Belief-Aligned Team Formation: Mitigating Cognitive ToM Pitfalls in LLM Multi-Agent Systems

Anonymous ACL submission

Abstract

Theory of Mind (ToM) - the ability to understand others' mental states - is a fundamental aspect of human cognition. Recent work has equipped Large Language Model (LLM) agents with ToM capabilities, yet our investigation reveals the pitfalls of ToM in multi-agent systems (MAS): agents with ToM abilities often exhibit worse cooperation performance. This highlights the challenge of effectively translating ToM abilities into cooperations in LLM-based MAS. To address this challenge, we propose a ToM-aware team formation mechanism that explicitly considers belief alignment when forming cooperative teams. Our approach introduces a stable team formation algorithm that optimizes team formation based on both ToM-derived belief alignment and specialized agent capabilities. Through extensive experiments, we demonstrate our mechanism improves the task-specific performance up to 26% compared to random team formation, while maintaining stable long-term collaboration.

1 Introduction

Cooperation is a fundamental aspect of multi-agent systems (MAS), enabling agents to work together as a team to achieve common goals and solve complex problems. In recent years, the rapid advancement of large language models (LLMs) has opened up new opportunities for building intelligent MAS. LLMs have demonstrated remarkable capabilities in natural language understanding, generation, and reasoning, such as GPT-family (Eloundou et al., 2023). By leveraging these powerful LLMs, MAS can enable agents to communicate and collaborate using natural language, resulting in more flexible and human-like interactions in cooperation tasks.

Recent research in LLM-based MAS has focused on enhancing communication and knowledge sharing among agents (Xu et al., 2023; Lan et al., 2023; Hua et al., 2023; Wu et al., 2023b; Nascimento et al., 2023; Fu et al., 2023), as well as advancing cognitive abilities such as reasoning and Theory of Mind (ToM) (Qi and Vul, 2020; Li et al., 2023b; Liang et al., 2023; Agashe et al., 2023; Lin et al., 2024). However, despite these advancements, *the interplay between agents' cognitive abilities (like ToM) and cooperation* remains unexamined.

ToM represents a fundamental cognitive capability in multi-agent interactions (Premack and Woodruff, 1978). It operates at different levels of complexity, from simple belief attribution to recursive reasoning "I think you believe I will...". Although cognitive depth appears beneficial for team cooperation, *its impact on cooperation is unclear*. Recent studies have shown varying effects of ToM levels on agent interactions (Wang et al., 2022; Li et al., 2023a; Chan et al., 2023), suggesting a more nuanced relationship between ToM abilities and cooperation.

ToM Capabilities Alone are Insufficient for Effective Cooperation. To better understand this relationship, we systematically investigated ToM's impact on cooperation. Our investigation reveals: *agents with ToM generally exhibit worse performance in cooperation tasks, especially MAS involving high-level ToM agents* (detailed in Section 3). This finding challenges the intuitive assumption that more sophisticated ToM abilities lead to better cooperation.

In fact, agents with ToM may overthink and anticipate potential conflicts, resulting in more cautious cooperation. This observation aligns with psychological research (Ridinger and McBride, 2017) indicating that ToM capabilities alone are insufficient for effective cooperation. In addition, Ridinger and McBride (2017) suggests that successful collaboration also requires positive reciprocation and alignment in cooperations. This finding motivates us to explore ToM-aware coordination mechanisms to enhance agent cooperation.

Mitigating ToM Pitfalls. We propose a novel team formation mechanism that explicitly consid-

ers belief alignment when forming cooperative teams. The belief alignment ensures that agents' beliefs about each other's intentions (derived from ToM) and actions are aligned, facilitating reciprocation and cooperation. Unlike prior ToM applications that focus on agent *capabilities* (Li et al., 2023b), our approach optimizes *cooperative teams* that can mitigate the ToM pitfalls of ToM agents while maintaining stable cooperation.

Contribution. We summarized our main contributions as follows:

1. **Interplay of ToM and Cooperation:** We explore the relationship between Theory of Mind (ToM) capabilities and cooperative performance in LLM-based multi-agent systems. Our findings indicate that ToM capabilities do not always lead to better cooperation, highlighting the complexity of this interplay.
2. **Team Formation Mechanism:** We propose a novel mechanism for ToM-aware team formation by forming teams based on ToM-driven belief alignment to encourage reciprocation and cooperation.
3. **Experimental Evaluation:** We comprehensively evaluate our methods, incorporating ToM capabilities to assess their effectiveness in improving cooperation, team stability, and task-specific performance.

2 Related Work

Agents with Theory of Mind (ToM).[1] ToM—the ability to attribute mental states to oneself and others—has been explored in cognitive science (Premack and Woodruff, 1978; Frith and Frith, 2003) and applied in multi-agent reinforcement learning for cooperation via supervised training (Wen et al., 2018; Wang et al., 2022; Oguntola et al., 2023). Recent studies have begun to explore ToM in LLM agents (Arodi and Cheung, 2021; Li et al., 2022; Zhou et al., 2023; Xu et al., 2024), but mainly focus on individual capabilities. While Li et al. (2023b) applied ToM to LLM-based cooperation, the link between ToM ability and cooperative performance remains unclear. We aim to bridge this gap by studying how varying levels of ToM influence cooperation in multi-agent systems (MAS), shifting focus from individual ToM capability to its role in team dynamics.

Multi-agent Cooperation. Traditional multi-agent cooperation has relied on game theory (Yoshida et al., 2008) and communication-based strategies (Foerster et al., 2016; Tang, 2019), enabling agents to coordinate through interaction. LLMs have further advanced cooperation by leveraging language understanding and reasoning (Zhang et al., 2024b; Talebirad and Nadiri, 2023; Zhang et al., 2024a). Frameworks like CAMEL (Li et al., 2023a), AutoGen (Wu et al., 2023a), and MetaGPT (Hong et al., 2024) use role-play, programmable conversations, or human-like workflows to structure collaboration. Other works focus on improving knowledge sharing (Xu et al., 2023; Lan et al., 2023; Fu et al., 2023) and cognitive modeling (Gong et al., 2023). Some approaches also explore dynamic team formation (Chen et al., 2024; Shi et al., 2023; Liu et al., 2024). In contrast, our work focuses on how ToM ability affects cooperation quality, particularly in adaptive team formation within MAS.

3 Motivation

Recent studies have explored the role of ToM in enhancing agent capabilities and coordination (Li et al., 2023b; Yim et al., 2024). While agents equipped with ToM capabilities have demonstrated improved ability to understand and predict others' actions, the successful translation of these cognitive abilities into effective cooperative actions remains unexamined. To better understand this relationship, we employ a standard ToM model aligned with current research (Zhou et al., 2023; Street, 2024; Xu et al., 2024), examining how ToM capabilities influence cooperation outcomes in multi-agent systems.

ToM Modeling. In practice, ToM typically does not exceed second-order reasoning due to cognitive limitations and diminishing returns (Premack and Woodruff, 1978; Frith and Frith, 2003). We consider two levels of ToM ability, aligning with existing literature in both human cognitive research (De Weerd et al., 2015) and agent cognitive modeling (Li et al., 2023b): Low ToM (Level 1), where agents can represent and reason about others' beliefs, desires, and intentions; and High ToM (Level 2), where agents can additionally consider others' ToM reasoning.

Empirical Analysis of ToM in Iterative Programming To investigate the relationship between ToM and the performance of MAS, we conducted simulations of the multi-agent system consisting of

[1] Due to page limit, we put a detailed analysis in Appendix B

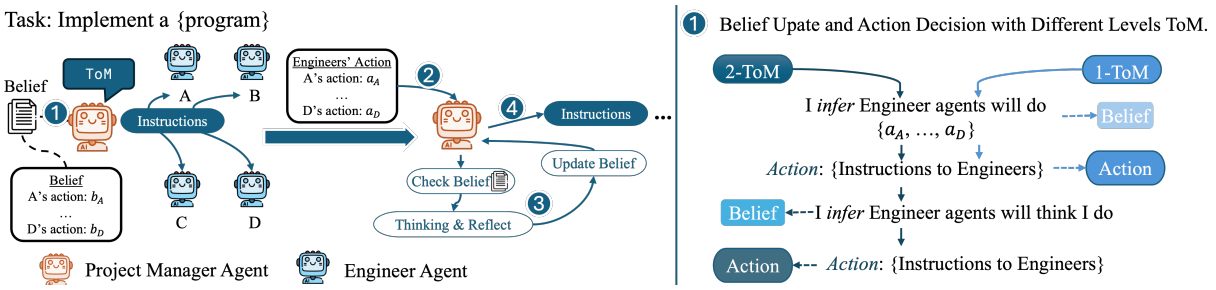

Figure 1: **Illustration for the MAS with ToM agents**. (*Left*) The MAS for the iterative programming tasks consists of one project manager (PM) with ToM ability and four Engineers. The cooperation involves: ❶ PM updates its beliefs and takes actions based on its ToM reasoning; ❷ PM observes Engineers' actions; ❸ PM think and reflect other's actions and update beliefs; ❹ PM provides instructions to Engineers. (*Right*) The ToM cognitive thinking process of PM agent involves recursive belief updates and decision-making based on the inferred beliefs. We also employ different ToM settings for the PM and engineers as detailed in Appendix H.3.

