

Faculty of Sciences

Department of Computer Science



UNIVERSITÉ LIBRE DE BRUXELLES

MASTER THESIS

Theory of Mind Agents Behavior in The Mod Game

Student: Grace Hanna

Academic Supervisor: Prof. Tom Lenaerts

Academic year 2024–2025

Abstract

With the rapid progress of integrating Artificial Intelligence into almost all aspects of human life, the ability for machines to understand and anticipate human behavior has become increasingly necessary. One promising direction in this context is investigating the capabilities of the Theory of Mind (ToM) in artificial agents, which is the ability to attribute mental states such as beliefs, intentions, and desires to others. Enabling machines to reason about others' mental states can significantly enhance their performance in interactive, strategic, or socially sensitive environments, either with humans or other machines. Much research in this field was conducted by simulating cooperative and competitive games using agent-based models.

This thesis work contributes to a deeper analysis of the modeling and behavior of ToM agents within the Mod Game, a decision-making environment that captures strategic interaction among multiple agents of homogeneous and heterogeneous groups, across structured experimental settings. This work extends a previous work on ToM in the Mod Game by Prof. Harmen de Weerd. This work's findings reveal that while higher-order reasoning has theoretical advantages, it is not always practical or beneficial in adaptive environments. Agents might not always manage to strike a balance between reasoning complexity and performance. At the same time, agents of low-order reasoning cannot align with complex activities or in overwhelming environments, which makes them unreliable in achieving the goal of ToM in AI systems. These insights highlight the importance of cognitive efficiency in agent design and clarify the complexity and the size of the aspects that should be considered when improving the social capabilities of artificial intelligence systems.

Keywords: Theory of Mind, agent-based model, the Mod Game, behavior analysis, competitive games.

Acknowledgments

I would like to express my deepest gratitude to Professor Tom Lenaerts for being a constant source of professionalism, guidance, and support throughout every stage of working on this thesis. Your leadership and encouragement have been truly invaluable.

I extend my gratitude to Professor Harmen de Weerd for his generous support in answering all my questions related to his research and the broader topic, as well as for providing valuable resources that greatly contributed to the completion of this work.

I would also like to sincerely thank Marco Saponara for his valuable comments and insightful feedback, which significantly contributed to improving the quality, clarity, and structure of this thesis.

I am grateful to the ULB and Daughters for Life association for believing in my potential, offering me this life-changing opportunity to pursue my studies, and empowering me through this journey.

Without the support of my family and dear ones, I would not be able to overcome all the hardships and make it to this point. Thank you from the bottom of my heart for all the love.

Computational resources have been provided by the Consortium des Équipements de Calcul Intensif (CÉCI), funded by the Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under Grant No. 2.5020.11 and by the Walloon Region.

Content Table

Abstract	i
Acknowledgments	ii
List of Figures	vi
List of Tables	vii
List of Notations	ix
List of Abbreviations	x
1 Introduction	1
1.1 What is the Theory of Mind?	1
1.2 ToM in the World of Artificial Intelligence	2
1.3 Background and Objective	3
2 State of the art	7
2.1 ToM Model Types	7
2.2 The Mod Game	9
2.3 The Mod Game Strategies	11
2.3.1 Behavior Based Strategies	11
2.3.2 Theory of Mind Based Strategies (Orders)	12
3 Methods and Configurations	15
3.1 The Model and the Game Key Factors	15

3.1.1	Key Representatives of the Model	15
3.1.2	Key Representatives of the Game	17
3.2	The Game and ToM Agents Modeling	19
3.2.1	The Game’s Representation	19
3.2.2	ToM Orders Models	19
3.2.2.1	ToM ₀ Model	19
3.2.2.2	ToM ₁ Model	22
3.2.2.3	ToM ₂ Model	24
3.2.2.4	Higher Orders Model	26
3.3	Practical Details	29
3.3.1	The Game Settings	29
3.3.2	Code Implementation	31
4	Experiments and Results	34
4.1	Visualizations Reading	34
4.2	Reproduction and Validation of Prior Findings	36
4.2.1	Findings Validation Divided by ToM Order	37
4.2.1.1	ToM ₀ related theories	37
4.2.1.2	ToM ₁ related theories	38
4.2.1.3	Higher-order related theories	40
4.3	Experiments Ordered by ToM Level	43
4.3.1	Groups of ToM ₀ Agents	43
4.3.2	Groups of ToM ₁ Agents	55
4.3.3	High-Order ToM Agents	62
4.3.3.1	Groups of ToM ₂ Agents	62
4.3.3.2	Groups of ToM ₃ Agents	69
4.3.4	Groups of Different ToM Orders	72
4.3.4.1	Comparison and Interpretation	81
4.3.5	Common Observations Among Different ToM Orders	81
4.3.6	Summary	84

5 Discussion and Concluding Remarks	87
5.1 Discussion	87
5.2 Future Work	89
5.3 Conclusion	89
A ChatGPT interventions in this thesis	91
References	93

List of Figures

2.1	The Mod Game Design	11
3.1	Actions' Likelihood assigning and selection process	21
4.1	Choices distribution visualization	34
4.2	Payoffs distribution visualization	35
4.3	Five-agent ToM ₀ group simulation	38
4.4	Isolated choice pattern of ToM ₀ agent	38
4.5	Distribution of choices made by five ToM ₁ agents	39
4.6	Isolated choice pattern of ToM ₁ agent	39
4.7	Choices distribution of five ToM ₁ agents	40
4.8	Choices distribution of five ToM ₃ agents	40
4.9	Isolated choice pattern of ToM ₂ agent	41
4.10	Decision pattern of a ToM ₃ agent over 100 rounds	42
4.11	Two ToM ₀ agents' choices distribution - Identical LS Experiment	44
4.12	Three ToM ₀ agents' choices distribution - Identical LS Experiment	45
4.13	Four ToM ₀ agents - identical LS Experiment	45
4.14	ToM ₀ agents' behavior pattern over 150 rounds	48
4.15	ToM ₀ agents' behavior pattern over 150 rounds	48
4.16	Four-agent ToM ₀ choices' pattern	51
4.17	Behavior patterns of three ToM ₀ agents	51
4.18	Agents' behavior and payoff distribution	54
4.19	Two ToM ₁ agents' choices distribution - Identical LS Experiment	56
4.20	Three ToM ₁ agents' choices distribution - Identical LS Experiment	56

4.21	Behavior differences between ToM ₀ & ToM ₁ agents	58
4.22	Behavior differences between ToM ₀ & ToM ₁ agents	58
4.23	Four-agent ToM ₁ group - Random LS Values	62
4.24	Learning Speed categorized groups	65
4.25	Total payoff scores of three ToM ₂ agents	66
4.26	Total payoff scores of three ToM ₂ agents	66
4.27	Total payoff comparison among five ToM ₂ agents playing the game for 150 rounds	68
4.28	Comparative performance of ToM ₂ & ToM ₃ agents under identical simulation settings	70
4.29	Total scores achieved by five ToM ₃ agents	70
4.30	Comparison of choices made by ToM ₀ , ToM ₁ , and ToM ₂ agents	73
4.31	Isolated behavior of the ToM ₃ agent	73
4.32	ToM ₂ agent behavior in four-agent different ToM group	74
4.33	ToM ₃ agent behavior in four-agent different ToM group	74
4.34	All agents actions pattern over 100 rounds	75
4.35	Agents' choices over 100 rounds - No LS	76
4.36	Agents' payoffs	76
4.37	Total payoff comparison among four agents with different ToM levels	77
4.38	Choices distribution of a ToM ₃ agent - Low LS	78
4.39	ToM ₃ agent choices distribution and payoff throughout the game	79
4.40	ToM ₂ agent choices distribution and payoff throughout the game	79
4.41	ToM ₀ , ToM ₁ , ToM ₂ , and ToM ₃ agents' behavior throughout the game	80
4.42	Illustration of irregular behavior exhibited by a ToM ₃ agent	82
4.43	Illustration of irregular behavior exhibited by a ToM ₂ agent	82
4.44	Illustration of irregular behavior exhibited by a ToM ₂ agent	83
4.45	Actual used level extent - All ToM orders	84

List of Tables

4.1	Tested ToM ₀ Agent Configurations	44
4.2	Experimental Results of Zero-Order ToM Agents for High-High Configuration	46
4.3	Experimental Results of Zero-Order ToM Agents for Low-High Configuration	49
4.4	Experimental Results of Zero-Order ToM Agents for Low-Mid-High Configuration	52
4.5	Extra experiment on a four-agent ToM ₀ group	53
4.6	Experimental Results of First-Order ToM Agents for High-High Configuration	59
4.7	Experimental Results of First-Order ToM Agents for Low-High Configuration	60
4.8	Experimental Results of First-Order ToM Agents for Low-Mid-High Configuration	61
4.9	Results of two-agent ToM ₂ simulations	63
4.10	Results of games played by three ToM ₂ agents of different learning speed values	65
4.11	Results of extended experiments on three-agent ToM ₂ group	65
4.12	Four ToM ₂ agents simulation's results	67
4.13	Five ToM ₂ agents simulation's results	68
4.14	Results of two-agent ToM ₃ simulations	71

List of Notations

i, j	(i) denotes the main agent, (j) denotes the opponent
$\Phi_{(.)}$	The Value Function
$t^*_{(.)}$	Decision Function
$b_{(.)}$	Beliefs
U	Belief integrating function
$C_{(.)}$	Agent's confidence
$\lambda_{(.)}$	Learning Speed
$\pi_{i,j}$	Payoff Function
T	Transition Function
$\mathcal{N} = \{i, j\}$	The set of agents
\mathcal{S}	The set of possible states of the game
s'	Future game state
$\mathcal{A} = \mathcal{A}_i \times \mathcal{A}_j$	The set of possible actions
$\hat{a}_{i,j}^{(.)}$	Action prediction
$\tilde{a}_{(.)}$	Actual action
k	Drift parameter
P	Choice probability
l	Level of order
r	Rounds (number of rounds)
n	Number of players
m	Number of choices (a range in [1, ..., m])

List of Abbreviations

You will find here the explanation of all the abbreviations (letters, words, etc) used in this thesis.

ToM Theory of Mind

ToM₀ Zero-Order Theory of Mind

ToM₁ First-Order Theory of Mind Agent

ToM₂ Second-Order Theory of Mind

ToM₃ Third-Order Theory of Mind

ToM_{*l*} Theory of Mind of order/level *l*

LS Learning Speed

Chapter 1

Introduction

1.1 What is the Theory of Mind?

The first mention of the term 'Theory of Mind (ToM)' in Science and its first formal definition was introduced by David Premack and Guy Woodruff in 1978 [1]; says an individual has a Theory Of Mind if they can attribute mental states to themselves and others, it is called Theory since these states (i.e., intentions, beliefs, thinking, etc) are not directly observable. Before that, ToM was only discussed in Philosophy. Through their experiments on an adult chimpanzee, they tried to determine whether chimpanzees could attribute mental states (e.g., intentions, beliefs, and desires) to others. The chimpanzee was shown multiple videotaped scenes of a human struggling with different problems; some were simple (i.e., food out of reach), and some were more complex (i.e., the human actor is unable to free himself from a locked cage). Several photographs containing one solution to each problem were given to the chimpanzees to choose from. Due to the chimpanzee's repeated selection of the correct pictures, the authors claimed that the chimpanzee's behavior came from his recognition of the videotape as a problem, understanding the human's intention, and choosing options aligned with that intention. Thus, the chimpanzee solved those problems by imputing at least two states of mind to the human actor: intention or purpose on the one hand, and knowledge or belief, on the other hand.

The proposed advantage for chimpanzees to have ToM abilities lies in enabling those ToM chimpanzees to interpret others' behavior, predict their actions and possible outcomes, and make sense of life-observed situations.

Later, Wimmer and Perner (1983) [2] conducted experiments on humankind; children of different ages, to discover their reasoning abilities about situations and others. They examined children's understanding of one person's belief about reality, which was referred to as first-order false beliefs. They found out that children around the ages of 4 to 6 years use this level of thinking. In addition to the idea that others may hold beliefs different from one's own, which was referred to as second-order reasoning, they stated that children around age 7 or 8 are capable of using this more complex reasoning capacity [3]. However, other researchers also confirmed that claim, saying that a child has immediate access to their mental states but starts forming others' mental states around the age of 4 years old, through social interactions, observing others' behavior, and understanding their thinking [4].

It was proven through different studies that having ToM abilities equips individuals with the required cognitive tools for successful social interactions and understanding others' perspectives. On the other hand, the lack of ToM ability, which includes struggling to understand others' behavior, inability to impute beliefs to others, inability to predict their future actions, etc, causes crucial difficulties. Another research compared the abilities of autistic and neurotypical children [5], they found that the group of autistic children failed to employ a ToM, which causes a disorder in understanding and coping with the social environment, resulting in social ineptness.

1.2 ToM in the World of Artificial Intelligence

The concept of ToM in AI was first explicitly discussed in a landmark paper by Yoav Shoham's work on "Agent-Oriented Programming" (1993) [6]; you can notice that ToM in AI is a renewed field. That paper discussed ToM concepts (i.e., beliefs, choices, mental state, etc) and how agents in multi-agent systems could benefit from reasoning about other agents' mental states. Still, the term 'Theory of Mind' was not mentioned directly in this research. However, many papers and articles following this one talked about ToM and its importance in AI systems [7] [8] [9] [10].

Why ToM is crucial in AI systems?

The ToM feature enables AI systems to reason about the mental states, intentions, and beliefs of other agents (human or artificial), which is a foundation for many systems involving interaction,

prediction, and cooperation. For example, in Multi-Agent systems (e.g., robotic environments), ToM allows each agent to model the goals, beliefs, and strategies of others, leading to more sophisticated and adaptive behavior [11]. In Human-Computer Interaction environments like Personal Assistants, ToM enables the system to understand users' context and intent, making interactions more natural and intuitive [12]. Moreover, in Autonomous Systems, such as Self-Driving Cars [10], ToM is critical for safety, efficiency, and seamless integration with human-operated systems. If the Self-Driving Car systems do not have the ToM feature, they would not be able to recognize that a cyclist might turn or understand when a driver is changing lanes.

Without enabling ToM abilities in AI systems, they cannot infer goals or intentions, make predictions, correctly interpret the surroundings and situations, or navigate the social and interactive aspects of multi-agent systems and human-agent interactions, making it harder for humans to trust or work with those systems [8] [13].

In order to create reliable systems that mirror competitive [14] [15], cooperative [16] [17], or mixed real-life environments [18], many experiments that simulate ToM in artificial systems using agent-based models were conducted. Those studies span game theory, social learning, multi-agent reinforcement learning, etc.

