ENHANCING DECISION-MAKING WITH LARGE LANGUAGE MODELS THROUGH MULTI-AGENT FICTITIOUS PLAY

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM)-based multi-agent systems (MAS) have demonstrated great potential in solving tasks with execution complexity, by distributing subtasks across cooperative agents. However, this divide-and-conquer paradigm falls short in mixed-motive environments such as negotiation and competitive markets, where we control only one agent and the others pursue their own payoffs. In these environments, the quality of our strategy depends on how the others respond, so the strategies are mutually dependent and leave no independently solvable subtasks to divide. Reasoning over this dependence within a single LLM call unfolds recursive mutual anticipation, which quickly exceeds the model's capability. Inspired by the classic learning dynamic of Fictitious Play, we propose Multi-Agent Fictitious Play (MAFP), a novel MAS paradigm that converts this recursive anticipation into iterative strategy updates. To obtain a policy for a distinguished agent, MAFP instantiates a simulated agent for all agents. At each iteration, every agent generates a new decision that seeks to maximize its expected utility against other agents' past decisions. After several rounds, MAFP synthesizes a policy from the distinguished agent's history for deployment. Throughout the process, each agent only needs first-order reasoning, as it best responds to a fixed pool of observed decisions. We evaluate MAFP on decision-making tasks that require deciding a strategy for competitive scenarios prior to acting. Across scenarios spanning competitive games and negotiation, MAFP outperforms other multi-round baselines, and retains higher payoff when others adapt against it, validating it as an effective MAS framework for decision-making.

1 INTRODUCTION

Large language model (LLM)-based multi-agent systems (MAS) (Zhang et al., 2024b; Tran et al., 2025) have emerged as a powerful paradigm for solving complex tasks that exceed the capability of a single LLM call (Wang et al., 2024a), with applications in software engineering (Deng et al., 2025), deep research (Xu & Peng, 2025), and scientific discovery (Zhao et al., 2026). The core of these MAS is divide and conquer (Gu et al., 2025), where each agent corresponds to a part of the task, accomplishing the task collaboratively. By doing so, MAS reduces the difficulty faced by each individual agent and alleviates the context-window pressure that bottlenecks single-agent reasoning (Team et al., 2026), thereby increasing the overall performance. The complexity that existing MAS address is primarily *execution complexity*: tasks are difficult because their execution requires long reasoning chains (Lei et al., 2024), broad information coverage (Xu & Peng, 2025; Manus, 2025), or heterogeneous skills (Zhao et al., 2026), thus these burdens can be distributed across cooperative agents, as demonstrated in Fig. 1 (Left).

However, the situation is more complex in many real-world environments, including those such as negotiation (Bianchi et al., 2024), game-playing (Duan et al., 2024), and competitive markets (Backlund & Petersson, 2025; Zhang et al., 2026). Here, our goal is to design a policy for one distinguished agent operating among others. At deployment, we choose only this agent's policy; the other agents pursue objectives of their own and may adapt as they observe play. Consequently, the policy's payoff depends on responses we cannot directly specify. Designing it requires accounting for the behavior of strategically motivated agents beyond our control. Game theory makes this dependence

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.

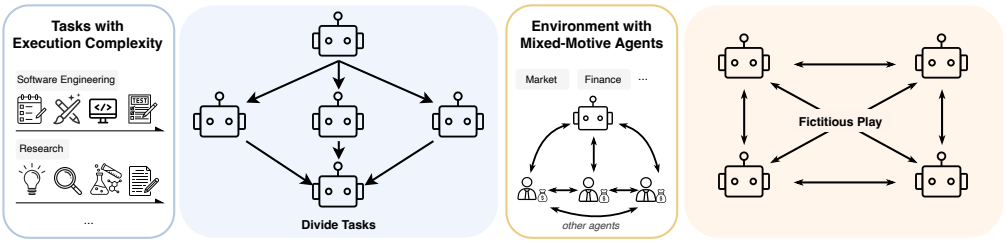


Figure 1: Existing MAS address execution complexity (left), by dividing a task into subtasks across cooperative agents. In contrast, MAFP targets environments with mixed-motive agents (right), where we control only one agent and the agents' decisions are mutually dependent. It derives a good strategy for our agent through fictitious play.

explicit. At a Nash equilibrium, each agent's strategy is a best response to the others' strategies. These conditions are coupled, since changing one strategy can change what is optimal for another agent. More generally, we are faced with a coupled policy-design problem: changing our agent's policy may change how the others respond, and those responses determine its payoff. We therefore need to explore possible responses while choosing *only* the policy we will deploy.

Inspired by the classic learning dynamic of Fictitious Play (FP, Brown (1951)), we design *Multi-Agent Fictitious Play (MAFP)*, a multi-agent framework with mixed-motive agents (Schelling, 1960; Dafoe et al., 2020) that each maximize their individual payoff rather than directly optimize for a shared objective. To obtain a policy for a distinguished agent, MAFP instantiates a simulated agent for all agent. At each iteration, every agent generates a new decision that seeks to maximize its expected utility against other agents' past decisions. After several rounds, MAFP synthesizes a policy from the distinguished agent's history for deployment. Interaction among agents is thus explored across simulated updates, with each policy-generation step conditioned on a concrete pool of opponent policies. Throughout the process, each agent only needs first-order reasoning, as it best responds to a fixed pool of observed decisions. This avoids having to form complicated, higher-order beliefs about how the others would behave or respond (Wu et al., 2023; Kim et al., 2023). MAFP represents a *de-centralized* form of *co-evolution* (Hillis, 1990; Rosin & Belew, 1997), in which each player iteratively probes the exploitable weaknesses of the others and patches its own flaws.

We evaluate MAFP on decision-making tasks across two-player scenarios (Duan et al., 2024; Bianchi et al., 2024), multi-player scenarios (Light et al., 2023; Zha et al., 2019), and a long-horizon economic simulation (Sugiura et al., 2026). These tasks require the model to decide an open-ended strategy for each scenario before acting, which is challenging due to the vast space of possible strategies. We represent strategies by a *code-based decision tree* for simulation and evaluation. We evaluate the resultant decision quality with two complementary metrics, the *tournament score* and the *Stackelberg value*. The former measures a decision's average payoff against all alternative candidates being evaluated, while the latter measures the payoff our agents retains when other agents adapt against it (von Stackelberg, 1934; Conitzer & Sandholm, 2006). Experimental results show that MAFP outperforms other multi-round baselines on both metrics, validating it as an effective MAS framework for decision-making. Our contributions are summarized as follows.

Contributions. We study LLM-based policy generation for a *distinguished agent in mixed-motive multiagent environments*, where the other agents' policies cannot be chosen at deployment. Our main contribution is MAFP: a fictious-play based method that simulates all agents, iteratively generates and tests policy programs against empirical pools of opponent policies, and finally synthesizes a deployable policy for the distinguished agent. We evaluate MAFP on 13 two-player scenarios, three multiplayer games, and a supply-chain simulation based on tournament scores and Stackelberg values. Our results show that MAFP achieves higher average scores than other baselines.

2 Related Work

Classical Fictitious Play. Classical fictitious play (FP) has each player best respond to the empirical distribution of opponents' past actions (Brown, 1951). Unlike solvers that compute coupled strategy

profiles, in FP players update independently based on individual payoffs and observed play. Empirical strategies approach the Nash equilibrium set in finite two-player zero-sum games (Robinson, 1951) and finite weighted potential games (Monderer & Shapley, 1996), but convergence may fail more generally (Shapley, 1964). The proposed MAFP adopts similar updates in the setting of LLM-generated policies but may not inherit the classical guarantees. FP itself belongs to a class of methods involving *self-play*, which also features both online-learning and population-based methods such as PSRO (Freund & Schapire, 1999; Zinkevich et al., 2007; Lanctot et al., 2017).

LLM-based Multi-Agent System. MAS (Liu et al., 2024; Li et al., 2023) demonstrates effectiveness on a wide range of complex tasks (Xu & Peng, 2025; Deng et al., 2025; Merrill et al., 2026; Wei et al., 2025) by having multiple LLM-based agents work collaboratively. These systems target tasks with execution complexity and operate through a divide-and-conquer paradigm. Tasks that demand long execution chains, such as software engineering (Deng et al., 2025) and deep research (Xu & Peng, 2025), are decomposed across role-specialized agents and executed sequentially (Qian et al., 2024; Hong et al., 2024); those that demand broad information coverage or divergent exploration, such as wide research (Manus, 2025) and multi-agent debate (Liang et al., 2024), are executed in parallel (Du et al., 2024; Wang et al., 2025).

