makes the final peer selection.
- **Algorithmic UCB:** UCB1 directly selects the peer without LLM involvement. This control follows the same principle of algorithmically guided peer selection used by MACE.

Specifically, we evaluate Qwen-7B at multiple temperatures in a setting where peers A and B succeed with probabilities 0.6 and 0.4, respectively. We measure the *inferior-peer lock-in rate*, defined as the percentage of runs in which the selector chooses peer B in a majority of the final ten rounds, and cumulative regret incurred by selecting the inferior peer.

Table 1 shows that providing the UCB formula reduces the lock-in rate from 50.0% to 23.3% and cumulative regret from 20.0 to 13.5. Providing externally computed UCB scores further reduces the observed lock-in rate to 0.0%, with cumulative regret closely matching Algorithmic UCB (9.9 versus 9.8). The calculator result shows that the LLM can correctly interpret and act on externally computed UCB values, suggesting that the LLM can use externally computed exploration scores to make effective selections in this setting, but providing the formula alone is insufficient for reliable execution of the policy. The gap between the formula and calculator conditions highlights the value of computational support for implementing an explicit exploration strategy. More broadly, these findings motivate MACE's use of algorithmic peer selection to support systematic exploration.

Table 1: **Explicit exploration policies and computational support.** Comparison of peer selectors in the two-arm delegation experiment. Lower values indicate better performance for both metrics.

Selector	Inferior-Peer Lock-In Rate	Cumulative Regret
Prompt-only	50.0%	20.0
LLM + UCB Formula	23.3%	13.5
LLM + UCB Calculator	**0.0%**	9.9
Algorithmic UCB	**0.0%**	**9.8**

B Appendix for Multi-Agent Contextual Exploration (MACE)

B.1 Feature Design

We describe the relational features used by MACE to represent each candidate interaction between an agent i and a peer a at round t. To reduce the complexity of learning, we do not directly use high-dimensional response embeddings as contextual features. Instead, we construct a compact relational feature vector that captures the pairwise relationship between agent i and peer a, together with peer-level utility signals. For each possible interaction (i, a), we define

$$\mathbf{x}_{i,a,t} = \left[d_{i,a}^{(1)},\ d_{i,a}^{(2)},\ d_{i,a}^{(3)},\ c_a^{(1)},\ c_a^{(2)},\ c_a^{(3)},\ \hat{p}_a,\ \tau_t,\ 1 \right]^\top \in \mathbb{R}^9, \tag{8}$$

where $d_{i,a}^{(n)}$ denotes the response diversity between agent i and peer a measured using n-gram similarity, $c_a^{(n)}$ denotes the centrality of peer a in the response graph constructed from n-gram similarities, $\hat{p}_a$ denotes the peer's historical performance, τ_t denotes the normalized interaction round t, and the final constant 1 is a bias feature. These features are normalized before being used by the bandit or value-based policy. Further details are described below for each feature type.

Response diversity. The response diversity feature measures how much peer a's current response differs from agent i's current response. Let y_i and y_a denote the answers produced by agent i and peer a at round t, respectively. For each $n \in \{1, 2, 3\}$, we first compute the Jaccard similarity between their n-gram sets:

$$s_{i,a}^{(n)} = \frac{|\mathcal{N}_n(y_i) \cap \mathcal{N}_n(y_a)|}{|\mathcal{N}_n(y_i) \cup \mathcal{N}_n(y_a)|}, \tag{9}$$

where $\mathcal{N}_n(y)$ denotes the set of n-grams extracted from response y. The corresponding diversity feature is defined as

$$d_{i,a}^{(n)} = 1 - s_{i,a}^{(n)}. \tag{10}$$

Thus, higher values of $d_{i,a}^{(n)}$ indicate that peer a provides a response that is lexically more distinct from agent i's current answer. This feature captures the potential information gain from querying a peer whose answer differs from the agent's own belief.

Peer distinctiveness. In addition to pairwise diversity between agent i and peer a, we include a network-level distinctiveness feature that measures how much peer a's response is distinct compared to the overall agent population. For each $n \in \{1, 2, 3\}$, we define the divergence degree of peer a as

$$q_a^{(n)} = \sum_{u \in \mathcal{V},\, u \neq a} \left(1 - s_{a,u}^{(n)}\right), \tag{11}$$

where

$$s_{a,u}^{(n)} = \frac{|\mathcal{N}_n(y_a) \cap \mathcal{N}_n(y_u)|}{|\mathcal{N}_n(y_a) \cup \mathcal{N}_n(y_u)|}. \tag{12}$$

We then normalize this quantity by the total divergence mass across all agents:

$$c_a^{(n)} = \frac{q_a^{(n)}}{\sum_{p \in \mathcal{V}} q_p^{(n)}}, \tag{13}$$

with $c_a^{(n)} = 0$ when the denominator is zero. This feature can be interpreted as peer a's share of the total response-divergence mass in the population.

Historical performance. The historical performance feature provides an exploitation signal based on the empirical utility of peer a in previous interactions. Let N_a denote the number of previous times peer a has been selected, and let $r_a^{(s)}$ denote the reward observed from the s-th selection of peer a. We use the plain running mean

$$\hat{p}_a = \begin{cases} \frac{1}{N_a} \sum_{s=1}^{N_a} r_a^{(s)}, & N_a > 0, \\ 0, & N_a = 0. \end{cases} \tag{14}$$

This feature signals the policy to exploit peers based on historical performance.

Interaction round. We include the normalized per-sample interaction round

$$\tau = \frac{r}{R}, \tag{15}$$

where $r \in [0, R]$ is the current round and R is the total number of interaction rounds for each sample. This feature allows the policy to condition its decision on the stage of the interaction process for a shared task sample, enabling different behavior in earlier versus later rounds. For the contextual diversity setting, $R = 5$, and $R = 3$ for the parametric diversity setting.

Bias feature. Finally, we append a constant bias feature,

$$x_{\text{bias}} = 1. \tag{16}$$

This allows the learned policy to include an intercept term independent of the relational features.

B.2 Reward Function

At each interaction round t, agent i first updates its response after observing the selected peer. We then assign a scalar reward based on the change in a task-specific quality score. Let $s_i^{(t)} \in [0, 1]$ denote the score of agent i's response after round t, evaluated against the ground-truth answer. Depending on the task, we use token-level F1 for question answering, and the binary exact match score for multiple-choice and mathematical reasoning tasks.

A natural reward is the improvement in task score:

$$r_{i,\text{imp}}^{(t)} = s_i^{(t)} - s_i^{(t-1)}. \tag{17}$$

This reward is positive when the agent improves after interaction, negative when its answer degrades, and zero when the score remains unchanged.

However, using only improvement has an important limitation: an agent that is already correct and remains correct receives zero reward, even though the interaction successfully preserved a high-quality answer. To address this, we use a blended reward that combines improvement with the agent's absolute post-interaction score:

$$r_i^{(t)} = \frac{1}{2} \left[\left(s_i^{(t)} - s_i^{(t-1)} \right) + s_i^{(t)} \right]. \tag{18}$$

The first term credits interactions that improve the agent's answer, while the second term credits interactions that lead to a high-quality final response.

B.3 ALGORITHM

In Algorithm 1, we provide the pseudo-code for Multi-Agent Contextual Exploration (MACE). Note that the total interaction step for the contextual bandit problem is $T = |\mathcal{Q}| \times R$, λ is set to 1.0 across all settings.

Algorithm 1 Multi-Agent Contextual Exploration (MACE)

Require: Questions $\mathcal{Q}$, N agents, R rounds, α (exploration)

Initialize for all i, j: $\mathbf{A}_{ij} \leftarrow \lambda \mathbf{I}_9,\ \mathbf{b}_{ij} \leftarrow \mathbf{0},\ n_{ij} \leftarrow 0,\ \bar{r}_{ij} \leftarrow 0,\ t \leftarrow 0$
for each task $q \in \mathcal{Q}$ **do**
  Round 0: each agent i generates $y_i^{(0)}$ independently given q
  for $r = 1, \ldots, R$ **do**
    $t \leftarrow t + 1$
    // Peer selection
    for each agent i **do**
      for each candidate peer j **do**
        Compute context $\mathbf{x}_{ij}^{(t)} \in \mathbb{R}^9$
        $\boldsymbol{\theta}_{ij} \leftarrow \mathbf{A}_{ij}^{-1}\mathbf{b}_{ij}$
      end for
      $a_i^* \leftarrow \arg\max_j \left[\boldsymbol{\theta}_{ij}^\top \mathbf{x}_{ij}^{(t)} + \alpha\sqrt{\mathbf{x}_{ij}^{(t)\top}\mathbf{A}_{ij}^{-1}\mathbf{x}_{ij}^{(t)}}\right]$
    end for
    // Response generation
    for each agent i **do**
      Agent i reads $y_{a_i^*}^{(t-1)}$ and produces updated response $y_i^{(t)}$
    end for
    // Reward computation and LinUCB update
    for each agent i **do**
      $r_i \leftarrow \frac{1}{2}\left[\left(s_i^{(t)} - s_i^{(t-1)}\right) + s_i^{(t)}\right]$
      $\mathbf{A}_{i,a_i^*} \leftarrow \mathbf{A}_{i,a_i^*} + \mathbf{x}_{i,a_i^*}^{(t)}\mathbf{x}_{i,a_i^*}^{(t)\top}$
      $\mathbf{b}_{i,a_i^*} \leftarrow \mathbf{b}_{i,a_i^*} + r_i\,\mathbf{x}_{i,a_i^*}^{(t)}$
      $n_{i,a_i^*} \mathrel{+}= 1, \quad \bar{r}_{i,a_i^*} \leftarrow \bar{r}_{i,a_i^*} + (r_i - \bar{r}_{i,a_i^*}) \,/\, n_{i,a_i^*}$
    end for
  end for
end for

C Appendix for the Main Experiments

In this section, we provide more detailed explanations on the benchmark datasets (Appendix C.1), backbone LLMs (Appendix C.3), and prompt templates (Appendix C.5).

C.1 Dataset Details

We evaluated MACE under two representative forms of agent heterogeneity: *contextual diversity* and *parametric diversity*. These settings are designed to capture complementary sources of uncertainty in realistic multi-agent systems. In contextual diversity, agents share the same backbone model but observe different pieces of task-relevant context. In parametric diversity, agents differ in their underlying model family and scale, inducing heterogeneity in reasoning capabilities.

HotpotQA for the contextual diversity setting. For contextual diversity, we use the distractor setting of HotpotQA (Yang et al., 2018), a multi-hop question answering benchmark in which each example consists of multiple context passages required to answer an open-ended question. Each instance contains two gold evidence passages required to answer the question, together with eight distractor passages. We construct a multi-agent environment with $N = 10$ agents, where each agent receives a distinct passage as its local context. Thus, only a subset of agents has access to the information necessary for solving the task, while the remaining agents observe irrelevant or potentially misleading context. To avoid a case where a single agent consistently receives the gold evidence passage, we randomly assign the passage to each agent for every sample question. This setting requires each agent to explore its peers in order to identify which agents possess useful evidence. We evaluate on the first 600 examples labeled as hard, as the full dataset takes an excessive amount of time to evaluate on, where the first 300 samples are used for the trial-and-error phase evaluation and the latter 300 samples are used for the exploitation phase where the MACE parameters are frozen to test generalizability. The agents interact for $R = 5$ rounds per example, totaling $300 \times 5 = 1500$ steps of interaction for each evaluation phase. Answer quality is measured using Exact Match (EM) and token-level F1 against the ground-truth answer.

Math500 for the parametric diversity setting. For mathematical reasoning, we use Math500 (Lightman et al., 2023), focusing on Level 4–5 problems to emphasize challenging examples that require non-trivial reasoning. This results in a total of 262 questions, where the first 131 samples are used for the trial-and-error phase, and the remaining 131 is evaluated for the exploitation phase. Unlike the contextual-diversity setting, all agents receive the same problem statement, but differ in their underlying backbone models. This setting tests whether agents can learn which peers are more reliable for difficult mathematical reasoning problems. The agents interact for $R = 3$ rounds per sample, totaling $131 \times 3 = 393$ steps of interaction for each evaluation phase. We evaluate performance using a binary exact match between the extracted final answer and the ground-truth solution.

GPQA for the parametric diversity setting. We also evaluate on the Graduate-Level Google-Proof Question Answering (GPQA) benchmark (Rein et al., 2024), a challenging multiple-choice question answering dataset designed to test expert-level scientific reasoning. We use the Diamond split, which contains 198 particularly difficult questions. Similar to other benchmarks, we use the first 99 samples for trial-and-error, and the latter 99 samples for the exploitation phase evaluation. Also, as in Math500, all agents receive the same question, but differ in their model backbones, and the agents interact for $R = 3$ rounds per sample, totaling $99 \times 3 = 297$ steps of interaction for each evaluation phase. The evaluation metric is a binary exact match after extracting the selected answer option.

2WikiMultihopQA for the parameter transfer experiment. To evaluate cross-task transfer of learned exploration behavior, we use 2WikiMultihopQA (Ho et al., 2020), another multi-hop question answering benchmark that requires combining evidence from multiple documents, also providing the distractor mode. Generally, this benchmark is considered more challenging compared to HotpotQA. As in the HotpotQA setup, we construct a contextual-diversity environment with $N = 10$ agents, where each agent observes a different local context and only a subset of agents has access to the information required to answer the question correctly. The learned MACE parameters are transferred

without further training, and agents interact for $R = 5$ rounds across a subset of 300 examples. We evaluate answer quality using Exact Match (EM) and token-level F1 against the ground-truth answer.