1.3 Background and Objective

This thesis focuses on investigating an existing agent-based model by Prof. de Weerd from his research on the interactions between humans and artificial systems. In his wider research named 'If you know what I mean' [19], Prof. de Weerd worked on examining the development of the ToM in humans and its existence in artificial agents. Also, he explored the potential advantages of using higher ToM orders¹ through agent-based models in multiple game settings, including competitive games like Rock-Paper-Scissors and its variations, The Mod Game, cooperative games like Tacit Communication Game, and negotiation scenarios like Colored Trails. The paper distinguishes different theoretical perspectives on the ToM and its orders' functionality and efficiency in artificial agents. It also discusses how computational models highlight human behavior in social interac-

¹**Higher ToM Orders:** ToM orders start from order zero until higher orders that can be unlimited. For instance, a ToM of order zero (ToM_0) means no reasoning about the other's mental state. (ToM_1) denotes one level of reasoning. (ToM_2) denotes two levels of reasoning, and so on for higher levels of ToM.

tions and strategic games. Prof. de Weerd's work [19] investigated the emergence of the cognitive abilities of higher-order ToM in humans by discussing three main hypotheses: the Machiavellian intelligence hypothesis, the Vygotskian intelligence hypothesis, and the Mixed-Motive interaction hypothesis.

By simulating those games and comparing the results to empirical data on humans, the findings suggest that the higher ToM orders offer significant advantages in mixed-motive interactions involving cooperation and competition, even though they also show benefits in diverse situations, which somehow explains their emergence in humans.

For this thesis, one of the simulated games from Prof. Harmen's research was chosen to expand and further understand the agents' behavior. The Mod Game (see section[2.2]) is a generalization of the rock-paper-scissors game. The game was first introduced by Frey and Goldstone [20], who studied complex collective dynamics in human higher-level reasoning by simulating the game with human participants as well as artificial agents. Prof. de Weerd and his colleagues in their specific study on the Mod Game [21] executed experiments using their agent-based model [22] to examine strategic decision-making in competitive games (i.e., the Mod Game) and investigated the extent to which higher-order ToM can enhance an agent's performance in such games. Then, the results were compared with human participants' results of playing the game under similar conditions from Frey and Goldstone's [20].

Prof. de Weerd claimed that their agent-based model is different from previous similar models in the sense that the zero-order agents change their actions based on their observation of other agents' behavior [21], where they learn the behavior of other agents through repeated games. The experiments were conducted on multiple ToM order groups, each consisting of five agents. Each simulation of the game was played for 100 repeated rounds. The key measures that were used to compare agents' outcomes with humans' were choice, rate, and acceleration; where **choice** represents the agents' choices distribution, while the **rate** is the rate at which agents and participants change their choices, and **acceleration** represents the first difference in the rate.

Human results from Frey and Goldstone's experiments showed that most people use first- or second-order ToM, but very few use thinking orders beyond that. Prof. Harmen's experiments' results on the agent-based model suggested that agents with reasoning abilities up to ToM₂ perform well and, as for humans, going beyond that can lead to overthinking and does not result in any advantages.

What is the objective of this master's thesis?

The experiments in the main paper [21] were processed as follows: the overview results of each ToM order group (that is, the results of five agents of ToM_0 , five agents of ToM_1 , etc.) were compared to the results of the human participant groups. The primary focus of this thesis is to understand the details behind these general results and expand those experiments to examine the agents' performance under variant game settings. I am interested in finding the impact of the different game's characteristics, such as agents' learning speed, game's repeated rounds, number of agents in one group, on agents' behavior in groups of identical and different ToM orders, as well as their resulting performance (payoff).

To do that, first, a validation of the existing findings will be held, then additional structured experiments will be conducted and analyzed to get a better vision of the agents' performance and how it changes with the change of the parameters mentioned above. This will give a full image of the influence of the environment's attributes variation on different ToM agents of low and high orders and how they affect each other's behavior under various conditions. Since this is an extension of an existing work and a validation of previous results must be made, these experiments will be done depending on the functioning method of the agent-based model that was proposed in Prof. de Weerd's study [22] with some modifications to enhance the model functioning and visualize the results. In summary, this master's thesis will answer the following questions:

1. How does the learning speed affect agents' behavior in a group of ToM_0 , ToM_1 , or higher-order agents?
2. Does playing the game for a smaller/larger number of rounds affect the agents' behavior in a group of ToM_0 , ToM_1 , or higher-order agents?
3. How does the number of agents in the same group change their behavior in groups of ToM_0 , ToM_1 , or higher-order agents?
4. In a group of agents of different ToM orders, what observations on the players' behavior do we see? Are the players of higher/lower ToM orders performing better/worse?

The findings revealed that the learning speed (LS) had the greatest impact on agents' performance among all tested parameters. In low-order ToM groups, agents with high LS outperformed those

with low LS, while in higher-order ToM groups, low LS agents performed better. Agents with identical LS values consistently displayed identical behavior, regardless of their ToM level. The number of agents per group also significantly affected performance; larger groups enabled more stable strategies in specific settings. Running the game for 100 or 150 rounds with four or five agents provided a clearer view of performance gaps. In higher-order ToM groups, the agents' performance depended not only on the learning speed factor but also on other parameters alongside the LS. In mixed ToM-level groups, higher-order agents (with ToM₂ performing best) consistently outperformed lower-order agents across all settings. However, they were also more vulnerable to random behavior due to increased environmental complexity.

The remainder of this thesis is structured as follows: in the next Chapter[2] you will find explanations of the ToM model types, the Mod Game, its strategies and regulations, as well as a description of the logic of the ToM orders in general and a detailed explanation of the math behind their modeling in the Mod Game in particular. In Chapter[3], you can find a description of the logic of the ToM orders in general, a detailed explanation of the math behind their modeling in the Mod Game in particular, and some practical details about the game settings and the code implementation. Then, in Chapter[4], you will find the contribution to this topic, experiments, analysis, and results. Finally, in Chapter[5] there is a discussion of the outcomes and some suggestions for future work. An appendix[A] is also added, which presents an overview of how AI chatbots were utilized in enhancing the writing of this thesis, including specific examples and a personal reflection on the experience.

Chapter 2

State of the art

2.1 ToM Model Types

There are multiple types of ToM models, each of which works in different mechanisms and can be applied in different applications or situations. In the following, you can find an explanation of the difference between Model-free learning and Model-based learning, with a wider explanation of the Model-based one, since it is essential in this research. Christelle Langley et. al [23] provided a detailed overview of the ToM models and the challenges of each.

1. Model-free learning

Model-free learning in ToM involves mapping actions directly to rewards instead of reasoning about the mental states or intentions of other agents, and does not consider the structure of the environment. In other words, Agents adjust their strategies according to the observations of others' behavior in different situations, optimizing their behavior based on received rewards [24]. Model-free ToM is based on trial-and-error learning, mostly using reinforcement learning (RL) techniques, such as Q-learning or deep RL.

2. Model-based Learning

In general, ToM model-based learning refers to approaches where an agent behaves depending on a pre-implemented internal model that follows a specific method.

A. Theory-Theory modeling

This approach is based on an idea coming from philosophy: that people have an implicit

theory about how the human mind is structured and functioning [25]. Applying this method in agent modeling entails building a theoretical model that equips agents with beliefs about other agents' mental states, such as their desires, goals, and intentions, as well as the reasoning processes they follow. The agent then applies cognitive reasoning to predict the others' mental states and anticipate their actions. This method aligns with a commonly used framework for modeling agent-based systems: the BDI (Belief-Desire-Intention).

B. Simulation-Theory modeling

This approach is proposed as an alternative to theory-theory modeling. It suggests that an agent constructs an internal model of the mental states of other agents (beliefs, intentions, emotions, etc) using their own mental abilities as a simulation to understand others. In other words, they imagine themselves in another person's situation; thus, they can suspect what the other person might have in mind [19]. This approach represents what an agent thinks or knows about the feelings or thoughts of others. This approach is preferred from a developer's perspective due to multiple programming advantages, such as: reusing the same architecture (i.e., reasoning, decision-making, etc) for both the agent and its opponent, thus it does not require building a full theory of another mind consequently easier in implementing, as well as the ability to apply it to agent models beyond the BDI [25] since its core idea is functional; it requires an agent to internally simulate others' mental state using its own mechanisms which makes it possible for other agents' architectures (i.e., RL agents) to use simulation theory, not just the ones with explicit beliefs or intentions.

In the simulation theory approach, individuals apply the 'theory' they have about other agents' unobservable intentions to simulate their mental states and then predict their future actions based on observed behavior and contextual information over time [16]. Methods such as Bayesian inference or reinforcement learning are used to generate predictions. When this predictive strategy involves hierarchical reasoning levels, it is called **Recursive Modeling**, which represents Higher ToM Orders .

Agents in this modeling adjust their level of recursive reasoning in reaction to others' behavior. A level ' l ' agent considers others to be agents of any level from ' $l - 1$ '

up to any higher level than ' l ' based on their behavior [26]. This requires explicit representation of the mental states of others.

'Higher ToM Orders': ToM orders start from order zero until higher orders that can be unlimited. For instance, a ToM of order zero (ToM_0) means no reasoning about the other's mental state. ToM of order one (ToM_1) denotes one level of reasoning, and ToM of order two (ToM_2) denotes two levels of reasoning, and so on for higher levels of ToM. The following is an example of using the recursive ToM in a chess game, which clarifies how the orders work.

Assume that two players (e.g., Sara and Jad) are playing chess. If Sara plays using her zero-order abilities (ToM_0), she depends on her own strategy without considering what her opponent (Jad) might do. If she sees that taking a piece is good for her, she captures it without thinking about Jad's response. Later during the game, she starts to reason about Jad's thoughts (ToM_1) and observes what he is doing. For example, if Jad moved his knight to a specific spot, she would think that maybe he was setting up an attack, so she would think she should defend her queen. When the game gets to a critical level, she might start using her second ToM order (ToM_2), which means reasoning about what her opponent (Jad) thinks she is thinking. Therefore, she would say: "He knows I see his attack coming, so he might tend to set a trap. Thus, to surprise him, I will make a move he does not expect instead of defending normally". As noted above, each level adds more strategic depth to the decision-making process.

2.2 The Mod Game

As mentioned earlier, the Mod Game was first introduced by Frey and Goldstone [20]. It is a non-zero-sum version of the Rock-Paper-Scissors game. Non-zero sum games are games where if one player wins, it does not necessarily mean the other player has to lose [27].

The Mod Game (see Figure[2.1]) consists of n players, and each has to choose a number from the range of choices $[1, \dots, m]$, where $n, m > 1$. The game can be played for ' r ' number of rounds; in each round, all players play against each other. A player earns one point when he chooses a number that is *exactly* +1 higher than another player's choice. For instance, when a player ' P_1 '

chooses number ‘3’ and another player ‘ P_2 ’ chooses number ‘4’, then P_2 will earn 1 point and P_1 will not earn any points, nor lose any points. There is only one choice exception to this rule, which is the choice of number ‘1’ that beats choice ‘ m '; i.e. if the range of choices is $[1, \dots, 24]$ then choosing number ‘1’ by P_1 beats choosing number ‘24’ by P_2 , thus P_1 will get a point.

If there are more than two players in the game (i.e., 3 players) and two of them choose the same choice that is one higher than the third player’s choice, each player gets a point. For example, we have three players (P_1 , P_2 , and P_3) where P_1 and P_2 choose the number ‘12’ while P_3 chooses the number ‘11’, then the earned points will be $[1, 1, 0]$ respectively.

In case a player chooses a number that is more than +1 higher than the other player’s choice, both players get no points. For instance, if player P_1 chooses the number ‘8’ and player P_2 chooses the number ‘13’, no points are gained by anyone.

A player gets a point for each other player they defeat. For example, if we have a group of four players (P_1 , P_2 , P_3 , and P_4) where P_1 , P_2 , and P_4 choose the number ‘20’ while P_3 chooses the number ‘21’, then P_3 will earn 3 points (payoff = $[0, 0, 3, 0]$); a point for each agent.

The name of the game (The Mod Game) refers to the player’s intention in choosing a number that ‘ $+1 \bmod m$ ’ than their opponent(s) [28]. Similarly to the Rock-Paper-Scissors game, the Mod Game has a mixed-strategy Nash equilibrium where each choice has an equal probability of being chosen by the players [20]. You can find below formal definitions of some main terms from Game Theory in order to clarify the concept of **Mixed Strategy Nash equilibrium**:

- **Pure Strategy:** is the set of actions A_i available to player i [29]. It specifies the action a_i a player takes for each situation they might face during the game.
- **Mixed Strategy:** it is a distribution on the player’s set of actions. If the action set is finite, this means that the mixed strategy is the probability that action a_i is chosen [30]. For example, in the Rock-Paper-Scissors game, a player can have a mixed strategy of 50%, 25%, or 25%, respectively, defining the random probability of each pure strategy that a player might choose.
- **Nash equilibrium:** A set of strategies and the corresponding payoffs constitutes a Nash equilibrium if each player has chosen a strategy and none of the players can get a higher expected payoff by changing their chosen strategy while the other players keep theirs unchanged [30].
- **Mixed Strategy Nash Equilibrium:** A set of mixed strategies creates a mixed strategy Nash equilibrium if no player has any other pure strategy that would deliver a better payoff than

the expected payoff from their mixed strategy when played against the mixed strategies of the other players [29].

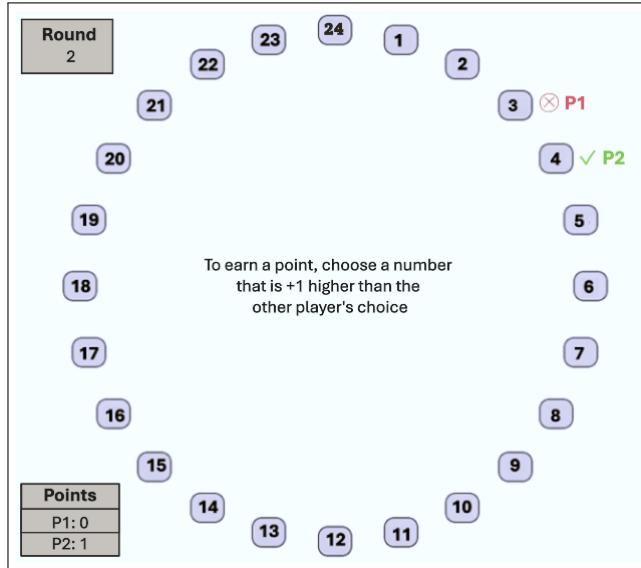


Figure 2.1: The Mod Game Design

2.3 The Mod Game Strategies

Multiple strategies can be used to play the Mod Game. Here, you can find a brief explanation of some of these strategies. They are divided into two categories: Behavior-Based strategies and Theory-of-Mind-based strategies. Agents of the used model in this research use ToM-based strategies and one type of behavioral strategy, the 'Other-regarding strategy'.