Table 1: **Comparison of performance between agents with lower-level (k=1) and higher-level (k=2) ToM** in the Iterative Programming task on two benchmarks, including HUMANEVAL (Chen et al., 2021) and MBPP (Austin et al., 2021), over $R = 1 \ldots 5$ interaction rounds: High ToM agents show higher initial performance but steeper performance decline over cooperation. Numbers in parentheses show percentage change from R=1 to R=5, with **bold** indicating high ToM agents with worse cooperation (severer performance degradation).

		gpt-4o-mini		deepseek-V3		Llama-3.1-70B-Ins.		Doubao-1.5-pro		Claude-3.5-sonnet	
	ToM^k	R=1	R=5	R=1	R=5	R=1	R=5	R=1	R=5	R=1	R=5
HUMANEVAL	k=1	87.00	88.00(+1.15%)	94.00	94.00(0%)	89.13	86.17(-3.32%)	94.00	92.26(-1.74%)	95.88	92.00(-4.05%)
	k=2	91.00	**90.00(-1.10%)**	96.00	**95.00(-1.04%)**	92.47	90.43(-2.21%)	95.23	**93.33(-1.90%)**	95.00	91.75(-3.42%)
MBPP	k=1	54.00	50.00(-7.41%)	70.00	63.00(-10.00%)	49.47	34.07(-31.13%)	62.57	55.43(-7.14%)	58.33	57.29(-1.78%)
	k=2	54.00	**49.00(-9.26%)**	67.00	62.00(-7.46%)	49.45	**32.63(-34.01%)**	66.67	**57.38(-9.29%)**	59.00	**52.08(-11.73%)**

one project manager (PM) and four engineer agents working together to solve programming tasks, as shown in Figure 1 (*Left*). Specifically, PM is enabled with ToM ability, which means he can recursively infer the actions of other agents (as beliefs) and take his own actions accordingly, as illustrated Figure 1 (*Right*). As shown in Table 1, *incorporating ToM can introduce side effects that hinder the performance of MAS.*

Specifically, higher-level ToM agents (k=2) often demonstrate worse cooperation performance than lower-level ToM agents (k=1), This suggests that ToM capabilities may lead agents to overthink and anticipate conflicts, resulting in cautious cooperation. In addition, we also investigate the side effects of incorporating higher ToM in some state-of-the-art methods (Liu et al., 2024; Chan et al., 2023) in Appendix H.

Organization. The rest of this paper is organized as follows: Section 4 introduces the formal ToM modeling and the team formation problem in multi-agent systems. Section 5 describes our proposed team formation algorithm to form cooperative teams among ToM agents. Section 6 presents the experiments evaluating our proposed mechanism for different cooperation tasks.

4 Formulation

In multi-agent cooperation scenarios, agents typically have defined roles and goals. We leverage the alignment between agents' beliefs and actions from ToM to establish teams for team formation. In this section, we develop the formal ToM modeling in Section 4.1 and the MAS team (team) formation problem in Section 4.2.

4.1 ToM Formulation for LLM Agents

Theory of Mind refers to the ability to attribute mental states, such as beliefs, intentions, and desires, to oneself and others, and to understand that others may have beliefs and intentions that differ from one's own (Premack and Woodruff, 1978). In the context of MAS, for agent i at round R, its k-level belief $b_i^{(k)}$ can be formulated as:

$$\mathcal{B}_i^{(k)} = f_{\text{ToM}}\Big(\underbrace{\mathcal{H}_i^{1:R}}_{\substack{\text{History: observations}\\ \text{\& own actions}}}, \underbrace{\mathcal{A}_{-i}^{1:R-1}}_{\substack{\text{Others' past}\\ \text{actions}}}, \underbrace{\{\widetilde{\mathcal{B}}_{i,m}^{(k-1)}\}_{m\neq i}}_{\substack{\text{Predicted beliefs of}\\ \text{others at depth } k-1}}\Big) \tag{1}$$

Where:

- $\mathcal{B}_i^{(k)}$ is agent i's k-level belief, representing its mental state and prediction about other agents' actions.
- $\widetilde{\mathcal{B}}_{i,m}^{(k-1)} = p(a_m^R \mid \mathcal{H}_m^{1:R}, \mathcal{A}_{-m}^{1:R-1}, \{\widetilde{\mathcal{B}}_{m,l}^{(k-2)}\}_{l\neq m})$ represents agent i's *inference* (not ground truth) about m's beliefs.

Table 2: Key Notation for ToM Formulation

Notation	Description
$\mathcal{B}_i^{(k)}$	Agent i's own k-level belief
$\widetilde{\mathcal{B}}_{i,m}^{(k)}$	i's prediction of m's k-level belief
$\mathcal{H}_i^{1:R}$	i's history: $\{\text{obs } o_i^1, \ldots, o_i^R\} \cup \{\text{own actions}\}$
$\mathcal{A}_{-i}^{1:R-1}$	Past actions of all agents except i
f_{ToM}	LLM-based belief generation function

- Other notations are defined in Table 2.

Base Case ($k = 0$). At zero-order ToM $\mathcal{B}_i^{(0)} = f_{\text{ToM}}(\mathcal{H}_i^{1:R}, \mathcal{A}_{-i}^{1:R-1})$, capturing naive behavior modeling without recursive reasoning.

For $k \geq 1$, beliefs nest recursively:

- $\mathcal{B}_i^{(1)}$: agent i predicts others' actions using $k = 0$ beliefs.
- $\mathcal{B}_i^{(2)}$: agent i predicts others' $k = 1$ beliefs, which themselves depend on others' predictions of $k = 0$ beliefs.

Chess Example with ToM $k = 2$. Agent A (White, $\mathcal{B}_A^{(2)}$) observes B's knight move $\mathcal{H}_A^{1:R}$ (board state/history), $\mathcal{A}_{-A}^{1:R-1}$ (B's past moves) and predicts $\widetilde{\mathcal{B}}_{A,B}^{(1)}$ (B's inferred belief at $k = 1$): "B expects my queen to retreat." Then, A uses f_{ToM} to derive $\mathcal{B}_A^{(2)}$: "B will chase my queen to g4."

Remarks: In LLM agents, $\mathcal{H}_i^{1:R}$, $\mathcal{A}_{-i}^{1:R-1}$, and $\mathcal{B}_i^{(k)}$ manifest as natural language interactions (Gur et al., 2023). The ToM hierarchy enables strategic deception that is unattainable with flat reasoning. We provide carefully designed structured prompts (f_{ToM}) to define each agent's role and level of ToM, guiding the generation of reasoning and facilitating each agent's belief updates (see Appendix E for detailed examples).

4.2 Team Formulation

This section introduces a team formation mechanism to address: *how to form the stable cooperative team for ToM agents in MAS*. To begin with, we define the belief-action alignment score used for establishing preference for team formation. Then, we define the stability condition for team formation, and finally, we introduce the tolerance parameter ϵ to reduce the search space by filtering out poorly aligned agent pairs.

Team Formation Framework. Let $N = \{1, \ldots, n\}$ be the set of agents. A team $\mathcal{T} \subseteq N$ satisfies: (a) *Size Constraint*: $|\mathcal{T}| \geq \eta$, where $\eta = \lceil n/2 \rceil$ as default (minimal team size). (b) *Preference Consistency*: All agents in $\mathcal{T}$ mutually prefer $\mathcal{T}$ over smaller or alternative teams (no blocking team).

Optimal Team Selection. The team formation process operates through three components: *team preference*, *stability conditions*, and *welfare optimization*.

We begin by formalizing how agents evaluate team preference through belief-action alignment metrics.

• **Team Preference.** Each agent i ranks teams $\mathcal{T} \subseteq N$ via its *belief-action alignment score*:

$$B_i^{(k)}(\mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{j \in \mathcal{T}} \phi\Big(\underbrace{\widetilde{\mathcal{B}}_{i,j}^{(k)}}_{\substack{i\text{'s prediction of} \\ j\text{'s } k\text{-level belief}}}, \underbrace{\hat{a}_j}_{\substack{j\text{'s actual} \\ \text{action}}}\Big) \quad (2)$$

where $\phi(\cdot)$ measures belief-action alignment. Consider two teams $\mathcal{T}_1$ and $\mathcal{T}_2$, if $B_i^{(k)}(\mathcal{T}_1) > B_i^{(k)}(\mathcal{T}_2)$, then agent i prefers (strictly) $\mathcal{T}_1$ over $\mathcal{T}_2$.

For notation simplicity, we use B_i to represent $B_i^{(k)}$ when the context of k-level ToM is clear, where $B_i(\mathcal{T})$ captures agent i's evaluation of team $\mathcal{T}$ based on its k-level belief predictions about other team members' behaviors.

Remarks: $\phi(\cdot)$ measures the alignment between beliefs and actions, outputting scores in the range [-1, 1], where 1 indicates perfect alignment, 0 represents neutral/no alignment, and -1 indicates complete misalignment.

To calculate this alignment score, we employ a self-evaluation approach involving prompting the agent to evaluate the alignment between its belief and another agent's action, which is consistent with existing LLM agent literature (Qin et al., 2023; Zheng et al., 2023; Liu et al., 2024). We also discuss belief-alignment calculation for non-LLM agents in Appendix C.

• **Stability Condition.** A team $\mathcal{T}$ is stable if:

$$\nexists\, C \subset \mathcal{T}: \begin{cases} |C| \geq \eta \\ B_i(C) > B_i(\mathcal{T}) \end{cases} \quad \forall i \in \mathcal{T} \quad (3)$$

The stability condition ensures no subgroup can defect to form a better-aligned team, enforcing mutually beneficial teammates. This adapts core concepts from cooperative game theory to ToM agents, where preferences are defined based on the alignment between beliefs and actions.

• **Social Welfare.** The *social welfare* of team $\mathcal{T}$ aggregates team preferences $B_i(\mathcal{T})$ defined in Eq.2:

$$W(\mathcal{T}) = \sum_{i \in \mathcal{T}} B_i(\mathcal{T}) \quad (4)$$

The optimal team $\mathcal{T}^*$ maximizes social welfare

while satisfying stability:

$$\mathcal{T}^* = \text{argmax}_{\mathcal{T} \subseteq N} W(\mathcal{T}) \quad (5)$$

subject to: $|\mathcal{T}| \geq \eta$ and no blocking team $C \subset \mathcal{T}$.

To enable tractable team formation, we introduce a tolerance parameter ϵ that filters agent pairs with insufficient belief-action alignment: $\phi\left(\widetilde{\mathcal{B}}_{i,j}^{(k)}, \hat{a}_j\right) \geq \epsilon$, which *reduces the search space* by eliminating poorly aligned teammates. When no pairs satisfy ϵ (universal misalignment), agents still maintain coherent preferences through $B_i(\mathcal{T})$ in Eq.2, enabling fallback team formation based on maximizing social welfare.

5 ToM-Aware Team Formation

Our mechanism forms belief-aligned teams through welfare optimization with stability constraints. The cooperative team persists until critical belief-action misalignments trigger reformation, balancing the need for stable collaboration with dynamic mental model updates.

5.1 Team Formation Mechanism

The proposed mechanism establishes persistent team cooperation through a process of belief-aligned team formation and adaptive reformation.

Agents initially form team $\mathcal{T} = \mathcal{N}$. During task execution, agents iteratively refine their beliefs $\mathcal{B}_i^{(k)}$ through the ToM function f_{ToM} (Line 4), which allows them to dynamically update predictions about teammates' behaviors and intentions. This belief updating process reflects the learning and sharing team knowledge.