C.2 Empirical Regret Computation

For each method m, we measure its cumulative reward shortfall relative to the Random baseline:

$$D_m(T) = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \left(r_{i,t}^{\text{Random}} - r_{i,t}^{(m)} \right), \tag{19}$$

where $r_{i,t}^{(m)}$ is the blended reward defined in Appendix B.2, and t indexes interactions across task samples and rounds within the evaluation phase. Each method follows its own peer-selection and response-update trajectory on the same task sequence. Negative values indicate higher cumulative reward than Random; positive values indicate lower cumulative reward. This metric compares realized rewards without estimating counterfactual rewards for unselected peers and is therefore distinct from the oracle-based pseudo-regret in Section 3.2. The plotted curves average the cumulative reward differences across agents.

C.3 Backbone Model Details

We use different backbone configurations depending on the source of heterogeneity being evaluated.

Contextual diversity. In the contextual-diversity experiments, all agents use the same backbone model, `Qwen2.5-7B-Instruct` (Yang et al., 2024). Since we instantiate ten independent Qwen agents in this setup, we use stochastic decoding with a temperature of 1.2 and nucleus sampling with `top_p`=0.95 to encourage response diversity across agents.

Parametric diversity. In the parametric-diversity experiments, we construct a heterogeneous agent pool consisting of four model types: { `GPT-5` (OpenAI, 2025), `Qwen2.5-7B-Instruct` (Yang et al., 2024), `Llama3.1-8B-Instruct` (Grattafiori et al., 2024), `Mistral-7B-v0.3` (Jiang et al., 2023) }, each with (temperature, `top_p`) values of (`default`, `default`), (1.2, 0.95), (1.0, 0.95), (1.0, 0.9), respectively. Each model corresponds to one agent in the multi-agent system. This setup reflects realistic open-agent ecosystems in which agents may differ substantially in model family, scale, training data, and reasoning ability.

C.4 Resources

All contextual diversity experiments were done on a single NVIDIA RTX A6000 GPU, and the parametric diversity experiments were done on three NVIDIA RTX A6000 GPUs (GPT-5 used Azure API calls).

C.5 Prompt Templates

Prompt templates for the Contextual Diversity setup

Prompt Template Under Contextual Diversity (initial response generation)

Context:
`<context passage or sentence>`

Question: `<question>`

First, briefly reason over the provided context. Then, place your final answer between <answer> and </answer> tags. Keep the answer concise – a word, phrase, or short sentence. Example: <answer>Paris</answer>

Prompt Template Under Contextual Diversity (interaction round response generation)

Context:
`<context passage or sentence>`

Question: `<question>`

YOUR PREVIOUS RESPONSE:
`<agent's previous round response>`

RESPONSE FROM `<chosen peer's name>`:
`<chosen peer's previous round response>`

INSTRUCTIONS:
Review your previous response and the peer's response carefully. You may agree, disagree, or selectively incorporate the peer's reasoning. Then provide your updated, complete response to the QUESTION.

First, briefly reason over the provided context. Then, place your final answer between <answer> and </answer> tags. Keep the answer concise – a word, phrase, or short sentence. Example: <answer>Paris</answer>

Prompt templates for the Parametric Diversity setup

Note that the specific instructions on how to format answers differ between the datasets: Math500 and GPQA.

Prompt Template Under Parametric Diversity (initial response generation)

Question: `<question>`

If Math500:
First, briefly state your step-by-step reasoning. Then, make sure to place your final answer between <answer> and </answer> tags. Example: <answer>\boxed{123}</answer>
If GPQA:
First, briefly state your step-by-step reasoning. Then, make sure to place only your final answer label between <answer> and </answer> tags. Example: <answer>(A)</answer>

Prompt Template Under Parametric Diversity (interaction round response generation)

Question: `<question>`

YOUR PREVIOUS RESPONSE:
`<agent's previous round response>`

RESPONSE FROM `<chosen peer's name>`:
`<chosen peer's previous round response>`

INSTRUCTIONS:
Review your previous response and the peer's response carefully. You may agree, disagree, or selectively incorporate the peer's reasoning. Then provide your updated, complete response to the QUESTION.

If Math500:
First, briefly state your step-by-step reasoning. Then, make sure to place your final answer between <answer> and </answer> tags. Example: <answer>\boxed{123}</answer>
If GPQA:
First, briefly state your step-by-step reasoning. Then, make sure to place only your final answer label between <answer> and </answer> tags. Example: <answer>(A)</answer>

Prompt template for peer selection of In-Context Exploration

Prompt Template for Peer Selection (In-Context Exploration)

QUESTION: `<question>`

HISTORY:
`<agent 1 name>` [chosen `<n>` times, mean reward: `<r>` | features: [Diversity (unigram): `<f1>`, Diversity (bigram): `<f2>`, Diversity (trigram): `<f3>`, Centrality (unigram): `<f4>`, Centrality (bigram): `<f5>`, Centrality (trigram): `<f6>`, Hist. Reward: `<f7>`, Norm. Round: `<f8>`, Bias: 1.0]]

`<agent 2 name>` [chosen `<n>` times, mean reward: `<r>` | features: [Diversity (unigram): `<f1>`, Diversity (bigram): `<f2>`, Diversity (trigram): `<f3>`, Centrality (unigram): `<f4>`, Centrality (bigram): `<f5>`, Centrality (trigram): `<f6>`, Hist. Reward: `<f7>`, Norm. Round: `<f8>`, Bias: 1.0]]

⋮

`<agent` N `name>` [chosen `<n>` times, mean reward: `<r>` | features: [Diversity (unigram): `<f1>`, Diversity (bigram): `<f2>`, Diversity (trigram): `<f3>`, Centrality (unigram): `<f4>`, Centrality (bigram): `<f5>`, Centrality (trigram): `<f6>`, Hist. Reward: `<f7>`, Norm. Round: `<f8>`, Bias: 1.0]]

Based on the historical information, whose full response would you like to read to help refine your answer? You may also choose your own.
Reply with only the agent number (1 to <N>).

For the features used in In-Context Exploration, the feature values used for MACE are identically shown to each agent as context for selection.

D Further Experiments and Analyses

D.1 Round-wise Performance

Round-wise task performance. In Figure 16, we show the average task performance across task samples during both the exploration and exploitation phases, evaluated for each per-sample interaction round $r \in [0, R]$. Compared to the global step regret evaluation in Figure 4, this analysis provides a

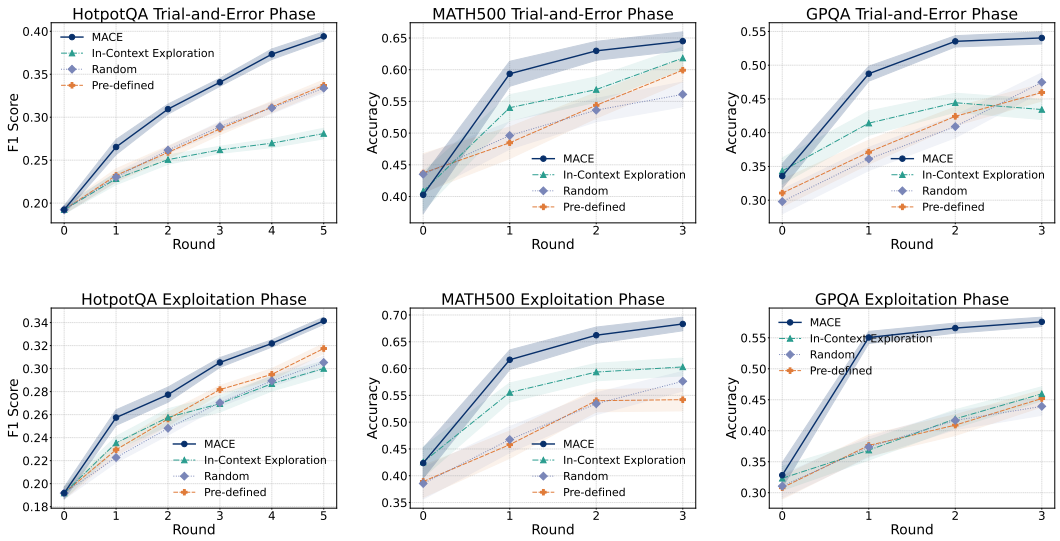


Figure 9: Comparison of exploration and exploitation performance for each interaction round. Shaded are the standard errors across participating agents; standard error * 0.2 are shown for Math500 and GPQA to avoid visual clutter. Also, note that the variance in the 0-th round is induced by GPT-5.

concrete intuition of how proper exploration can significantly impact task performance. Across all three benchmarks, MACE consistently improves as the interaction round proceeds, indicating that the learned peer-selection policy is able to convert additional communication rounds into better task performance. This trend is especially clear on Math500 and GPQA, where MACE quickly separates from Random, Pre-defined, and In-Context Exploration after the first interaction round and maintains the largest gains through the final round. Generally, MACE yields steady improvements in F1 score across rounds, suggesting that explicit exploration helps agents identify peers with useful evidence under contextual diversity. In contrast, In-Context Exploration often improves only marginally after the initial rounds and can remain close to or below Random, reinforcing that prompting agents to explore is insufficient for reliable peer discovery. Importantly, the same pattern persists in the exploitation phase, where MACE parameters are frozen: the policy learned during trial-and-error continues to produce stronger round-wise gains than all baselines. This suggests that MACE does not merely overfit to transient exploration feedback, but learns reusable interaction strategies that remain effective when deployed without further updates.

Round-wise task performance of the parameter transfer experiment. Figure 10 shows the round-wise performance of parameter transfer to the 2WikiMultihopQA benchmark. Across interaction rounds, MACE consistently outperforms the baselines, indicating that the exploration strategy learned on HotpotQA transfers effectively to a different multi-hop QA environment. These results support that MACE captures a transferable exploration policy rather than overfitting to the source benchmark.

Full tables for the round-wise performance. Here, we provide a full table of the round-wise performances, corresponding to the results shown in Figure 16. The tables also include the performance of each individual agent.

- Table 2 – 5: Contextual diversity experiment results on HotpotQA for MACE, In-Context Exploration, Pre-defined, and Random.
- Table 10 – 17: Parametric diversity experiment results on GPQA and Math500, for MACE, In-Context Exploration, Pre-defined, and Random.
- Table 6 – 9: Parameter Transfer experiment results on 2WikiMultihopQA for MACE, In-Context Exploration, Pre-defined, and Random.

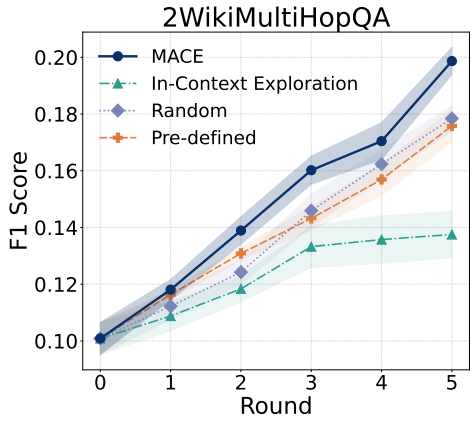


Figure 10: Per-sample interaction round-wise performance

Table 2: Round-wise HotpotQA (contextual diversity) performance of **MACE** across exploration and exploitation phases. We report Exact Match (EM) and token-level F1 for each agent over interaction rounds, where round 0 corresponds to the initial response before peer interaction. The highlighted Mean column reports the average performance across all 10 agents (all Qwen2.5-7B-Instruct models).

(a) Exploration Phase: Exact Match (EM)

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6	Agent 7	Agent 8	Agent 9	Agent 10	**Mean**
0 (init)	0.1633	0.1433	0.1533	0.1667	0.1200	0.1567	0.1600	0.1533	0.1300	0.1500	**0.1497**
1	0.2267	0.2333	0.2033	0.2100	0.1667	0.2333	0.2067	0.2233	0.1733	0.2100	**0.2087**
2	0.2300	0.2333	0.2667	0.2500	0.2467	0.2600	0.2667	0.2400	0.2133	0.2433	**0.2450**
3	0.2700	0.2833	0.2767	0.2667	0.2733	0.2867	0.2767	0.2433	0.2533	0.2500	**0.2680**
4	0.3233	0.3067	0.3167	0.2900	0.2800	0.2967	0.2933	0.2800	0.2700	0.2867	**0.2943**
5	0.3433	0.3000	0.3167	0.2900	0.3133	0.3267	0.3267	0.2900	0.3267	0.3000	**0.3133**

(b) Exploration Phase: Token-level F1

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6	Agent 7	Agent 8	Agent 9	Agent 10	**Mean**
0 (init)	0.2032	0.1929	0.2094	0.2035	0.1588	0.1917	0.2039	0.1923	0.1718	0.1955	**0.1923**
1	0.2823	0.3037	0.2641	0.2744	0.2027	0.2957	0.2583	0.2654	0.2327	0.2735	**0.2653**
2	0.2901	0.3074	0.3357	0.3253	0.2979	0.3137	0.3378	0.3032	0.2753	0.3078	**0.3094**
3	0.3302	0.3540	0.3626	0.3542	0.3330	0.3466	0.3561	0.3182	0.3264	0.3250	**0.3406**
4	0.3995	0.3940	0.4054	0.3762	0.3470	0.3619	0.3737	0.3622	0.3491	0.3642	**0.3733**
5	0.4223	0.3888	0.3958	0.3792	0.3827	0.4143	0.4117	0.3678	0.3987	0.3797	**0.3941**

(c) Exploitation Phase: Exact Match (EM)