2.3.1 Behavior Based Strategies

1. **Self-regarding strategy:** In this strategy, the players ignore their opponent's actions and take the decision based on their own previous actions [28], depending on two parameters:

- Drift parameter ' kk higher modulo 24 than the previous action).
- Choice probability ' p

For example, an agent with a value of ' $k = 3$ ' selects the action that is 3 higher than its

previous choice with probability p and selects any other action with probability $\frac{1}{23}(1 - p)$.

2. **Other-regarding strategy:** contrary to the latter strategy, here the player reacts to the previous action of the opponent [28], depending on the same parameters mentioned in the self-regard strategy, the drift parameter ‘ k ’ and the choice probability ‘ p ’. For example, an agent with a value of ‘ $k = 3$ ’ selects the action that is 3 higher than its opponent’s previous choice with probability p and selects any other action with a probability of $\frac{1}{23}(1 - p)$.
3. **Win-Stay, Lose-Shift (WSLS) strategy:** The decision in this strategy is based on the result of the previous round [28]. If the agent won, they repeat their last action with a probability p , and with a probability of $\frac{1}{23}(1 - p)$ to any other action. Otherwise, if the agent lost, they would repeat their previous choice with a probability of $(1 - p)$ and with a value of $\frac{p}{23}$ for any other choice.

2.3.2 Theory of Mind Based Strategies (Orders)

1. Zero-Order Theory of Mind strategy (ToM_0)

When an agent follows the zero-order ToM strategy, this agent represents an inexperienced player who does not consider the mental states of their opponent. This type of agent cannot recognize that the opponent may have different goals or try to get a higher score. In contrast, it relies solely on its memory of past events to predict the future behavior of its opponents [21]. A ToM_0 agent tends to generate, each round, a *Likelihood* of each choice of the 24 choices that denotes the probability of choosing a specific number by other agents, and depending on those likelihood values, the agent selects the number that he thinks will maximize his score. For instance, if a ToM_0 agent recalls that its opponent frequently chose the number ‘10’ in previous game rounds, it will assume that the opponent will likely play the same choice (number 10) in the next round. Thus, the agent decides to play number ‘11’. Therefore, a ToM_0 agent’s beliefs can be seen as a linear memory updated according to the agent’s original beliefs combined with newly observed actions by its opponent.

2. First-Order Theory of Mind strategy (ToM_1)

Agents that play using the First-Order ToM strategy take into consideration their opponents' intentions to increase their score. This type of agent reasons about its opponents' mental states and beliefs and reacts accordingly [21]. To make accurate predictions, ToM_1 agent applies its way of thinking to its opponent in order to understand what she (his opponent) might tend to play in the next round [31]. The ToM_1 agent can also utilize the zero-order level, so it has two models to use throughout the game. By observing repeated interactions, the ToM_1 agent adapts its behavior to the one that best defines its opponent's behavior; if the opponent uses zero or first-order reasoning, the agent responds accordingly. For example, if the ToM_1 agent believes that most players will choose the number '7', then it should select the number '8' based on its zero-order response. After applying these thoughts to its opponents, the agent recognizes that they will be thinking the same way and, according to its first-order response, the agent concludes that playing the number '9' will maximize its score. After observing its opponent's action over repeated rounds, the agent determines whether its opponent is of a zero or first ToM order, and starts using the suitable order accordingly.

3. Second-Order Theory of Mind (ToM_2 - higher orders strategy)

A ToM_2 agent follows a strategy that is similar to that of ToM_1 , but adds one reasoning level about its opponent's intentions. Clarifying this with an example: if the ToM_2 agent believes that most players will choose the number '20', then it should choose the number '21' based on its zero-order response. As explained in the ToM_1 strategy, the agent will apply its way of thinking to its opponents, considering that they might also decide to pick the number '21'. According to its first-order response, the agent will start thinking of playing number '22' to maximize its score. The second-order of thinking allows our agent to think deeper and understand that other agents recognize its plan of playing number '22', so the agent concludes that selecting number '23' is the choice that will increase its payoff. Hence, ToM_2 or any higher-order agent, similarly to the ToM_1 agent, has more than one level of thinking in the range of $[0, \dots, \text{up to } l\text{-order}]$ ToM [31]. While playing the game, and after making a few random choices at the beginning to explore and store observations about the opponent's actions, the agent decides on the model that best suits its opponent's behavior and enables

it to outperform that opponent. Then, the agent trusts the chosen model choices throughout the game and ignores the other models' insights unless a change in the opponent's behavior is noticed, and then the agent adapts accordingly. Therefore, the agent might use multiple theory-of-mind orders during the same game's rounds. For instance, the ToM_2 agent might notice that its opponent has no ToM ($= \text{ToM}_0$), then it chooses to play as a ToM_1 agent to maximize its payoff. Later during the game, its opponent changes its behavior to act as a ToM_1 agent; hence, the agent will adapt and change its behavior to react as a ToM_2 agent.

Chapter 3

Methods and Configurations

3.1 The Model and the Game Key Factors

Several crucial elements could change the game's outcome and affect the players' performance. To expand the experiments and get deeper into analyzing the agents' behavior under different conditions, it is possible to calibrate the values of some of these elements and interpret the outcomes, in order to observe the real influence of these elements on the agents' behavior, strategies, and scores. Some of these characteristics are related to the agents' modeling (i.e., Likelihood and Learning Speed), and other parameters are related to the game itself (number of rounds and number of agents in the group). The one that is generated by the agent and cannot be modified manually is the likelihood. The other three parameters played a major role in the experiments of this research, examining the effect of their variations on the agents' behavior revealed interesting results that could not be seen otherwise. In this section, you will find an explanation of each of these elements.

3.1.1 Key Representatives of the Model

1. Likelihood, or Beliefs ($b^{(\cdot)}$)

In the context of decision-making games, the likelihood parameter is crucial for agents with ToM capabilities, as it allows them to model and anticipate their opponents' behaviors. It represents the probability that the opponent will select a particular action a_j , given the agent's thoughts about that opponent's mental state. In other words, the likelihood is the agent's

internal guess about how likely certain actions might be selected by its opponent, depending on its own simulation of the opponent's reasoning process. When an agent assigns likelihood values to actions, the following conditions are taken into consideration [22]:

$$b^{(0)}(a_j; s) \geq 0 \quad \text{for all } a_j \in \mathcal{A}_j, s \in \mathcal{S}. \quad (3.1)$$

$$\sum_{a_j \in \mathcal{A}_j} b^{(0)}(a_j; s) = 1 \quad \text{for all } s \in \mathcal{S}. \quad (3.2)$$

Equation[3.1] represents the non-negative probability that the agent assigns to their opponent for playing a specific action in a particular game state, and Equation[3.2] denotes that, for each possible game state, the total probabilities an agent assigns to all possible actions of each opponent must sum up to 1.

2. Learning Speed (λ)

The concept of the agent's learning speed (sometimes in the text it will be abbreviated to 'LS'). It controls an agent's speed in adapting its strategy based on new information during the game. In other words, the learning speed represents the weight an agent assigns to the most recent observation when updating its beliefs about its opponents' behavior. Its value varies in the range of: $\lambda \in [0, 1]$. In more detail:

- a. No Learning Speed: $\lambda = 0 \rightarrow$ No Update:

If an agent has a learning speed of 0, it ignores new information and sticks to its initial strategy that comes from its memory in all rounds.

- b. Highest Learning Speed: $\lambda = 1 \rightarrow$ Full Update:

In this case, an agent assumes that the last observation is the best predictor. Thus, it forgets everything from earlier rounds and only uses the most recent result to make the next decision.

- c. Intermediate Learning Speed: $0 < \lambda < 1$:

If an agent has an intermediate LS value, it considers the past and the most recent observations to decide its next action. The smaller the LS value is, the less weight it gives to the recent observation, and the more weight it gives to the old information, and

vice versa.

Generally, a higher learning speed allows the agent to learn faster and adapt quicker to its opponent's behavior changes. However, a higher learning speed value does not always mean a better performance; sometimes, it could cause a negative score for the agent. A low learning speed value could lead to more stable predictions since it keeps long-term information from past rounds. At the same time, the agent with a very low LS has a weak adaptation ability, which means it will take the agent a long time to adapt to its opponent's behavior changes. In the Mod Game, the goal is to predict the opponent's next choice and select the best response based on that prediction. Since we are working with agents of different ToM orders, no specific LS value is the best; it depends on the agent's and its opponents' ToM order of the same group, as well as the values of other parameters in the whole environment.

3. Theory of Mind order

Since the ToM agents have an advantage over non-ToM agents in reasoning about their opponents' mental states, this advantage differs depending on the environment, its requirements, and the order in which an agent reasons(see section[2.3.2]). Deciding on which ToM agents perform better differs between cooperative and competitive games. For instance, in cooperative games, agents have the motive to be more predictable to each other; thus, they tend to reduce the vagueness of the environment to gain more points. Therefore, researchers found that higher ToM orders' effectiveness is less than for first or second order [32]. Whereas, in competitive games like the Mod Game, players tend to be unpredictable to their zero-order opponents; thus, higher benefits to agents that reason at higher-order ToM [32].

In the next chapter, the results of the experiments will show the behavior of each ToM order agent (i.e., zero-order, first-order, second-order, etc.) when playing the game under different conditions.

3.1.2 Key Representatives of the Game

1. Number of agents

The number of agents in the game significantly affects the agents' performance. In competitive games like the Mod Game, since the agent needs to remember its opponents' past ac-

tions to optimize its individual score, increasing/decreasing the number of agents impacts the complexity of strategies, specifically the memory strategies needed, where the agent needs to remember, analyze, and adapt to each opponent's changes, then act accordingly. Although the number of agents tested so far was limited, it still obviously affects the outcome, which you will see later in the experiments.

In cooperation games, this factor is even more sensitive, where the agents need to increase their own as well as the whole group's score. In multi-agent repeated games that address real-world dilemmas, such as the iterated prisoner's dilemma, the effect of the number of agents relates to serious computational and equilibrium determination challenges [33].

2. Number of rounds

The number of rounds of a game is a key parameter in Game Theory; in cooperative and competitive games, it represents the times a game is repeated before getting the final result. In games (like the Mod Game) where the agents learn over repeated rounds to improve their behavior and increase their score, a small number of rounds could result in an unrealistic outcome of the real capabilities the agents can show otherwise. In cooperative repeated games, agents can develop strategies that promote sustainable cooperation and higher payoffs for all players, which makes the choices over time become rational according to the common rational knowledge developed by agents over rounds(i.e., Prisoner's Dilemma, Tit-For-Tat, etc) [34]. On the contrary, in some games, if the number of rounds played was higher than a specific limit, the game could be unsuccessful and cause negative scores to the players (i.e., negotiation games [26]).

In general, in the games where agents have a ToM, specifically through experiments on the Mod Game by Frey and Goldstone [20], the findings showed that the effect of the number of rounds was significant.

3.2 The Game and ToM Agents Modeling

3.2.1 The Game's Representation

The implemented model used to simulate this game is inspired by Prof. de Weerd et al. [22] research; this model considers agents that use simulation-theory modeling(see section [B.]) to win the game against their opponents. Also, our ToM_l agents use recursive modeling to simulate their opponents' thinking and predict their possible choices.

Agents' modeling, including all math equations and symbolic representations mentioned in this section, is derived from the cited paper above [22].

The game is represented as a tuple $G = \{\mathcal{N}, \mathcal{S}, \mathcal{A}, T, \pi\}$, where:

- $\mathcal{N} = \{i, j\}$ is the set of agents, where i denotes the main agent, and j denotes the opponent; either this opponent is an individual (1 agent) or a group of agents.
- \mathcal{S} : is the set of possible states of the game.
- $\mathcal{A} = \mathcal{A}_i \times \mathcal{A}_j$: is the set of the game's possible actions combinations; where \mathcal{A}_i denotes the actions of agent i , and \mathcal{A}_j denotes the set of the opponent/s' j actions.
- $T : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}'$, which is a transition function that represents the resulting state \mathcal{S}' when playing specific actions \mathcal{A} while in the current state \mathcal{S} .
- π : represents the payoff functions of the agent and its opponent: $\pi_i, \pi_j : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$.

The numeric value of each agent's payoff in each round of the game can be any value from the range of $\{0, \dots, n - 1\}$, where n is the number of players.

3.2.2 ToM Orders Models

We assume that all agents of any order do not have conflicting beliefs ' $b^{(\cdot)}$ ' of the game's rules and assignments, either about other agents or themselves. In other words, all agents are playing according to the common knowledge of the game.

3.2.2.1 ToM_0 Model

ToM_0 agents have limited abilities for using ToM; thus, they do not simulate their opponent's mental state, as explained before (see section[1]).

ToM₀ agents form beliefs $b^{(0)}$; or 'Likelihoods', as a probability of each action of the other agent's set of actions \mathcal{A}_j for each game state \mathcal{S} such that $b^{(0)}(a_j; s)$. The likelihood value is necessary for agents in decision-making games to model their opponents' possible next actions and make predictions. Depending on the conditions of equations[3.1, 3.2], the agent computes the likelihood of choosing each action from the Mod Game actions' space {1, ..., 24} by their opponent/s. An agent then computes the value of ϕ , which is a subjective utility function, that represents choosing a specific action (a_i) in a game state (s) based on the main agent's beliefs $b^{(0)}$ of their opponent's behavior. That function is named the 'Value Function' and is represented as follows:

$$\Phi_i(a_i; b^{(0)}, s) = \sum_{a_j \in \mathcal{A}_j} b^{(0)}(a_j; s) \cdot \left(\pi_i(s, (a_i, a_j)) + \max_{a' \in \mathcal{A}_i} \Phi_i(a'; b^{(0)}, T(s, (a_i, a_j))) \right) \quad (3.3)$$

To determine this value, the agent considers the likelihood of their opponent choosing action $a_j \in \mathcal{A}_j$. If the opponent plays a_j , the agent immediately receives a payoff $\pi_i(s, (a_i, a_j))$, but the game also transitions to a new state $s' = T(s, (a_i, a_j))$. By anticipating future states, the agent plans ahead, estimating the best possible value that can be achieved in s' . The value of the value function is equal to the total value of the agent's zero-order beliefs of playing an action $a_j \in A_j$ in a state s by its opponent, multiplied by the combination of:

- The immediate payoff $\pi_i(s, (a_i, a_j))$.
- The maximum future value the agent can achieve in state s' if the agent plays action a' as a future action; where $a' = a_i \in A_i$.

Thus, $\Phi_i(a_i; b^{(0)}, s)$ reflects the agent's expected outcome based on its prediction of its opponent's behavior.