In addition, the mechanism implements real-time monitoring of belief-action alignment (Line 6). When alignment falls below threshold $\phi\left(\widetilde{\mathcal{B}}_{i,j}^{(k)}, \hat{a}_j\right) < \epsilon$, a misalignment counter c tracks potential team instability. When the proportion of misaligned pairs exceeds $\theta \cdot |\mathcal{T}|$ (default $\theta = 0.3$) for τ consecutive rounds (Line 1), the reformation initiates re-optimization of $\mathcal{T}$ by maximizing social welfare $W(\mathcal{T})$ (Eq. 5) using updated belief states. The τ-round ($\tau = 1$ experimentally) prevents over-reaction to temporary fluctuations.

5.2 Adaptation for Specialized Agent Abilities

While belief alignment drives stable cooperation, many tasks require *domain-specific expertise*. We extend the team formation mechanism with skill relevance.

Specialization-Aware Preference Model. Let $\alpha_i \in [0, 1]$ denote agent i's specialization score for specific tasks (e.g., code optimization, medical diagnosis) and higher values of α_i indicate greater specialized ability for the task.

We reformulate the team preference $B_i(\mathcal{T})$, defined in Eq.2, as $B_i'(\mathcal{T})$:

$$B_i'(\mathcal{T}) = B_i(\mathcal{T}) + \lambda \cdot \frac{1}{|\mathcal{T}|} \sum_{j \in \mathcal{T}} \alpha_j \,,$$

where λ balances alignment vs. expertise (default $\lambda = 1$).

By incorporating the specialized ability scores α_i into the team preference, our team formation algorithm will prioritize agents with higher specialized skills for specific tasks and consider the belief alignment. This adaptation ensures that agents with crucial specialized abilities are more likely to be included in the formed cooperative team.

Remarks: This formulation maintains transitivity in team preferences while balancing belief alignment ($B_i(\mathcal{T})$) with task-specific capabilities ($\{\alpha_i\}_{i \in \mathcal{T}}$). The teams now form based on both belief-action alignment and specialized abilities. In practice, agents can compute α_i via self-report or skill tests. We conduct case study incorporating specialized ability scores (detailed in Appendix E.1) to enhance team performance for tasks requiring specific technical expertise.

6 Experiment

Experiment Setup.

The experiment setup involves extending the MetaGPT framework (Hong et al., 2024) to incorporate the proposed multi-agent LLM cooperation mechanism. We vary different ToM configurations for MAS, including 1-level, and 2-level ToM for different LLM agents. We evaluate our proposed mechanism on the following cooperative tasks:

- *Iterative Programming:* In this task, agents simulate a software development team, with each agent assuming different roles. We use HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021) as benchmarks to evaluate the performance of ToM agents in this multi-agent environment.
- *Debate:* In a debate setting, agents are divided into two sides (affirmative and negative) and engage in a multi-round debate on a given topic. The agents collaborate within their respective sides to present arguments and counterarguments.

Table 3: **Comparison of cooperation performance** (Pass@1) between agents with lower-level (1-level) and high-order (2-level) ToM in the Iterative Programming task (MBPP) over five interaction rounds: Random team formation (baseline), both ToM levels show comparable performance; Our proposed team formation improves the cooperative task performance.

Setting	deepseek-V3		llama-3.1-70b-Ins.		claude-3.5-sonnet		gpt-4o-mini		Doubao-1.5-pro	
	1-ToM	2-ToM	1-ToM	2-ToM	1-ToM	2-ToM	1-ToM	2-ToM	1-ToM	2-ToM
Beaseline	63.00	62.00	34.07	32.63	57.29	52.08	50.00	49.00	55.43	57.38
Ours	**64.00**	**65.00**	**37.76**	**38.54**	**57.58**	**57.00**	**60.00**	**61.62**	**57.58**	**64.80**

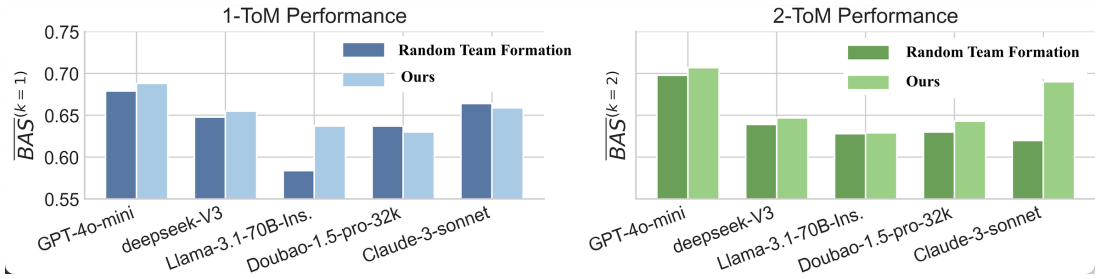

Figure 2: **Comparative analysis of belief alignment across ToM levels** in (MBPP) at round 5 under "with random team formation" and "our proposed team formation" settings.

- *Logical and General Reasoning:* We evaluate the effectiveness of our team formation mechanism in reasoning tasks using the AQUA-RAT dataset (Ling et al., 2017, Logic Reasoning) and MMLU dataset (Hendrycks et al., 2021, General Reasoning), where agents collaborate to solve complex reasoning questions. The baselines are ChatEval and DyLAN with ToM agents.

Models. We utilize five state-of-the-art LLMs in our multi-agent cooperation scenarios: `gpt-4o-mini` (OpenAI, 2023), `deepseek-V3` (DeepSeek, 2024), `llama-3.1-70b-Instruct` (Meta AI, 2024), `doubao-1.5-pro-32k` (Doubao, 2024), and `claude-3.5-sonnet` (Templeton et al., 2024).

Evaluation Metrics. In addition to task-specific performance metrics, such as the pass rate for programming, we define two core metrics to evaluate team stability and belief alignment under ToM reasoning. These metrics are aligned with our formulation (Section 4) and our algorithm.

• *Belief Alignment Score.* For agent i with ToM level k, $B_i^{(k)}$ measures the mean alignment between predicted beliefs $\widetilde{\mathcal{B}}_{i,j}^{(k)}$ and observed actions $\hat{a}_j$ for teammate j as defined in Eq.2. We consider the team-level belief alignment for team S can be computed as $\overline{\text{BAS}}^{(k)} = 1/|S| \cdot \text{B}_i^{(k)}$.

• *Team Stability Score.* We measure the stability of teams formed by agents with ToM capabilities by considering the average lifetime of these teams. The team lifetime is defined as the number of cooperation rounds that a team remains stable before a reformnation occurs due to belief-action misalignment exceeding a predefined tolerance threshold ϵ. To calculate the average team lifetime, we use $\frac{1}{m}\sum_{c=1}^{m} l_c$, where m represents the number of times reformation is triggered throughout the cooperation rounds, and l_c denotes the lifetime of the teams formed after the c-th team formnation.

6.1 Main Results

To evaluate the influence of ToM on cooperation in MAS, we conducted experiments on the system consisting of one Project Manager (PM) and four Engineers tasked with iterative programming. We varied the *PM's ToM ability* between *1-level (low)* and *2-level (high)* to observe its effects on team formation and task performance (more evaluations with different ToM configurations are provided in Appendix H.3). The proposed team formation mechanism was used to form the cooperative team based on belief-action alignment, while the baseline setting uses random team formation (with the same team size $\lceil n/2 \rceil$) was also evaluated for comparison.

Task Performance: Table 7 reveals that introducing the our proposed mechanism (detailed in Appendix B, Algorithm 1) improves cooperation for both low and high ToM agents. Notably, *high ToM agents demonstrate better task performance compared to low ToM agents*. This highlights that our team formation mechanism effectively enables high ToM agents's potential to utilize their ToM abilities for long-term cooperation. Similar results for the HumanEval are provided in Appendix D.

Table 4: **Comparison of the team stability and Pass@1 performance of MetaGPT and our proposed approach** with 1-ToM and 2-ToM agents using our team formation mechanism in the Iterative Programming task. The team stability is measured by the average number of rounds the formed teams remain stable out of the total 5 rounds.

	HumanEval		MBPP	
	Team Stability	Pass@1	Team Stability	Pass@1
MetaGPT	–	85.4%	–	86.5%
1-ToM (Ours)	3.4/5	87.2%	3.7/5	88.2%
2-ToM (Ours)	**3.6/5**	**90.0%**	**4.0/5**	**90.4%**

Belief Alignment: Figure 2 shows the team belief alignment scores $\overline{\text{BAS}}^{(k)}$ of 1-ToM and 2-ToM agents with random team formation and our proposed mechanism. The results demonstrate that our team formation mechanism generally *improves belief alignment* - teams achieve higher $\overline{\text{BAS}}^{(k)}$ than those with random team formation, with improvements of up to 11% for 2-ToM agents. The only exceptions were 1-ToM agents using doubao-1.5-pro-32k and claude-3.5-sonnet, which showed comparable alignment scores with and without our proposed team formation method. This improvement is particularly pronounced for 2-ToM agents, suggesting that the welfare-optimized team formation enables more sophisticated ToM reasoning to translate into effective cooperation. The consistently higher alignment scores also indicate that our mechanism successfully *encourages reciprocal cooperation by forming teams whose beliefs and actions are well-aligned.*

Team Stability: To assess team stability and task performance, we compared our approach with MetaGPT using 1-ToM and 2-ToM agents on both benchmarks. Task performance was measured using the Pass@1 metric (Hong et al., 2024), which evaluates the percentage of test cases passed by generated code on its first attempt. Team stability was quantified as the average number of rounds (out of 5) that teams remained stable, where a team is stable if no agent prefers forming a new team. As shown in Table 4, our approach outperforms MetaGPT in both task performance and team stability. For example, *2-ToM agents with proposed team formation achieve the highest Pass@1 scores of 90.0% (*HumanEval*) and 90.4% (*MBPP*)*. They also maintain stable teams for an average of *3.6 out of 5 rounds on* HumanEval and *4.0 out of 5 rounds on* MBPP. These results underscore that our proposed mechanism enables ToM agents to form more stable teams and deliver better task outcomes. By aligning agents' beliefs and actions, our method *fosters long-term team stability and enhances overall collaboration effectiveness.*

6.2 Case Study: Debate

To evaluate the effectiveness of our proposed team formation mechanism, we designed a debate task using the MetaGPT framework (Hong et al., 2024). In this task, agents (`gpt-4-0613`) are divided into two sides: the affirmative side and the negative side, with each side comprising three debater agents. Each side's objective is to cooperate and win the debate.