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6	Agent 7	Agent 8	Agent 9	Agent 10	**Mean**
0 (init)	0.1533	0.1267	0.1600	0.1833	0.1600	0.1300	0.1500	0.1767	0.1467	0.1567	**0.1543**
1	0.2100	0.2067	0.2300	0.2167	0.1767	0.1867	0.1733	0.2367	0.1867	0.2100	**0.2033**
2	0.2100	0.2133	0.2300	0.2400	0.1967	0.2400	0.1967	0.2233	0.2200	0.2100	**0.2180**
3	0.2333	0.2300	0.2733	0.2500	0.2267	0.2200	0.2333	0.2400	0.2467	0.2333	**0.2387**
4	0.2467	0.2600	0.2500	0.2567	0.2367	0.2533	0.2433	0.2467	0.2467	0.2667	**0.2507**
5	0.2767	0.2600	0.2633	0.2833	0.2767	0.2733	0.2500	0.2800	0.2600	0.2633	**0.2687**

(d) Exploitation Phase: Token-level F1

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6	Agent 7	Agent 8	Agent 9	Agent 10	**Mean**
0 (init)	0.1901	0.1656	0.2029	0.2213	0.1783	0.1699	0.1876	0.2093	0.1959	0.1971	**0.1918**
1	0.2721	0.2633	0.2904	0.2758	0.2263	0.2336	0.2248	0.2742	0.2508	0.2632	**0.2575**
2	0.2723	0.2726	0.2890	0.3101	0.2413	0.3001	0.2442	0.2829	0.2881	0.2729	**0.2774**
3	0.2966	0.2969	0.3356	0.3203	0.2920	0.2933	0.2973	0.2983	0.3070	0.3157	**0.3053**
4	0.3174	0.3299	0.3235	0.3350	0.3106	0.3240	0.3154	0.3104	0.3121	0.3401	**0.3218**
5	0.3487	0.3329	0.3401	0.3571	0.3351	0.3494	0.3274	0.3547	0.3280	0.3418	**0.3415**

Table 3: Round-wise HotpotQA (contextual diversity) performance of **In-Context Exploration** across exploration and exploitation phases. We report Exact Match (EM) and token-level F1 for each agent over interaction rounds, where round 0 corresponds to the initial response before peer interaction. The highlighted Mean column reports the average performance across all 10 agents.

(a) Exploration Phase: Exact Match (EM)

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6	Agent 7	Agent 8	Agent 9	Agent 10	**Mean**
0 (init)	0.1633	0.1433	0.1533	0.1667	0.1200	0.1567	0.1600	0.1533	0.1300	0.1500	**0.1497**
1	0.2167	0.1800	0.2000	0.1767	0.1500	0.1967	0.1800	0.1500	0.1967	0.1767	**0.1823**
2	0.2133	0.2167	0.2133	0.2133	0.1933	0.1900	0.2067	0.1667	0.1967	0.1933	**0.2003**
3	0.2200	0.2267	0.2200	0.2133	0.1900	0.2133	0.2033	0.1933	0.2233	0.1833	**0.2087**
4	0.2333	0.2433	0.2133	0.2033	0.2267	0.2133	0.1867	0.2067	0.2067	0.2200	**0.2153**
5	0.2433	0.2533	0.2367	0.2133	0.2167	0.2167	0.2233	0.2233	0.1867	0.2433	**0.2257**

(b) Exploration Phase: Token-level F1

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6	Agent 7	Agent 8	Agent 9	Agent 10	**Mean**
0 (init)	0.2032	0.1929	0.2094	0.2035	0.1588	0.1917	0.2039	0.1923	0.1718	0.1955	**0.1923**
1	0.2659	0.2320	0.2439	0.2201	0.1910	0.2448	0.2267	0.1955	0.2441	0.2194	**0.2284**
2	0.2655	0.2739	0.2560	0.2621	0.2453	0.2364	0.2573	0.2096	0.2567	0.2432	**0.2506**
3	0.2731	0.2767	0.2668	0.2715	0.2457	0.2638	0.2548	0.2449	0.2793	0.2433	**0.2620**
4	0.2848	0.2951	0.2768	0.2592	0.2829	0.2721	0.2370	0.2528	0.2628	0.2742	**0.2698**
5	0.3065	0.3081	0.2948	0.2745	0.2663	0.2716	0.2735	0.2815	0.2399	0.2931	**0.2810**

(c) Exploitation Phase: Exact Match (EM)

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6	Agent 7	Agent 8	Agent 9	Agent 10	**Mean**
0 (init)	0.1533	0.1267	0.1600	0.1833	0.1600	0.1300	0.1500	0.1767	0.1467	0.1567	**0.1543**
1	0.2067	0.1800	0.2200	0.1700	0.1733	0.1633	0.1733	0.2067	0.1867	0.2033	**0.1883**
2	0.2267	0.2133	0.2500	0.2133	0.2067	0.1833	0.1733	0.2100	0.1967	0.2267	**0.2100**
3	0.2333	0.2133	0.2633	0.2333	0.2133	0.1867	0.1967	0.2333	0.2033	0.2167	**0.2193**
4	0.2467	0.2467	0.2767	0.2500	0.2067	0.2067	0.2200	0.2400	0.2100	0.2333	**0.2337**
5	0.2567	0.2600	0.2800	0.2467	0.2067	0.2267	0.2167	0.2500	0.2433	0.2467	**0.2433**

(d) Exploitation Phase: Token-level F1

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6	Agent 7	Agent 8	Agent 9	Agent 10	**Mean**
0 (init)	0.1901	0.1656	0.2029	0.2213	0.1783	0.1699	0.1876	0.2093	0.1959	0.1971	**0.1918**
1	0.2478	0.2242	0.2658	0.2220	0.2246	0.2093	0.2206	0.2416	0.2337	0.2644	**0.2354**
2	0.2699	0.2555	0.3021	0.2615	0.2499	0.2316	0.2146	0.2595	0.2473	0.2843	**0.2576**
3	0.2903	0.2558	0.3161	0.2849	0.2693	0.2317	0.2428	0.2755	0.2633	0.2651	**0.2695**
4	0.3002	0.2974	0.3292	0.3065	0.2639	0.2637	0.2677	0.2835	0.2743	0.2823	**0.2869**
5	0.3024	0.3256	0.3402	0.3062	0.2693	0.2871	0.2717	0.2930	0.3078	0.2974	**0.3001**

Table 4: Round-wise HotpotQA (contextual diversity) performance of the **Pre-defined** baseline across exploration and exploitation phases. We report Exact Match (EM) and token-level F1 for each agent over interaction rounds, where round 0 corresponds to the initial response before peer interaction. The highlighted Mean column reports the average performance across all 10 agents.

(a) Exploration Phase: Exact Match (EM)

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6	Agent 7	Agent 8	Agent 9	Agent 10	**Mean**
0 (init)	0.1633	0.1433	0.1533	0.1667	0.1200	0.1567	0.1600	0.1533	0.1300	0.1500	**0.1497**
1	0.2133	0.1867	0.2100	0.1833	0.1633	0.1933	0.1733	0.1533	0.1600	0.1900	**0.1827**
2	0.2267	0.2367	0.2133	0.1833	0.1933	0.2033	0.2067	0.1867	0.1833	0.2267	**0.2060**
3	0.2633	0.2500	0.2267	0.2100	0.2033	0.2333	0.2267	0.2133	0.2300	0.2400	**0.2297**
4	0.2733	0.2700	0.2400	0.2200	0.2267	0.2533	0.2367	0.2500	0.2533	0.2700	**0.2493**
5	0.2900	0.2667	0.2700	0.2533	0.2567	0.2700	0.2500	0.2767	0.2800	0.2867	**0.2700**

(b) Exploration Phase: Token-level F1

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6	Agent 7	Agent 8	Agent 9	Agent 10	**Mean**
0 (init)	0.2032	0.1929	0.2094	0.2035	0.1588	0.1917	0.2039	0.1923	0.1718	0.1955	**0.1923**
1	0.2696	0.2470	0.2623	0.2310	0.2044	0.2474	0.2154	0.1937	0.2108	0.2410	**0.2323**
2	0.2922	0.2969	0.2696	0.2385	0.2479	0.2535	0.2401	0.2382	0.2296	0.2832	**0.2590**
3	0.3267	0.3179	0.2881	0.2720	0.2600	0.2782	0.2709	0.2616	0.2875	0.3008	**0.2864**
4	0.3401	0.3459	0.3108	0.2899	0.2784	0.3046	0.2929	0.3128	0.3111	0.3326	**0.3119**
5	0.3686	0.3468	0.3506	0.3102	0.3113	0.3298	0.3119	0.3359	0.3442	0.3609	**0.3370**

(c) Exploitation Phase: Exact Match (EM)

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6	Agent 7	Agent 8	Agent 9	Agent 10	**Mean**
0 (init)	0.1533	0.1267	0.1600	0.1833	0.1600	0.1300	0.1500	0.1767	0.1467	0.1567	**0.1543**
1	0.1767	0.1667	0.2033	0.1933	0.1700	0.1667	0.2033	0.1900	0.1767	0.1767	**0.1823**
2	0.1967	0.2033	0.2400	0.1900	0.1767	0.2033	0.2267	0.1933	0.1867	0.1900	**0.2007**
3	0.2200	0.2267	0.2267	0.2067	0.2200	0.2400	0.2400	0.1933	0.2200	0.2100	**0.2203**
4	0.2300	0.2233	0.2267	0.2267	0.2300	0.2667	0.2300	0.2100	0.2233	0.2200	**0.2287**
5	0.2433	0.2433	0.2500	0.2467	0.2700	0.2633	0.2300	0.2500	0.2367	0.2367	**0.2470**

(d) Exploitation Phase: Token-level F1

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6	Agent 7	Agent 8	Agent 9	Agent 10	**Mean**
0 (init)	0.1901	0.1656	0.2029	0.2213	0.1783	0.1699	0.1876	0.2093	0.1959	0.1971	**0.1918**
1	0.2189	0.2200	0.2575	0.2334	0.2097	0.2110	0.2484	0.2332	0.2444	0.2185	**0.2295**
2	0.2556	0.2599	0.2985	0.2499	0.2245	0.2562	0.2701	0.2562	0.2526	0.2382	**0.2562**
3	0.2863	0.2874	0.2913	0.2741	0.2678	0.2954	0.2962	0.2631	0.2833	0.2725	**0.2817**
4	0.2941	0.2972	0.2985	0.2884	0.2814	0.3380	0.2914	0.2779	0.2887	0.2945	**0.2950**
5	0.3149	0.3110	0.3188	0.3094	0.3377	0.3391	0.2925	0.3267	0.3179	0.3065	**0.3174**

Table 5: Round-wise HotpotQA (contextual diversity) performance of the **Random** baseline across exploration and exploitation phases. We report Exact Match (EM) and token-level F1 for each agent over interaction rounds, where round 0 corresponds to the initial response before peer interaction. The highlighted Mean column reports the average performance across all 10 agents.

(a) Exploration Phase: Exact Match (EM)

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6	Agent 7	Agent 8	Agent 9	Agent 10	**Mean**
0 (init)	0.1633	0.1433	0.1533	0.1667	0.1200	0.1567	0.1600	0.1533	0.1300	0.1500	**0.1497**
1	0.2067	0.1700	0.1900	0.1733	0.1900	0.2067	0.1767	0.1600	0.1800	0.1700	**0.1823**
2	0.2300	0.1933	0.1900	0.2067	0.1967	0.2233	0.1967	0.2233	0.2233	0.2033	**0.2087**
3	0.2600	0.2167	0.2133	0.2533	0.2133	0.2400	0.2233	0.2367	0.2467	0.2367	**0.2340**
4	0.2900	0.2667	0.2300	0.2600	0.2500	0.2733	0.2267	0.2433	0.2600	0.2467	**0.2547**
5	0.3000	0.2933	0.2600	0.2567	0.2633	0.2600	0.2633	0.2967	0.2700	0.2700	**0.2733**

(b) Exploration Phase: Token-level F1

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6	Agent 7	Agent 8	Agent 9	Agent 10	**Mean**
0 (init)	0.2032	0.1929	0.2094	0.2035	0.1588	0.1917	0.2039	0.1923	0.1718	0.1955	**0.1923**
1	0.2531	0.2237	0.2469	0.2217	0.2431	0.2434	0.2189	0.1967	0.2296	0.2249	**0.2302**
2	0.2785	0.2532	0.2413	0.2657	0.2565	0.2707	0.2470	0.2690	0.2742	0.2606	**0.2617**
3	0.3168	0.2792	0.2666	0.3166	0.2605	0.2856	0.2771	0.2979	0.2978	0.2918	**0.2890**
4	0.3399	0.3381	0.2842	0.3195	0.3038	0.3212	0.2722	0.3138	0.3120	0.3033	**0.3108**
5	0.3564	0.3561	0.3080	0.3241	0.3290	0.3157	0.3281	0.3560	0.3311	0.3339	**0.3338**

(c) Exploitation Phase: Exact Match (EM)

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6	Agent 7	Agent 8	Agent 9	Agent 10	**Mean**
0 (init)	0.1533	0.1267	0.1600	0.1833	0.1600	0.1300	0.1500	0.1767	0.1467	0.1567	**0.1543**
1	0.1667	0.1633	0.2100	0.1700	0.1833	0.1600	0.1633	0.1900	0.1700	0.1800	**0.1757**
2	0.2000	0.1867	0.1833	0.1900	0.2000	0.1967	0.2000	0.1733	0.2200	0.2000	**0.1950**
3	0.2033	0.1933	0.2167	0.2233	0.2033	0.2267	0.2100	0.2233	0.2167	0.2167	**0.2133**
4	0.2233	0.2133	0.2533	0.2400	0.2267	0.2467	0.2033	0.2200	0.2233	0.2133	**0.2263**
5	0.2300	0.2267	0.2267	0.2400	0.2367	0.2533	0.2233	0.2467	0.2733	0.2267	**0.2383**

(d) Exploitation Phase: Token-level F1

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6	Agent 7	Agent 8	Agent 9	Agent 10	**Mean**
0 (init)	0.1901	0.1656	0.2029	0.2213	0.1783	0.1699	0.1876	0.2093	0.1959	0.1971	**0.1918**
1	0.2074	0.2072	0.2516	0.2191	0.2310	0.2130	0.2146	0.2415	0.2200	0.2218	**0.2227**
2	0.2554	0.2387	0.2317	0.2360	0.2442	0.2655	0.2582	0.2166	0.2873	0.2476	**0.2481**
3	0.2599	0.2619	0.2673	0.2900	0.2527	0.2809	0.2657	0.2709	0.2776	0.2765	**0.2703**
4	0.2860	0.2905	0.3071	0.3011	0.2779	0.3162	0.2592	0.2820	0.2897	0.2835	**0.2893**
5	0.3001	0.2952	0.2844	0.3126	0.2979	0.3233	0.2798	0.3069	0.3412	0.3117	**0.3053**

Table 6: Round-wise performance on 2WikiMultihopQA (parameter transfer experiment) of **MACE** measured by Exact Match (EM) and token-level F1. We report performance for each agent over interaction rounds, where round 0 corresponds to the initial response before peer interaction. The highlighted Mean column reports the average performance across all 10 agents.