Finally, the agent makes its final decision depending on the decision function t_i^* , where the agent chooses the action a_i that maximizes the value of the Φ_i function. This decision function is given by:

$$t_i^*(b^{(0)}, s) = \arg \max_{a_i \in \mathcal{A}_i} \Phi_i(a_i; b^{(0)}, s) \quad (3.4)$$

Illustrative example:

In case we have two ToM₀ agents playing the Mod Game for 100 rounds. The set of states in each

round of playing the game is $S = \{s_{current}, s'\}$, where $s_{current}$ is the current state of each round and s' is the 'end of the round' state or the future state. The action space is the range of $\{1, \dots, 24\}$. The payoffs in state s_0 of the first round are zero for all, and then they change by playing more rounds and with the payoff accumulation. In state s' , the payoff depends on the agents' choices; the agent that selects a number exactly +1 higher than the other agent will get 1 point. We simulated a game for 100 rounds between two ToM_0 agents. Two screenshots from the terminal are provided below (see Figure[3.1]) showing the process of assigning likelihood values to the actions¹ by each agent:

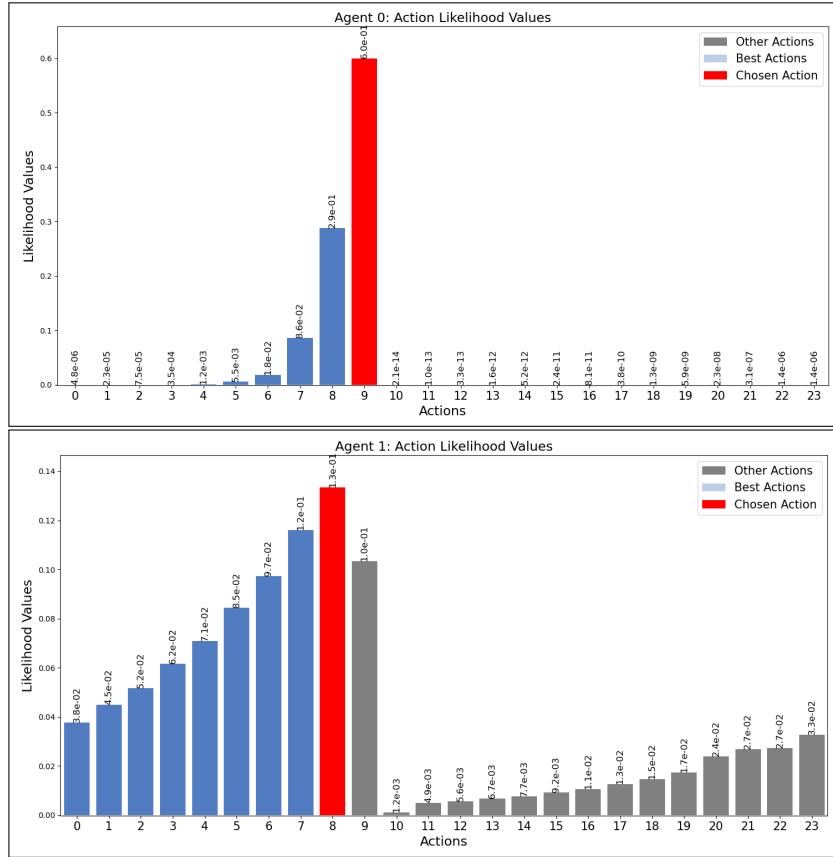


Figure 3.1: Actions' Likelihood assigning and selection process by two ToM_0 agents playing the Mod Game for 100 rounds. Each plot illustrates how an agent assigns likelihoods to all possible actions in a given round, Blue bars represent the actions that became best values through the process until the max value was assigned to action number 9 and chosen at the end of the round. Gray bars are the standard values of other actions that were not selected as the best at any point. At the top is the detailed process by agent₀, and at the bottom is by agent₁

¹**Note:** In the implementation, the indexing starts from 0, not 1. Thus, the choices range from [0,..., 23], where 0 represents the number 24. The same applies to agents, where the counter will start from zero (i.e. agent₀, agent₁, agent₂, etc)

As clarified in Figure[3.1], each agent computes the likelihood of choosing a specific action by its opponent, while computing the values, the agent updates the desired action when a new best value that it believes will maximize their payoff appears until they reach the best choice among the 24 choices set, and that would be their desired choice to pick in that round. One of the two agents, namely agent₁, made the correct prediction that enabled it to excel over agent₀ and got 1 point in that round, while agent₀ got zero.

3.2.2.2 ToM₁ Model

For the First-order ToM agents, one reasoning level is added to the agent's mental abilities in simulating their opponent's mental state. As mentioned in section[2], ToM₁ agent has two levels of ToM orders (0 & 1), thus it has two beliefs ($b^{(0)}$ & $b^{(1)}$). The ToM₁ agent decision-making process consists of 3 main stages:

1. Predict the opponent's decision \hat{a}_j depending on the agent's first-order beliefs $b^{(1)}$.

Depending on its first order, ToM₁ agent predicts the opponent's behavior using the decision function t_j^* . To do that, the agent determines the action $\hat{a}_j \in A_j$ that maximizes the value function in state s . This prediction is given in the following equation:

$$\hat{a}_j^{(1)} = t_j^*(b^{(1)}, s) = \arg \max_{a_j \in A_j} \Phi_j(a_j; b^{(1)}, s) \quad (3.5)$$

Although equations [3.4] and [3.5] look similar, in equation [3.5], ToM₁ agent makes a single prediction of the opponent's possible choice by specifying \hat{a}_j instead of assigning probability to each action of the actions' set, while in equation [3.4] ToM₀ agent determines his own behavior depending on his beliefs $b^{(0)}$. Even though having a single prediction makes it easier for the agent to compare the opponent's behavior with his prediction to validate the quality of the predictions, it also means that any slight difference between the two values will make the predictions incorrect, which might confuse ToM₁ agent between his first-order predictions' quality and its zero-order beliefs. The ToM₁ agent uses its confidence value ($0 \leq c_1 \leq 1$), which reflects its belief in the accuracy of its first-order predictions about the opponent's behavior, to determine how strongly this prediction influences its own decision-making. The question is: How is the value of c_1 determined?

When an agent plays the game against an opponent for the first time, the agent starts playing with random choices depending on random beliefs and zero confidence. After each round, the actual choices selected by the agent \tilde{a}_i and his opponent \tilde{a}_j are revealed. The agent then updates their confidence based on how accurate their predictions were in the previous round. If the agent's first-order prediction was correct, their confidence in their ToM increases. Otherwise, it decreases. The following equation describes this updating process:

$$c_1 = \begin{cases} (1 - \lambda) \cdot c_1 & \text{if } \tilde{a}_j \neq \hat{a}_j^{(1)}, \\ \lambda + (1 - \lambda) \cdot c_1 & \text{if } \tilde{a}_j = \hat{a}_j^{(1)} \end{cases} \quad (3.6)$$

λ is the value of the agent's learning speed (see section[2]).

2. Integrate the result of the previous step with the agent's zero-order beliefs $b^{(0)}$ using the belief integrating function U .

The value of c_1 enables the agent to distinguish between the different opponents' types and determines whether its opponent is ToM_0 or ToM_1 . Accordingly, the agent weighs his zero-order beliefs against his first-order predictions. This process is indicated in the belief integrating function equation below, which incorporates the agent's first-order prediction and the agent's zero-order beliefs about his opponent's behavior. If the actual action played by the opponent a_j is equal to the predicted action \hat{a}_j by the agent, then the value of the agent's integrated belief increases. Otherwise, it decreases. The equation can be described as follows:

$$U(b^{(0)}, \hat{a}_j^{(1)}, c_1)(a_j; s) = \begin{cases} (1 - c_1) \cdot b^{(0)}(a_j; s) & \text{if } a_j \neq \hat{a}_j^{(1)}, \\ (1 - c_1) \cdot b^{(0)}(a_j; s) + c_1 & \text{if } a_j = \hat{a}_j^{(1)}. \end{cases} \quad (3.7)$$

3. Make a decision depending on the decision function t_i^* .

After making the prediction and the integration process in the first two steps, we reach the last stage where the agent has to decide on the action it has to play. ToM_1 agent decides depending on his integrated beliefs U instead of his zero-order beliefs $b^{(0)}$ directly, like ToM_0 .

The decision function of ToM₁ is given as follows:

$$t_i^*(U(b^{(0)}, \hat{a}_j^{(1)}, c_1); s) = t_i^*(U(b^{(0)}, t_j^*(b^{(1)}; s), c_1); s). \quad (3.8)$$

3.2.2.3 ToM₂ Model

Two reasoning levels are added to the ToM₂ agent's mental abilities in simulating their opponent's mental state. As explained in section[3], A ToM₂ agent assumes its opponent is a ToM₁ agent, which means the opponent is modeling the ToM₂ agent as if it were a ToM₀ agent. To predict the opponent's behavior, the ToM₂ agent must simulate the entire ToM₁ reasoning process from the opponent's perspective. Thus, ToM₂ agents form three levels of belief:

- $b^{(0)}$: **Zero-Order belief** — probability distribution over opponent actions.
- $b^{(1)}$: **First-Order belief** — what the agent believes the opponent believes about *his own* actions.
- $b^{(2)}$: **Second-Order belief** — what the agent believes the opponent believes *the agent* believes about *their opponent's* behavior.

The ToM₂ agent decision-making process consists of 3 main stages:

1. Predicts the opponent's action, where the ToM₂ agent forms a second-order prediction, assuming the opponent uses first-order ToM. This results in the following nested reasoning:

$$\hat{a}_j^{(2)} = t_j^*(U(b^{(1)}, t_i^*(b^{(2)}; s), c_1); s) \quad (3.9)$$

Where:

- $b^{(1)}$: the agent's belief about what the opponent believes the agent will do.
- $b^{(2)}$: the agent's belief about what the opponent believes the agent believes about the opponent's behavior.
- t_i^* : the agent's own decision function.
- t_j^* : the opponent's simulated decision function.
- U : the belief integration function.
- c_1 : the confidence assumed by the ToM₂ agent regarding the opponent's first-order

- prediction accuracy.
2. Combines predictions into an integrated belief using the belief integration function U to blend the first and the second order predictions into an updated belief over the opponent's actions.

$$U(b, \hat{a}, c)(a_j; s) = \begin{cases} (1 - c) \cdot b(a_j; s), & \text{if } a_j \neq \hat{a} \\ (1 - c) \cdot b(a_j; s) + c, & \text{if } a_j = \hat{a} \end{cases} \quad (3.10)$$

Using the belief integration function U , the agent applies this function in two steps:

First, the ToM₂ agent integrates the zero-order belief $b^{(0)}$ with the first-order prediction $\hat{a}_j^{(1)}$:

$$\text{First integration: } b^{(0,1)} = U\left(b^{(0)}, \hat{a}_j^{(1)}, c_1\right) \quad (3.11)$$

Then, integrate the result with the second-order prediction $\hat{a}_j^{(2)}$:

$$\text{Final integration: } \tilde{b} = U\left(b^{(0,1)}, \hat{a}_j^{(2)}, c_2\right) \quad (3.12)$$

Here, $c_2 \in [0, 1]$ is the ToM₂ agent's confidence in the accuracy of its second-order prediction. The value of c_2 determines how much weight is given to this deeper reasoning in decision-making.

3. Makes a decision; the agent selects the action that maximizes its expected utility under the integrated belief \tilde{b} :

$$t_i^*(b^{(0)}, b^{(1)}, b^{(2)}; s) = \arg \max_{a_i \in A_i} \Phi_i(a_i; \tilde{b}, s) \quad (3.13)$$

Confidence and Belief Rule Update for the Second-Order Prediction

- **Confidence Update:**

After the agent observes the opponent's actual action \tilde{a}_j , it updates its confidence in the

second-order prediction c_2 as follows:

$$c_2 := \begin{cases} (1 - \lambda) \cdot c_2, & \text{if } \tilde{a}_j \neq \hat{a}_j^{(2)} \\ c_2, & \text{if } \exists : l < 2 \text{ such that } \tilde{a}_j = \hat{a}_j^{(l)} \\ \lambda + (1 - \lambda) \cdot c_2, & \text{otherwise} \end{cases} \quad (3.14)$$

Remember: The letter 'l' denotes the agent's ToM level (order).

$\lambda \in [0, 1]$ is the agent's learning rate that governs how quickly confidence and beliefs are adjusted.

This rule[3.14] increases c_2 only when the second-order prediction is correct and none of the lower-order predictions ($l < 2$) match the observed action.

- **Belief Update Rule:**

Beliefs $b^{(l)}$ are updated through exponential smoothing depending on the observed actions:

- For even l values (i.e., $l = 0, 2, \dots$), $b^{(l)}$ is updated based on the opponent's action \tilde{a}_j .
- For odd l values (i.e., $l = 1, 3, \dots$), $b^{(l)}$ is updated based on the agent's own action \tilde{a}_i .

$$b^{(l)}(a; s) := \begin{cases} (1 - \lambda) \cdot b^{(l)}(a; s) + \lambda, & \text{if } a = \text{observed action} \\ (1 - \lambda) \cdot b^{(l)}(a; s), & \text{otherwise} \end{cases} \quad (3.15)$$

This update applies only to the belief in the current game state s .

3.2.2.4 Higher Orders Model

This section generalizes the decision-making model for all higher-order ToM agents, including but not limited to ToM_3 . These agents recursively simulate their opponent's reasoning process by adding successive levels of belief about beliefs. Generally, a ToM_l agent assumes its opponent is a ToM_{l-1} agent, and uses simulation theory to model that opponent's behavior. For each increase in ToM order l , the agent:

- Introduces a new belief level $b^{(l)}$.
- Computes a new prediction $\hat{a}_j^{(l)}$ by simulating the opponent's ToM_{l-1} reasoning.
- Integrates $\hat{a}_j^{(l)}$ into the belief distribution using a confidence parameter c_l .

The structure of decision-making, belief integration, and belief/confidence updating follows a recursive and scalable pattern across increasing orders. For clarity, we present the explicit model for a ToM_3 agent as an example since it is used in this research experiment, with the understanding that higher orders extend this pattern by adding one further level of nested beliefs and predictions. A **ToM₃ agent** assumes that its opponent is a **ToM₂ agent**. This means the ToM_3 agent must simulate a reasoning chain in which its opponent simulates the agent as a ToM_1 thinker, who in turn simulates the opponent as a ToM_0 agent. Consequently, the ToM_3 agent must maintain and reason over four nested belief structures.

ToM_3 agents form and update the following belief levels:

- $b^{(0)}$: Zero-Order belief — beliefs about the opponent's actions.
- $b^{(1)}$: First-Order belief — what the agent believes the opponent believes about the agent's actions.
- $b^{(2)}$: Second-Order belief — what the agent believes the opponent believes the agent believes about her actions.
- $b^{(3)}$: Third-Order belief — what the agent believes the opponent believes the agent believes the opponent believes the agent will do.