Existing research primarily operates within this cooperative paradigm, focusing on how to split and arrange tasks effectively, spanning orchestrator optimization (Wu et al., 2024; Chen et al., 2024b; Team et al., 2026) and topology optimization (Zhuge et al., 2024; Yang et al., 2026; Xu et al., 2026). In all of these systems, the agents optimize one shared objective. This premise does not hold in mixed-motive environments (Duan et al., 2024; Chen et al., 2023; Backlund & Petersson, 2025), where we control only our own agent and the other agents pursue their own payoffs. The quality of our strategy then depends on how the others respond, leaving no independently solvable subtasks to divide. Even in multi-agent debate (Liang et al., 2024; Du et al., 2024), where agents debate for different views, they still share the goal. In contrast, MAFP uses a MAS with mixed-motive agents to simulate the game and evolve a good strategy for our agent.

LLM Agents in Mixed-Motive Environments. As LLMs (Yang et al., 2025; Singh et al., 2025; Liu et al., 2026) grow more capable, they are increasingly deployed in mixed-motive environments, and a wave of benchmarks evaluates this capability across games (Duan et al., 2024; Zhang et al., 2024a; Costarelli et al., 2024; Akata et al., 2025; Chen et al., 2024a), negotiation (Bianchi et al., 2024; , FAIR; Rivera et al., 2024), social deduction (Guo et al., 2023; Bailis et al., 2024; Light et al., 2023), and competitive market (Chen et al., 2023; Backlund & Petersson, 2025; Wang & Zhang, 2025; Yun et al., 2025). Existing attempts deal with such environments through strategic reasoning, performing explicit theory-of-mind (Frith & Frith, 2005) (ToM) reasoning at inference time. Their solutions range from one-step perspective-taking on opponents (Wilf et al., 2024; Cross et al., 2024), to two-order reasoning that additionally anticipates how others perceive the agent (Wang et al., 2024b), to recursive level-k mutual anticipation that extends ToM to arbitrary depth (Zhang et al., 2025). What unites these methods is the assumption that an LLM can carry out higher-order belief reasoning within a single inference pass.

However, recursive mutual anticipation is a structural weak point of LLMs, which are essentially autoregressive next-token predictors (McCoy et al., 2024a;b). This limitation is verified empirically by Hi-ToM (Wu et al., 2023) and FANToM (Kim et al., 2023), showing that LLM accuracy drops rapidly in higher-order belief reasoning (e.g., "I think that you think") and exhibits an illusory theory of mind. As the number of agents grows, the depth and breadth of such reasoning quickly exceed the capability of a single LLM call. MAFP avoids this recursion by instantiating every agent with an LLM and letting their strategies co-evolve in simulated games.

3 Policy Design for a Distinguished Agent

We consider environments that involve multiple mixed-motive agents (Schelling, 1960; Dafoe et al., 2020), such as negotiations (Bianchi et al., 2024), market competitions (Backlund & Petersson, 2025; Zhang et al., 2026; Sugiura et al., 2026), and financial trading (Yun et al., 2025).

In these environments, the strategy an agent generates will be played against the other agents, each pursuing its own goal. We formulate such environments as games, where an agent needs to commit to a *strategy* before the interaction begins. Specifically, the game is described in natural language by $\mathcal{D}$,

which defines the set of agents $\mathcal{N} = \{1, \ldots, N\}$ and each agent i's own information ω_i, covering its goal and private context. Without loss of generality, we let agent 1 be the distinguished agent whose strategy we design.

A strategy specifies how to act in every situation (Shoham & Leyton-Brown, 2008), and thus takes the form of a decision tree. For evaluation, we represent this decision tree as an executable program, called the *code-based decision tree* of agent i and denoted by $\pi_i \in \Pi_i$, where Π_i is the space of admissible strategies. Concretely, a strategy is a mapping from observations to actions:

$$a_i \;=\; \pi_i(o_i;\, \xi)\,, \tag{1}$$

where o_i is agent i's observation, comprising its view of the current state together with the currently legal actions, a_i is the action it takes, and ξ is a seeded random source through which the strategy can randomize over actions, realizing mixed strategies (Shoham & Leyton-Brown, 2008).

We evaluate a strategy by executing it in the game and computing payoffs from the outcomes it realizes. The execution starts from the initial state constructed from $\mathcal{D}$, and each agent repeatedly chooses its action through its own strategy as in Eq. 1 until a terminal state is reached. We write $\tau \sim \mathrm{Env}(\mathcal{D};\, \pi_i,\, \boldsymbol{\pi}_{-i})$ for the realized trajectory, where $\mathrm{Env}(\cdot)$ denotes this execution process, $\boldsymbol{\pi}_{-i} = (\pi_j)_{j \neq i}$ collects the other agents' strategies, and the randomness of τ comes from both the strategies and the environment's own random events. From the realized trajectory, the environment computes a payoff $R_i(\tau)$ for each agent i according to the game's payoff rules.

Since the payoff of a strategy inherently depends on what the others play, we define the utility of π_i against a distribution $\mathcal{P}$ over the other agents' strategies:

$$U_i(\pi_i,\, \mathcal{P}) \;=\; \mathbb{E}_{\boldsymbol{\pi}_{-i} \sim \mathcal{P}}\, \mathbb{E}_{\tau \sim \mathrm{Env}(\mathcal{D};\, \pi_i, \boldsymbol{\pi}_{-i})}[R_i(\tau)]\,, \tag{2}$$

and when $\mathcal{P}$ is a point mass at a specific $\boldsymbol{\pi}_{-i}$, we write the utility as $U_i(\pi_i, \boldsymbol{\pi}_{-i})$.

Evaluation metrics. A good strategy must achieve a high average payoff against the strategies it currently faces, and retain its payoff when the others adapt. This is because the other agents are goal-driven, and the strategies they play are not fixed a priori. Once they observe or infer the committed π_1, they adjust their strategies toward whatever better serves their own goals. We measure the two aspects with two complementary metrics, the *tournament score* and the *Stackelberg value*.

The tournament score instantiates the utility $U_1(\pi_1, \mathcal{P})$ of Eq. 2, with $\mathcal{P}$ being the distribution over the strategies under evaluation. The Stackelberg value (SV) is the utility a strategy retains when the others observe it and respond in their own interest (von Stackelberg, 1934; Conitzer & Sandholm, 2006). We state it for the two-agent case, in which it is evaluated:

$$\mathrm{SV}(\pi_1) \;=\; U_1(\pi_1,\, \pi_j^{\mathrm{br}}), \qquad \pi_j^{\mathrm{br}} \;\in\; \arg\max_{\pi_j \in \Pi_j}\; U_j(\pi_j,\, \pi_1), \tag{3}$$

where j denotes the other agent, $U_j(\pi_j, \pi_1)$ denotes its own utility.

Deciding such a strategy differs structurally from the tasks existing MAS solve. In cooperative tasks, all agents optimize a single shared objective, and the typical approach is to apply a divide and conquer approach, distributing smaller subtasks across available agents. In contrast, for our mixed-motive setting we control only our own agent, while other agents pursue their own goals payoffs independently. Consequently, we are only able to select our own strategy π_1, yet its quality is defined only relative to the others' strategies $\boldsymbol{\pi}_{-1}$, which are in turn chosen in response to π_1. The strategies are therefore mutually dependent and cannot be decided in isolation, leaving no independently solvable subtasks to divide. Reasoning over this mutual dependence within a single LLM call instead unfolds recursive "what-if-they-think-that-I-think" anticipation, whose depth grows with the number of agents and quickly exceeds the model's capability (Wu et al., 2023; Kim et al., 2023). We therefore design MAFP, which uses the multi-agent system in a different way (Section 4).