(a) Exact Match (EM)

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6	Agent 7	Agent 8	Agent 9	Agent 10	**Mean**
0 (init)	0.0767	0.1000	0.0967	0.0867	0.1267	0.0800	0.0800	0.0667	0.0800	0.0867	**0.0880**
1	0.1033	0.1100	0.1267	0.1000	0.1267	0.0933	0.0867	0.1000	0.1000	0.0933	**0.1040**
2	0.1467	0.1100	0.1300	0.1133	0.1167	0.1300	0.1033	0.1067	0.1433	0.1233	**0.1223**
3	0.1500	0.1567	0.1467	0.1433	0.1367	0.1367	0.1100	0.1567	0.1533	0.1167	**0.1407**
4	0.1500	0.1567	0.1867	0.1533	0.1100	0.1600	0.1200	0.1600	0.1600	0.1700	**0.1527**
5	0.1700	0.1800	0.1900	0.1633	0.1767	0.2000	0.1667	0.1967	0.1833	0.1933	**0.1820**

(b) Token-level F1

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6	Agent 7	Agent 8	Agent 9	Agent 10	**Mean**
0 (init)	0.0854	0.1161	0.1030	0.0993	0.1453	0.0849	0.1026	0.0808	0.0963	0.0949	**0.1009**
1	0.1095	0.1275	0.1347	0.1155	0.1400	0.1114	0.1063	0.1093	0.1175	0.1091	**0.1181**
2	0.1617	0.1277	0.1453	0.1295	0.1400	0.1521	0.1208	0.1147	0.1612	0.1362	**0.1389**
3	0.1619	0.1763	0.1613	0.1629	0.1588	0.1538	0.1310	0.1811	0.1778	0.1370	**0.1602**
4	0.1654	0.1726	0.2035	0.1739	0.1353	0.1788	0.1328	0.1763	0.1873	0.1785	**0.1705**
5	0.1823	0.1978	0.2017	0.1729	0.1939	0.2164	0.1829	0.2232	0.2049	0.2107	**0.1987**

Table 7: Round-wise performance on 2WikiMultihopQA (parameter transfer experiment) of **In-Context Exploration** measured by Exact Match (EM) and token-level F1. We report performance for each agent over interaction rounds, where round 0 corresponds to the initial response before peer interaction. The highlighted Mean column reports the average performance across all 10 agents.

(a) Exact Match (EM)

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6	Agent 7	Agent 8	Agent 9	Agent 10	**Mean**
0 (init)	0.0767	0.1000	0.0967	0.0867	0.1267	0.0800	0.0800	0.0667	0.0800	0.0867	**0.0880**
1	0.0833	0.1100	0.0933	0.0767	0.1167	0.1000	0.0933	0.0967	0.0667	0.1133	**0.0950**
2	0.1133	0.1033	0.1000	0.0833	0.1100	0.1233	0.1067	0.1133	0.0767	0.0967	**0.1027**
3	0.1367	0.1167	0.0967	0.1133	0.1067	0.1467	0.1100	0.1133	0.0767	0.1500	**0.1167**
4	0.1300	0.1167	0.1067	0.0967	0.1167	0.1533	0.1333	0.1167	0.0733	0.1633	**0.1207**
5	0.1333	0.1100	0.1033	0.0967	0.1133	0.1633	0.1300	0.1433	0.0833	0.1500	**0.1227**

(b) Token-level F1

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6	Agent 7	Agent 8	Agent 9	Agent 10	**Mean**
0 (init)	0.0854	0.1161	0.1030	0.0993	0.1453	0.0849	0.1026	0.0808	0.0963	0.0949	**0.1009**
1	0.0890	0.1303	0.1036	0.0928	0.1335	0.1130	0.1089	0.1105	0.0830	0.1232	**0.1088**
2	0.1205	0.1243	0.1097	0.0994	0.1250	0.1387	0.1262	0.1361	0.0917	0.1121	**0.1184**
3	0.1489	0.1386	0.1053	0.1310	0.1221	0.1670	0.1251	0.1325	0.0924	0.1696	**0.1333**
4	0.1455	0.1307	0.1127	0.1136	0.1328	0.1721	0.1506	0.1350	0.0860	0.1784	**0.1357**
5	0.1467	0.1259	0.1094	0.1121	0.1256	0.1795	0.1516	0.1532	0.0994	0.1719	**0.1375**

Table 8: Round-wise performance on 2WikiMultihopQA (parameter transfer experiment) of the **Pre-defined** baseline measured by Exact Match (EM) and token-level F1. We report performance for each agent over interaction rounds, where round 0 corresponds to the initial response before peer interaction. The highlighted Mean column reports the average performance across all 10 agents.

(a) Exact Match (EM)

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6	Agent 7	Agent 8	Agent 9	Agent 10	**Mean**
0 (init)	0.0767	0.1000	0.0967	0.0867	0.1267	0.0800	0.0800	0.0667	0.0800	0.0867	**0.0880**
1	0.1000	0.1067	0.1067	0.1033	0.1133	0.1000	0.0900	0.0967	0.1000	0.1067	**0.1023**
2	0.1267	0.1100	0.1233	0.1167	0.1067	0.1167	0.1167	0.1167	0.1033	0.1300	**0.1167**
3	0.1300	0.1200	0.1233	0.1133	0.1133	0.1333	0.1300	0.1200	0.1400	0.1600	**0.1283**
4	0.1433	0.1433	0.1200	0.1167	0.1300	0.1567	0.1233	0.1300	0.1633	0.1667	**0.1393**
5	0.1533	0.1567	0.1500	0.1233	0.1767	0.1533	0.1433	0.1633	0.1833	0.1700	**0.1573**

(b) Token-level F1

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6	Agent 7	Agent 8	Agent 9	Agent 10	**Mean**
0 (init)	0.0854	0.1161	0.1030	0.0993	0.1453	0.0849	0.1026	0.0808	0.0963	0.0949	**0.1009**
1	0.1087	0.1199	0.1162	0.1226	0.1244	0.1137	0.1070	0.1150	0.1154	0.1203	**0.1163**
2	0.1392	0.1229	0.1386	0.1345	0.1230	0.1284	0.1313	0.1275	0.1219	0.1413	**0.1309**
3	0.1470	0.1351	0.1346	0.1295	0.1307	0.1413	0.1426	0.1385	0.1599	0.1730	**0.1432**
4	0.1596	0.1634	0.1323	0.1359	0.1431	0.1723	0.1415	0.1564	0.1807	0.1839	**0.1569**
5	0.1744	0.1754	0.1668	0.1347	0.1920	0.1682	0.1705	0.1858	0.2058	0.1845	**0.1758**

Table 9: Round-wise performance on 2WikiMultihopQA (parameter transfer experiment) of the **Random** baseline measured by Exact Match (EM) and token-level F1. We report performance for each agent over interaction rounds, where round 0 corresponds to the initial response before peer interaction. The highlighted Mean column reports the average performance across all 10 agents.

(a) Exact Match (EM)

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6	Agent 7	Agent 8	Agent 9	Agent 10	**Mean**
0 (init)	0.0767	0.1000	0.0967	0.0867	0.1267	0.0800	0.0800	0.0667	0.0800	0.0867	**0.0880**
1	0.1033	0.1067	0.1100	0.1133	0.1233	0.0767	0.0833	0.1033	0.0800	0.1000	**0.1000**
2	0.1400	0.1133	0.0967	0.1100	0.1233	0.1067	0.1033	0.0900	0.1000	0.1200	**0.1103**
3	0.1433	0.1433	0.0933	0.1233	0.1600	0.1300	0.1167	0.1300	0.1333	0.1400	**0.1313**
4	0.1433	0.1600	0.1233	0.1267	0.1767	0.1433	0.1433	0.1333	0.1567	0.1433	**0.1450**
5	0.1433	0.1600	0.1600	0.1633	0.1700	0.1567	0.1700	0.1500	0.1633	0.1867	**0.1623**

(b) Token-level F1

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6	Agent 7	Agent 8	Agent 9	Agent 10	**Mean**
0 (init)	0.0854	0.1161	0.1030	0.0993	0.1453	0.0849	0.1026	0.0808	0.0963	0.0949	**0.1009**
1	0.1094	0.1245	0.1193	0.1221	0.1390	0.0857	0.1032	0.1143	0.0961	0.1087	**0.1122**
2	0.1468	0.1295	0.1133	0.1226	0.1393	0.1165	0.1187	0.1011	0.1163	0.1384	**0.1243**
3	0.1607	0.1551	0.1053	0.1350	0.1761	0.1410	0.1373	0.1384	0.1530	0.1575	**0.1459**
4	0.1573	0.1775	0.1424	0.1416	0.1912	0.1547	0.1627	0.1494	0.1838	0.1628	**0.1623**
5	0.1566	0.1828	0.1749	0.1730	0.1871	0.1663	0.1946	0.1603	0.1866	0.2020	**0.1784**

Table 10: Round-wise Math500 (parametric diversity) performance of **MACE** across exploration and exploitation phases. We report accuracy for each agent and the mean accuracy across agents over interaction rounds. Round 0 corresponds to the initial response before peer interaction. Agent 1 through 4 correspond to GPT-5, Qwen-7B, Llama-8B, Mistral-7B, respectively.

(a) Exploration Phase

Round	Agent 1	Agent 2	Agent 3	Agent 4	**Mean**
0 (init)	0.8626	0.3969	0.3206	0.0305	**0.4027**
1	0.8779	0.4504	0.6718	0.3740	**0.5935**
2	0.8702	0.5038	0.6412	0.5038	**0.6298**
3	0.8931	0.5191	0.5573	0.6107	**0.6450**

(b) Exploitation Phase

Round	Agent 1	Agent 2	Agent 3	Agent 4	**Mean**
0 (init)	0.8397	0.4733	0.3206	0.0611	**0.4237**
1	0.8550	0.4962	0.7252	0.3893	**0.6164**
2	0.8779	0.4809	0.7252	0.5649	**0.6622**
3	0.8779	0.5267	0.6947	0.6336	**0.6832**

Table 11: Round-wise GPQA (parametric diversity) performance of **MACE** across exploration and exploitation phases. We report accuracy for each agent and the mean accuracy across agents over interaction rounds. Round 0 corresponds to the initial response before peer interaction. Agent 1 through 4 correspond to GPT-5, Qwen-7B, Llama-8B, Mistral-7B, respectively.

(a) Exploration Phase

Round	Agent 1	Agent 2	Agent 3	Agent 4	**Mean**
0 (init)	0.6061	0.3535	0.2828	0.1010	**0.3359**
1	0.6465	0.5354	0.3939	0.3737	**0.4874**
2	0.6465	0.5859	0.4545	0.4545	**0.5354**
3	0.6667	0.5758	0.4949	0.4242	**0.5404**

(b) Exploitation Phase

Round	Agent 1	Agent 2	Agent 3	Agent 4	**Mean**
0 (init)	0.6465	0.3535	0.1919	0.1212	**0.3283**
1	0.6364	0.5859	0.5960	0.3838	**0.5505**
2	0.6566	0.5960	0.5758	0.4343	**0.5657**
3	0.6566	0.6162	0.5758	0.4545	**0.5758**

Table 12: Round-wise Math500 (parametric diversity) performance of **In-Context Exploration** across exploration and exploitation phases. We report accuracy for each agent and the mean accuracy across agents over interaction rounds. Round 0 corresponds to the initial response before peer interaction. Agent 1 through 4 correspond to GPT-5, Qwen-7B, Llama-8B, Mistral-7B, respectively.

(a) Exploration Phase

Round	Agent 1	Agent 2	Agent 3	Agent 4	**Mean**
0 (init)	0.8855	0.3969	0.3206	0.0305	**0.4084**
1	0.8931	0.4046	0.4504	0.4122	**0.5401**
2	0.8931	0.3511	0.4809	0.5496	**0.5687**
3	0.8779	0.3893	0.6031	0.6031	**0.6183**

(b) Exploitation Phase

Round	Agent 1	Agent 2	Agent 3	Agent 4	**Mean**
0 (init)	0.8473	0.4733	0.3206	0.0611	**0.4256**
1	0.8626	0.4580	0.4580	0.4427	**0.5553**
2	0.8702	0.4427	0.5267	0.5344	**0.5935**
3	0.8702	0.4198	0.5344	0.5878	**0.6031**

Table 13: Round-wise GPQA (parametric diversity) performance of **In-Context Exploration** across exploration and exploitation phases. We report accuracy for each agent and the mean accuracy across agents over interaction rounds. Round 0 corresponds to the initial response before peer interaction. Agent 1 through 4 correspond to GPT-5, Qwen-7B, Llama-8B, Mistral-7B, respectively.