The ToM_3 agent decision-making process consists of 3 main stages:

1. Predict Opponent's Action: The ToM_3 agent predicts the opponent's action $\hat{a}_j^{(3)}$ by simulating its full ToM_2 reasoning process. This involves two stages of nested belief modeling and two applications of the belief integration function $U(\cdot)$.

$$\hat{a}_j^{(3)} = t_j^* \left(U \left(U \left(b^{(1)}, t_i^*(b^{(2)}; s), c_1 \right), t_i^* \left(U \left(b^{(2)}, t_j^*(b^{(3)}; s), c_1 \right); s \right), c_2 \right); s \right) \quad (3.16)$$

This equation captures the full prediction process from the perspective of the ToM_3 agent. The innermost call simulates the opponent's belief that the agent is a ToM_2 reasoner. The use of c_1 and c_2 represents the ToM_3 agent's assumptions about the confidence levels used by its opponent during belief integration.

2. Integrate Predictions into Final Belief using the Belief Integration Function U : after creating its third-order prediction $\hat{a}_j^{(3)}$, the ToM_3 agent integrates this with its first and second-order

predictions. This yields an updated probability distribution \tilde{b} over the opponent's next action.

The agent applies this function in two steps:

First, integrate the zero-order belief with the first-order prediction:

$$b^{(0,1)} = U(b^{(0)}, \hat{a}_j^{(1)}, c_1) \quad (3.17)$$

Then, incorporate the second-order prediction into the result:

$$b^{(0,2)} = U(b^{(0,1)}, \hat{a}_j^{(2)}, c_2) \quad (3.18)$$

Finally, incorporate the third-order prediction:

$$\tilde{b} = U(b^{(0,2)}, \hat{a}_j^{(3)}, c_3) \quad (3.19)$$

The resulting \tilde{b} is the integrated belief used for decision-making. The parameter $c_3 \in [0, 1]$ represents the ToM₃ agent's confidence in the accuracy of its third-order prediction.

3. Select the Best Action Using the belief distribution \tilde{b} from the previous step, the agent evaluates the expected utility of each possible action and selects the one that maximizes it:

$$t_i^*(b^{(0)}, b^{(1)}, b^{(2)}, b^{(3)}; s) = \arg \max_{a_i \in A_i} \Phi_i(a_i; \tilde{b}, s) \quad (3.20)$$

This decision function reflects how the agent selects its action in the current game state (s), based on recursive belief reasoning up to the third order.

Confidence and Belief Rule Update for the Third-Order Prediction

- **Confidence Update:**

Once the opponent's actual action \tilde{a}_j is observed, the agent adjusts its confidence in the accuracy of its third-order prediction c_3 .

The update rule ensures that c_3 increases only when the third-order prediction is uniquely successful (i.e., the prediction was correct and does not match any of the lower-levels' pre-

dictions ($l < 3$):

$$c_3 := \begin{cases} (1 - \lambda) \cdot c_3, & \text{if } \tilde{a}_j \neq \hat{a}_j^{(3)} \\ c_3, & \text{if } \exists : l < 3 \text{ such that } \tilde{a}_j = \hat{a}_j^{(l)} \\ \lambda + (1 - \lambda) \cdot c_3, & \text{otherwise} \end{cases} \quad (3.21)$$

- **Belief Update Rule:**

Each belief $b^{(l)}$ is updated after each game round based on whether it models the opponent's behavior or the agent's own. The update follows exponential smoothing and applies only within the current game state s .

- For even l values (i.e., $l = 0, 2, \dots$): is updated based on the opponent's action \tilde{a}_j
- For odd l values (i.e., $l = 1, 3, \dots$): is updated based on the agent's action \tilde{a}_i

$$b^{(l)}(a; s) := \begin{cases} (1 - \lambda) \cdot b^{(l)}(a; s) + \lambda, & \text{if } a = \text{observed action} \\ (1 - \lambda) \cdot b^{(l)}(a; s), & \text{otherwise} \end{cases} \quad (3.22)$$

To clarify this idea: for instance, if the agent is a ToM₃ agent (i.e., maintains $b^{(0)}, b^{(1)}, b^{(2)}, b^{(3)}$), then the belief updates follow this structure:

- $b^{(0)}$ and $b^{(2)}$ are updated based on the opponent's action \tilde{a}_j .
- $b^{(1)}$ and $b^{(3)}$ are updated based on the agent's own action \tilde{a}_i .

3.3 Practical Details

3.3.1 The Game Settings

The Mod Game is a non-zero-sum version of the Rock-Paper-Scissors game, as mentioned in section[2.2], but with a wider action space. There are key parameters (see section[3.1]) whose values can be adjusted to examine their influence on the agents' behavior within the game. The following section outlines the specific values assigned to each of these parameters in the implementation used for this thesis.

- **Number of agents (number of players in one group):** The Mod game can be played, at

minimum, by two players. The max limit is up to the highest number ' m ' of game choices, in our case it is ' $m = 24$ ', thus the players' number space is of the range $\{2, \dots, 24\}$. In the experiments, various group sizes were tested; the game was run for each ToM level with groups of two, three, and four agents. Additionally, groups of five agents were used to replicate and validate the findings reported in the main study by Prof. de Weerd et al [21].

- **Number of rounds:** The game can be repeated for an optional number of rounds. The experiments on the Mod Game, by Prof. de Weerd [21], were simulated for 100 rounds. In this thesis's experiments, three different values (a smaller value, 50 rounds, and higher values, 100 and 150 rounds) are used in simulating the game to see in more depth the impact of ' r ' repeated rounds on the evolution of the agents' behavior. A higher number of rounds was tested aside and showed that the results of playing the game for more than 150 rounds (i.e., $r = 200$ or 300) do not reveal any new observations other than the ones shown in 150 or fewer tested rounds of repetition.
- **Learning Speeds:** The agent's learning speed (see section[2]) influences the way an agent learns and updates their beliefs through the rounds. The learning speed parameter plays an essential role in changing the agents' behavior, thus changing the choices an agent might take and even the ToM order an agent uses while simulating their opponent's mental state. The learning speed has values in the range $(0 < \lambda < 1)$. The learning speed of each agent can be different from or identical to that of other agents in the group. For example, a group of 5 agents playing the game can have either random learning speed values (i.e., $[0.2, 0.5, 0.7, 0.1, 0.4]$) or identical values (i.e., $[0.2, 0.2, 0.2, 0.2, 0.2]$).
- **The main condition of the game:** In the Mod Game (see section[2.2]), the only way for a player to earn a point is when they choose a number that is exactly +1 higher than another player's choice. The only exception to this rule is the choice of number '1' that beats choice ' m ', which is the highest value of the action space. Since the action space of the Mod Game is $[1, \dots, 24]$, then $m = 24$, and choosing number '1' by an agent beats choosing number '24' by its opponent. The agent who makes a good choice that is +1 than other choices gets 1 point for every agent who chose the number that is 1 lower.

3.3.2 Code Implementation

To conduct the experiments, I extended the model from Prof. de Weerd's research on the Mod Game [21]. The code is in Java.² It can only be shared for further extending research and under the permission of Prof. de Weerd. To better understand the game's implementation and the agent's logic, two pseudocode snippets are provided: one illustrates the logic behind the player's behavior, and the other describes the sequence of steps in the game's execution.

1. The player's logic, including their beliefs updating, decision-making process, and interaction with the game environment:

- **Reset Player (Lines 1–2):**

This step resets all internal player variables. “Reset the memory state” refers to clearing any previously stored beliefs, payoffs, or choices made in earlier simulations. It ensures that each simulation is independent.

- **Initialize Player (Lines 3–7):**

Each player is assigned a unique ID and linked to the current game instance to access shared parameters. Initial beliefs represent the agent's subjective probability distribution over the opponent's possible actions. These beliefs are typically initialized with random values and then normalized so that they form a valid probability distribution (i.e., all values are between 0 and 1 and sum up to 1). Learning speed (λ) controls how much the player adapts its beliefs in response to observed actions, it can be adjusted in the code as desired.

- **Play a Round (Lines 8–21):** Each round, the player must decide which action to take.

It first initializes an empty list to store the most favorable actions (best actions). In each state, it evaluates the value of all actions [0, 23] (see section [3.2.2.1]), one action at a time. This value reflects the expected payoff of taking that action. The agent compares values; if the evaluated action has a value that is better than the current best, the player updates the best value and resets the list of best actions to only include this one. If the

²**Note:** In the implementation, indexing begins at zero rather than one. Thus, all values shown in the results, figures, and examples follow zero-based numbering. The action space spans from 0 to 23 instead of 1 to 24; number zero effectively represents action 24. Similarly, a game of 50 rounds will be indexed from 0 to 49. Agents are also indexed starting from zero; they will be referred to as $agent_0$, $agent_1$, .. rather than starting from $agent_1$.

value is equal to the current best, the action is added to the list (see example [3.2.2.1]).

Finally, one action is selected at random from the list of best actions.

- **Observe Opponent's Action (Lines 22–28):**

After each round ends, the choices made by all players are revealed. The player updates its beliefs for each possible action. The memory state is updated to store this new information.

- **Repeat (Line 30):**

Steps 3 and 4 are repeated for all rounds of the game.

Algorithm 1 Player Logic

```
1: 1. Reset Player: reset the memory state
2: 2. Initialize Player
3:   Set player ID
4:   Set game instance
5:   Initialize beliefs (randomized and normalized)
6:   Set learning speed
7: for each round in the game do
8:   3. Play a Round
9:     Initialize list of best actions
10:    for each possible action do
11:      Calculate the value of the action based on beliefs and payoffs
12:      if value > current best value then
13:        Update best value
14:        Clear list of best actions and add this action
15:      else if value = current best value then
16:        Add this action to the list of best actions
17:      end if
18:    end for
19:    Randomly select one of the best actions
20:    Return the chosen action
21: 4. Observe Opponent's Action
22:   Receive the likelihoods of the opponent's actions
23:   Update beliefs based on the likelihoods and the player's learning speed
24:   for each possible action do
25:     Update belief using a weighted average of the current belief and the likelihood
26:   end for
27:   Update memory state based on observed action
28: end for
29: 4. Repeat Steps 3 and 4 for all rounds of the game.
```

2. The overall game logic, including the main conditions, loops, and processed information:

Algorithm 2 Game Logic

- 1: **1. Initialize the Game**
 - 2: Reset all players to start fresh
 - 3: Set the number of players
 - 4: Set the number of rounds
 - 5: Initialize the game instance with a payoff matrix
 - 6: Create players with specified ToM levels and learning speeds
- 7: **2. Play the Game**
 - 8: **for** each round **do**
 - 9: Initialize an array to store players' choices
 - 10: Initialize a choice distribution array
 - 11: **for** each player **do**
 - 12: Player makes a choice (play a round)
 - 13: Update choice distribution based on player's choice
 - 14: **end for**
 - 15: Store the players' choices for this round
 - 16: **for** each player **do**
 - 17: Calculate payoff using modular arithmetic:
 - 18: Add 1 to the payoff if the player beats another
 - 19: Update the player's total payoff
 - 20: Store the player's payoff for this round
 - 21: **end for**
 - 22: **for** each player **do**
 - 23: Update player's beliefs based on the choice distribution
 - 24: **end for**
 - 25: **end for**
 - 26: **3. Save Results**
 - 27: Save players' choices and payoffs to CSV files
 - 28: **4. Visualize Results**
 - 29: Generate charts to show players' choices and payoffs over time

Chapter 4

Experiments and Results

4.1 Visualizations Reading

To make understanding and reading the main visualizations that are mostly used in this thesis easier, this section provides detailed explanations of how to read each one with examples.

1. Agents' choices distribution visualization

The following scatter plot visualization illustrates the chosen actions by five players (agents) throughout 100 rounds game.

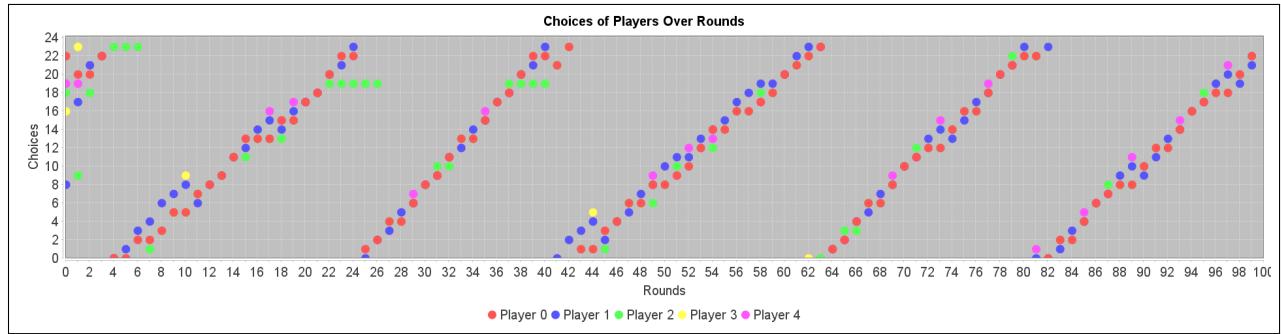


Figure 4.1: This figure shows a distribution of player choices over 100 rounds in a five-agent game. The x-axis represents game rounds (from 0 to 99), while the y-axis denotes the selected choices ranging from 0 to 23, reflecting the action space. Each player is distinguished by a unique color: Player 0 (Red), Player 1 (Blue), Player 2 (Green), Player 3 (Yellow), and Player 4 (Magenta). Each colored dot corresponds to a single player's choice during a specific round.

Interpreting the Patterns: The alignment of dots of the same color across consecutive rounds reveals each agent's behavior over time.

- A smooth ascending diagonal pattern (e.g., red or blue) indicates that the player is increasing their choice value each round, likely attempting to exhibit incremental behavior in a game like the Mod Game.
- A clustered or flat pattern would suggest repeated choices or stagnant behavior.
- When multiple dots align horizontally in a round, this suggests tied preferences. This is when you see some players' colors in the legend down the figure, but you do not notice this color on the display, because this player's choices overlap with another player's, so one player's color is hiding the other.
- The random choices at the beginning of any game is caused by the following: agents at the beginning of each game start by playing random choices at the first few rounds in order to discover their opponents' behavior, formulate their beliefs, and determines the values of some parameters they need (i.e., confidence value; c_k) then they start to have a stable/clear playing pattern.
- When the choices wrap from 23 back to 0, you may observe a sudden drop to the bottom of the Y-axis, followed by a new ascent—this is not a mistake but part of the cyclic structure of the game.

2. Agents' payoffs distribution visualization

The following scatter and line visualization displays the payoffs earned by each player (agent) over game rounds. The connected lines provide a clearer view of the fluctuations in each player's payoffs over time.