4 Multiagent Fictitious Play (MAFP)

To decide a good strategy under such mutual dependence, we draw inspiration from *fictitious play* (Brown, 1951), a classic learning dynamic in game theory. As illustrated in Fig. 2, fictitious play operates by having each player iteratively best respond to the empirical distribution of the opponents'

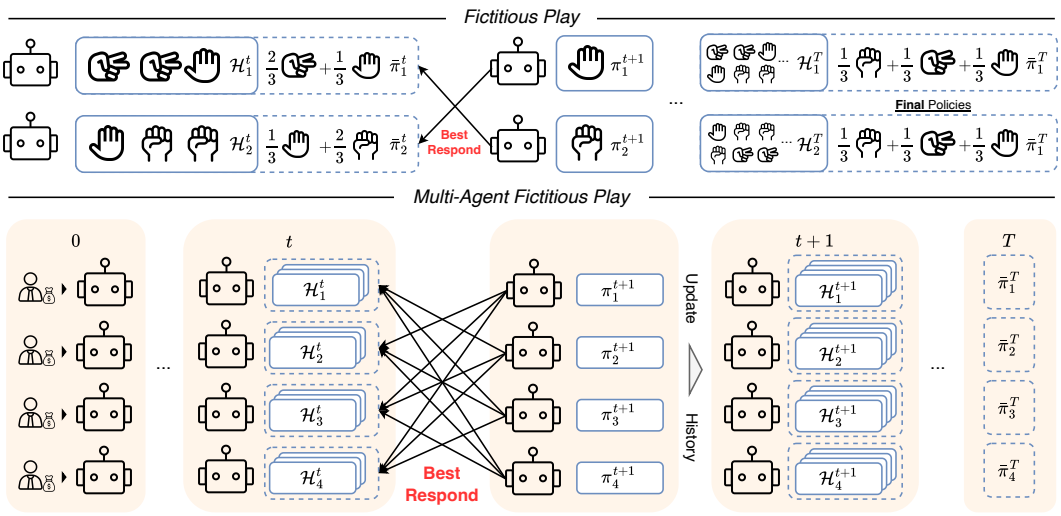


Figure 2: **Illustration of MAFP algorithm.** *Fictitious play* finds equilibrium through an iterative process in which each player best responds to the empirical average of others' past actions. Inspired by this, MAFP derives strategies through multi-agent fictitious play, where agents iteratively update strategies by best responding to others' past strategies.

past plays. The mutually dependent solving is thereby converted into iterative local best responses, where each update conditions on the opponents' observed plays alone, without anticipating how they would counter.

Following this learning dynamic, we design MAFP, which finds a good strategy by letting different agents' strategies co-evolve in simulated games before deployment. Although we control only our own agent at deployment, in simulation MAFP can instantiate every agent $i \in \mathcal{N}$ with an LLM. Each instantiated agent maximizes its own payoff as its counterpart in the game does, forming a mixed-motive multi-agent system. The co-evolution proceeds through K iterative rounds (Fig. 2). In each round, every agent generates a new strategy that maximizes its utility against the pool of the other agents' past strategies, exploiting the weaknesses exposed in the others' play while patching those the others have found in its own. After the final round, each agent aggregates its own pool into its final strategy for deployment. We detail each component in the following.

Multi-agent initialization. Each agent is initialized with a strategy π_i^0 and a history pool $\mathcal{H}_i$, a multiset that collects the strategies it generates. The first strategy is generated from the game description $\mathcal{D}$ and the agent's own information ω_i,

$$\pi_i^0 \sim \mathrm{Init}_M(\mathcal{D}, \omega_i), \qquad \mathcal{H}_i^0 = \{\pi_i^0\}, \tag{4}$$

where Init_M prompts M to draft an initial strategy for agent i. Throughout, Init_M, BR_M, and Agg_M are realized by prompting the same frozen LLM M, and we write $\sim$ since each call samples its output from M, with prompt templates given in Appendix C.

Iterative updates. At iteration t, each agent best responds based on how the opponents have played so far. Following fictitious play, the agent treats each opponent j as playing a strategy drawn uniformly at random from its pool $\mathcal{H}_j^t$, so the pools serve as the agent's belief about the opponents. We realize the best response with BR_M, which conditions on the opponents' history pools $\mathcal{H}_{-i}^t$,

$$\pi_i^{t+1} \sim \mathrm{BR}_M\big(\mathcal{D},\, \omega_i,\, \mathcal{H}_{-i}^t\big)\,, \qquad \mathcal{H}_{-i}^t = \big(\mathcal{H}_j^t\big)_{j \neq i}\,, \tag{5}$$

and the result is appended to the pool, $\mathcal{H}_i^{t+1} = \mathcal{H}_i^t \cup \{\pi_i^{t+1}\}$, with the union taken in the multiset sense so that $|\mathcal{H}_i^{t+1}| = t + 2$.

The quality of the best response can be further improved through a best-of-n scheme, where the agent samples n candidates from BR_M and adopts the one with the highest utility measured in simulated

Algorithm 1 Multi-Agent Fictitious Play

Require: Game description $\mathcal{D}$, agents' $\{\omega_i\}_{i\in\mathcal{N}}$, frozen LLM M, iterations K, candidates n
Ensure: Final strategies $\boldsymbol{\pi}^{\text{out}}$

```
for each agent i ∈ N do
    Initialize π_i^0 ~ Init_M(D, ω_i) and H_i^0 = {π_i^0}
end for
for t = 0 to K − 1 do
    for each agent i ∈ N in parallel do
        Generate n best-response candidates:  π_i^(1), ..., π_i^(n) ~ BR_M(D, ω_i, H_{-i}^t)
        Select the best candidate by Eq. 6:  π_i^{t+1} = arg max_{π ∈ {π_i^(1),...,π_i^(n)}} Û_i(π, μ_{-i}^t)
        Update pool:  H_i^{t+1} = H_i^t ∪ {π_i^{t+1}}
    end for
end for
for each agent i ∈ N do
    π_i^out ~ Agg_M(D, ω_i, H_i^K)
end for
return π^out = (π_i^out)_{i∈N}
```

matches against the opponents' mixtures,

$$\pi_i^{t+1} \;=\; \arg\max_{\pi\in\{\pi_i^{(1)},\ldots,\pi_i^{(n)}\}} \hat{U}_i(\pi,\, \mu_{-i}^t), \qquad \pi_i^{(1)},\ldots,\pi_i^{(n)} \;\sim\; \mathrm{BR}_M\big(\mathcal{D},\, \omega_i,\, \mathcal{H}_{-i}^t\big)\,, \tag{6}$$

where μ_j^t denotes the uniform mixture over pool $\mathcal{H}_j^t$, $\mu_{-i}^t = (\mu_j^t)_{j\neq i}$ draws each opponent's strategy independently from its own pool, and $\hat{U}_i$ denotes the estimate of the utility U_i from the simulated matches. The selection brings the realized update closer to an exact best response against the belief.

Final strategy aggregation. In fictitious play, it is the empirical mixture of past plays, rather than the last play, that carries the convergence guarantees (Robinson, 1951; Monderer & Shapley, 1996). After K rounds, we therefore take the empirical mixture of all strategies generated by each agent as its final strategy. This mixture also captures the adaptations the agent accumulated in response to every weakness that the others exploited across the rounds. Concretely, we realize this aggregation with Agg_M, which reads the pool and writes one strategy representing the uniform mixture over it,

$$\pi_i^{\text{out}} \sim \mathrm{Agg}_M\big(\mathcal{D},\, \omega_i,\, \mathcal{H}_i^K\big)\,, \tag{7}$$

which is the final strategy for agent i. MAFP returns the final strategies of all agents, $\boldsymbol{\pi}^{\text{out}} = (\pi_i^{\text{out}})_{i\in\mathcal{N}}$, and an agent commits its own strategy at deployment. The full procedure is presented in Algorithm 1.

5 Experiments

In this section, we conduct experiments to investigate three research questions. **RQ1:** How does MAFP perform compared with existing multi-round frameworks? **RQ2:** How do strategies evolve across iterations under different multi-round methods? **RQ3:** How does MAFP perform on games with more than two players and on realistic economic scenarios?

5.1 Experimental Setup

Scenarios. We evaluate MAFP on 13 scenarios spanning two strategic decision-making categories: competitive games (Duan et al., 2024) and natural-language negotiation (Bianchi et al., 2024), representing strategic games with clear rules and language-based real-world tasks. Among them, 7 scenarios are zero-sum, in which the seats' payoffs sum to zero and an optimal strategy typically exists. The other 6 are general-sum, in which the payoffs no longer cancel out; these scenarios are more challenging, as no fixed optimal strategy exists and the desirable play depends on how the others play. Detailed scenario description can be found in Appendix B.1. Each match payoff is

Table 1: **Tournament score (TS).** Each method's mean normalized payoff against the other six methods under a round-robin schedule, grouped by game class. Best per column in bold.