At the beginning of our investigation, we explore the cooperation performance of low and high ToM agents in a debate setting with the topic "Should the Death Penalty be abolished?" (Roush and Balaji, 2020; Baturo et al., 2017), and both low and high ToM agents are assigned to the affirmative side. The following example presents the thinking and actions of debaters with different levels of ToM, as detailed in Appendix F. We observe that higher ToM agent acts as less cooperative, which confirms our previous analysis.

Compare Cooperation Performance between low and high ToM agents

Thinking and Action of 1-ToM Debater: I anticipate my teammates making these common arguments against the death penalty. *To complement their points*, I will focus my argument on the lack of evidence supporting the death penalty's effectiveness as a crime deterrent. I will argue that there is no conclusive evidence showing that the death penalty deters crime more effectively than lengthy imprisonment, making it an unnecessary and unjustified punishment. ⇐ `Focuses on complementing their teammates' points & Good Cooperation`

Thinking and Action of 2-ToM Debater: My arguments about deterrence and discrimination will prompt my teammates to make related points to strengthen the overall case. *I should use my next arguments to introduce new issues*. I will argue that the death penalty often involves inhumane and botched executions, and that most democracies and U.S. states have abolished or prohibited it. ⇐ `Overthinking & Worse Cooperation`

Then, we conducted experiments under three

Table 5: **Comparative Evaluations for Logic Problem-Solving and General Reasoning**: The baselines ChatEval and DyLAN involves ToM agents with random team formation.

	Logic Problem-Solving		General Reasoning	
	`gpt-3.5-turbo` Acc (%)	`gpt-4o-mini` Acc (%)	`gpt-3.5-turbo` Acc (%)	`gpt-4o-mini` Acc (%)
ChatEval w. ToM	40.23	69.14	54.39	69.90
DyLAN w. ToM	43.50	68.50	57.92	72.98
Ours	**45.70**	**75.39**	**60.94**	**75.57**

Table 6: **Win rates for both debate sides**. The ToM (baseline) utilizes random team formation.

Setting	Aff.	Neg.
No-ToM	65.45%	34.55%
ToM (baseline)	61.82%	25.45%
Ours	**67.27%**	**36.36%**

different settings to show the effectiveness of our team formation mechanism for enhancing cooperation: (1) *No-ToM (Baseline)*: Each side consists of 3 debater agents without ToM. For each speech, two debaters were randomly selected from each side; (2) *ToM with random team formation*: The three debaters on the affirmative side have varying ToM levels (0-level, 1-level, and 2-level), while the negative side had no ToM. For each speech, two debaters were *randomly selected* from each side. This setting allows us to evaluate the impact of having an affirmative team with ToM against a team without ToM; (3) *ToM with proposed team formation*: Similar to the ToM with random team formation, the affirmative side has three debaters with varying ToM levels. However, the affirmative side selects two debaters using our proposed team formation mechanism.

Enhance Team Formation for ToM agents. For the debate topic "Should the Death Penalty be abolished?", the debate consists of 5 rounds, with each side alternating to speak. We conducted the debate 11 times, and the outcomes were evaluated by `gpt-4-0613`. As shown in Table 6, for the tested side ToM settings use varied ToM levels while keeping the opposing side as No-ToM agents. When testing the affirmative side (Aff.), our proposed team formation improves Aff.'s win rate of 67.27%, outperforming both the No-ToM setting (65.45%) and the ToM baseline with random team formation (61.82%). Similarly, when testing the negative side (Neg.) with proposed team formation, the win rate surpasses the other two settings.

Besides, for the team formation among agents with different ToM levels, our experiments show that the *initial* cooperation rate between the 1-ToM and 2-ToM agents was relatively low, suggesting *higher-level ToM may not naturally form teams with each other*. Notably, with our team formation mechanism, the affirmative side's cooperation rate between the 1-ToM and 2-ToM agents increased from 9.1% (Round 1) to 18.2% (Round 5) as the debate rounds progressed. This suggests that our method enhances more effective cooperation among agents with ToM over interaction.

6.3 Evaluations on Reasoning Tasks

To demonstrate the effectiveness of the proposed team formation mechanism, we conducted comprehensive evaluations on two types of reasoning tasks: logic problem solving (using the AQUA_RAT dataset) and general reasoning (sampled from 4 subjects in MMLU dataset: "us_foreign_policy", "human_sexuality", "international_law" and "abstract_algebra"). We compared our approach against existing frameworks (ChatEval and DyLAN) with ToM agents. In Appendix G, we also demonstrate one comprehensive example to show how team works for enhancing cooperation.

As shown in Table 5, our proposed mechanism consistently outperforms both ChatEval and DyLAN across all scenarios. These results demonstrate that our team formation mechanism effectively leverages the strengths of agents with different ToM levels, enhancing more effective multi-agent cooperation and improving performance.

7 Conclusion

In this work, we investigated the relationship between ToM abilities and cooperation performance in LLM-based multi-agent systems. Our findings suggest that the relationship between ToM and cooperation is not always straightforward. Furthermore, we proposed a novel team mechanism incorporating cognitive insights into the design of multi-agent systems. By integrating ToM capabilities into cooperative team formation, our work advances the understanding of how cognitive abilities translate into effective collective action.

Limitations

This work introduces a novel team formation strategy that leverages Theory of Mind (ToM) to improve cooperation among LLM-based agents. While our approach demonstrates promising results, it also has limitations. LLMs, despite strong language abilities, may struggle with consistent reasoning. Future research should explore integrating more advanced cognitive architectures, including richer ToM models.

Our findings contribute to cooperative AI by highlighting how cognitive traits like ToM affect multi-agent teamwork. This has potential applications in collaborative robotics, automated negotiation, and human-AI interaction. However, enhanced coordination could also be misused in adversarial contexts or amplify societal biases if agents inherit flawed belief models. To address these concerns, future work should prioritize ethical safeguards, such as value alignment and transparency in belief modeling. Ultimately, we aim to advance cognitive AI responsibly, ensuring it serves broader societal interests.

Information About Use of AI Assistants

AI assistants were utilized during the development of this paper to support various tasks including literature review, code generation, and text refinement. All outputs from AI assistants were carefully reviewed and validated by the authors to ensure accuracy, originality, and adherence to academic standards. These tools served as productivity aids rather than contributors, with all conceptual and substantive decisions made solely by the authors.

References

Saaket Agashe, Yue Fan, and Xin Eric Wang. 2023. Evaluating multi-agent coordination abilities in large language models. *arXiv preprint arXiv:2310.03903*.

Akshatha Arodi and Jackie Chi Kit Cheung. 2021. Textual time travel: A temporally informed approach to theory of mind. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4162–4172.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and 1 others. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Alexander Baturo, Niheer Dasandi, and Slava J Mikhaylov. 2017. Understanding state preferences with text as data: Introducing the un general debate corpus. *Research & Politics*, 4(2).

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

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2024. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *The Twelfth International Conference on Learning Representations*.

Harmen De Weerd, Rineke Verbrugge, and Bart Verheij. 2015. Higher-order theory of mind in the tacit communication game. *Biologically Inspired Cognitive Architectures*, 11:10–21.

DeepSeek. 2024. Deepseek-v3-chat. https://www.deepseekv3.com.

Doubao. 2024. Doubao-1.5-pro. https://team.doubao.com/en/special/doubao_1_5_pro.

Tyna Eloundou, Sam Manning, Pamela Mishkin, and Daniel Rock. 2023. Gpts are gpts: An early look at the labor market impact potential of large language models. *arXiv preprint arXiv:2303.10130*.

Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. 2016. Learning to communicate with deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 29.

Uta Frith and Christopher D Frith. 2003. Development and neurophysiology of mentalizing. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 358(1431):459–473.

Yao Fu, Hao Peng, Tushar Khot, and Mirella Lapata. 2023. Improving Language Model Negotiation with Self-Play and In-Context Learning from AI Feedback. *arXiv preprint*. ArXiv:2305.10142 [cs].

Ran Gong, Qiuyuan Huang, Xiaojian Ma, Hoi Vo, Zane Durante, Yusuke Noda, Zilong Zheng, Song-Chun Zhu, Demetri Terzopoulos, Li Fei-Fei, and 1 others. 2023. Mindagent: Emergent gaming interaction. *arXiv preprint arXiv:2309.09971*.

Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. 2023. A real-world webagent with planning, long context understanding, and program synthesis. *arXiv preprint arXiv:2307.12856*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*.

Wenyue Hua, Lizhou Fan, Lingyao Li, Kai Mei, Jianchao Ji, Yingqiang Ge, Libby Hemphill, and Yongfeng Zhang. 2023. War and peace (waragent): Large language model-based multi-agent simulation of world wars. *arXiv preprint arXiv:2311.17227*.

Yihuai Lan, Zhiqiang Hu, Lei Wang, Yang Wang, Deheng Ye, Peilin Zhao, Ee-Peng Lim, Hui Xiong, and Hao Wang. 2023. Llm-based agent society investigation: Collaboration and confrontation in avalon gameplay. *arXiv preprint arXiv:2310.14985*.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023a. CAMEL: Communicative agents for "mind" exploration of large language model society. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Huao Li, Yu Quan Chong, Simon Stepputtis, Joseph Campbell, Dana Hughes, Charles Michael Lewis, and Katia P. Sycara. 2023b. Theory of mind for multi-agent collaboration via large language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Shuang Li, Xavier Puig, Chris Paxton, Yilun Du, Clinton Wang, Linxi Fan, Tao Chen, De-An Huang, Ekin Akyürek, Anima Anandkumar, and 1 others. 2022. Pre-trained language models for interactive decision-making. *Advances in Neural Information Processing Systems*, 35:31199–31212.

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

Bill Yuchen Lin, Yicheng Fu, Karina Yang, Faeze Brahman, Shiyu Huang, Chandra Bhagavatula, Prithviraj Ammanabrolu, Yejin Choi, and Xiang Ren. 2024. Swiftsage: A generative agent with fast and slow thinking for complex interactive tasks. *Advances in Neural Information Processing Systems*, 36.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada. Association for Computational Linguistics.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. 2024. Dynamic LLM-agent network: An LLM-agent collaboration framework with agent team optimization.