(a) Exploration Phase

Round	Agent 1	Agent 2	Agent 3	Agent 4	**Mean**
0 (init)	0.6364	0.3535	0.2828	0.1010	**0.3434**
1	0.6869	0.2424	0.2626	0.4646	**0.4141**
2	0.6566	0.3434	0.2929	0.4848	**0.4444**
3	0.6566	0.3333	0.2727	0.4747	**0.4343**

(b) Exploitation Phase

Round	Agent 1	Agent 2	Agent 3	Agent 4	**Mean**
0 (init)	0.6263	0.3535	0.1919	0.1212	**0.3232**
1	0.6465	0.2626	0.2626	0.3030	**0.3687**
2	0.6667	0.2828	0.3434	0.3838	**0.4192**
3	0.6566	0.3838	0.3939	0.4040	**0.4596**

Table 14: Round-wise Math500 (parametric diversity) performance of the **Pre-defined** baseline across exploration and exploitation phases. We report accuracy for each agent and the mean accuracy across agents over interaction rounds. Round 0 corresponds to the initial response before peer interaction. Agent 1 through 4 correspond to GPT-5, Qwen-7B, Llama-8B, Mistral-7B, respectively.

(a) Exploration Phase

Round	Agent 1	Agent 2	Agent 3	Agent 4	**Mean**
0 (init)	0.8779	0.3969	0.3206	0.0305	**0.4065**
1	0.8702	0.4809	0.1756	0.4122	**0.4847**
2	0.8702	0.4427	0.3282	0.5344	**0.5439**
3	0.8779	0.4351	0.4427	0.6412	**0.5992**

(b) Exploitation Phase

Round	Agent 1	Agent 2	Agent 3	Agent 4	**Mean**
0 (init)	0.8473	0.4733	0.3206	0.0611	**0.4256**
1	0.8702	0.3740	0.1756	0.4122	**0.4580**
2	0.8779	0.3664	0.3740	0.5420	**0.5401**
3	0.8626	0.3206	0.3969	0.5878	**0.5420**

Table 15: Round-wise GPQA (parametric diversity) performance of the **Pre-defined** baseline across exploration and exploitation phases. We report accuracy for each agent and the mean accuracy across agents over interaction rounds. Round 0 corresponds to the initial response before peer interaction. Agent 1 through 4 correspond to GPT-5, Qwen-7B, Llama-8B, Mistral-7B, respectively.

(a) Exploration Phase

Round	Agent 1	Agent 2	Agent 3	Agent 4	**Mean**
0 (init)	0.6263	0.3535	0.2828	0.1010	**0.3409**
1	0.6566	0.3030	0.1717	0.3535	**0.3712**
2	0.6566	0.3131	0.3232	0.4040	**0.4242**
3	0.6566	0.3131	0.4040	0.4646	**0.4596**

(b) Exploitation Phase

Round	Agent 1	Agent 2	Agent 3	Agent 4	**Mean**
0 (init)	0.6061	0.3535	0.1919	0.1212	**0.3182**
1	0.6465	0.3030	0.1818	0.3737	**0.3763**
2	0.6667	0.2323	0.2929	0.4444	**0.4091**
3	0.6566	0.2828	0.4040	0.4646	**0.4520**

Table 16: Round-wise Math500 (parametric diversity) performance of the **Random** baseline across exploration and exploitation phases. We report accuracy for each agent and the mean accuracy across agents over interaction rounds. Round 0 corresponds to the initial response before peer interaction. Agent 1 through 4 correspond to GPT-5, Qwen-7B, Llama-8B, Mistral-7B, respectively.

(a) Exploration Phase

Round	Agent 1	Agent 2	Agent 3	Agent 4	**Mean**
0 (init)	0.8702	0.3969	0.3206	0.0305	**0.4046**
1	0.8473	0.4733	0.3893	0.2748	**0.4962**
2	0.8473	0.4656	0.4656	0.3664	**0.5363**
3	0.8855	0.4809	0.5115	0.3664	**0.5611**

(b) Exploitation Phase

Round	Agent 1	Agent 2	Agent 3	Agent 4	**Mean**
0 (init)	0.8321	0.4733	0.3206	0.0611	**0.4218**
1	0.8702	0.4427	0.3130	0.2443	**0.4676**
2	0.8855	0.4275	0.4427	0.3817	**0.5344**
3	0.8855	0.4885	0.5191	0.4122	**0.5763**

Table 17: Round-wise GPQA (parametric diversity) performance of the **Random** baseline across exploration and exploitation phases. We report accuracy for each agent and the mean accuracy across agents over interaction rounds. Round 0 corresponds to the initial response before peer interaction. Agent 1 through 4 correspond to GPT-5, Qwen-7B, Llama-8B, Mistral-7B, respectively.

(a) Exploration Phase

Round	Agent 1	Agent 2	Agent 3	Agent 4	**Mean**
0 (init)	0.5758	0.3535	0.2828	0.1010	**0.3283**
1	0.6465	0.3434	0.2424	0.2121	**0.3611**
2	0.6768	0.3939	0.3333	0.2323	**0.4091**
3	0.7071	0.4444	0.4040	0.3434	**0.4747**

(b) Exploitation Phase

Round	Agent 1	Agent 2	Agent 3	Agent 4	**Mean**
0 (init)	0.6162	0.3535	0.1919	0.1212	**0.3207**
1	0.6263	0.3737	0.3131	0.1818	**0.3737**
2	0.6566	0.3939	0.3636	0.2525	**0.4167**
3	0.6566	0.4747	0.3535	0.2727	**0.4394**

D.2 Sensitivity Analysis of Exploration Coefficient α

We analyze the sensitivity of MACE to the exploration coefficient α on the HotpotQA benchmark (contextual diversity), as shown in Figure 11. The coefficient α controls the strength of the uncertainty bonus in the LinUCB selection rule (equation 3), thereby determining how aggressively agents explore under-tested peers.

Overall, we find that MACE is sensitive to this trade-off: both insufficient and excessive exploration lead to worse regret behavior. When α is too small, such as $\alpha = 0.1$, the uncertainty bonus becomes negligible, and the policy effectively reduces to a greedy peer-selection strategy. In this regime, agents are more likely to prematurely commit to peers based on noisy early observations, leading to poor exploration and regret even worse than the random baseline, indicated by the dashed horizontal line in the figure. This is consistent with our broader observation that non-exploring or weakly exploring policies can fail severely in heterogeneous multi-agent environments. On the other hand, overly large values of α also degrade exploratory performance. Although a higher α encourages broader peer exploration, excessive exploration slows down the decrease in cumulative regret because agents continue to prioritize uncertain peers even after sufficient evidence has been collected. In other words, the policy overweights exploration relative to exploitation, delaying convergence toward effective collaborators. Empirically, intermediate values around $\alpha = 0.5$ to $\alpha = 1.0$ achieve the strongest regret reduction, suggesting that MACE benefits from a moderate exploration bonus that is large enough to prevent premature commitment but not so large that it prevents efficient exploitation.

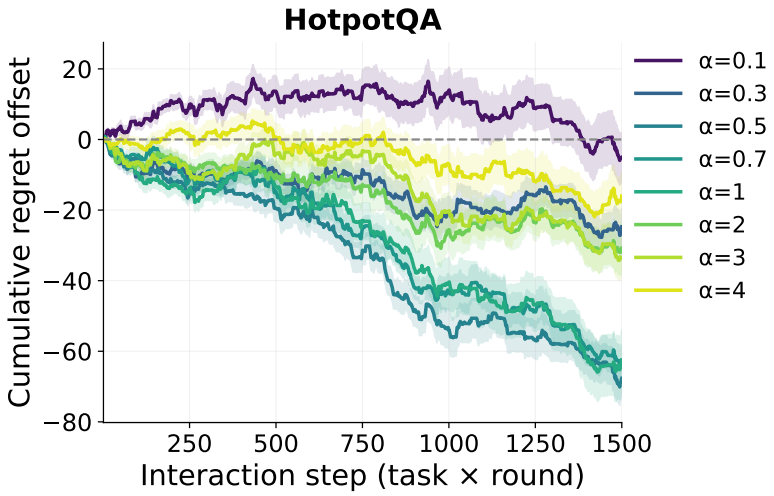


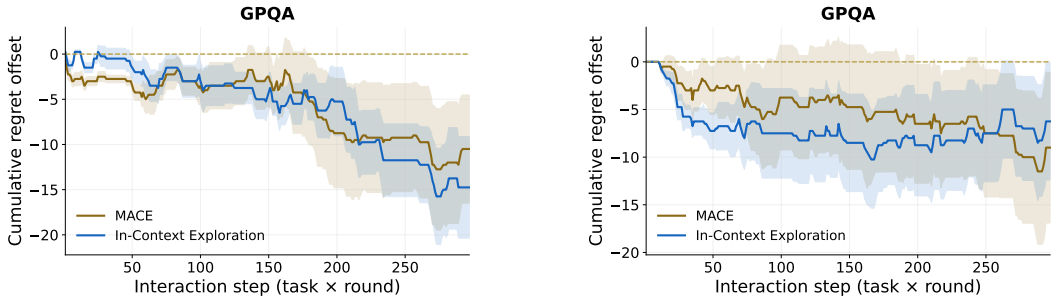
Figure 11: Sensitivity of α on the HotpotQA benchmark.

D.3 Experiments on Stronger LLMs

To further examine the role of model capabilities, we revisit the parametric diversity setting using a set of stronger frontier models: GPT-5, GPT-4, GPT-5.4-mini, and GPT-5.4-nano. Although these models vary in scale, they are all sufficiently capable of making meaningful progress on GPQA, enforcing more homogeneity in the model capabilities.

Figure 12 compares the cumulative regret offset of MACE and In-Context Exploration in this setting. The two methods exhibit broadly similar regret trends in both the Trial-and-Error and Exploitation phases. We conjecture that this is because most agents are comparably capable, and identifying the "best" peer becomes less consequential, and exploration contributes less to cumulative regret reduction. This is consistent with our theoretical intuition from Definition 1: when the capability diversity δ is small, the penalty for selecting one peer over another is reduced, and the benefit of explicit exploration correspondingly diminishes.

Interestingly, however, the round-wise task performance in Table 18 reveals a more nuanced picture. Despite the similar regret trajectories, MACE still yields stronger downstream accuracy than In-Context Exploration, with the advantage becoming especially clear in the Exploitation phase. MACE



(a) Trial-and-Error Phase (b) Exploitation Phase

Figure 12: **Cumulative regret offset on a set of stronger LLMs.** The average cumulative regret offset (Baseline – Random) across interaction steps are shown (lower the better). The horizontal dashed line at Offset $= 0$ is the Random baseline, and the shaded areas indicate the standard error across participating agents.

achieves the best final-round mean accuracy in both phases, reaching 0.7475 during Trial-and-Error and 0.7652 during Exploitation, compared to 0.7273 and 0.7222 for In-Context Exploration, respectively. This suggests that even when capability diversity is not large enough to produce a pronounced regret gap, structured exploration can still improve the quality of information exchange and lead to better final predictions. Overall, these results support the view that the regret benefit of exploration depends strongly on diversity, while the performance benefit can persist even in relatively homogeneous pools of strong agents. In light of these observations, we further provide experimental results on a fully homogeneous setting in Appendix D.4, where agents have identical capabilities and contexts, all instructed to solve shared tasks.

Table 18: Round-wise GPQA performance across Trial-and-Error and Exploitation phases, using a set of stronger models. We report accuracy for each agent and the mean accuracy across agents over interaction rounds. Round 0 corresponds to the initial response before peer interaction. Agent 1 through 4 correspond to GPT-5, GPT-4, GPT-5.4-mini, GPT-5.4-nano respectively. Best final-round performance is underlined.