	Zero-sum							General-sum						
Method	TicTac	Nim	Conn4	Pig	BrkThru	KuhnPk	LiarDice	IPD	BlindAuc	Negot	BuySell	Ultimat	ResExch	Avg.(13)
CoT	0.061	0.047	-0.169	-0.171	-0.224	-0.007	-0.133	0.453	0.229	-0.155	-0.117	0.378	**0.979**	0.090
SR	-0.064	0.036	-0.150	-0.044	-0.065	0.008	-0.029	0.470	0.253	-0.280	0.028	0.450	0.917	0.118
Debate	-0.001	**0.141**	-0.094	-0.068	0.013	0.037	**0.281**	0.451	0.239	-0.107	-0.041	0.465	**0.979**	0.177
ToM	0.062	-0.047	-0.419	-0.332	0.188	-0.066	-0.188	**0.563**	0.229	-0.069	0.064	0.239	**0.979**	0.093
MAFP	**0.070**	0.065	0.438	0.224	**0.232**	**0.075**	0.224	0.471	**0.256**	**0.549**	**0.383**	0.323	**0.979**	**0.330**
w/o BoN	-0.065	-0.099	**0.449**	**0.258**	-0.029	-0.027	-0.008	0.465	0.247	0.167	0.140	0.327	**0.979**	0.216
w/o pool	-0.064	-0.143	-0.055	0.133	-0.115	-0.021	-0.148	0.445	0.247	0.221	0.298	**0.479**	0.802	0.160

Table 2: **Stackelberg value (SV).** For each scenario, strategies are evaluated by letting an adaptive opponent observe and evolve its strategy over 6 rounds. Best per column in bold.

	Zero-sum							General-sum						
Method	TicTac	Nim	Conn4	Pig	BrkThru	KuhnPk	LiarDice	IPD	BlindAuc	Negot	BuySell	Ultimat	ResExch	Avg.(13)
CoT	**0.000**	-0.234	-0.688	-0.406	-0.672	-0.223	-0.500	**0.450**	0.192	-0.382	-0.340	0.197	**1.000**	-0.123
SR	-0.156	-0.062	-0.570	-0.312	-0.844	-0.211	-0.359	0.357	0.181	-0.408	-0.143	0.167	**1.000**	-0.105
Debate	-0.086	**0.000**	-0.562	-0.391	-0.469	-0.258	**-0.188**	0.313	**0.199**	-0.446	-0.172	**0.258**	**1.000**	-0.062
ToM	**0.000**	-0.125	-0.578	-0.469	**-0.406**	-0.297	-0.719	0.200	0.151	-0.495	-0.010	-0.015	**1.000**	-0.136
MAFP	**0.000**	-0.125	**0.086**	-0.188	-0.531	**-0.195**	-0.391	0.157	0.185	**0.167**	0.144	0.130	**1.000**	**0.034**
w/o BoN	-0.125	-0.250	-0.219	-0.125	-0.453	-0.359	-0.391	0.180	0.182	-0.311	0.019	0.088	**1.000**	-0.059
w/o pool	-0.109	-0.250	-0.672	**-0.109**	-0.516	-0.363	-0.625	0.137	0.189	-0.203	**0.187**	0.237	0.812	-0.099

normalized by the largest payoff reachable for that seat in the match, so that utilities are comparable across scenarios (Appendix B.2). To ensure the reliability of the results, we play 16 matches with seat exchange for each pair, averaging the normalized payoff of each scenario. And the full pipeline is run with 8 random seeds and we report seed-averaged means. All strategies are generated by Qwen3.7-Flash (Yang et al., 2025) with reasoning, and every multi-round method starts from an initial strategy and runs $K{=}3$ iterations.

Metrics. We report the tournament score (TS) and the Stackelberg value (SV) defined in Section 3, and describe here only how they are estimated. TS is estimated with a round-robin schedule, in which each method plays pairwise matches against all the others, and $\mathcal{P}$ in Eq. 2 is the empirical distribution over the competing methods' strategies. SV is estimated with an adaptive opponent approximating the best response in Eq. 3. An opponent LLM evolves a counter-strategy over 6 rounds, each round playing simulated matches against the evaluated strategy π_1 and revising the counter-strategy based on the trajectories and payoffs of these matches, without ever seeing π_1's source code. Every frozen opponent iterate is scored on a disjoint batch, and we report π_1's utility at the iterate where the opponent's own payoff is highest, with ties broken against π_1.

5.2 RQ1: MAFP versus Existing Multi-Round Frameworks

To evaluate the effectiveness of MAFP, we compare it against representative multi-round baselines using the same LLM backbone, self-reflection (SR) (Shinn et al., 2023), debate (Liang et al., 2024), and theory-of-mind (ToM) with recursive level-k reasoning (Wilf et al., 2024; Zhang et al., 2025), together with the single-round CoT reasoning (Wei et al., 2022). Implementation details of the multi-round baselines can be found in Appendix B.3. We run MAFP with $n{=}3$ best-response candidates per iteration (Eq. 6).

To investigate the contribution of the best-of-n selection and the history pool, we additionally evaluate two MAFP ablations, shown as the indented rows in the tables. The ablation without best-of-n keeps a single candidate per iteration ($n{=}1$), and the ablation without the history pool best responds to a single program representing the opponent's current strategy rather than the pool of its past strategies. Across these 7 configurations, we calculate the TS and SV metrics, and per-scenario results are summarized in Tables 1 and 2.

From the results, MAFP outperforms all existing methods on average on both metrics. This validates our main claim that MAFP produces strategies that both prevail against the existing competitors and withstand the others' adaptation. Even without the best-of-n selection, MAFP still outperforms

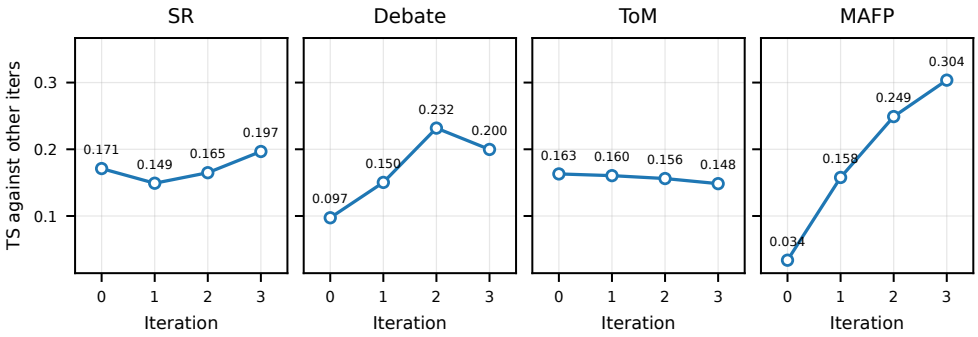


Figure 3: **Per-iteration comparison of strategies produced by each iterative method.** For each method, we run an internal tournament among the strategies after iterations 0 to 3, where iteration 0 is the initial draft, and report each iteration's tournament score against the other three.

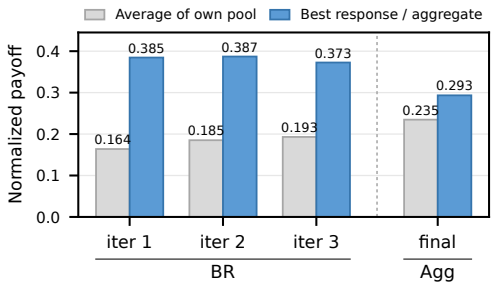


Figure 4: **Effectiveness of best response and aggregation.** Both outperform the average of the agent's own pool.

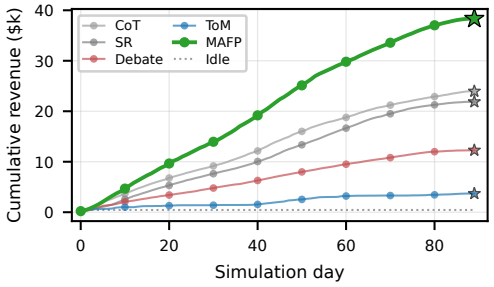


Figure 5: **Results on CoffeeBench.** MAFP achieves higher cumulative revenue than baseline methods over the 90-day period.

all existing methods on average on both metrics. Adding the selection through simulated matches (Eq. 6) further improves the strategy quality by a clear margin. In contrast, removing the history pool clearly degrades both metrics, as reacting to the opponent's latest strategy alone easily overfits to the opponent's current weaknesses, leaving it vulnerable to exploitation.

5.3 RQ2: Strategy Evolution Across Iterations

We study the strategy evolution from two views, an outcome view that compares the strategies each method delivers across iterations, and a mechanism view that checks the agent behaviors inside the MAFP loop.

Improvement across iterations. Figure 3 visualizes the per-iteration performance of multi-round methods, computed by an internal round-robin tournament among their per-iteration strategies. From the results, iteration by itself does not lead to improvement. SR and Debate fail to improve consistently, and ToM declines slightly from the initial 0.163 to 0.148. In contrast, MAFP improves strategies at every iteration, which further validates the effectiveness of our fictitious play method.