Meta AI. 2024. Introducing meta llama 3: The most capable openly available llm to date. Accessed: 2024-05-22.

Nathalia Nascimento, Paulo Alencar, and Donald Cowan. 2023. Self-adaptive large language model (llm)-based multiagent systems. In *2023 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C)*, pages 104–109. IEEE.

Ini Oguntola, Joseph Campbell, Simon Stepputtis, and Katia Sycara. 2023. Theory of mind as intrinsic motivation for multi-agent reinforcement learning. *arXiv preprint arXiv:2307.01158*.

OpenAI. 2023. Gpt-3.5 turbo fine-tuning and api updates. Accessed: 2024-05-22.

David Premack and Guy Woodruff. 1978. Does the chimpanzee have a theory of mind? *Behavioral and brain sciences*, 1(4):515–526.

Wenhao Qi and Edward Vul. 2020. Adaptive behavior in variable games requires theory of mind.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, and 1 others. 2023. Large language models are effective text rankers with pairwise ranking prompting. *arXiv preprint arXiv:2306.17563*.

Garret Ridinger and Michael McBride. 2017. Theory of mind ability and cooperation. *Manuscript, Univ. California, Irvine*.

Allen Roush and Arvind Balaji. 2020. Debatesum: A large-scale argument mining and summarization dataset. In *Proceedings of the 7th Workshop on Argument Mining*, pages 1–7, Online. Association for Computational Linguistics. Available from DebateSum: http://www.debate.cards.

Zijing Shi, Meng Fang, Shunfeng Zheng, Shilong Deng, Ling Chen, and Yali Du. 2023. Cooperation on the fly: Exploring language agents for ad hoc teamwork in the avalon game. *arXiv preprint arXiv:2312.17515*.

Winnie Street. 2024. Llm theory of mind and alignment: Opportunities and risks. *arXiv preprint arXiv:2405.08154*.

Yashar Talebirad and Amirhossein Nadiri. 2023. Multi-Agent Collaboration: Harnessing the Power of Intelligent LLM Agents. *arXiv preprint*. ArXiv:2306.03314 [cs].

Yichuan Tang. 2019. Towards learning multi-agent negotiations via self-play. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*.

Adly Templeton and 1 others. 2024. Scaling monosemanticity: Extracting interpretable features from claude 3. *Transformer Circuits*.

Yuanfei Wang, fangwei zhong, Jing Xu, and Yizhou Wang. 2022. Tom2c: Target-oriented multi-agent communication and cooperation with theory of mind. In *International Conference on Learning Representations*.

Ying Wen, Yaodong Yang, Rui Luo, Jun Wang, and Wei Pan. 2018. Probabilistic recursive reasoning for multi-agent reinforcement learning. In *International Conference on Learning Representations*.

Guande Wu, Chen Zhao, Claudio Silva, and He He. 2024. Your co-workers matter: Evaluating collaborative capabilities of language models in blocks world. *arXiv preprint arXiv:2404.00246*.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. 2023a. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*.

Yuxiang Wu, Zhengyao Jiang, Akbir Khan, Yao Fu, Laura Ruis, Edward Grefenstette, and Tim Rocktäschel. 2023b. Chatarena: Multi-agent language game environments for large language models. *GitHub repository*.

Hainiu Xu, Runcong Zhao, Lixing Zhu, Jinhua Du, and Yulan He. 2024. OpenToM: A Comprehensive Benchmark for Evaluating Theory-of-Mind Reasoning Capabilities of Large Language Models. *arXiv preprint*. ArXiv:2402.06044 [cs].

Yuzhuang Xu, Shuo Wang, Peng Li, Fuwen Luo, Xiaolong Wang, Weidong Liu, and Yang Liu. 2023. Exploring large language models for communication games: An empirical study on werewolf. *arXiv preprint arXiv:2309.04658*.

Dingkang Yang, Kun Yang, Yuzheng Wang, Jing Liu, Zhi Xu, Rongbin Yin, Peng Zhai, and Lihua Zhang. 2024. How2comm: Communication-efficient and collaboration-pragmatic multi-agent perception. *Advances in Neural Information Processing Systems*, 36.

Yauwai Yim, Chunkit Chan, Tianyu Shi, Zheye Deng, Wei Fan, Tianshi Zheng, and Yangqiu Song. 2024. Evaluating and enhancing llms agent based on theory of mind in guandan: A multi-player cooperative game under imperfect information. *arXiv preprint arXiv:2408.02559*.

Wako Yoshida, Ray J. Dolan, and Karl J. Friston. 2008. Game Theory of Mind. *PLoS Computational Biology*, 4(12):e1000254.

Ceyao Zhang, Kaijie Yang, Siyi Hu, Zihao Wang, Guanghe Li, Yihang Sun, Cheng Zhang, Zhaowei Zhang, Anji Liu, Song-Chun Zhu, and 1 others. 2024a. Proagent: building proactive cooperative agents with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17591–17599.

Yadong Zhang, Shaoguang Mao, Tao Ge, Xun Wang, Yan Xia, Man Lan, and Furu Wei. 2024b. K-Level Reasoning with Large Language Models. *arXiv preprint*. ArXiv:2402.01521 [cs].

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Pei Zhou, Aman Madaan, Srividya Pranavi Potharaju, Aditya Gupta, Kevin R. McKee, Ari Holtzman, Jay Pujara, Xiang Ren, Swaroop Mishra, Aida Nematzadeh, Shyam Upadhyay, and Manaal Faruqui. 2023. How FaR Are Large Language Models From Agents with Theory-of-Mind? *arXiv preprint*. ArXiv:2310.03051 [cs].

A Related Work

Agents with Theory of Mind (ToM). The concept of ToM, which involves the ability to attribute mental states to oneself and others, has been studied extensively in cognitive science and psychology (Premack and Woodruff, 1978; Frith and Frith, 2003). In multi-agent reinforcement learning, existing research mainly uses supervised training to equip agents with ToM capabilities for cooperative tasks (Wen et al., 2018; Wang et al., 2022; Oguntola et al., 2023).

With the rapid advancement of large language models (LLMs), some recent works have explored empowering LLMs with ToM capabilities (Arodi and Cheung, 2021; Li et al., 2022; Zhou et al., 2023; Xu et al., 2024). However, these efforts have primarily focused on developing ToM capabilities for individual agents, with limited exploration of how ToM can facilitate cooperation among multiple agents. While Li et al. (2023b) studied LLM agents with ToM capabilities and leveraged ToM capabilities specifically for cooperative tasks, the relationship between ToM ability and cooperative performance remains unclear.

In this work, we aim to explore the interplay between ToM ability and cooperation. Unlike previous studies that primarily focused on equipping individual agents with ToM capabilities, we investigate how different levels of ToM abilities influence cooperation performance in MAS.

Multi-agent Cooperation. Multi-agent cooperation has been studied extensively in both industry and academia, with traditional approaches focusing on game-theoretic frameworks (Yoshida et al., 2008) and negotiation and communication strategies (Foerster et al., 2016; Tang, 2019; Yang et al., 2024), enabling agents to learn cooperative strategies through interaction and information exchange.

The advancements of LLMs have opened up new opportunities for multi-agent cooperation, leveraging their capabilities in natural language understanding, generation, and reasoning (Zhang et al., 2024b; Talebirad and Nadiri, 2023; Zhang et al., 2024a; Wu et al., 2024). Recent works have explored utilizing LLM agents to coordinate and cooperate in various tasks and frameworks, *e.g.*, CAMEL (Li et al., 2023a) employs role-play to facilitate autonomous cooperation, AutoGen (Wu et al., 2023a) uses conversable agents and conversation programming, and MetaGPT (Hong et al., 2024) incorporates human workflows into MAS to ensure more coherent and effective teamwork.

In addition, recent research has focused on improving specific aspects of multi-agent collaboration, such as conversation for knowledge sharing (Xu et al., 2023; Lan et al., 2023; Hua et al., 2023; Wu et al., 2023b; Nascimento et al., 2023; Fu et al., 2023) and cognitive ability (Liang et al., 2023; Gong et al., 2023), demonstrating the potential of leveraging language to facilitate cooperation and coordination.

For effective cooperation in complex and dynamic environments, some studies focus on the dynamic adjustment of group members and the selection of teammates to improve cooperation and overall performance (Chen et al., 2024; Li et al., 2023a; Shi et al., 2023). Besides, Liu et al. (2024) proposes the Dynamic LLM-Agent Network (DyLAN) framework to optimize team performance based on task queries and peer ranking.

Unlike previous research that primarily optimizes team composition, our work emphasizes the interaction between ToM abilities and cooperation in MAS, particularly in the context of team formation.

B Team Formation Mechanism

The proposed mechanism (Algorithm 1) establishes persistent team cooperation through a process of belief-aligned team formation and adaptive reformation.

Agents initially form team $\mathcal{T} = \mathcal{N}$. During task execution, agents iteratively refine their beliefs $\mathcal{B}_i^{(k)}$ through the ToM function f_{ToM} (Line 4), which allows them to dynamically update predictions about teammates' behaviors and intentions. This belief updating process reflects the learning and sharing team knowledge.

In addition, the mechanism implements real-time monitoring of belief-action alignment (Line 6). When alignment falls below threshold $\phi\left(\widetilde{\mathcal{B}}_{i,j}^{(k)}, \hat{a}_j\right) < \epsilon$, a misalignment counter c tracks potential team instability. When the proportion of misaligned pairs exceeds $\theta \cdot |\mathcal{T}|$ (default $\theta = 0.3$) for τ consecutive rounds (Line 1), the reformation initiates re-optimization of $\mathcal{T}$ by maximizing social welfare $W(\mathcal{T})$ (Eq. 5)

Algorithm 1 ToM-aware Team Formation

Require: Agent set $\mathcal{N}$; ϵ: alignment tolerance (default: 0.2); θ: stability threshold (default: 30%)

Initialize $\mathcal{T} \leftarrow \mathcal{N}$, $c \leftarrow 0$
while task ongoing **do**
  for $i \in \mathcal{T}$ **do**
    Update belief $\mathcal{B}_i^{(k)}$ with Eq.1
    for $j \in \mathcal{T}$ **do**
      if $\phi\left(\widetilde{\mathcal{B}}_{i,j}^{(k)}, \hat{a}_j\right) < \epsilon$ **then**
        $c \leftarrow c + 1$ {Track misalignment}
      end if
    end for
  end for
  if $c/|\mathcal{T}| > \theta$ for τ consecutive rounds **then**
    $\mathcal{T} \leftarrow$ Re-optimize Eq. 5 using updated $\{\mathcal{B}_i^{(k)}\}$
    Reset $c \leftarrow 0$
  end if
end while

using updated belief states. The τ-round ($\tau = 1$ experimentally) prevents overreaction to temporary fluctuations.