Trial-and-Error Phase

(a) MACE

Round	Agent 1	Agent 2	Agent 3	Agent 4	**Mean**
0 (init)	0.6263	0.7172	0.6263	0.5556	**0.6313**
1	0.6667	0.7172	0.6869	0.6061	**0.6692**
2	0.6566	0.7475	0.7374	0.7273	**0.7172**
3	0.6970	0.7576	0.7879	0.7475	**0.7475**

(c) In-Context Exploration

Round	Agent 1	Agent 2	Agent 3	Agent 4	**Mean**
0 (init)	0.6162	0.7677	0.6465	0.4747	**0.6263**
1	0.6869	0.7677	0.6768	0.6566	**0.6970**
2	0.6869	0.7475	0.7374	0.7172	**0.7222**
3	0.6970	0.7576	0.7273	0.7273	**0.7273**

Exploitation Phase

(b) MACE

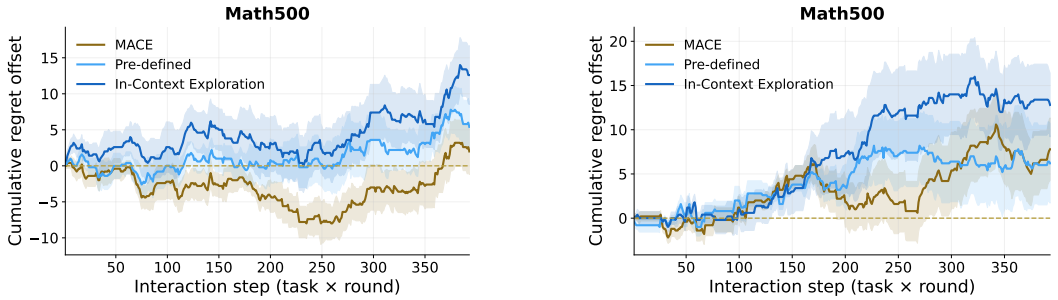
Round	Agent 1	Agent 2	Agent 3	Agent 4	**Mean**
0 (init)	0.6263	0.7374	0.6667	0.4545	**0.6212**
1	0.6566	0.7475	0.7778	0.6465	**0.7071**
2	0.6667	0.7677	0.7980	0.7879	**0.7551**
3	0.6768	0.7677	0.8081	0.8081	**0.7652**

(d) In-Context Exploration

Round	Agent 1	Agent 2	Agent 3	Agent 4	**Mean**
0 (init)	0.5758	0.7273	0.6364	0.5556	**0.6237**
1	0.6465	0.7172	0.7374	0.6566	**0.6894**
2	0.6869	0.7576	0.7374	0.7273	**0.7273**
3	0.6768	0.7374	0.7071	0.7677	**0.7222**

D.4 Fully Homogeneous Agent Setup

To isolate the role of capability diversity, we consider a strictly homogeneous setting on Math500 using five copies of Qwen2.5-7B-Instruct. In this setup, all agents share the same backbone model, receive the same input content, and solve the same task distribution, so the capability diversity δ in Definition 1 is effectively zero. As shown in Figure 13, the cumulative regret trends are indeed much less profound than in the heterogeneous settings. MACE generally maintains a lower regret offset than In-Context Exploration, especially during the Trial-and-Error phase, but the advantage is modest and often close to the dashed random-baseline line. In the Exploitation phase, the regret offset of MACE also rises above zero in later rounds, indicating that explicit exploration offers little regret advantage once agents become nearly interchangeable. This mirrors, and further amplifies, the low-diversity trend observed with the stronger frontier-model pool in Appendix D.3: when peers



(a) Trial-and-Error Phase (b) Exploitation Phase

Figure 13: **Cumulative regret offset on a set of homogeneous LLMs.** The average cumulative regret offset (Baseline – Random) across interaction steps are shown (lower the better). The horizontal dashed line at Offset $= 0$ is the Random baseline, and the shaded areas indicate the standard error across participating agents.

have similar capabilities, identifying a particular collaborator matters much less, so exploration has limited room to reduce regret.

That said, Table 19 shows that MACE still yields small but consistent improvements in round-wise task performance. Although the gains are subtle, MACE attains the best final-round mean accuracy in both phases: in Trial-and-Error, it reaches 0.4779, compared to 0.4687 for In-Context Exploration and 0.4611 for Pre-defined; in Exploitation, it reaches 0.5053, compared to 0.4962 and 0.4641, respectively. Thus, even when explicit exploration does not substantially lower cumulative regret, structured peer selection can still improve the quality of interaction and lead to slightly better final predictions. Overall, these results are consistent with our theory: as $\delta \to 0$, the regret benefit of exploration largely vanishes, but modest performance gains can still remain through better coordination over otherwise similar agents.

Table 19: Round-wise Math500 performance across Trial-and-Error and Exploitation phases, using a set of stronger models. We report accuracy for each agent and the mean accuracy across agents over interaction rounds. Round 0 corresponds to the initial response before peer interaction. Agent 1 through 5 correspond to the five agents in the homogeneous-model setup, where each agent is an instance of the Qwen2.5-7B-Instruct model. Best final-round performance is underlined.

Trial-and-Error Phase

(a) MACE

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Mean
0 (init)	0.4809	0.4351	0.4504	0.4122	0.4122	**0.4382**
1	0.4656	0.4122	0.4275	0.4504	0.4504	**0.4412**
2	0.4351	0.4275	0.4809	0.4962	0.4427	**0.4565**
3	0.4885	0.4809	0.4885	0.5267	0.4046	**0.4779**

(c) In-Context Exploration

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Mean
0 (init)	0.4809	0.4351	0.4504	0.4122	0.4122	**0.4382**
1	0.4580	0.4580	0.4275	0.4351	0.4351	**0.4427**
2	0.4733	0.4885	0.4580	0.4580	0.4962	**0.4748**
3	0.4656	0.5573	0.4427	0.4275	0.4504	**0.4687**

(e) Pre-defined

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Mean
0 (init)	0.4809	0.4351	0.4504	0.4122	0.4122	**0.4382**
1	0.4962	0.4504	0.4122	0.4122	0.4198	**0.4382**
2	0.4504	0.4351	0.4809	0.4046	0.4427	**0.4427**
3	0.5038	0.5038	0.3969	0.4504	0.4504	**0.4611**

Exploitation Phase

(b) MACE

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Mean
0 (init)	0.4580	0.4733	0.4122	0.4351	0.5267	**0.4611**
1	0.4733	0.4198	0.5038	0.4351	0.4656	**0.4595**
2	0.5038	0.4962	0.4809	0.4733	0.4275	**0.4763**
3	0.5038	0.4885	0.5344	0.5267	0.4733	**0.5053**

(d) In-Context Exploration

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Mean
0 (init)	0.4580	0.4733	0.4122	0.4351	0.5267	**0.4611**
1	0.4733	0.4656	0.4656	0.4351	0.4046	**0.4489**
2	0.4885	0.4656	0.4351	0.4351	0.4733	**0.4595**
3	0.5038	0.4580	0.4046	0.4504	0.5038	**0.4641**

(f) Pre-defined

Round	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Mean
0 (init)	0.4580	0.4733	0.4122	0.4351	0.5267	**0.4611**
1	0.4656	0.4580	0.4427	0.4580	0.4504	**0.4550**
2	0.4504	0.4656	0.5191	0.4656	0.4962	**0.4794**
3	0.4580	0.4885	0.5267	0.5115	0.4962	**0.4962**

D.5 MACE-TD: A Temporal-Difference Extension

MACE-TD. The main MACE formulation treats each interaction as a one-step contextual bandit decision. However, if agents engage in multiple rounds of discussion for each task question (like the setting in our work), an early peer query may improve not only the current response but also the

agent's future information state. To capture this delayed effect, we naturally consider a temporal-difference (Sutton, 1988) extension, MACE-TD, which models the R interaction rounds for each task as a finite-horizon sequential decision process.

Let $h_{i,r}$ denote agent i's interaction history before interaction round $r \in [0, R]$, and let $\mathbf{x}_{i,a,r}$ be the relational feature vector for querying peer a. We approximate the action-value function by a linear model

$$Q_i(h_{i,r}, a) \approx \mathbf{x}_{i,a,r}^\top \boldsymbol{\theta}_{i,a}.$$

After agent i selects peer $a_{i,r}$, observes reward $\rho_{i,r}$, and transitions to history $h_{i,r+1}$, we form the temporal-difference target

$$y_{i,r} = \rho_{i,r} + \gamma \max_{a' \in [N]} Q_i(h_{i,r+1}, a'),$$

where $\rho_{i,r}$ is the reward agent i received in round r, γ is set to 0.9. The linear parameters are then updated by replacing the immediate reward target in MACE with the temporal-difference target:

$$\mathbf{A}_{i,a_{i,r}} \leftarrow \mathbf{A}_{i,a_{i,r}} + \mathbf{x}_{i,a_{i,r},r}\mathbf{x}_{i,a_{i,r},r}^\top, \qquad \mathbf{b}_{i,a_{i,r}} \leftarrow \mathbf{b}_{i,a_{i,r}} + y_{i,r}\mathbf{x}_{i,a_{i,r},r}.$$

Action selection follows the same optimistic rule as MACE, using the estimated temporal-difference target together with an uncertainty bonus. This extension allows peer selection to account for delayed benefits of interaction, such as early exploration that improves information propagation or enables better decisions in later rounds.

Selection distribution comparison. To begin with, we compare the overall peer selection distribution of MACE-TD to the original MACE method. Interestingly, MACE-TD exhibits a slightly sharper distribution compared to MACE. We conjecture that this arises from its ability to internalize multi-turn interaction effects, potentially promoting emergent cooperative structures (see the last paragraph of this section). In particular, agents learn to favor peers that not only yield immediate gains but also lead to improved downstream outcomes, resulting in role specialization (i.e., each agent focuses on exploring a small subset of agents to reach maximum coverage across agents) and more stable interaction patterns over rounds.

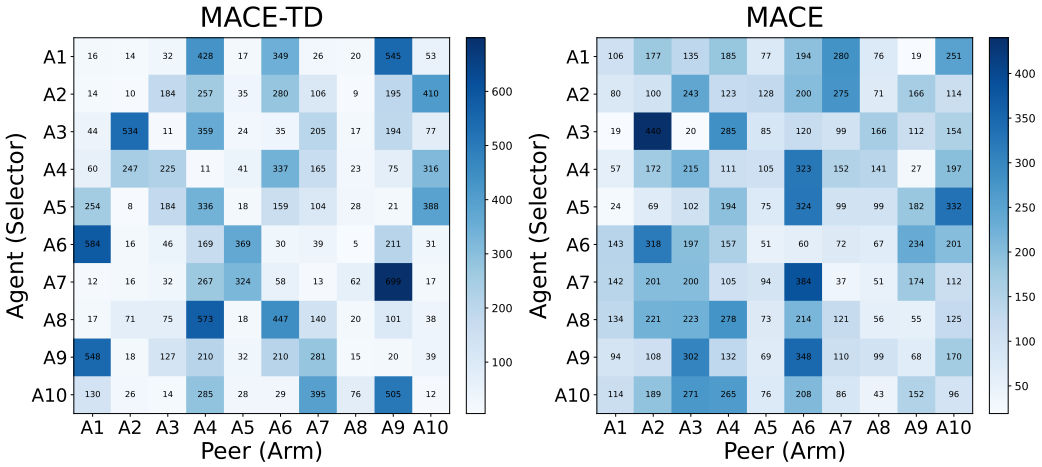


Figure 14: **Peer Selection distribution comparison: MACE vs MACE-TD**

Cumulative regrets and round-wise task performance. Figure 15 compares MACE-TD against the original MACE formulation in terms of cumulative regret offset and round-wise downstream performance. Overall, MACE-TD does not exhibit a clear advantage over MACE in cumulative regret. On HotpotQA and Math500, the regret trajectories of the two methods are broadly comparable, while on GPQA, MACE-TD underperforms MACE by accumulating noticeably higher regret, suggesting that temporal-difference updates may introduce instability when the reward signal is sparse or noisy. Interestingly, however, this disadvantage in regret does not translate into worse final task performance. In Figure 16, in GPQA, MACE-TD reaches round-wise performance that is similar to, or even slightly better than, MACE after multiple interaction rounds. This suggests that while MACE-TD may be

less efficient in minimizing cumulative regret during peer selection, its sequential value estimation can still support effective multi-round information aggregation, leading to comparable downstream performance after sufficient interaction. On the other hand, HotpotQA presents a contrasting case: although the cumulative regret trend is comparable to MACE, the exploration-phase F1 score remains noticeably lower.

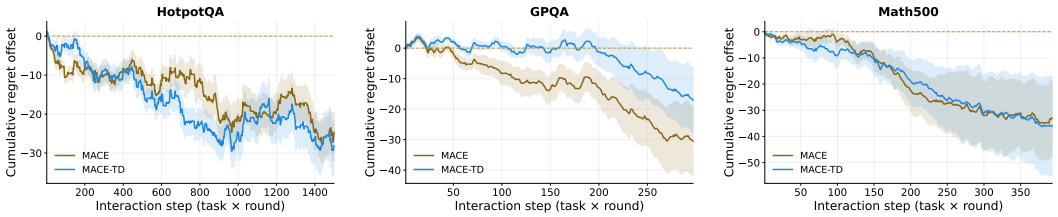


Figure 15: **Cumulative regret offset across tasks and phases: MACE-TD vs MACE.** The average cumulative reward offset (Baseline – Random) across interaction steps are shown (lower the better). The horizontal dashed line at Offset = 0 is the Random baseline, and the shaded areas indicate the standard error across participating agents.

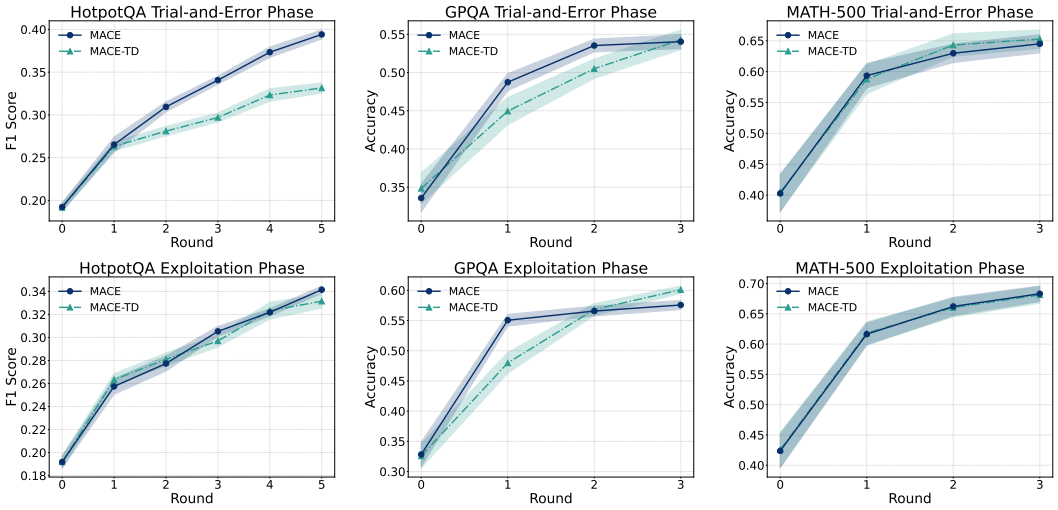


Figure 16: **Comparison of exploration and exploitation performance for each interaction round: MACE-TD vs MACE.** Shaded are the standard errors across participating agents; standard error * 0.2 are shown for Math500 and GPQA to avoid visual clutter. Also, note that the variance in the 0-th round is induced by GPT-5.