Effectiveness of best response and aggregation. The effectiveness of MAFP relies on two agent behaviors, the best response at every iteration and the aggregation after the final iteration. To understand how MAFP works, we check the two behaviors in the MAFP loop. Figure 4 compares each strategy generated by best response and aggregation with the average of the agent's own pool, studying whether they achieve above-average performance. At every iteration, the best response outperforms the average of the agent's own pool by a large margin, with improvements of +0.221, +0.202, and +0.180, confirming that best responding to a fixed pool is a prediction task the LLM handles reliably. Meanwhile, the final aggregated strategy outperforms the average of the full pool (+0.059), showing that aggregation effectively combines the strengths accumulated across iterations. With both conditions holding, the pool quality rises round by round, as formalized in Theorem 3 of Appendix A, which explains the steady gains of MAFP.

Table 3: **Results on multi-player games.** Each method's mean normalized payoff against the rest of the field on Avalon, Doudizhu, and three-player Texas Hold'em, averaged over 8 seeds, with every multi-round method run for $K=3$ or $K=7$ iterations. Best per column in bold.

	$K=3$				$K=7$			
Method	Avalon	Doudizhu	Hold'em 3p	Avg.(3)	Avalon	Doudizhu	Hold'em 3p	Avg.(3)
CoT	0.005	-0.063	-0.007	-0.021	-0.022	-0.100	-0.001	-0.041
SR	**0.080**	-0.096	-0.002	-0.006	-0.016	-0.095	-0.011	-0.041
Debate	-0.006	0.090	0.004	0.029	0.049	0.055	-0.005	0.033
ToM	-0.146	-0.053	-0.018	-0.072	-0.088	-0.076	-0.012	-0.058
MAFP	0.067	**0.122**	**0.023**	**0.071**	**0.077**	**0.216**	**0.029**	**0.107**

5.4 RQ3: Multi-Player Games and Realistic Tasks

To answer RQ3, we extend MAFP beyond two-player settings along two directions. For multi-player games, we evaluate on Avalon, a five-player hidden-identity team game (Light et al., 2023), Doudizhu, a three-player card game in which one landlord plays against two cooperating peasants (Zha et al., 2019), and three-player Texas Hold'em (Zha et al., 2019). For evaluation, each method in turn supplies the strategy it generates for each seat, while the remaining seats are filled with strategies drawn at random from the other methods, and we average the resulting normalized payoffs as the method's tournament score. From Table 3, MAFP attains the highest average payoff at $K=3$, ranking the best on Doudizhu and Hold'em and the second best on Avalon. Extending the iterations to $K=7$ widens this advantage, as MAFP rises from 0.071 to 0.107 and attains the best score in all three games. In contrast, the other methods struggle on these decision-making scenarios involving more agents and gain little from the extended iterations, as Debate improves merely from 0.029 to 0.033, SR even declines from -0.006 to -0.041, and ToM stays the lowest among all methods at both budgets. This indicates that MAFP scales with the number of agents, as the update of each LLM call stays a first-order best response to observed strategies.

We further evaluate MAFP on CoffeeBench (Sugiura et al., 2026), a 90-day supply-chain economy with three roles, farmer, roaster, and retailer. Each role is played by two competing agents, and the six agents produce, negotiate, and trade along the supply chain. Each method generates one code-based decision tree per role to control one of the two competing agents, while the other agent of each role runs the CoT produced strategy as baseline. Figure 5 visualizes the cumulative revenue of the method's three agents over the 90 days, averaged over 8 seeds. From the results, MAFP shows a clear advantage in this realistic scenario, reaching about \$38k cumulative revenue versus \$24k for CoT, while none of the other methods surpasses the CoT baseline. This further validates MAFP's ability to handle complex multi-agent decision-making tasks and its potential for real-world applications.

6 Conclusion

In this work, we study strategy design for a controlled agent in mixed-motive environments, where the other agents pursue their own payoffs and the strategies are mutually dependent. This challenge lies beyond the execution complexity that existing MAS are designed for. To tackle it, we propose MAFP, a multi-agent framework that instantiates every agent with an LLM and evolves a good strategy for our agent through fictitious-play co-evolution in simulation. On 13 scenarios spanning competitive games and negotiation, MAFP attains the best average tournament score and Stackelberg value among all evaluated baselines, and its advantage extends to multi-player games and a realistic supply-chain economy. More broadly, MAFP explores a different way of using MAS: with mixed-motive agents, the system serves not as a division of labor but as a simulator, conducting simulated play between agents so that the strategy emerges from their interaction. We hope this perspective inspires MAS designs beyond the fully cooperative paradigm.

Limitations. In the current implementation, MAFP restricts each decision to a code-based decision tree. This restriction supports accurate evaluation, but the expressiveness of code limits the range of tasks MAFP applies to. For example, CoffeeBench's free-text negotiation channel is not utilized in our experiments. Our next step is to extend MAFP to more expressive decision representations, and to apply it to richer commercial settings (Backlund & Petersson, 2025; Yun et al., 2025), which involve more agents and more intricate strategic structure.

AI USE STATEMENT

We used generative AI tools to aid and polish the writing, where the tools suggested improved framing, alongside grammar and phrasing edits on author-written text. We also used them for retrieval and discovery, where the tools helped search for candidate related work, and we read and verified every paper before citing it. We did not use generative AI tools for research ideation or execution, for generating synthetic datasets, or for proving mathematical claims. We have reviewed all AI-assisted output. 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 uses publicly available datasets and open-weight models for all agents and does not collect private or sensitive personal information. Our approach shares the potential risks of LLMs and LLM-based agent systems while introducing no additional risk, and its use should follow the existing risk guidance for these models and systems.

REPRODUCIBILITY STATEMENT

We build all scenarios on open-source environments, with the two-player scenarios from GT-Bench (Duan et al., 2024) and Negotiation Arena (Bianchi et al., 2024), the multi-player games from AvalonBench (Light et al., 2023) and RLCard (Zha et al., 2019), and the economic simulation from CoffeeBench (Sugiura et al., 2026). Our method is described in Section 4 and Algorithm 1, and the assumptions and complete proof of Theorem 3 are given in Appendix A. The experimental configurations, including the LLM backbone, the numbers of iterations and candidates, and the evaluation protocol, are reported in Sections 5.1 and 5.2. The scenario rules, payoff normalization, and baseline implementations are detailed in Appendix B, and the prompt templates of MAFP are presented in Appendix C. The source code will be publicly released upon acceptance.

REFERENCES

Elif Akata, Lion Schulz, Julian Coda-Forno, Seong Joon Oh, Matthias Bethge, and Eric Schulz. Playing repeated games with large language models. *Nature Human Behaviour*, 9(7):1380–1390, 2025.

Axel Backlund and Lukas Petersson. Vending-bench: A benchmark for long-term coherence of autonomous agents. *arXiv preprint arXiv:2502.15840*, 2025.

Suma Bailis, Jane Friedhoff, and Feiyang Chen. Werewolf arena: A case study in llm evaluation via social deduction. *arXiv preprint arXiv:2407.13943*, 2024.

Federico Bianchi, Patrick John Chia, Mert Yuksekgonul, Jacopo Tagliabue, Dan Jurafsky, and James Zou. How well can llms negotiate? negotiationarena platform and analysis. In *International Conference on Machine Learning*, pp. 3935–3951. PMLR, 2024.

George W Brown. Iterative solution of games by fictitious play. *Act. Anal. Prod Allocation*, 13(1): 374, 1951.

Jiangjie Chen, Siyu Yuan, Rong Ye, Bodhisattwa Prasad Majumder, and Kyle Richardson. Put your money where your mouth is: Evaluating strategic planning and execution of llm agents in an auction arena. *arXiv preprint arXiv:2310.05746*, 2023.

Junzhe Chen, Xuming Hu, Shuodi Liu, Shiyu Huang, Wei-Wei Tu, Zhaofeng He, and Lijie Wen. Llmarena: Assessing capabilities of large language models in dynamic multi-agent environments. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13055–13077, 2024a.

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 *International Conference on Learning Representations*, volume 2024, pp. 20094–20136, 2024b.

Vincent Conitzer and Tuomas Sandholm. Computing the optimal strategy to commit to. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, pp. 82–90, 2006.

Anthony Costarelli, Mat Allen, Roman Hauksson, Grace Sodunke, Suhas Hariharan, Carlson Cheng, Wenjie Li, Joshua Clymer, and Arjun Yadav. Gamebench: Evaluating strategic reasoning abilities of llm agents. *arXiv preprint arXiv:2406.06613*, 2024.