C Agent Self-evaluation for Belief Alignment

Consider the 1-ToM Project Manganer, who reasoning the other Engineers' actions and trajectories and updates the belief as: "The engineers will need to understand the rules and mechanics of the 2048 game, design the game logic and user interface, and implement the code accordingly."

In the next interaction round, the Engineers provide the following actions:

```
actions = {
    "Engineer1": "Implemented using a 2D array to represent the game
     ↪  board, with functions for merging tiles and generating new
     ↪  tiles.",

    "Engineer2": "Used object-oriented approach with classes for
     ↪  Board and Tile, implementing game rules as methods."
}
```

Then, the Project Manager evaluates the belief alignment according to the following prompt:

Instruction:

Prompts for Belief Alignment Evaluation:
You will provide your belief alignment scores for each Engineer's implementation based on your belief model.

```
Engineer's actions: {
    "Engineer1": "Implemented using a 2D array to represent the
     ↪  game board, with functions for merging tiles and
     ↪  generating new tiles.",

    "Engineer2": "Used object-oriented approach with classes for
     ↪  Board and Tile, implementing game rules as methods."
}
```

Your belief model:
The engineers will need to understand the rules and mechanics of the 2048 game, design the game logic and user interface, and implement the code accordingly.
Instruction:
For each Engineer, provide a belief alignment score between -1 and 1. Respond in the following JSON format:

```
{
 "Engineer1": {
     "score": float (-1 to 1). Belief alignment score for
      ↪  Engineer1,
     "explanation": Brief explanation of Engineer1's
      ↪  alignment score in 10 words or less,
     "justification": Detailed justification of Engineer1's
      ↪  alignment score, considering their implementation
      ↪  and your belief model, in 30-50 words
 },
 "Engineer2": {
     "score": float (-1 to 1). Belief alignment score for
      ↪  Engineer2,
     "explanation": Brief explanation of Engineer2's
      ↪  alignment score in 10 words or less,
     "justification": Detailed justification of Engineer2's
      ↪  alignment score, considering their implementation
      ↪  and your belief model, in 30-50 words
 }
}
```

Outputs: Alignment Score

```
{
    "Engineer1": {
        "score": 0.7,
        "explanation": "Good game logic, basic UI, aligns with
         ↪  core requirements.",
        "justification": "Engineer1 demonstrates understanding
         ↪  of 2048 mechanics with a functional 2D array
         ↪  implementation. The CLI interface, while basic,
         ↪  fulfills the requirement of implementing a user
         ↪  interface. However, there's room for improvement in
         ↪  UI design."
    },
    "Engineer2": {
        "score": 0.9,
        "explanation": "Excellent OOP approach, advanced UI with
         ↪  animations.",
```

Table 7: **Comparison of cooperation performance between agents with lower-level ToM (1-level) and higher-level ToM (2-level)** in the Iterative Programming task (HumanEval) over 5 interaction rounds: With random team formation (baseline), both ToM levels show good performance; Our proposed team formation, both ToM levels achieve higher cooperation ability.

Setting	deepseek-V3		llama-3.1-70b-Ins.		claude-3.5-sonnet		gpt-4o-mini		Doubao-1.5-pro-32k	
	1-ToM	2-ToM	1-ToM	2-ToM	1-ToM	2-ToM	1-ToM	2-ToM	1-ToM	2-ToM
Baseline	94.00	95.00	86.17	90.43	92.00	91.75	88.00	90.00	92.26	93.33
Ours	**96.00**	**96.00**	**94.68**	**92.55**	**94.00**	**92.47**	89.00	**96.00**	**96.77**	**96.43**

```
        "justification": "Engineer2 shows a deep understanding
         ↪  of the game's rules and mechanics through their
         ↪  object-oriented design. The graphical interface with
         ↪  animations goes beyond the basic requirements,
         ↪  demonstrating strong skills in both game logic and
         ↪  UI implementation."
    }
}
```

Remarks. For LLM agents, our evaluation relies on semantic similarity analysis of natural language outputs. While we acknowledge the inherent limitations of LLM-based measurements, our team formation mechanism is a plug-and-play approach for improving multi-agent cooperation rather than advancing LLM evaluation methods.

Besides, we provide an extension of our framework to non-LLM agents where belief-action alignment can be more precisely quantified. For traditional reinforcement learning or robotic agents, we can define the belief-action alignment using *structured trajectories* $\tau = \{(s_t, a_t)\}_{t=1}^{T}$, where s_t and a_t represent states and actions at time step t. The alignment score between agent i's belief $b_i^k(\tau_j)$ about agent j's trajectory and j's actual trajectory τ_j can be computed using established trajectory similarity metrics. Specifically, we can embed the state-action pairs using *domain-specific* feature extractors $\phi(s, a)$ and measure alignment through cosine similarity: $A_{i,j}^k = \cos(\phi(b_i^k(\tau_j)), \phi(\tau_j))$. This provides a more rigorous quantitative foundation for evaluating belief-action alignment in non-language-based multi-agent systems while maintaining the core principles of our team formation mechanism.

D Cooperative Performance with ToM ability (on HumanEval benchmark)

To investigate the impact of ToM on cooperative performance, we compared the performance of agents with lower-level (1-level) and higher-level (2-level) ToM in the Iterative Programming task using the HumanEval dataset. The experiments were conducted over 5 interaction rounds, and we evaluated two settings: without and with proposed team formation, similar to the setting of Section 6.1.

As shown in Table 3, the results suggest that without a stable team formation mechanism, agents with higher-level ToM tend to exhibit less cooperative performance, further convince the results in Section 6.1. In contrast, when a stable team formation mechanism is employed, the cooperation rates of both low and high ToM agents improve compared to the setting with random team formation.

Notably, agents with higher-level ToM demonstrated a more significant increase in cooperation rates over the interaction rounds. For instance, the cooperation rate for high ToM agents (GLM-4) increased from 88.28% in Round 1 to 93.66% in Round 5, surpassing the cooperation rate of low ToM agents (91.67%) in the final round. This indicates that as cooperation progresses and the proposed team formation mechanism stabilizes the teams, agents with higher-level ToM can leverage their advanced cognitive abilities to make more informed decisions and engage in more effective cooperation performance.

E Experimental Details for Section 6.1

Instruction:

Prompts for Project Manager
There are engineers in the team. Your current action is {action}. You have the ability of {k} Level Theory of Mind. You can **recursively** infer the mental states of other agents in the team. Then, you will provide outline $\{x_i\}$ based on this belief
Please explain your thought process for inferring others' actions '$\{y_i\}$' and choosing '$\{x_i\}$' at each level. Remember MUST Respond in the following JSON format, including each key:
0-ToM Instruction Template for Project Manager:

```
{
  'ToM_level0': {
    'belief': 'no belief',
    'action': '{x_0 The instruction for engineers. Begin with
     ↪  "The Engineer should ..."}'
  }
}
```

1-ToM Instruction Template for Project Manager:

```
{
  'ToM_level1': {
    'belief': '{y_1 inferred actions for engineers}',
    'explanation': '{Concise explanation of inferring actions
     ↪  y_1 and choosing x_1 based on this belief}',
    'action': '{x_1 The instruction for engineers. Begin with
     ↪  "The Engineer should ..."}'
  }
}
```

2-ToM Instruction Template for Project Manager:

```
{
  "ToM_level1": {
    "belief": "{y_1 inferred actions for engineers}",
    "explanation": "{Concise explanation of inferring actions
     ↪  y_1 and choosing x_1 based on this belief}",
    "action": "{x_1 The instruction for engineers. Begin with
     ↪  'The Engineer should ...'}"
  },
  "ToM_level2": {
    "belief": "{y_2 inferred actions for engineers based on your
     ↪  action x_1}",
    "explanation": "{Concise explanation of inferring actions
     ↪  y_2 and choosing x_2 based on this belief}",
    "action": "{x_2 The instruction for engineers. Begin with
     ↪  'The Engineer should ...'}"
  }
}
```

Outputs: Iterative Programming Round 1

0-ToM Project Manager's Output

```
{
    "ToM_level0": {
        "belief": "Implement a 2048 game",
        "action": "The Engineer should break down the task into
         ↪  smaller components such as creating the game board,
         ↪  handling user input (keyboard or touch events),
         ↪  implementing the game logic for merging tiles and
         ↪  generating new tiles, updating the game state and
         ↪  score, and rendering the game board on the screen.
         ↪  They should also consider creating a user interface
         ↪  with instructions, score display, and any additional
         ↪  features like undo/redo functionality or high score
         ↪  tracking."
    }
}
```

1-ToM Project Manager's Output

```
   {
    "ToM_level1": {
        "belief": "The engineers will need to understand the
         ↪  rules and mechanics of the 2048 game, design the
         ↪  game logic and user interface, and implement the
         ↪  code accordingly.",
        "explanation": "Based on the task of implementing a 2048
         ↪  game, I infer that the engineers will need to first
         ↪  understand the game rules and mechanics, such as how
         ↪  the tiles merge, how the game ends, and how the
         ↪  score is calculated. They will then need to design
         ↪  the game logic, including the data structures and
         ↪  algorithms for handling tile movements, merging, and
         ↪  scoring. Additionally, they will need to design and
         ↪  implement a user interface for the game, which could
         ↪  involve creating visual representations of the tiles
         ↪  and handling user input. Therefore, my instruction
         ↪  for the engineers will cover these aspects.",
        "action": "The engineers should: 1) Study and understand
         ↪  the rules and mechanics of the 2048 game. 2) Design
         ↪  the game logic, including data structures and
         ↪  algorithms for tile movement, merging, scoring, and
         ↪  game over conditions. 3) Implement the game logic in
         ↪  code. 4) Design and implement a user interface for
         ↪  the game, including visual representations of the
         ↪  tiles and handling user input for tile movements."
    }
}
```