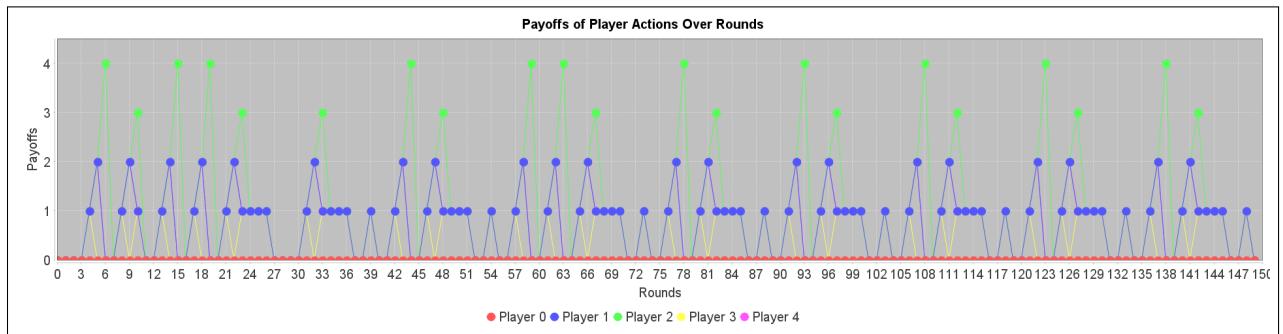


Figure 4.2: This figure shows the Payoff distribution of a five-agent group over 150 game rounds. The x-axis indicates the round number (from 0 to 149), and the y-axis represents the payoff earned by each agent per round, ranging from 0 to 4. Since the group size is five, the maximum attainable payoff in a round is 4 points. Each colored dot reflects the payoff achieved by a specific agent in a specific round, with color codes consistent with the legend: Player 0 (Red), Player 1 (Blue), Player 2 (Green), and Player 3 (Magenta).

Interpreting the Patterns: The height of the dots and lines indicates the value of the payoff.

- **Higher peaks:** for example, the green dots suggest that player 2 earned high payoffs (3 or 4 points) in those rounds.
- **Lower or flat lines:** for example, red dots indicate that player 0 earned little to no payoff (zero points throughout the game).
- **Overlapping dots:** Yellow lines indicate that player 3 earned points are hidden under the ones of player 1 (mostly 1 point). For instance, in round 4, both player 1 and player 3 each got 1 point, but the visible dot is only the blue one (player 1's dot).

4.2 Reproduction and Validation of Prior Findings

In this section, you will find a reproduction and validation of the findings and theories that were found by de Weerd et al. [21] paper that studied ToM on the Mod Game, which is the baseline of this thesis. The paper titled 'An agent-based model of strategic reasoning' studied the similarities and the differences between the agent-based model results and human-based experiment outcomes in the Mod Game. In their experiments, groups of 5 agents from different ToM orders were tested to get the agents' average *choices*; the agents' choices distribution, *rate*; the rate at which agents and participants change their choices, and *acceleration*; the first difference in the rate. They compared their results from the artificial agents to results derived from similar experiments that were conducted on human participants by Frey and Goldstone [35].

Although the same experimental settings described in the main paper were used to reproduce the results, one parameter, the agents' learning speeds, was not explicitly specified. The original work only noted that agents were assigned "random learning speed values." While random values were similarly applied in our replication, it is not possible to confirm whether these match those originally used by the authors.

Game settings from the main paper [21]:

Number of Agents = 5.

Number of rounds = 100 rounds.

Tested ToM orders: ToM_0 , ToM_1 , ToM_2 , ToM_3 .

Learning Speed = Random values (Not clearly announced).

To regenerate the results, the same settings were used along with three different sets of random learning speed values to examine whether the theories on five-agent groups would still be valid for any combination of learning speed values.

4.2.1 Findings Validation Divided by ToM Order

For each ToM order, the hypotheses from the main paper [21] will be quoted, followed by a validation or a denial depending on our reproduced results.

4.2.1.1 ToM₀ related theories

Theory 1: "The ToM₀ agent will select the number they expect to maximize their own score. For example, if they believe most will play '4', they will choose '5' to gain a point from each player. Thus, ToM₀ agents typically select the number that is one higher than the number chosen most often in the previous round."

Discussion: Observing the behavior of each agent during the game, it was indeed as described in Theory 1. All ToM₀ agents behave following the same pattern, with a difference in the exact choices related to each agent's adaptation speed to other agents, which enables some agents to outperform others. For example, the figures down below[4.4] show the agents' behavior from one of the simulations, the first figure shows the choices of five ToM₀ agents that played the game for 100 rounds. The second figure clarifies the pattern of one of the agents (agent₀), you can see how a ToM₀ agent either repeats a previous choice or chooses one number higher in the next rounds.

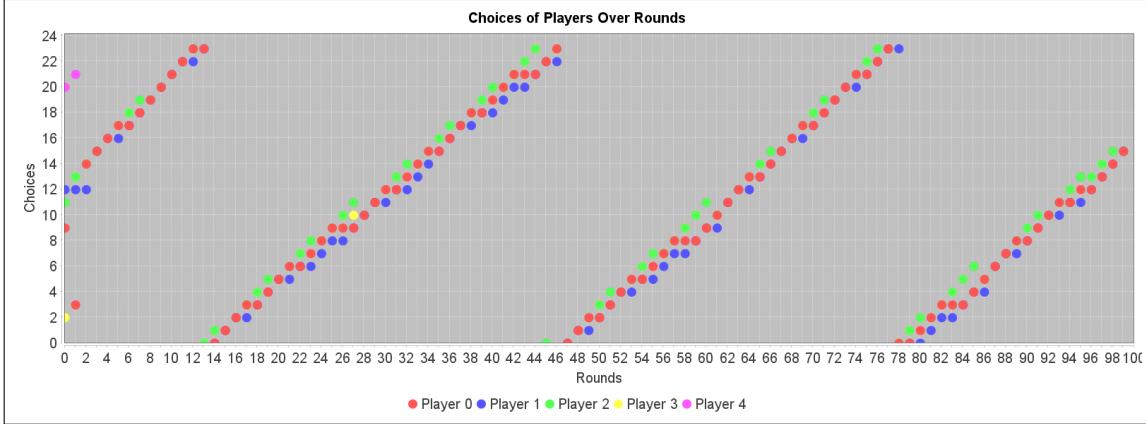


Figure 4.3: Distribution of choices made by five ToM_0 agents playing the game for 100 rounds. The agents were assigned the following learning speed values: $\lambda_0 = 0.3$, $\lambda_1 = 0.1$, $\lambda_2 = 0.9$, $\lambda_3 = 0.5$, $\lambda_4 = 0.75$. The figure highlights how all agents follow a similar progression pattern, with differences in exact choices due to individual adaptation speeds.

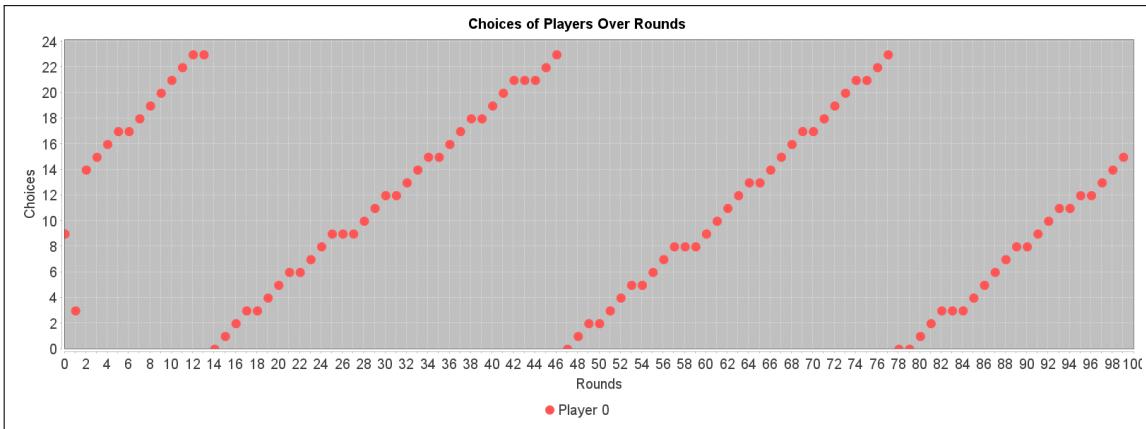


Figure 4.4: Isolated choice pattern of agent_0 from the same simulation shown in Figure 4.3. This focused view reveals the agent's strategy of either repeating its previous choice or incrementing by one, exemplifying the typical behavior of ToM_0 agents in sequential rounds.

4.2.1.2 ToM_1 related theories

- **Theory 1:** "A ToM_1 agent maintains two models of opponent behavior: one based on zero-order ToM and one based on first-order ToM . Through repeated interaction, a ToM_1 agent learns which of his models best describes the behavior of his opponents. Based on this information, a ToM_1 agent may therefore choose to play as if he were a ToM_0 agent, and ignore the predictions of his first-order theory of mind."
- **Theory 2:** "A ToM_1 agent typically selects the number that is either one or two higher than

the number that was chosen most often in the previous round.”

Discussion: The outcomes of our simulations on ToM₁ agents confirm these two theories. Looking at the agents’ choice patterns from one of the simulations (see Figure[4.5]), you can see that a ToM₁ agent varies its behavior throughout the game between reacting as a ToM₀ zero agent (repeating a choice from previous rounds or choosing a number that is +1 higher) or as a ToM₁ agents (selects a choice that is +2 higher than the previous round). For example, following agent₁ choices throughout the rounds (see Figure[4.6]), you can see that in round 17 the agent chose number 5, it repeated the same choice in the next round (ToM₀), then for the next round it selected choice number 7 (ToM₁).

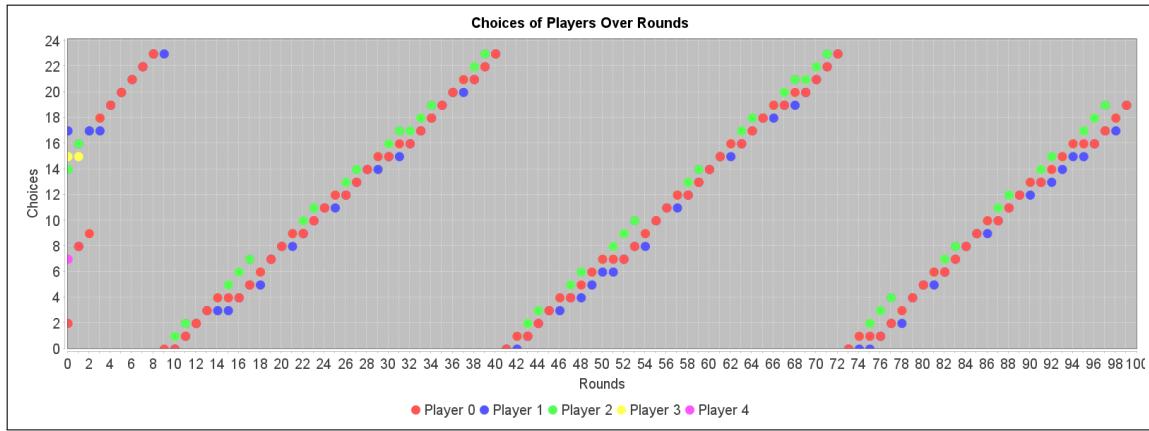


Figure 4.5: Distribution of choices made by five ToM₁ agents over 100 rounds. The agents are assigned the following LS values: $\lambda_0 = 0.3$, $\lambda_1 = 0.1$, $\lambda_2 = 0.9$, $\lambda_3 = 0.5$, $\lambda_4 = 0.75$. This figure captures the hybrid behavior of ToM₁ agents, which occasionally act based on their ToM₀ strategies (repeating or incrementing by one) or ToM₁ reasoning (choosing actions +2 ahead).

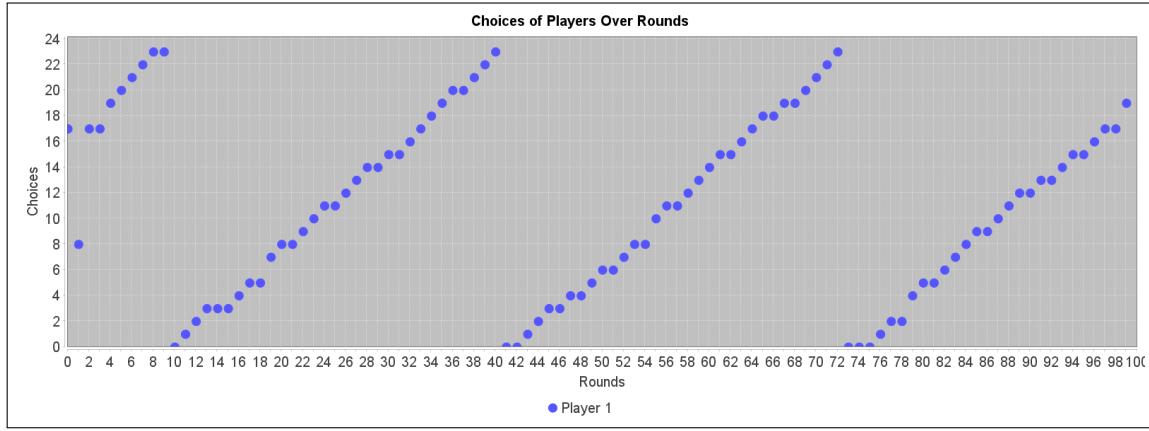


Figure 4.6: Detailed view of agent₁ choices from the same simulation shown in Figure 4.5. The plot reveals the agent’s strategic shifts between rounds: repeating previous actions, incrementing by one (ToM₀-like behavior), or selecting two steps ahead (ToM₁ behavior), exemplifying adaptive reasoning across rounds.

4.2.1.3 Higher-order related theories

- **Theory 1:** "For higher orders of theory of mind, agent behavior shows increasingly more variation."

Discussion: Indeed, playing the game under the same settings shows more variant choices for agents of higher orders than the ones of lower orders, where we see similar patterns by most agents. As an example, a simulation of the game for 100 rounds was played under the same conditions by two ToM groups of five-agent; a group of ToM₁ (see Figure[4.7]) and a group of ToM₃ (see Figure[4.8]), where the agents of both groups were assigned the same learning speed values [$\lambda_0 = 0.7, \lambda_1 = 0.9, \lambda_2 = 0.1, \lambda_3 = 0.2, \lambda_4 = 0.4$]. You can directly notice the difference in choice variations for both groups.