Logan Cross, Violet Xiang, Agam Bhatia, Daniel LK Yamins, and Nick Haber. Hypothetical minds: Scaffolding theory of mind for multi-agent tasks with large language models. *arXiv preprint arXiv:2407.07086*, 2024.

Allan Dafoe, Edward Hughes, Yoram Bachrach, Tantum Collins, Kevin R. McKee, Joel Z. Leibo, Kate Larson, and Thore Graepel. Open problems in cooperative AI. *arXiv preprint arXiv:2012.08630*, 2020.

Xiang Deng, Jeff Da, Edwin Pan, Yannis Yiming He, Charles Ide, Kanak Garg, Niklas Lauffer, Andrew Park, Nitin Pasari, Chetan Rane, et al. Swe-bench pro: Can ai agents solve long-horizon software engineering tasks? *arXiv preprint arXiv:2509.16941*, 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.

Jinhao Duan, Renming Zhang, James Diffenderfer, Bhavya Kailkhura, Lichao Sun, Elias Stengel-Eskin, Mohit Bansal, Tianlong Chen, and Kaidi Xu. Gtbench: Uncovering the strategic reasoning capabilities of llms via game-theoretic evaluations. *Advances in Neural Information Processing Systems*, 37:28219–28253, 2024.

Meta Fundamental AI Research Diplomacy Team (FAIR)†, Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, et al. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022.

Yoav Freund and Robert E Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, 1999.

Chris Frith and Uta Frith. Theory of mind. *Current biology*, 15(17):R644–R645, 2005.

Zhouhong Gu, Xiaoxuan Zhu, Yin Cai, Hao Shen, Xingzhou Chen, Qingyi Wang, Jialin Li, Xiaoran Shi, Haoran Guo, Wenxuan Huang, et al. Agentgroupchat-v2: Divide-and-conquer is what llm-based multi-agent system need. *arXiv preprint arXiv:2506.15451*, 2025.

Jiaxian Guo, Bo Yang, Paul Yoo, Bill Yuchen Lin, Yusuke Iwasawa, and Yutaka Matsuo. Suspicion-agent: Playing imperfect information games with theory of mind aware gpt-4. *arXiv preprint arXiv:2309.17277*, 2023.

W. Daniel Hillis. Co-evolving parasites improve simulated evolution as an optimization procedure. *Physica D: Nonlinear Phenomena*, 42(1-3):228–234, 1990. doi: 10.1016/0167-2789(90)90076-2.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Steven Yau, Zijuan Lin, Liyang Zhou, et al. Metagpt: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations*, volume 2024, pp. 23247–23275, 2024.

Hyunwoo Kim, Melanie Sclar, Xuhui Zhou, Ronan Bras, Gunhee Kim, Yejin Choi, and Maarten Sap. Fantom: A benchmark for stress-testing machine theory of mind in interactions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 14397–14413, 2023.

Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

Bin Lei, Yi Zhang, Shan Zuo, Ali Payani, and Caiwen Ding. Macm: Utilizing a multi-agent system for condition mining in solving complex mathematical problems. *Advances in Neural Information Processing Systems*, 37:53418–53437, 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.

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.

Jonathan Light, Min Cai, Sheng Shen, and Ziniu Hu. Avalonbench: Evaluating llms playing the game of avalon. *arXiv preprint arXiv:2310.05036*, 2023.

Xiaohao Liu, Xiaobo Xia, Weixiang Zhao, Manyi Zhang, Xianzhi Yu, Xiu Su, Shuo Yang, See-Kiong Ng, and Tat-Seng Chua. L-mtp: Leap multi-token prediction beyond adjacent context for large language models. *Advances in Neural Information Processing Systems*, 38:102569–102600, 2026.

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*, 2024.

Manus. Introducing wide research, 2025. URL `https://manus.im/blog/introducing-wide-research`.

R Thomas McCoy, Shunyu Yao, Dan Friedman, Mathew D Hardy, and Thomas L Griffiths. Embers of autoregression show how large language models are shaped by the problem they are trained to solve. *Proceedings of the National Academy of Sciences*, 121(41):e2322420121, 2024a.

R Thomas McCoy, Shunyu Yao, Dan Friedman, Mathew D Hardy, and Thomas L Griffiths. When a language model is optimized for reasoning, does it still show embers of autoregression? an analysis of openai o1. *arXiv preprint arXiv:2410.01792*, 2024b.

Mike A. Merrill, Alexander G Shaw, Nicholas Carlini, Boxuan Li, Harsh Raj, Ivan Bercovich, Lin Shi, Jeong Hyeob Shin, Thomas Walshe, Estefany Kelly Buchanan, Junhong Shen, Guanghao Ye, Hao Lin, Jason Poulos, Maoyu Wang, Marianna Nezhurina, Jenia Jitsev, Di Lu, Orfeas Menis Mastromichalakis, Zhiwei Xu, Zizhao Chen, Yue Liu, Robert Zhang, Leon Liangyu Chen, Anurag Kashyap, Jan-Lucas Uslu, Jeffrey Li, Jianbo Wu, Minghao Yan, Song Bian, Vedang Sharma, Ke Sun, Steven Dillmann, Akshay Anand, Andrew Lanpouthakoun, Bardia Koopah, Changran Hu, Etash Kumar Guha, Gabriel H. S. Dreiman, Jiacheng Zhu, Karl Krauth, Li Zhong, Niklas Muennighoff, Robert K. Amanfu, Shangyin Tan, Shreyas Pimpalgaonkar, Tushar Aggarwal, Xia Lin, Xin Lan, Xuandong Zhao, Yiqing Liang, Yuanli Wang, Zilong Wang, Changzhi Zhou, David Heineman, Hange Liu, H. Trivedi, John Yang, Junhong Lin, Manish Shetty, Michael Yang, Nabil Omi, Negin Raoof, Shanda Li, Terry Yue Zhuo, Wu Lin, Yiwei Dai, Yuxin Wang, Wenhao Chai, Shang Zhou, Dariush Wahdany, Ziyu She, Jiaming Hu, Zhikang Dong, Yuxuan Zhu, Sasha Cui, Ahson Saiyed, Arinbjörn Kolbeinsson, Jesse Hu, Christopher Rytting, Ryan Marten, Yixin Wang, Alexandros G. Dimakis, Andy Konwinski, and Ludwig Schmidt. Terminal-bench: Benchmarking agents on hard, realistic tasks in command line interfaces. *arXiv preprint arXiv:2601.11868*, 2026.

Dov Monderer and Lloyd S Shapley. Fictitious play property for games with identical interests. *Journal of economic theory*, 68(1):258–265, 1996.

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.

Juan-Pablo Rivera, Gabriel Mukobi, Anka Reuel, Max Lamparth, Chandler Smith, and Jacquelyn Schneider. Escalation risks from language models in military and diplomatic decision-making. In *Proceedings of the 2024 ACM conference on fairness, accountability, and transparency*, pp. 836–898, 2024.

Julia Robinson. An iterative method of solving a game. *Annals of mathematics*, 54(2):296–301, 1951.

Christopher D. Rosin and Richard K. Belew. New methods for competitive coevolution. *Evolutionary Computation*, 5(1):1–29, 1997. doi: 10.1162/evco.1997.5.1.1.

Thomas C. Schelling. *The Strategy of Conflict*. Harvard University Press, Cambridge, MA, 1960.

Lloyd S. Shapley. Some topics in two-person games. In Melvin Dresher, Lloyd S. Shapley, and Albert W. Tucker (eds.), *Advances in Game Theory*, volume 52 of *Annals of Mathematics Studies*, pp. 1–28. Princeton University Press, 1964.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Yoav Shoham and Kevin Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Issa Sugiura, Daichi Hattori, Kazuo Araragi, Keita Ogawa, Shota Onose, Taro Makino, Teppei Usuki, and Takashi Ishida. Coffeebench: Benchmarking long-horizon llm agents in heterogeneous multi-agent economies. *arXiv preprint arXiv:2606.16613*, 2026.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Y Charles, HS Che, Cheng Chen, Guanduo Chen, et al. Kimi k2.5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 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.

Heinrich von Stackelberg. *Marktform und Gleichgewicht*. Springer, Vienna, 1934.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Y Zou. Mixture-of-agents enhances large language model capabilities. In *International Conference on Learning Representations*, volume 2025, pp. 33944–33963, 2025.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024a.

Shenzhi Wang, Chang Liu, Zilong Zheng, Siyuan Qi, Shuo Chen, Qisen Yang, Andrew Zhao, Chaofei Wang, Shiji Song, and Gao Huang. Boosting llm agents with recursive contemplation for effective deception handling. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 9909–9953, 2024b.