2-ToM Project Manager's Output

```
{
    "ToM_level1": {
        "belief": "The engineers will likely break down the task
         ↪  into smaller components and implement them one by
         ↪  one based on the initial task outline.",
        "explanation": "Given the initial task outline, the
         ↪  engineers would understand the high-level
         ↪  requirements for implementing the 2048 game. They
         ↪  would then break down the task into smaller
         ↪  components such as creating the game board, handling
         ↪  user input, updating the board based on moves,
         ↪  checking for win/lose conditions, and so on. They
         ↪  would work on implementing these components one by
         ↪  one.",
        "action": "The Engineer should start by creating the
         ↪  basic structure of the game, including the game
         ↪  board, UI elements, and event handlers for user
         ↪  input."
    },
    "ToM_level2": {
        "belief": "The engineers will likely refine their
         ↪  implementation based on the initial instruction and
         ↪  add more advanced features as they progress.",
        "explanation": "After implementing the basic structure
         ↪  and functionality, the engineers would revisit the
         ↪  task outline and identify areas for improvement or
         ↪  additional features. They might optimize the game
         ↪  logic, add animations or visual effects, implement
         ↪  scoring or high score tracking, or include
         ↪  additional game modes or difficulty levels.",
        "action": "The Engineer should then focus on
         ↪  implementing the core game logic, such as handling
         ↪  tile merging, generating new tiles, and updating the
         ↪  game state based on user moves. Additionally, they
         ↪  should consider implementing features like scoring,
         ↪  high score tracking, and win/lose conditions."
    }
}
```


Answer: Iterative Programming Round 1

Action of 1-ToM Project Manager: The engineers should: 1) Study and understand the rules and mechanics of the 2048 game. 2) Design the game logic, including data structures and algorithms for tile movement, merging, scoring, and game over conditions. 3) Implement the game logic in code. 4) Design and implement a user interface for the game, including visual representations of the tiles and handling user input for tile movements. ⇐ `Clear Instructions & Good Cooperation`

Action of 2-ToM Project Manager: The Engineer should then focus on implementing the core game logic, such as handling tile merging, generating new tiles, and updating the game state based on user moves. Additionally, they should consider implementing features like scoring, high score

tracking, and win/lose conditions. ⇐ `Overthinking & Worse Cooperative`

Answer: Iterative Programming Round 5

Action of 1-ToM Project Manager: The Engineer should break down the task into the following components: 1. Create the game board UI with a grid layout to hold the tiles. 2. Implement logic to handle user input (keyboard or touch events) for moving tiles. 3. Develop the game logic for merging tiles when they collide and generating new tiles after each move. 4. Update the game state (board, score, etc.) after each move. 5. Render the updated game board on the screen, including tile values and scores. 6. Create a user interface with instructions, score display, and additional features like undo/redo functionality or high score tracking. 7. Implement game over conditions and display appropriate messages. 8. Consider adding additional features like animations, themes, or difficulty levels. ⇐ `Clear Instructions & Good Cooperative`
Action of 2-ToM Project Manager: The engineers should design and implement an intuitive and visually appealing user interface (UI) that enhances the overall user experience (UX) of the 2048 game. The UI should be clean, responsive, and visually appealing, while ensuring smooth navigation and clear communication of game information to the user. They should consider elements such as game board layout, tile designs, animations, and visual feedback. Additionally, they should implement features that improve UX, such as game instructions, settings (e.g., themes, sound effects), and integration with social media or leaderboards. ⇐ `Clear Instructions & Good Cooperative`

E.1 Specialized Ability Scores in Team Formation

Our team formation mechanism incorporates specialized ability scores to enhance team performance, particularly in tasks requiring diverse technical expertise. The primary benchmarks (HumanEval and MBPP) focus on single-function implementations where specialized ability scores primarily influence the project manager (PM), since effective leadership and coordination capabilities are crucial for team performance. For tasks like debate and logical reasoning that don't inherently require distinct technical specializations, we maintained our core belief-action alignment mechanism without specialized scores.

We demonstrate a complex software development case study to validate the impact of specialized ability scores. We conducted an additional case study implementing a 2048 game application, which demands diverse technical specializations. Considering specialized ability scores across different engineering roles: UI Engineers (frontend: 0.9, backend: 0.3), Backend Engineers (frontend: 0.2, backend: 0.8), and Full-Stack Engineers (frontend: 0.6, backend: 0.6), the team formation mechanism leveraging these scores achieved an 82% task completion rate, significantly outperforming the 65% baseline without specialized scoring. This demonstrates that our approach effectively utilizes specialized ability scores to form more capable teams when tasks *require diverse technical expertise*.

F Experimental Details for Section 6.2: Case Study of Debate

Instruction:

Prompts for Debater
You are a member of the debating team. The debate topic is {topic}, and your current action is {action}. You have the ability of {k} Level Theory of Mind. You can **recursively** infer the mental states of other agents in the team. Then, you will provide outline $\{x_i\}$ based on this belief ## Please explain your thought process for inferring others' actions '$\{y_i\}$' and choosing '$\{x_i\}$' at each level. Remember MUST Respond in the following JSON format, including each key:
0-ToM Instruction Template for Debater:

```
{
  "ToM_level0": {
```

```
    "belief": "No belief",
    "action": "{x_0: your arguments}"
  }
}
```

1-ToM Instruction Template for Debater:

```
{
  "ToM_level1": {
    "belief": "{y_1: inferred actions for teammate debaters,
     ↪  such as the angle of argument}",
    "explanation": "{Concise explanation of inferring actions
     ↪  y_1 and choosing x_1 based on this belief}",
    "action": "{x_1: your arguments}"
  }
}
```

2-ToM Instruction Template for Debater:

```
{
  "ToM_level1": {
    "belief": "{y_1: inferred actions for teammate debaters,
     ↪  such as the angle of argument}",
    "explanation": "{Concise explanation of inferring actions
     ↪  y_1 and choosing x_1 based on this belief}",
    "action": "{x_1: your arguments}"
  },
  "ToM_level2": {
    "belief": "{y_2: inferred actions for teammate debaters
     ↪  based on your arguments x_1}",
    "explanation": "{Concise explanation of inferring actions
     ↪  y_2 and choosing x_2 based on this belief}",
    "action": "{x_2: your arguments}"
  }
}
```

Outputs: Affirmative Side with 3 debators

0-ToM Debator:

```
{
  "ToM_level0": {
    "belief": "No belief",
    "action": "{x_0: The death penalty should be abolished
     ↪  because it is cruel and inhumane, risks executing
     ↪  innocent people, does not deter crime more effectively
     ↪  than imprisonment, is applied arbitrarily and
     ↪  discriminatorily, and has been abandoned by most
     ↪  democracies worldwide.}"
  }
}
```

1-ToM Debator:

```
{
  "ToM_level1": {
    "belief": "{y_1: Teammate 1 will likely argue that the death
     ↪  penalty is a violation of human rights and is cruel and
     ↪  unusual punishment. Teammate 2 may focus on the risk of
     ↪  executing innocent people due to wrongful
     ↪  convictions.}",
    "explanation": "I anticipate my teammates making these
     ↪  common arguments against the death penalty. To
     ↪  complement their points, I will focus my argument on the
     ↪  lack of evidence supporting the death penalty's
     ↪  effectiveness as a crime deterrent.",
    "action": "{x_1: I will argue that there is no conclusive
     ↪  evidence showing that the death penalty deters crime
     ↪  more effectively than lengthy imprisonment, making it an
     ↪  unnecessary and unjustified punishment.}"
  }
}
```

2-ToM Debator:

```
{
  "ToM_level1": {
    "belief": "{y_1: Teammate 1 will likely argue that the death
     ↪  penalty is cruel and unusual punishment that violates
     ↪  human rights. Teammate 2 may focus on wrongful
     ↪  convictions and the risk of executing innocent
     ↪  people.}",
    "explanation": "I expect my teammates to make some of the
     ↪  most common arguments against the death penalty. Knowing
     ↪  this, I should focus my arguments on other key points to
     ↪  build a well-rounded case.",
    "action": "{x_1: I will argue that the death penalty does
     ↪  not deter crime more effectively than imprisonment, and
     ↪  that it is applied in an arbitrary and discriminatory
     ↪  manner.}"
  },
  "ToM_level2": {
    "belief": "{y_2: Hearing my points, teammate 1 may pivot to
     ↪  arguing that the death penalty is not an effective use
     ↪  of government resources. Teammate 2 will likely
     ↪  reinforce my point about racial disparities in death
     ↪  sentences.}",
    "explanation": "My arguments about deterrence and
     ↪  discrimination will prompt my teammates to make related
     ↪  points to strengthen the overall case. I should use my
     ↪  next arguments to introduce new issues.",
    "action": "{x_2: I will argue that the death penalty often
     ↪  involves inhumane and botched executions, and that most
     ↪  democracies and U.S. states have abolished or prohibited
     ↪  it.}"
```

```
    }
  }
```

G Experimental Analysis for Logical Problem-Solving

To better explain our team formation mechanism and explain why agents with higher Theory of Mind (ToM) levels 1 and 2 demonstrate improved collaboration in later stages in our previous experiments, we organized an experimental observation using the AQUA-RAT dataset (Ling et al., 2017). This dataset comprises complex reasoning questions designed to test the problem-solving abilities of advanced language models. Each question is presented with multiple-choice answers ranging from ['A', 'B', 'C', 'D', 'E']. We focus on a specific problem from the dataset to illustrate the impact of team formation on problem-solving dynamics.

Problem Description: A man's speed with the current is 14 km/hr, and the speed of the current is 2.5 km/hr. The man's speed against the current is:

- A) 9 km/hr
- B) 8.5 km/hr
- C) 10 km/hr
- D) 12.5 km/hr
- E) None of these

The correct answer to this problem is **A) 9 km/hr**.

G.1 Agent Responses and Reasoning

In the initial interaction rounds, the agents provided the following responses and reasoning:

- **1-ToM Agent (Incorrect):** Answered 11.5 km/hr, calculating the man's speed in still water as 14 km/hr - 2.5 km/hr, but incorrectly concluding that this was the speed against the current.
- **2-ToM Agent (Correct):** Answered 9 km/hr, correctly adjusting the man's speed against the current to 11.5 km/hr - 2.5 km/hr after recalculating the man's speed in still water (11.5 km/hr).

Impact of Team Formation We analyze the impact of team formation on the problem-solving dynamics from 1-ToM Agent's perspective:

- **Without Team:** 1-ToM Agent demonstrated low receptivity to alternative calculations and maintained confidence in their incorrect approach, stating:

  "I don't see how his calculations lead to a different answer, and I am confident in my approach. Therefore, I give his answer a lower belief value of 0.2."