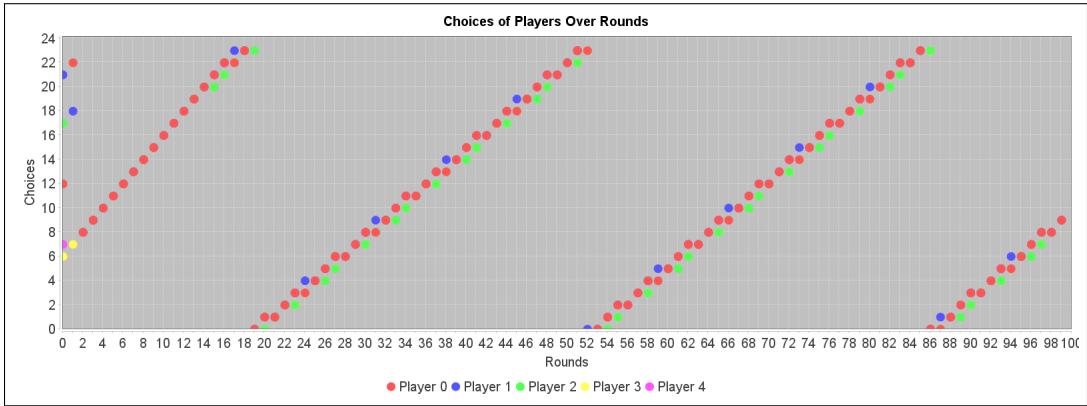


Figure 4.7: Choices distribution of five ToM₁ agents playing the game for 100 rounds. The figure illustrates the decision-making behavior typically observed among ToM₁ agents, with most agents following a nearly identical pattern of incrementing choices.



Figure 4.8: Choices distribution of five ToM₃ agents playing the game under the same conditions as in Figure 4.7. ToM₃ agents exhibit more diverse behaviors, highlighting the influence of higher-order ToM reasoning, which leads to less deterministic and more adaptive choice strategies.

- **Theory 2:** "ToM₂ agents tend to select a number that is up to 3 higher than their previous choice."

Discussion: Correct, ToM₂ agents usually tend to select a number that is +3 higher than their previous choice. Figure[4.9] below shows the behavior of one ToM₂ agent when playing the game in a group of five ToM₂ agents for 100 rounds using random learning speed values. You can see that, for example, in round number 33, the agent chose the number 7; in the next round, the agent chose the number 10. However, it does not mean that a ToM₂ agent will behave this way during the whole game. Similarly to the ToM₁ agent, a ToM₂ agent might switch between models and choose to act as a ToM₀ (repeats its choices or choose a +1 higher choice), ToM₁ (selects the choices that is +2 higher than the previous choice), or ToM₂ agent, depending on its adaptation to the opponent's changes. Moreover, it might decide to never use its second-order abilities at all during the whole game.

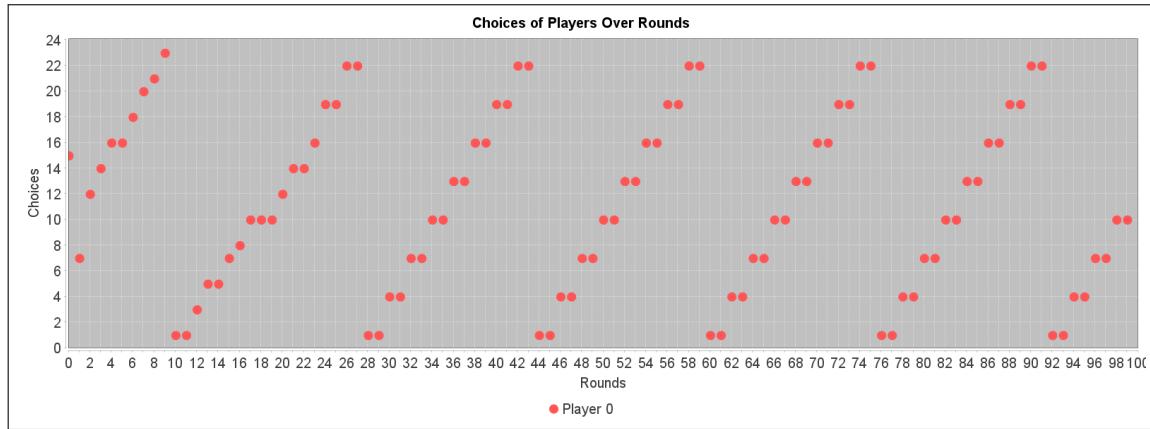


Figure 4.9: Choice trajectory of a ToM₂ agent over 100 rounds in a five-agent game. This figure demonstrates how ToM₂ agents can switch between ToM₀, ToM₁, and ToM₂ strategies depending on their adaptation to opponents' behaviors, or sometimes refrain from using higher-order reasoning entirely.

- **Theory 3:** "ToM₃ agents are also likely to select a number that is up to 4 higher than their previous choice."

Discussion: Same idea as in ToM₂, a ToM₃ agent might choose to play using different orders from its lower orders, and later you will see that a ToM₃ agent would be choosing to play using its lower orders' abilities much more than its actual order extent. In the figure below[4.10], you can see the choices' pattern of one agent in a group of five agents, assigned learning speeds of $[\lambda_0 = 0.7, \lambda_1 = 0.9, \lambda_2 = 0.1, \lambda_3 = 0.2, \lambda_4 = 0.4]$ that played the

game for 100 rounds; the agent used its third order only twice, in round 39, the agent chose number 12 to move to number 16 in the next round, which is +4 higher than its choice in the previous round. However, some other agents of this group, or even from other ToM₃ groups' experiments, never used their third-order abilities throughout the game.

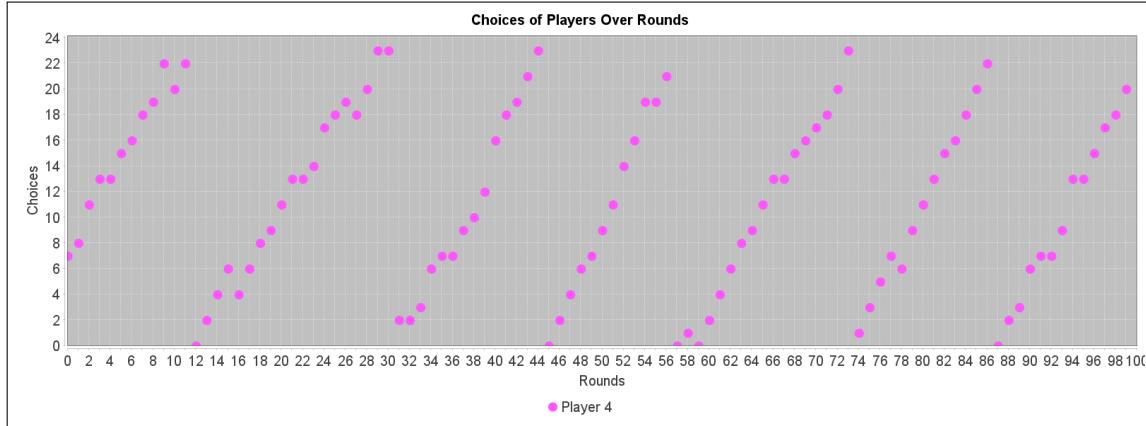


Figure 4.10: Decision pattern of a ToM₃ agent over 100 rounds in a five-agent group. This figure highlights how the agent, despite having access to third-order ToM reasoning, primarily relied on lower-order behaviors throughout the game.

4.3 Experiments Ordered by ToM Level

This section is divided into multiple subsections, each corresponding to experiments on a specific ToM order. The experiments explore the influence of various parameters on the performance of agents within each ToM group, such as the number of rounds, the number of agents playing the game, and the learning speed. The objective is to analyze how agents of each specific ToM order perform under defined conditions and to compare these performances in order to gain a thorough understanding of their similarities, differences, and the environmental factors that evoke optimal behavior from each type of agent.

Some experimental conditions were held constant across all ToM orders' experiments:

- The **number of rounds** was fixed across all tests, with each game being run for three different values: 50, 100, and 150 rounds.
- Each ToM order was evaluated using four distinct **group sizes**, consisting of 2, 3, 4, and 5 agents per group.

A large number of experiments were conducted to support the findings of this research. However, presenting every individual result would be impractical. Therefore, representative selections are provided, chosen specifically to highlight the core findings and address the main research questions. Regarding the **learning speed values**, selected configurations were applied consistently across all ToM orders to enable clear and meaningful comparisons. Detailed tables specifying the chosen learning speeds are included within each relevant subsection.

4.3.1 Groups of ToM_0 Agents

Remember: A ToM_0 agent is an agent with zero ToM that depends on its memory of past events to learn and predict the future behavior of its opponents (see section[1]).

Experiments on ToM_0 agents are divided into two parts according to the assigned learning speed values tested on the groups of agents: the first part contains experiments on groups of agents that are assigned identical learning speeds, and the second part uses random learning speeds.

1. **Part 1: Groups of identical Learning speeds:** To analyze the behavior of ToM_0 agents that have identical Learning speeds, a series of experiments was conducted across varying group

sizes and the number of repeated rounds. The game was played for 50, 100, and 150 rounds for each configuration. The tested configurations included:

Number of Agents	Learning Speed(s)
2	$\lambda_0 = \lambda_1 = 0.2$ $\lambda_0 = \lambda_1 = 0.9$
3	$\lambda_0 = \lambda_1 = 0.2$ $\lambda_0 = \lambda_1 = 0.9$
4	$\lambda_0 = \lambda_1 = 0.2$ $\lambda_0 = \lambda_1 = 0.9$
5	$\lambda_0 = \lambda_1 = 0.2$ $\lambda_0 = \lambda_1 = 0.9$

Table 4.1: Tested ToM₀ Agent Configurations

Overall, none of the agents gained any points for the eight tested combinations. Since the agents' learning speed values are equal ($\lambda_i = \lambda_j$), all games ended up with a zero result for agents in all group sizes. As you can see in the figures below [4.11][4.12][4.13], regardless of the random choices by the agents in the first few rounds, for the rest of the game, agents made the same decisions in all tested repeated rounds variations (50, 100, and 150 rounds), which explains the overlapping of their choices. We can conclude that when all players reason at the same level and adapt at the same speed, their behavior converges, and no agent gains an advantage. Later, we will see whether this observation also applies to other ToM orders.

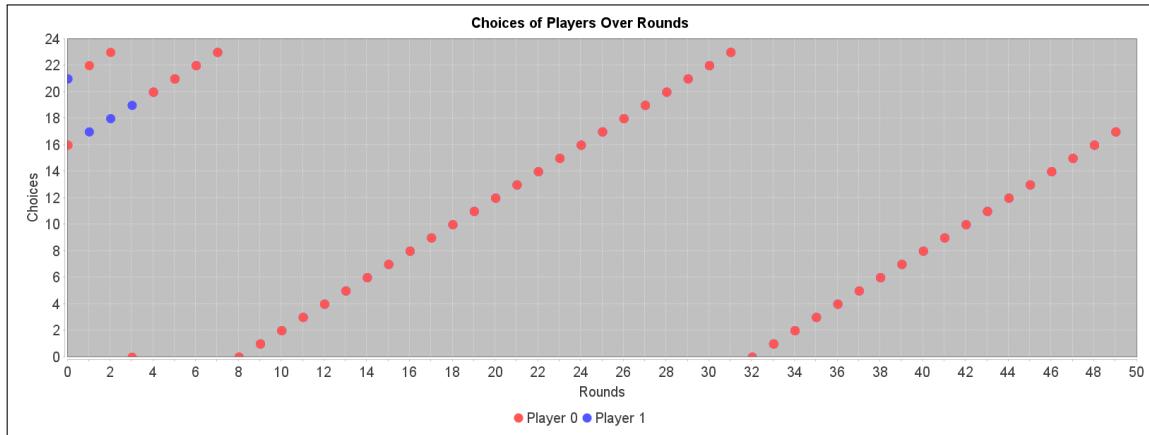


Figure 4.11: Choices distribution of two ToM₀ agents playing the game for 50 rounds. The agents were assigned the same LS value ($\lambda_0 = \lambda_1 = 0.9$)

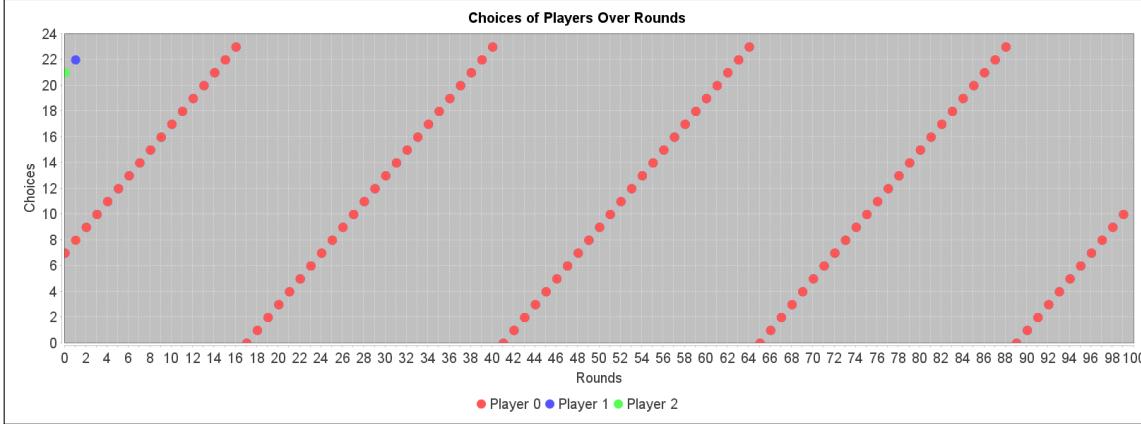


Figure 4.12: Choices distribution of three ToM₀ agents playing the game for 100 rounds. The agents were assigned the same LS value ($\lambda_0 = \lambda_1 = \lambda_2 = 0.2$)

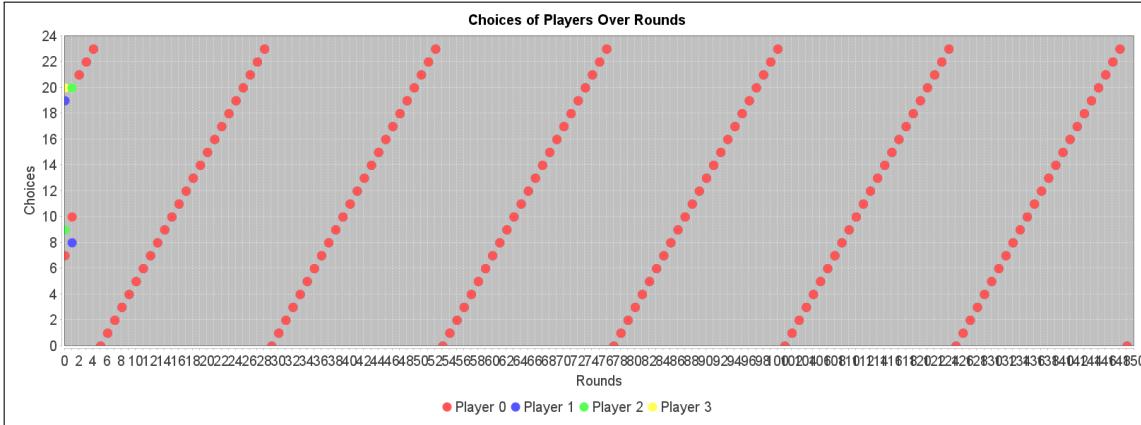


Figure 4.13: Choices distribution of four ToM₀ agents playing the game for 150 rounds. The agents were assigned the same LS value ($\lambda_0 = \lambda_1 = \lambda_2 = \lambda_3 = 0.9$)

2. Part 2: Groups of random Learning speed values.

In this part, different configurations were applied in the experiments across varying group sizes and the number of repeated rounds, but this time with assigning random learning speeds to the agents, which allows for a broader understanding of how agents with varying levels of adaptability interact in the game, leading to diverse outcomes. The experiments are divided into three distinct scenarios based on the distribution of learning speed values over the agents. All of the scenarios' experiments start with two agents, then add more agents each time, up to five agents per group, in order to monitor closely the changes that might occur under variant conditions through this process:

a. High-High Configuration

The first scenario investigates agents' behavior, starting with two agents, both assigned adjacent high learning speeds of ($\lambda_0 = 0.7$, $\lambda_1 = 0.9$). These values indicate that both agents can rapidly adapt and update their beliefs based on the game's outcomes. Then, additional agents with lower learning speed values ($\lambda_0 = 0.1$, $\lambda_1 = 0.2$, $\lambda_2 = 0.4$) are gradually added. This incremental setup helps observe how the addition of slower-learning agents affects the game's dynamics. Since this configuration starts with two fast-adapting agents, it could provide insights into how two agents with high learning speeds compete, how the addition of slower agents impacts their strategies, as well as the potential for a dominant strategy among the fast learners.