Zihao Wang and Junming Zhang. From bits to boardrooms: A cutting-edge multi-agent llm framework for business excellence. *arXiv preprint arXiv:2508.15447*, 2025.

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

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeff Han, Isa Fulford, Hyung Won Chung, Alexandre Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv: 2504.12516*, 2025.

Alex Wilf, Sihyun Lee, Paul Pu Liang, and Louis-Philippe Morency. Think twice: Perspective-taking improves large language models' theory-of-mind capabilities. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8292–8308, 2024.

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.

Yufan Wu, Yinghui He, Yilin Jia, Rada Mihalcea, Yulong Chen, and Naihao Deng. Hi-tom: A benchmark for evaluating higher-order theory of mind reasoning in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10691–10706, 2023.

Chen Xu, Yicheng Hu, Ruizi Wang, Xinyu Lin, Wenjie Wang, Dongrui Liu, and Fuli Feng. Tacomas: Test-time co-evolution of topology and capability in llm-based multi-agent systems. *arXiv preprint arXiv:2605.09539*, 2026.

Renjun Xu and Jingwen Peng. A comprehensive survey of deep research: Systems, methodologies, and applications. *arXiv preprint arXiv:2506.12594*, 2025.

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

Yingxuan Yang, Huacan Chai, Shuai Shao, Yuanyi Song, Siyuan Qi, Renting Rui, and Weinan Zhang. Agentnet: Decentralized evolutionary coordination for llm-based multi-agent systems. *Advances in Neural Information Processing Systems*, 38:107309–107336, 2026.

Junhyeog Yun, Hyoun Jun Lee, and Insu Jeon. Quantevolve: Automating quantitative strategy discovery through multi-agent evolutionary framework. *arXiv preprint arXiv:2510.18569*, 2025.

Daochen Zha, Kwei-Herng Lai, Yuanpu Cao, Songyi Huang, Ruzhe Wei, Junyu Guo, and Xia Hu. RLCard: A toolkit for reinforcement learning in card games. *arXiv preprint arXiv:1910.04376*, 2019.

Linghua Zhang, Jun Wang, Jingtong Wu, and Zhisong Zhang. Retailbench: Evaluating long-horizon autonomous decision-making and strategy stability of llm agents in realistic retail environments. *arXiv preprint arXiv:2603.16453*, 2026.

Yadong Zhang, Shaoguang Mao, Tao Ge, Xun Wang, Adrian de Wynter, Yan Xia, Wenshan Wu, Ting Song, Man Lan, and Furu Wei. Llm as a mastermind: A survey of strategic reasoning with large language models. *arXiv preprint arXiv:2404.01230*, 2024a.

Yadong Zhang, Shaoguang Mao, Tao Ge, Xun Wang, Yan Xia, Man Lan, and Furu Wei. K-level reasoning: Establishing higher order beliefs in large language models for strategic reasoning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7212–7234, 2025.

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

Weike Zhao, Chaoyi Wu, Yanjie Fan, Pengcheng Qiu, Xiaoman Zhang, Yuze Sun, Xiao Zhou, Shuju Zhang, Yu Peng, Yanfeng Wang, et al. An agentic system for rare disease diagnosis with traceable reasoning. *Nature*, 651(8106):775–784, 2026.

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

Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. *Advances in neural information processing systems*, 20, 2007.

A Further Analysis

To further understand how MAFP improves strategies over the rounds, we analyze its per-round dynamics under two mild assumptions on the agents' best response and aggregation. Throughout this section, we use the utility U_i of Eq. 2, extended linearly when its first argument is a mixture over agent i's own strategies.

A.1 Improvement

The key observation is that each update in MAFP is a plain prediction task. The agent best responds to a fixed and fully observed pool without unfolding any belief hierarchy, so the LLM rarely returns a strategy worse than what its current pool already achieves, and the iterations are thus likely to improve performance. We formalize this with two assumptions.

Assumption 1 (Best response improvement)**.** $U_i\big(\pi_i^{t+1}, \mu_{-i}^t\big) \;\geq\; U_i\big(\mu_i^t, \mu_{-i}^t\big) + \gamma_t$ with $\gamma_t \geq 0$, where μ_i^t and μ_{-i}^t are the uniform pool mixtures defined in Section 4.

Assumption 2 (Aggregation no worse than average)**.** Against the current opponent pool, $U_i\big(\bar{\pi}_i^{t+1}, \mu_{-i}^t\big) \;\geq\; \frac{1}{|\mathcal{H}_i^{t+1}|}\sum_{\pi\in\mathcal{H}_i^{t+1}} U_i\big(\pi, \mu_{-i}^t\big) - \delta$ with $\delta \geq 0$, where $\bar{\pi}_i^{t+1}$ denotes the strategy produced by $\mathrm{Agg}_M(\mathcal{D},\, \omega_i,\, \mathcal{H}_i^{t+1})$.

Remark. As a heuristic method, the expected reward of aggregation Agg_M should not be worse than random selection.

Both assumptions are verified empirically in the MAFP loop (Fig. 4 in Section 5.3), where the best-response margin of Assumption 1 stays large at every iteration and the final aggregation margin of Assumption 2 is positive.

Theorem 3 (Per-round improvement)**.** *Under Assumptions 1–2, with* $\alpha_t = \frac{1}{t+2}$, $U_i(\mu_i^{t+1}, \mu_{-i}^t) \;\geq\; U_i(\mu_i^t, \mu_{-i}^t) + \alpha_t\gamma_t$, *and* $U_i(\bar{\pi}_i^{t+1}, \mu_{-i}^t) \;\geq\; U_i(\mu_i^t, \mu_{-i}^t) + \alpha_t\gamma_t - \delta$, *where* $\bar{\pi}_i^{t+1}$ *is the aggregated strategy of Assumption 2.*

Remark. MAFP converts the mutual anticipation of decision making into the iterative improvement of strategies. Theorem 3 reveals that the quality of strategies can be improved over iterations against the opponents' current pools.

Proof of Theorem 3. Since $\mathcal{H}_i^{t+1} = \mathcal{H}_i^t \cup \{\pi_i^{t+1}\}$, the uniform mixtures satisfy $\mu_i^{t+1} = (1-\alpha_t)\,\mu_i^t + \alpha_t\,\pi_i^{t+1}$ with $\alpha_t = \frac{1}{t+2}$. As $U_i(\cdot\,, \mu_{-i}^t)$ is linear in player i's mixed strategy,

$$U_i\big(\mu_i^{t+1}, \mu_{-i}^t\big) = (1-\alpha_t)\,U_i\big(\mu_i^t, \mu_{-i}^t\big) + \alpha_t\,U_i\big(\pi_i^{t+1}, \mu_{-i}^t\big) \;\geq\; U_i\big(\mu_i^t, \mu_{-i}^t\big) + \alpha_t\,\gamma_t,$$

where the inequality applies Assumption 1. This proves the first claim. For the second, Assumption 2 and the linearity of $U_i(\cdot\,, \mu_{-i}^t)$ give

$$U_i\big(\bar{\pi}_i^{t+1}, \mu_{-i}^t\big) \;\geq\; \tfrac{1}{t+2}\textstyle\sum_{\pi\in\mathcal{H}_i^{t+1}} U_i\big(\pi, \mu_{-i}^t\big) - \delta \;=\; U_i\big(\mu_i^{t+1}, \mu_{-i}^t\big) - \delta,$$

and the first claim concludes. □

B Implementation Details

B.1 Scenarios

We evaluate MAFP across 13 two-player scenarios, comprising 10 strategic games from GTBench (Duan et al., 2024) and 3 negotiation settings from Negotiation Arena (Bianchi et al., 2024), together with three multi-player games and the CoffeeBench economy used in RQ3. These scenarios span a broad range of game-theoretic properties to comprehensively reflect the behavior of MAFP across different conditions. They include complete versus incomplete information, deterministic versus probabilistic dynamics, and zero-sum versus general-sum payoffs. We provide a per-scenario description of their rules and characteristics in Table 4. Notably, our task is formulated differently from GTBench and Negotiation Arena, which treat the system being tested as the policy, prompting it to select an action at each step. Instead, we test the capability to generate strategies: the tested system produces a strategy specifying how to act before the game begins.