Potential Emergence of Cooperative Behaviors. Although MACE-TD does not yield a consistent advantage over MACE in cumulative regret or final task performance, it reveals an interesting qualitative pattern in how agents distribute their interactions. In particular, MACE-TD appears to encourage agents to specialize over different subsets of peers, which can expand the system-level "receptive field" of information exchange. This behavior is desirable from a cooperative perspective: if all agents repeatedly query the same peer, then only that peer's information is propagated through the system, limiting the diversity of evidence available to the group and making the problem more difficult to solve through interaction. In contrast, when different agents query different peers, information from a broader portion of the agent population can be indirectly incorporated through subsequent interactions.

To quantify this effect, we measure *peer selection coverage*, defined as the fraction of peers that are selected by at least one agent in a given interaction round. We average this quantity across agents and samples, and report its round-wise trend in Figure 17. Interestingly, MACE-TD consistently achieves higher peer selection coverage than In-Context Exploration, whereas standard MACE does not exhibit the same trend. This suggests that temporal-difference learning may induce a more system-aware interaction pattern: by optimizing for delayed rewards rather than only immediate gains, agents may learn that diversifying peer coverage improves downstream information flow. While

this observation does not by itself establish the emergence of cooperation, it provides preliminary evidence that sequential credit assignment in MACE-TD can encourage interaction structures that are more favorable for collective information sharing.

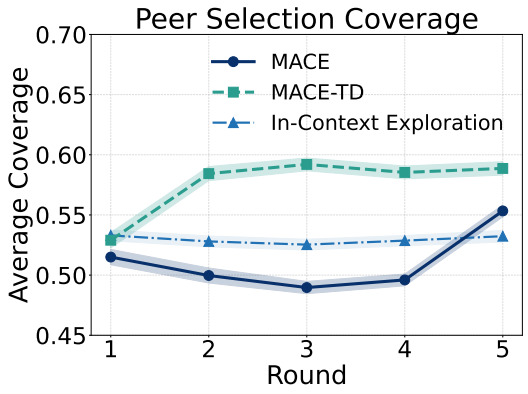


Figure 17: **The trend of peer coverage throughout interaction rounds**

D.6 Feature Analyses

In this section, we analyze the feature weights learned by MACE across agents and experimental settings.

D.6.1 Natural-Language versus Numerical Exploration Signals

We investigate whether the performance of In-Context Exploration depends on how peer information is represented. The original baseline receives the same numerical evidence used by MACE, enabling a controlled comparison between LLM-based and algorithmic peer selection. However, numerical features may be a less natural interface for LLMs than textual descriptions.

To examine this possibility, we evaluate a *Natural-Language In-Context Exploration* baseline that converts numerical statistics into qualitative descriptions of each peer's query frequency, response similarity to the selector, response distinctiveness relative to the agent population, and historical helpfulness. The selector is explicitly instructed to balance exploring uncertain peers against consulting historically helpful peers.

As shown in Table 20, the numerical and natural-language variants achieve similar performance in both phases on the HotpotQA benchmark, while MACE consistently outperforms both. These findings suggest that the gap is not primarily attributable to the numerical presentation of evidence. Instead, they support the value of an explicit exploration mechanism beyond providing peer information and prompting the LLM to balance exploration and exploitation.

Table 20: **Effect of exploration-signal representation.** In-Context Exploration performs similarly with numerical features and qualitative natural-language descriptions. MACE achieves the highest F1 and exact-match scores in both phases.

	Trial-and-Error		Exploitation	
Method	**F1**	**EM**	**F1**	**EM**
In-Context (Numerical)	0.2810	0.2257	0.3001	0.2433
In-Context (Natural Language)	0.2773	0.2193	0.3016	0.2427
MACE	**0.3941**	**0.3133**	**0.3415**	**0.2687**

D.6.2 Feature Importance

Figure 18 reports the mean learned weights $\boldsymbol{\theta}_{i,a}$ for each feature, averaged across agent pairs (i, a). Overall, the n-gram-based diversity features largely received positive weights, suggesting that MACE learns to favor peers whose responses differ from the agent's own response. This is expected, since disagreement can signal complementary information that may be useful for exploration.

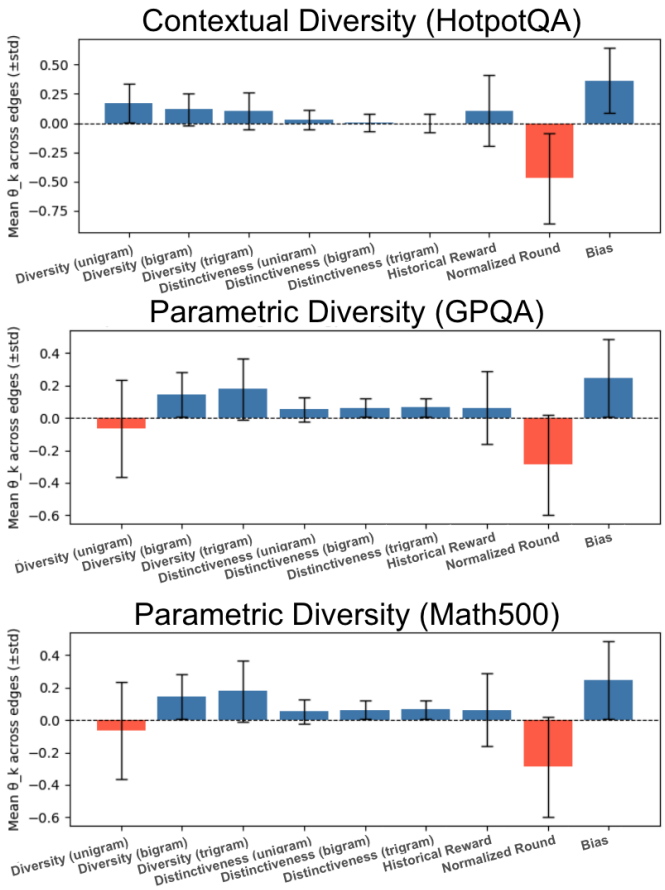


Figure 18: Learned feature weights averaged across agents.

Interestingly, the learned weights are broadly similar across contextual and parametric diversity settings. The main exception is that distinctiveness features are close to zero in the contextual diversity setting. This is intuitive: in HotpotQA, agents differ primarily by the passage they observe, rather than by their intrinsic reasoning ability. Thus, merely producing a distinctive response does not necessarily indicate that an agent is reliable or informative. It may simply reflect access to a different, possibly irrelevant, context. In contrast, under parametric diversity, distinctiveness is more useful because differences in responses can reflect differences in model capability or reasoning behavior.

D.6.3 Feature-Group Ablations

MACE uses a simple relational representation comprising intuitive signals that may inform peer selection. These features were not extensively curated or optimized; our goal is to examine whether explicit exploration can be effective with readily available peer information. To assess the contribution of each feature group, we conduct leave-one-group-out ablations on HotpotQA.

Table 21 reports performance during the Trial-and-Error phase. Full MACE achieves the highest F1 score and ties for the highest exact-match score. Removing response diversity produces the largest decrease in both metrics, reducing F1 from 0.3941 to 0.3736 and exact match from 0.3133 to 0.2963. Removing historical performance or interaction round also reduces both metrics, whereas removing peer distinctiveness has a smaller effect.

Table 21: **Feature-group ablations on HotpotQA.** Performance during the Trial-and-Error phase when each feature group is individually removed from MACE.

Method	F1 Score	Exact Match
Full MACE	**0.3941**	**0.3133**
w/o Peer distinctiveness	0.3900	**0.3133**
w/o Interaction round	0.3801	0.2997
w/o Historical performance	0.3768	0.3047
w/o Response diversity	0.3736	0.2963

These results suggest that response diversity provides particularly useful information for learning the peer-selection policy. Although this observable signal is distinct from the capability diversity δ considered in our theoretical analysis, its contribution is qualitatively consistent with the importance of distinguishing peers when guiding exploration.

E Theoretical Assumptions and Proofs

E.1 Assumptions

Here, we state the assumptions used in the theoretical analysis of MACE. For a fixed agent i, let $\mathbf{x}_{i,a,t} \in \mathbb{R}^d$ denote the contextual feature vector associated with querying peer a at round t. MACE estimates the utility of each candidate peer using a linear reward model:

$$\mathbb{E}[r_{i,t} \mid \mathcal{F}_{t-1}, a_{i,t} = a] = \mu_{i,a,t} = \mathbf{x}_{i,a,t}^\top \boldsymbol{\theta}_{i,a}, \tag{20}$$

where $\boldsymbol{\theta}_{i,a} \in \mathbb{R}^d$ is the unknown parameter associated with agent i's interaction with peer a, and $\mathcal{F}_{t-1}$ denotes the history observed before round t. Let

$$a_{i,t}^* \in \arg\max_{a \in \mathcal{N}} \ \mu_{i,a,t} \tag{21}$$

denote the optimal peer at round t. The cumulative regret of agent i is

$$\mathrm{Regret}_i = \sum_{t=1}^{T} \left(\mu_{i,a_{i,t}^*,t} - \mu_{i,a_{i,t},t} \right). \tag{22}$$

We use standard assumptions for contextual linear bandits, along with our several mild assumptions.

Assumption 1. Bounded features and parameters. For all agents i, peers a, and rounds t,

$$\|\mathbf{x}_{i,a,t}\|_2 \le 1, \qquad \|\boldsymbol{\theta}_{i,a}\|_2 \le S. \tag{23}$$

Assumption 2. Sub-Gaussian reward noise. The observed reward satisfies

$$r_{i,t} = \mathbf{x}_{i,a_{i,t},t}^\top \boldsymbol{\theta}_{i,a_{i,t}} + \eta_{i,t}, \tag{24}$$

where the noise $\eta_{i,t}$ is conditionally R-sub-Gaussian with respect to $\mathcal{F}_{t-1}$:

$$\mathbb{E}\left[\exp(\lambda \eta_{i,t}) \mid \mathcal{F}_{t-1}\right] \le \exp\left(\frac{\lambda^2 R^2}{2}\right), \qquad \forall \lambda \in \mathbb{R}. \tag{25}$$

E.2 Proof of Theorem 1

Fix a selector i. Assume that the contextual features are available before peer selection and that

$$\mu_{i,a,t} = \mathbb{E}[r_{i,t} \mid \mathcal{F}_{t-1}, a_{i,t} = a] = x_{i,a,t}^\top \theta_{i,a},$$

where $\theta_{i,a}$ is fixed over time, $\|x_{i,a,t}\|_2 \le 1$, and $\|\theta_{i,a}\|_2 \le S$. Here, $\mathcal{F}_{t-1}$ contains the information available before selection, including the current contextual features. The reward noise is conditionally σ-sub-Gaussian.

For each peer a, define

$$A_{i,a}(t) = \lambda I + \sum_{\substack{s<t \\ a_{i,s}=a}} x_{i,a,s} x_{i,a,s}^\top,$$

$$b_{i,a}(t) = \sum_{\substack{s<t \\ a_{i,s}=a}} r_{i,s} x_{i,a,s},$$

$$\widehat{\theta}_{i,a}(t) = A_{i,a}(t)^{-1} b_{i,a}(t).$$

We choose

$$\alpha_T = \sqrt{\lambda}\, S + \sigma \sqrt{d \log\left(1 + \frac{T}{d\lambda}\right) + 2\log(NT)}. \tag{26}$$

Step 1: Simultaneous confidence bounds. For each peer, the self-normalized concentration inequality for linear bandits (Abbasi-Yadkori et al., 2011) gives, with failure probability at most $1/(NT)$, simultaneously over $t \le T$,

$$\left\| \widehat{\theta}_{i,a}(t) - \theta_{i,a} \right\|_{A_{i,a}(t)} \le \sqrt{\lambda}\, S + \sigma \sqrt{\log \frac{\det A_{i,a}(t)}{\det(\lambda I)} + 2\log(NT)}.$$

The feature norm bound and the determinant–trace inequality imply

$$\log \frac{\det A_{i,a}(t)}{\det(\lambda I)} \le d \log\left(1 + \frac{T}{d\lambda}\right).$$

A union bound over the N peers therefore yields an event $\mathcal{G}$ with $\Pr(\mathcal{G}) \ge 1 - 1/T$ on which, for every peer and round,

$$\left| x_{i,a,t}^\top \left(\widehat{\theta}_{i,a}(t) - \theta_{i,a} \right) \right| \le \alpha_T \| x_{i,a,t} \|_{A_{i,a}(t)^{-1}}. \tag{27}$$

Step 2: Instantaneous regret. Let $a_{i,t}^\star \in \arg\max_a \mu_{i,a,t}$ and define

$$U_{i,a,t} = x_{i,a,t}^\top \widehat{\theta}_{i,a}(t) + \alpha_T \| x_{i,a,t} \|_{A_{i,a}(t)^{-1}}.$$

MACE selects $a_{i,t} \in \arg\max_a U_{i,a,t}$. On $\mathcal{G}$, optimism and the selection rule give

$$\mu_{i,a_{i,t}^\star,t} \le U_{i,a_{i,t}^\star,t} \le U_{i,a_{i,t},t}.$$

Applying equation 27 to the selected peer yields

$$\mu_{i,a_{i,t}^\star,t} - \mu_{i,a_{i,t},t} \le 2\alpha_T \| x_{i,a_{i,t},t} \|_{A_{i,a_{i,t}}(t)^{-1}}. \tag{28}$$

Step 3: Cumulative regret on the confidence event. Write

$$w_t = \| x_{i,a_{i,t},t} \|_{A_{i,a_{i,t}}(t)^{-1}},$$

and let T_a denote the number of selections of peer a through round T. Since $\lambda \ge 1$ and the features have norm at most one, $w_t^2 \le 1$.