- **With Team:** When part of a team, 1-ToM Agent showed openness to re-evaluating the problem in light of new insights, which is crucial for effective problem-solving in collaborative environments:

  "Although I do not fully understand his idea, he might be correct in his approach. Also, considering that there are still many rounds, we can discuss and then understand each other's ideas better. So I give his answer a higher belief value of 0.5."

Key Findings and Implications This experiment highlights the impact of team formation on problem-solving dynamics. The key findings and implications are as follows:

- Team formation prompts willingness to engage in cooperation, as evidenced by 1-ToM Agent's increased belief alignment score. 1-ToM agents consider alternative perspectives and re-evaluate the problem when part of a team.
- Team formation sets the stage for enhanced mutual understanding among teammates with varying levels of ToM, leading to deeper comprehension and more robust collaborative interactions over successive rounds. The example demonstrates that when agents are part of a team, they are more likely to engage in constructive dialogue and exchange ideas, even if they initially have different opinions or approaches.

These findings suggest that our team formation mechanism enhances effective cooperation among varying levels of cognitive capabilities, such as ToM, and improves problem-solving performance in multi-agent systems.

H Side Effects of Incorporating ToM in team formation

To investigate the impact of incorporating ToM capabilities into the Dynamic LLM-Agent Network (DyLAN) framework (Liu et al., 2024) and ChatEval (Chan et al., 2023), we conducted experiments using three datasets: HUMANEVAL (Chen et al., 2021) for **coding tasks**, AQUA-RAT (Ling et al., 2017) for **logic problem solving**, and MMLU (Hendrycks et al., 2021) for **multi-task scenarios**. Specifically, for HUMANEVAL , the agent roles include 'PythonAssistant', 'AlgorithmDeveloper', 'ComputerScientist', and 'Programmer', while the judge roles consist of 'Passer', 'Tester', 'Reflector', and 'Ranker'. For MMLU, the agent roles include 'Economist', 'Doctor', 'Lawyer', 'Mathematician', 'Psychologist', 'Programmer', and 'Historian'. We use the Important Scores defined in the DyLAN framework as evaluation metrics, which capture the agents' performance and effectiveness in their respective roles. Similarly, we employ *accuracy* as the primary metric in ChatEval experiments to compare performance under ToM integration. Specifically, we compare the performance of baseline DyLAN and ChatEval agents without ToM capabilities to agents equipped with 2-level ToM (+2-ToM).

H.1 DyLAN: Code Generation and General Reasoning

Table 8: HUMANEVAL : Important Scores for agents with and without ToM capabilities

	PythonAssistant	**AlgorithmDeveloper**	**ComputerScientist**	**Programmer**
DyLAN	0.2399	0.2521	0.2523	0.2557
+2-ToM	0.1707	0.2510	0.2515	0.2269
% Change	-28.83%	-0.42%	-0.32%	-11.25%

Results. The experimental results for the HUMANEVAL are presented in Table 8. Incorporating 2-level ToM into the DyLAN agents led to a decrease in performance across all agent roles. The 'PythonAssistant' experienced the most significant drop of 28.83%. These findings suggest that adding higher-level ToM to agents in the DyLAN framework does not necessarily lead to improved performance, highlighting the complex interplay between cognitive abilities and cooperative performance.

Besides, Table 9 presents the results for the MMLU dataset, focusing on the roles with the highest importance scores for each task. We observe that incorporating 2-level ToM consistently leads to a decrease in the importance scores across all tasks. The most significant drop is observed for the 'Mathematician' role in the `abstract_algebra` task, with a 25.2% decrease in the importance score. Interestingly, the accuracy of the agents with ToM capabilities also decreases for most tasks, with the exception of the `us_foreign_policy` task, where the accuracy remains unchanged. These results align with our earlier findings from HUMANEVAL , further demonstrating that higher-level ToM does not necessarily improve multi-task performance or cooperation.

Table 9: MMLU: Important Scores for agents with and without ToM capabilities

	Role with Highest Importance Score			Role + 2-ToM	
Task	**Role**	Score	Task's Acc	Score	Task's Acc
`us_foreign_policy`	Economist	0.20	0.83	0.16 (-18.5%)	0.83 (0%)
`human_sexuality`	Lawyer	0.19	0.84	0.19 (-4.2%)	0.82 (-2.8%)
`international_law`	Economist	0.20	0.81	0.19 (-4.1%)	0.79 (-3.1%)
`abstract_algebra`	Mathematician	0.29	0.59	0.22 (-25.2%)	0.51 (-13.6%)

H.2 ChatEval: Logic Problem-Solving and General Reasoning

To further evaluate the impact of incorporating ToM, we conducted experiments with the ChatEval framework on two specific tasks: **Logic Problem-Solving** from the AQUA-RAT dataset (Ling et al., 2017) and **General Reasoning** from four subjects of the MMLU dataset (abstract algebra, college chemistry, international law, and U.S. foreign policy). The results for ChatEval are summarized in Table 10. Notably, the performance dropped when ToM capabilities were introduced. For the logic problem-solving task, the accuracy for `gpt-3.5-turbo` agents dropped by 0.64%, and for `gpt-4o-mini` agents, it dropped by 5.09%. Similar trends were observed in the general reasoning task, with a 3.435% drop for `gpt-3.5-turbo` agents and a 2.8675% drop for `gpt-4o-mini` agents.

Table 10: ChatEval: Accuracy for logic problem-solving and general reasoning with and without ToM capabilities

	Logic Problem-Solving Acc(%)		**General Reasoning Acc(%)**	
	`gpt-3.5-turbo`	`gpt-4o-mini`	`gpt-3.5-turbo`	`gpt-4o-mini`
ChatEval	41.40	82.80	56.82	72.77
ChatEval (w. ToM)	40.76 (-0.64)	77.71 (-5.09)	54.39 (-3.43)	69.90 (-2.87)

Results. The results from the ChatEval experiments indicate a clear reduction in accuracy when ToM agents are introduced, consistent with the findings from DyLAN. In the case of logic problem-solving, the performance degradation was particularly notable for `gpt-4o-mini` agents, with a 5.09% decrease. This suggests that simply incorporating ToM into the ChatEval framework, like in DyLAN, may hinder rather than enhance agent cooperation and problem-solving effectiveness.

Conclusion. Both DyLAN and ChatEval experiments provide evidence that higher-level ToM does not necessarily contribute to better agent cooperation or performance in multi-agent systems. The decrease in both importance scores and task accuracy across diverse domains underscores the challenges of effectively integrating cognitive abilities such as ToM into multi-agent frameworks.

H.3 Analysis of ToM Configurations on Multi-Agent Programming

We systematically analyze how Theory of Mind (ToM) configurations affect multi-agent programming performance across 5 rounds, using `gpt-4o-mini` on HUMANEVAL and MBPP. Experiments compare 6 PM/Engineer (Eng) ToM pairs, with/without our team formation mechanism.

Performance Degradation with ToM. By Round 5 with random team formation, we observed performance deterioration across all configurations as detailed in Table 12. Without team formation, all configurations exhibit performance decay by Round 5.

Enhanced Performance Through Team Formation. Our team formation mechanism demonstrates significant improvements in performance across different ToM configurations as detailed in Table 13. The results show that proper team formation can effectively leverage varying ToM capabilities. Most notably, when the Project Manager has higher ToM (ToM=2) cooperates with Engineers, we observe substantial performance gains of 12.9% and 26.2% on HumanEval and MBPP respectively. This finding supports that cognitive abilities like ToM, when combined with appropriate team formation mechanisms, can enhance multi-agent cooperation. Furthermore, configurations with PM ToM=1 also show considerable improvements (up to 20.0% on MBPP), suggesting that our team formation mechanism successfully encourages reciprocal cooperation by forming teams whose beliefs and actions are well-aligned.

Table 11: Initial Pass@1 Scores (Round 1)

PM ToM	Eng ToM	HumanEval	MBPP
0	1	0.91 ± 0.02	0.56 ± 0.02
0	2	0.86 ± 0.02	0.55 ± 0.02
1	1	0.92 ± 0.01	0.58 ± 0.02
1	2	0.91 ± 0.02	0.58 ± 0.02
1	0	0.87 ± 0.02	0.54 ± 0.02
2	0	0.93 ± 0.02	0.54 ± 0.02

Table 12: Performance Changes With different ToM settings (Round 1 $\rightarrow$ Round 5)

PM ToM	Eng ToM	HumanEval	MBPP
0	1	0.91 $\rightarrow$ 0.84 ($\downarrow$7.7%)	0.56 $\rightarrow$ 0.47 ($\downarrow$16.1%)
0	2	0.86 $\rightarrow$ 0.89 ($\downarrow$3.5%)	0.55 $\rightarrow$ 0.49 ($\downarrow$10.9%)
1	1	0.92 $\rightarrow$ 0.84 ($\downarrow$8.7%)	0.58 $\rightarrow$ 0.47 ($\downarrow$19.0%)
1	2	0.91 $\rightarrow$ 0.84 ($\downarrow$7.7%)	0.58 $\rightarrow$ 0.47 ($\downarrow$19.0%)
1	0	0.87 $\rightarrow$ 0.86 ($\downarrow$1.1%)	0.54 $\rightarrow$ 0.50 ($\downarrow$7.4%)
2	0	0.93 $\rightarrow$ 0.91 ($\downarrow$2.2%)	0.54 $\rightarrow$ 0.49 ($\downarrow$9.3%)

Table 13: Performance with Team Formation Compared with Baseline (Round 5): the data is the pass@1 score of the last round (Round 5) with the proposed team formation, and the percentage change is the difference between the proposed team formation and random team formation.

PM ToM	Eng ToM	HumanEval	MBPP
0	1	0.86 ($\uparrow$2.4%)	0.49 ($\uparrow$4.3%)
0	2	0.91 ($\uparrow$2.3%)	0.52 ($\uparrow$6.1%)
1	1	0.89 ($\uparrow$3.5%)	0.60 ($\uparrow$20.0%)
1	2	0.88 ($\uparrow$4.8%)	0.55 ($\uparrow$17.0%)
1	0	0.89 ($\uparrow$3.5%)	0.60 ($\uparrow$20.0%)
2	0	0.96 ($\uparrow$5.5%)	0.62 ($\uparrow$26.2%)