B.2 PAYOFF DEFINITION

All reported utilities are *normalized own payoffs* (NP). For each match, agent i's raw payoff $R_i(\tau)$ is read from the environment's own scoring rule, such as the pot won in Kuhn Poker, the value minus bid in Blind Auction, or the value retained after a deal in the negotiation scenarios. We then divide by the largest payoff S_i that seat could have obtained in this match, which is positive in all scenarios,

$$\mathrm{NP}_i \;=\; R_i(\tau) \,/\, S_i. \tag{8}$$

B.3 BASELINES

SR (self-reflection). We follow Shinn et al. (2023) to implement verbal self-critique at the strategy level. Starting from the initial strategy, each seat runs K refinement iterations, at each of which it is shown only its own previous strategy and is asked to critique and refine it; no opponent strategy and no game trace is introduced.

Debate. We follow Liang et al. (2024) to implement multi-agent debate. Initially, two agents independently propose a strategy profile from rules. In each of the K subsequent iterations, each agent refines its profile after seeing the other agent's previous profile. After the final iteration, a judge LLM aggregates the two final profiles into a single consensus profile.

ToM (K-level reasoning). We follow Zhang et al. (2025) to adapt their recursive K-level reasoning framework from per-step action selection to upfront strategy generation. The recursion is structured as a call stack with K best-response levels above a rules-only base, matching the iteration budget K of the other multi-round methods:

$$\texttt{KReason}(k) \;=\; \begin{cases} \mathrm{LLM}(\text{rules}) & k = 0, \\ \mathrm{LLM}\big(\text{rules},\ \texttt{KReason}(k-1)\big) & k \geq 1. \end{cases} \tag{9}$$

At each level k, all seats' strategies are jointly generated in a single LLM call conditioned on the level-$(k{-}1)$ profile: each seat's level-k strategy is the best response to the other seats' level-$(k{-}1)$ strategies. The recursion bottoms out at $k{=}0$ with a rules-only naive profile, mirroring Algorithm 1 of Zhang et al. (2025) with the recursive unit replaced from a predicted action to a written strategy profile for our strategy generation tasks.

C PROMPT TEMPLATES

We present the prompt templates for the three LLM-based components of our pipeline in Figs. 6, 7, and 8: the initialization Init_M, the best response BR_M, and the aggregation Agg_M. Every user prompt ends with the shared policy-as-code contract shown in Fig. 6, which pins the program interface, the real observation schema, and the action format. In the prompts, a code-based decision tree is referred to as a policy program, and the symbol K denotes the number of programs in the pool shown to the model rather than the round number in Algorithm 1.

System prompt

System Role: You are a game strategy engineer. You write COMPLETE, self-contained Python policy programs for one seat of a game. Your objective is to maximize that seat's OWN expected payoff as the game's rules define it (the payoff line in the task states exactly what counts). Think through the strategy first, then emit the final program.

User prompt

Game: ⟨task⟩
You are writing the policy program for the ⟨seat⟩ player (seat ⟨k⟩ of 2).

Game rules:
⟨game rules⟩

POLICY-AS-CODE CONTRACT (strict):
- Define exactly one entry point: `def act(observation, rng):` returning the move as a string.
- The returned string must match the move format: ⟨format⟩. Anything that is not in `observation['legal_moves']` loses the match immediately, so when in doubt pick from `observation['legal_moves']` directly.
- `observation` is a plain dict. Example payload (the REAL schema at the initial state; later states fill the same fields):
⟨example observation JSON⟩
- `rng` is a seeded `random.Random`. Randomized (mixed) strategies are allowed and encouraged where strategically appropriate; draw via `rng` so play is reproducible.
- Standard library only (`math`, `random`, `itertools`, `collections`, `re`). No I/O, no network, no files, no mutation of module globals between calls. The function must return within 0.5 seconds.
- Output your COMPLETE policy as ONE fenced Python code block. `act` must be self-contained with its helpers in the same block.

Figure 6: Prompt template for the initialization Init_M, together with the shared policy-as-code contract that is appended to every generation call. $\langle\cdot\rangle$ marks runtime-filled slots.

Table 4: Scenarios Description.

Scenario	Description
	Strategic Games
TicTacToe	A two-player game on a 3×3 grid; players alternate marking squares, and the first to align three marks horizontally, vertically, or diagonally wins.
Connect Four	Players alternately drop tokens into a 6×7 vertically suspended grid; the first to form a line of four wins.
Breakthrough	An abstract strategy game played on a 3×8 board; pieces move one space straight or diagonally forward and may capture diagonally. The first player to reach the opponent's home row wins.
Nim	Players alternately remove matches from one of four piles (initial sizes $1, 3, 5, 7$); a player must remove at least one match from a single pile, and the player forced to take the last match loses.
Iterated Prisoner's Dilemma (IPD)	Two players repeatedly choose between `Silent` and `Testify`; per-round payoffs follow the classic Prisoner's Dilemma matrix and accumulate over rounds.
Pig	A turn-based dice game in which a player repeatedly rolls a single die, accumulating points until they choose to stop or roll a 1 (losing the turn's gains); the first to reach the target score wins.
Kuhn Poker	A two-player imperfect-information poker variant with a three-card deck (King, Queen, Jack); players alternately `Bet` or `Pass`, and the showdown awards the pot to the higher card.
Blind Auction	Players simultaneously submit sealed bids for an item with private valuations; the higher bidder wins and pays their bid.
Liar's Dice	A two-player game with private dice; players alternately bid increasing quantity-or-value combinations, or challenge the previous bid. The losing side of a challenge loses a die.
Negotiation	Two players divide a pool of three item types with private value vectors, alternating between proposal turns and utterance turns; payoff is the total value of items each player ultimately receives.
	Negotiations
BuySell	A buyer with a private valuation and a seller with a private cost negotiate the price of a single item over multiple rounds; per-player surplus is the gap between the agreed price and the respective private value, with rejection zeroing both players.
Ultimatum	A proposer offers a split of a fixed pot, and a responder either accepts (both receive the proposed shares) or rejects (both receive zero).
Resource Exchange	Two players hold different bundles of resources with asymmetric private values and negotiate trades over multiple rounds to maximize their own utility.
	Multi-Player Games and Real Task
Avalon	A five-player hidden-identity team game with four role groups (Merlin, Servant, Minion, Assassin); players propose teams, vote, and run quests, and the evil side may win by assassinating Merlin.
Doudizhu	A three-player climbing card game in which one landlord plays against two cooperating peasants; players shed cards in legal combinations and the side that empties a hand first wins.
Texas Hold'em (3p)	A three-player no-limit poker variant with private hole cards and shared community cards; players bet over four streets and the best hand or last remaining player takes the pot.
CoffeeBench	A 90-day supply-chain economy with three roles, farmer, roaster, and retailer, each played by two competing agents; the six agents produce, set prices, negotiate in free text, and trade to maximize net income (Sugiura et al., 2026).

System prompt
System Role: You are a game strategy engineer. The opponent's strategy for the next game is drawn UNIFORMLY AT RANDOM from the pool of policy programs shown below. Write a fresh COMPLETE program that maximizes YOUR OWN expected payoff, as the game's rules define it, under this uniform mixture: a strategy that earns well against every program in the pool, not one that exploits a single program. Favor general, state-grounded logic over brittle move-by-move reaction tables.

User prompt

Game: ⟨task⟩
You are writing the policy program for the ⟨seat⟩ player (seat ⟨k⟩ of 2).

Game rules:
⟨game rules⟩

The opponent's (⟨opponent seat⟩ player) strategy for the next game is drawn UNIFORMLY AT RANDOM from this pool of programs (K=⟨K⟩):
[Opponent program #1]
⟨program 1⟩

[Opponent program #2]
⟨program 2⟩

⋮

Write a fresh COMPLETE program for YOUR seat that maximizes your own expected payoff under this uniform mixture.

⟨contract⟩

Figure 7: Prompt template for the best response BR_M, which reads the opponent's entire strategy pool in context as an exact uniform mixture.

System prompt

System Role: You are a strategy summarizer for policies written as Python programs. Given K policy programs written by one player for the same seat of the same game, produce ONE synthesized program that best represents the uniform mixture over that pool. Capture the decision logic the programs share, preserve tactical branches that recur across programs, and break idiosyncratic ties neutrally. Do NOT cherry-pick a single program, and do not invent tactics that no program endorses.

User prompt

Game: ⟨task⟩
You are writing the policy program for the ⟨seat⟩ player (seat ⟨k⟩ of 2).

Game rules:
⟨game rules⟩

K=⟨K⟩ policy programs written by the player for this seat, listed in order; treat them as a uniform mixture:
[Program #1]
⟨program 1⟩

[Program #2]
⟨program 2⟩

⋮

Synthesize the single program that best represents the uniform mixture over this pool.

⟨contract⟩

Figure 8: Prompt template for the aggregation Agg_M, applied once per agent after the final round to consolidate its pool into one executable strategy.