The matrix determinant lemma and the inequality $u \le 2\log(1+u)$ for $u \in [0,1]$ imply

$$\begin{aligned} \sum_{t=1}^{T} w_t^2 &\le 2 \sum_{a=1}^{N} \log \frac{\det A_{i,a}(T+1)}{\det(\lambda I)} \\ &\le 2d \sum_{a=1}^{N} \log\left(1 + \frac{T_a}{d\lambda}\right) \\ &\le 2Nd \log\left(1 + \frac{T}{d\lambda}\right). \end{aligned}$$

Summing equation 28 and applying Cauchy–Schwarz gives, on $\mathcal{G}$,

$$\begin{aligned}\mathrm{Regret}_i^{\mathrm{MACE}} &\le 2\alpha_T \sum_{t=1}^{T} w_t \\ &\le 2\alpha_T \sqrt{T \sum_{t=1}^{T} w_t^2} \\ &\le 2\alpha_T \sqrt{2TNd \log\left(1 + \frac{T}{d\lambda}\right)} =: B_T.\end{aligned}$$

Step 4: Expected regret. Rewards lie in $[-1/2, 1]$, as does the reward $r_{i,t} = s_i^{(t)} - \frac{1}{2} s_i^{(t-1)}$ used in our experiments. Hence, conditional expected rewards also lie in this interval, and cumulative pseudo-regret is bounded by $3T/2$ on every trajectory. Splitting the expectation according to $\mathcal{G}$,

$$\begin{aligned}\mathbb{E}\big[\mathrm{Regret}_i^{\mathrm{MACE}}\big] &\le B_T \Pr(\mathcal{G}) + \frac{3T}{2} \Pr(\mathcal{G}^c) \\ &\le 2\alpha_T \sqrt{2TNd \log\left(1 + \frac{T}{d\lambda}\right)} + \frac{3}{2}.\end{aligned}$$

This establishes the claimed bound. For fixed N, d, λ, S, σ, the choice in equation 26 satisfies $\alpha_T = O(\sqrt{\log T})$, giving $O(\sqrt{T} \log T)$ expected regret. □

E.3 Proof of Theorem 2

Fix $N \ge 2$. Consider N peers with independent Bernoulli rewards and means

$$\mu_1 = \frac{3}{4}, \qquad \mu_a = \frac{1}{4}, \quad a = 2, \dots, N.$$

Peer 1 is optimal, and every other peer has the same suboptimality gap $\Delta = 1/2$.

Set $c_a = \mu_a$. The mean capability is

$$\bar{c} = \frac{1}{4} + \frac{1}{2N}.$$

Consequently, Definition 1 gives

$$\begin{aligned}\delta &= \frac{1}{N}\left(\frac{N-1}{2N} + (N-1)\frac{1}{2N}\right) \\ &= \frac{N-1}{N^2}.\end{aligned}$$

Let $X_{a,1}$ denote the reward from the initialization query to peer a. Consider the event

$$\mathcal{E} = \left\{ X_{1,1} = 0, \ \max_{2 \le a \le N} X_{a,1} = 1 \right\}.$$

By independence,

$$\Pr(\mathcal{E}) = \frac{1}{4}\left[1 - \left(\frac{3}{4}\right)^{N-1}\right] \ge \frac{1}{16}.$$

On $\mathcal{E}$, the optimal peer's empirical mean is zero, while at least one inferior peer has an initial reward of one. Fix such an inferior peer b. After any finite number $n \ge 1$ of queries to this peer,

$$\widehat{\mu}_{i,b} = \frac{1 + \sum_{k=2}^{n} X_{b,k}}{n} \ge \frac{1}{n} > 0.$$

Thus, at every subsequent round, at least one inferior peer has a strictly positive empirical mean. As long as peer 1 is not queried again, its empirical mean remains zero. By induction, greedy selection never returns to peer 1.

Therefore, on $\mathcal{E}$, the optimal peer is queried exactly once during initialization, and all other $T-1$ queries select inferior peers. Hence,

$$\mathrm{Regret}_i^{\text{non-exploring}} = \Delta(T-1) = \frac{T-1}{2}.$$

Using nonnegativity of pseudo-regret,

$$\mathbb{E}\Big[\mathrm{Regret}_i^{\text{non-exploring}}\Big] \geq \Pr(\mathcal{E})\frac{T-1}{2} \geq \frac{T-1}{32}.$$

Finally,

$$\delta = \frac{N-1}{N^2} \leq \frac{1}{4},$$

so

$$\frac{T-1}{32} \geq \frac{\delta}{8}(T-1).$$

This proves the theorem. □

E.4 Discussion

Theorem 1 shows that MACE achieves sublinear regret in the number of interaction rounds, scaling as

$$\widetilde{O}\Big(\sqrt{T\log T}\Big). \tag{29}$$

Thus, the average regret per round vanishes as T grows. In contrast, Theorem 2 shows that a greedy non-exploring policy can suffer regret that grows linearly in T once it prematurely commits to a suboptimal peer.

The role of capability diversity is captured by the factor δ. When $\delta \approx 0$, peers are nearly interchangeable, and the cost of selecting the wrong peer is small. In this regime, explicit exploration may provide only limited benefit. When δ is large, however, agents are specialized, and peer selection becomes consequential. The regret gap in Corollary 1 therefore grows as

$$\Omega(\delta T) - \widetilde{O}(\sqrt{T\log T}), \tag{30}$$

showing that the value of exploration increases directly with the diversity of the agent pool.

F Appendix for Related Works

Here, we provide an extended version of the Related Works section of the manuscript. We give a broad overview of the landscape of multi-agent LLM systems, and discuss approaches that specifically target autonomous multi-agent systems (Section F.1). We further discuss works that conducted in-depth experiments and analysis on the exploratory ability of single LLM agents (Section F.2).

F.1 Autonomous Multi-Agent LLM Systems

Autonomous multi-agent Overview. Recent autonomous LLM systems have shifted from isolated, single-agent prompting toward coordinated groups of agents that divide roles, exchange information, and jointly plan or act. Recent overviews (Guo et al., 2024; Li et al., 2024a; Ferrag et al., 2025; Tran et al., 2025) collectively organize the space around agent profiling, communication structure, collaboration protocol, and system evolution.

Frameworks for multi-agent coordination. A major line of work builds general-purpose infrastructures for multi-agent coordination. CAMEL (Li et al., 2023) introduced role-playing communicative agents for autonomous cooperation, while AutoGen (Wu et al., 2024) cast multi-agent interaction as programmable conversation among agents, humans, and tools. ChatDev (Qian et al., 2024), MetaGPT (Hong et al., 2023), and AgentScope (Gao et al., 2024) further instantiated multi-agent orchestration in which specialized agents collaborate through structured workflows. Beyond these, DyLAN (Liu et al., 2024c) studied learnable designs for dynamic agent networks, while recent work on self-resource allocation (Amayuelas et al., 2025) examines how LLM-based planners and

orchestrators allocate tasks across worker agents under cost, efficiency, and capability constraints. AgentVerse (Chen et al., 2023) and MegaAgent (Wang et al., 2025) further proposed collaborative groups that are autonomously designed by LLM agents. In parallel, benchmark-oriented work has begun to isolate the coordination abilities required by such systems: LLM-Coordination (Agashe et al., 2025) evaluates LLM agents in pure coordination games and shows that, while LLMs can coordinate well when decisions depend on observable environmental state, they remain limited in theory-of-mind reasoning and joint planning. Self-Consistency (Wang et al., 2023) and ModeX (Choi & Li, 2026) instead devise multi-agent response aggregation methods based on voting and consensus. Moreover, Multi-Agent Debate (MAD) frameworks have been commonly explored as a hallmark of collective intelligence (Du et al., 2024; Chan et al., 2024; Liu et al., 2024a; Li et al., 2024b; Smit et al., 2023; Liang et al., 2024; Becker et al., 2025), while several recent works report failure modes and limitations of the MAD protocol (Smit et al., 2023; Huang et al., 2023; Estornell & Liu, 2024; Liu et al., 2024a; Cemri et al., 2025; Choi et al., 2025; Kaesberg et al., 2025; Choi et al., 2026). Subsequent works have also proposed methods to enhance self-improvement through group-evolution (Weng et al., 2026). Together, these works establish the engineering foundations of autonomous multi-agent LLM systems, although most still rely on externally specified roles, orchestration rules, or known worker capabilities rather than learning whom to interact with and how to coordinate from experience.

Granting Autonomy to Multi-Agent LLM Systems. Several works have focused on the autonomy aspect of multi-agent LLM systems. A work on iAgents (Liu et al., 2024b) discussed human-computer interaction paradigms for autonomous communication between agents, while a body of works studied approaches to autonomously design and assign roles to LLM agents. Specifically, HALO (Hou et al., 2025) introduced a system of hierarchical orchestration with high-level planning, role design, and workflow search. Puppeteer (Dang et al., 2025) proposes a reinforcement learning approach to train a central agent that orchestrates sub-agents dynamically and autonomously. MegaAgent (Wang et al., 2025) and AgentVerse (Chen et al., 2023), on the other hand, autonomously design the role specifications in recruiting sub-agents to solve the given task. Other works enforced autonomy in solving domain-specific tasks (Sanwal & Deva, 2024), while (Händler, 2023) attempted to assess the current state and taxonomy of autonomy in multi-agent LLM architectures.

F.2 LLM Exploration Capabilities

Single-agent LLM exploration capabilities. A recent line of work asks whether a *single* LLM can explore effectively when placed in sequential decision-making problems. (Krishnamurthy et al., 2024) studied in-context bandits and found that robust exploration is rare in LLMs. (Nie et al., 2025) deepened this evaluation perspective by introducing a bandit benchmark suite for measuring LLM exploration and showing that explicit algorithmic support, distillation, and fine-tuning can improve exploratory behavior. Beyond bandits, (Pan et al., 2025) study open-ended exploration and report that most LLMs underperform humans because they rely too heavily on uncertainty and make premature decisions, while (Zhang et al., 2025) compare LLMs and humans on standard multi-armed bandits and show that thinking-enabled models become more human-like in simple stationary settings but still struggle with directed exploration in more complex environments. (Arumugam & Griffiths, 2025) sharpens this picture by showing that explicitly instantiating a classical exploration algorithm such as PSRL inside an LLM can be substantially more effective than expecting exploration to emerge implicitly from prompting alone. Most recently, (Park et al., 2026) showed that exploration-exploitation errors are quantifiable in LLMs and revealed that there are systematic inefficiencies in how they balance exploration and exploitation.

Delegation, trust, and adjacent capabilities for exploration. A closely related strand studies the capabilities a single LLM agent needs in order to decide *whom* to rely on under uncertainty, which is adjacent to exploration in delegation-style settings. (Tomašev et al., 2026) frame intelligent AI delegation as an adaptive sequence of decisions over task allocation, transfer of authority, and trust, emphasizing that autonomous delegation requires judging when another agent is worth relying on. (Buyl et al., 2025) show that explicit self-reported trust between LLMs can be weakly aligned, or even negatively aligned, with implicit behavioral measures such as persuasion susceptibility and collaborative willingness, suggesting that naive verbalized confidence is a poor proxy for reliable partner selection. Relatedly, (Debnath et al., 2025) examine whether LLMs can reason about and actively foster trust in dyadic interactions. Although these papers are not exploration papers in the

bandit sense, they are highly relevant to our setting because effective exploration in multi-agent delegation depends not only on trying alternatives, but also on assessing partner reliability, calibrating trust, and avoiding over-commitment to noisy early impressions.

G BROADER IMPACT

Most discussions of AI reliability focus on single models acting in isolation. However, many future deployments may consist of populations of interacting agents that communicate, delegate, negotiate, and coordinate with one another while operating alongside humans. Despite this shift, the question of how multi-agent AI systems can integrate seamlessly into broader social and organizational environments remains comparatively underexplored. We view this as the challenge of building *socially reliable multi-agent AI*. This work contributes to this agenda by identifying a fundamental obstacle to reliable multi-agent autonomy: insufficient exploration of peers and interaction strategies.

A positive implication is that socially reliable multi-agent systems could better support domains such as scientific collaboration, digital workforces, education, healthcare coordination, and public-service decision support, where success depends on combining distributed expertise rather than relying on a single model. By improving how agents discover useful partners, adapt to changing environments, and allocate communication efficiently, methods such as MACE may help AI systems participate more naturally in human institutions and mixed human-AI teams. At the same time, stronger coordination among autonomous agents may introduce risks, including scalable manipulation, collusion, disinformation campaigns, or opaque collective behavior that becomes difficult to monitor. As multi-agent capabilities improve, governance and safety mechanisms become increasingly important. Future research should therefore pair advances in coordination with controllable communication protocols, fairness-aware objectives, and human oversight in high-stakes settings.

Overall, we hope this work helps broaden the conversation from reliable *models* to reliable *societies of models*. Enabling seamless and trustworthy integration of multi-agent AI systems into real-world social environments may become one of the central challenges of the next generation of AI.