The values of the first configuration's characteristics alongside the simulation's results can be found in the following Table[4.2].

ToM ₀ - First Configuration Experiments										
Number of Agents	Learning Speed					Number of Rounds	Payoff			
	λ_0	λ_1	λ_2	λ_3	λ_4		P_0	P_1	P_2	P_3
2	0.7	0.9				50	0	0		
						100	0	0		
						150	0	0		
3	0.7	0.9	0.1			50	4	3	0	
						100	2	2	0	
						150	5	4	0	
4	0.7	0.9	0.1	0.2		50	33	33	0	1
						100	62	62	0	2
						150	102	102	1	6
5	0.7	0.9	0.1	0.2	0.4	50	39	61	0	11 24
						100	81	133	0	15 41
						150	125	209	1	22 65

Table 4.2: Experimental Results for Zero-Order ToM Agents with High-High Configuration. This table presents the outcomes of the zero-order ToM agents across different group sizes (2, 3, 4, and 5 agents) and various learning speed combinations. The High-High configuration is characterized by assigning two agents with high learning speeds (0.7 and 0.9) and adding more agents with lower learning speeds (0.1, 0.2, 0.4). The table details the agents' learning speeds, the number of rounds played (50, 100, 150), and the final payoffs each agent in the group achieved.

Results Analysis:

The beginning was with a group of two agents: agent₀ and agent₁, each assigned high learning speed values ($\lambda_0 = 0.7$ and $\lambda_1 = 0.9$), which showed that regardless of the

number of played rounds (50, 100, or 150), both agents consistently achieved zero points ($P_0 = P_1 = 0$). This outcome aligns with the theory revealed in Part1 experiments[1], that agents with identical or adjacent high learning speeds behave almost identically in competitive environments. The two agents' identical payoffs confirm that agents with close learning speed values exhibit nearly indistinguishable behavior, despite slight differences in their learning rates.

Adding a third agent with a significantly lower learning speed ($\lambda_2 = 0.1$) slightly alters the outcome; in the 50-round game, agent₀ and agent₁ earned 3 and 4 points, respectively, while agent₂ ended with 0 points. The pattern is consistent across all three game repetitions (50, 100, 150 rounds), with agent₀ and agent₁ maintaining a minor lead over agent₂. The performance gap between agent₀ and agent₁ remains marginal, highlighting that their higher learning speeds continue to dominate the game dynamics, while agent₂ struggles to compete due to its extremely low learning speed. However, it is worth mentioning that the few points gained by the two agents were obtained during the first rounds, where the agents' behavior was still random.

Adding another low-speed agent ($\lambda_3 = 0.2$) resulted in further changes; agent₀ and agent₁ maintain superior performance, and consistently earn more points than the other two low-speed agents. The significant gap in learning speed values creates a clear hierarchy where high-speed agents consistently outperform their low-speed counterparts. To better understand the cause behind identical gains for agents of adjacent but different learning speed values, we analyzed the agents' actions throughout the game's rounds. Specifically, as shown in Figures[4.14][4.15], agents with adjacent LS values (i.e., agent₀ and agent₁, OR agent₂ and agent₃) demonstrate identical behavior, even though one should theoretically perform better due to its higher LS value. The indicators explain this observation as follows: in a group of four ToM₀ agents, if two agents have adjacent low LS values ($\lambda \leq 0.5$) or adjacent high LS values ($\lambda \geq 0.5$), they tend to exhibit the same behavior, leading to identical performance. This is primarily driven by the negligible differences in their learning speeds, which are not significant enough to create a competitive advantage.

Finally, the introduction of a fifth agent ($\lambda_4 = 0.4$) further highlights the established pattern of the high LS agents' superiority over the low LS agents, where the agent with the highest learning speed (agent₁, $\lambda_1 = 0.9$) achieves the highest score in all game lengths, agent₀ with ($\lambda_0 = 0.7$) consistently follows, while the three low-speed agents compete for lower scores. Interestingly, while the high-speed agents maintain a clear advantage, the low-speed agents occasionally achieve higher scores among each other, each according to its speed value.

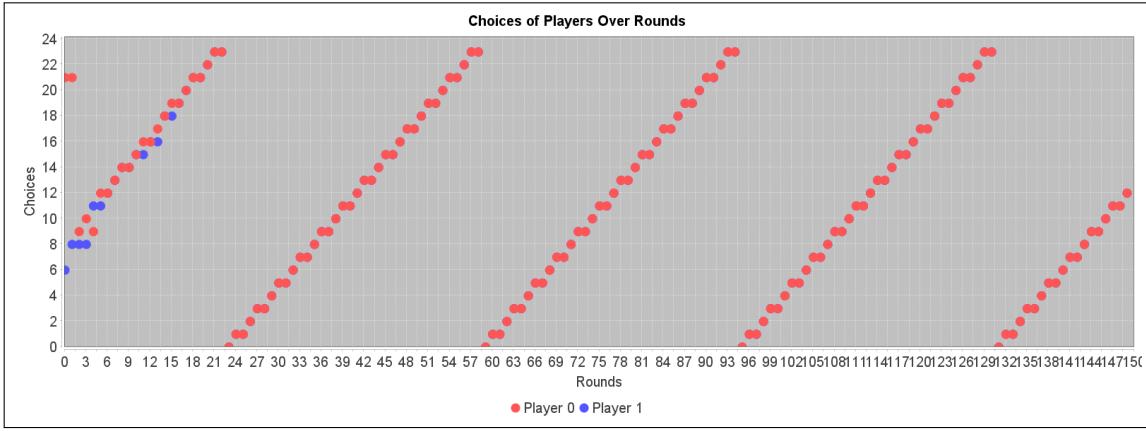


Figure 4.14: Choice patterns of agent₀ and agent₁ over 150 rounds in a four-agent group. This figure shows the behavior of two agents with adjacent high learning speed values ($\lambda_0 = 0.7$, $\lambda_1 = 0.9$), exhibiting nearly identical choices across the game, which illustrates the indistinguishable adaptation of agents with adjacent LS values.

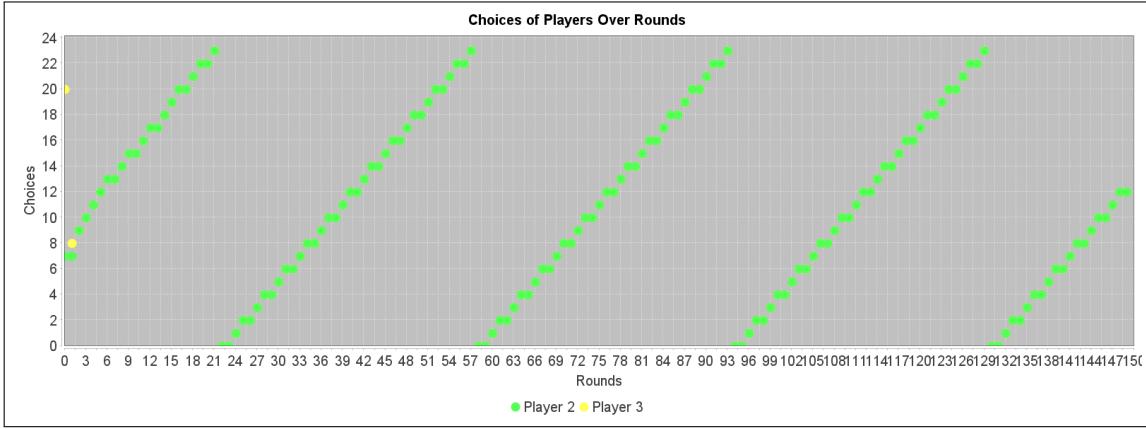


Figure 4.15: Choice patterns of agent₂ and agent₃ over 150 rounds in a four-agent group. These agents, with adjacent low learning speed values ($\lambda_2 = 0.1$, $\lambda_3 = 0.2$), also support the observation that agents with adjacent LS values behave similarly regardless of their performance expectations.

b. Low-High Configuration

The second scenario explores agents' behavior starting with two players, each having significantly different learning speeds; one with a low learning speed of $\lambda_0 = 0.2$ and the other with a high learning speed of $\lambda_1 = 0.9$. This configuration investigates the impact of combining fast and slow learners in a competitive environment. The results are presented in Table[4.3].

ToM ₀ - Second Configuration Experiments											
Number of Agents	Learning Speed					Number of Rounds	Payoff				
	λ_0	λ_1	λ_2	λ_3	λ_4		P_0	P_1	P_2	P_3	P_4
2	0.2	0.9				50	0	0			
						100	0	0			
						150	0	0			
3	0.2	0.9	0.1			50	17	51	0		
						100	36	110	0		
						150	57	171	0		
4	0.2	0.9	0.1	0.8		50	2	34	1	34	
						100	1	67	1	65	
						150	2	99	0	99	
5	0.2	0.9	0.1	0.8	0.6	50	11	67	1	63	36
						100	24	143	1	139	75
						150	38	217	0	218	113

Table 4.3: Experimental Results for Zero-Order ToM Agents with Low-High Configuration. This table presents the outcomes of the zero-order ToM agents across different group sizes (2, 3, 4, and 5 agents) and various learning speed combinations. The low-high configuration is characterized by assigning two agents with learning speeds of (0.2 and 0.9) and adding more agents with variant learning speeds (0.1, 0.8, 0.6). The table details the agents' learning speeds, the number of rounds played (50, 100, 150), and the final payoffs each agent in the group achieved.

Results Analysis:

Initially, with only two agents, the outcomes mirror the first scenario's "High-High" configuration. Both agents consistently achieve zero points regardless of the number of rounds (50, 100, or 150). This raises an observation that maybe not the adjacent LS values were the reason behind the zero score for a two-agent group simulation in the first scenario.

A notable change occurs when a third agent with a low learning speed ($\lambda_2 = 0.1$) is added to the group, which did not happen in the three-agent group experiment in the first configuration. The high-speed agent agent_1 starts to accumulate points, agent_0 fol-

lows, but with a big gap in both agents' scores, while the third slow agent cannot adapt as effectively. This pattern is consistent across all three game lengths. However, it is worth mentioning that this result only showed when the added agent had a value of $\lambda_3 = 0.1$, while for any other value, all agents were getting zero points.

Adding a fourth agent with a learning speed of $\lambda_3 = 0.2$ further clarifies this dynamic. The two high-speed agents (agent_1 with $\lambda_1 = 0.9$ and agent_3 with $\lambda_3 = 0.8$) maintain their superior performance, while the two low-speed agents (agent_0 and agent_2) each struggle to gain points. You can also notice that agent_0 was getting a much better payoff before adding the last agent of the high LS value; after that, the agents of adjacent LS values started to get identical. Figure[4.16] shows that the pattern for agent_0 & agent_2 , as well as agent_1 & agent_3 , where the choices of each pair of agents are overlapping, denoting identical behavior.

In the five-agent game, similarly to the first configuration finding about the five-agent group, the performance hierarchy becomes even more evident. The high-speed agents continue to dominate, while the low-speed agents compete among each other for lower scores. Although agent_1 and agent_3 are outperforming agent_4 , looking at these three agents' behavior (see Figure[4.17]) aligns with the finding that agents with the most adjacent LS values in the group will behave more similarly.

Eventually, in a group of 5 agents, the 'number of agents' parameter plays an effective role in stabilizing the agents' behavior and shows scores that make more sense according to each agent's learning speed value.

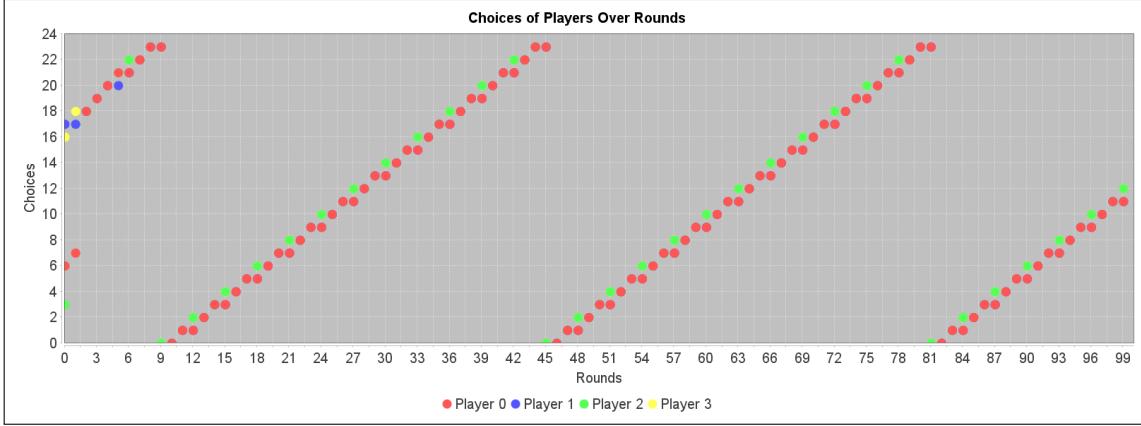


Figure 4.16: Behavior patterns of four ToM₀ agents over 100 rounds. The agents were assigned the following learning speeds: $\lambda_0 = 0.2$, $\lambda_1 = 0.9$, $\lambda_2 = 0.1$, $\lambda_3 = 0.8$. The overlapping paths between agent₀ and agent₁, and between agent₂ and agent₃, demonstrate identical behavior among agents with adjacent learning speed values.

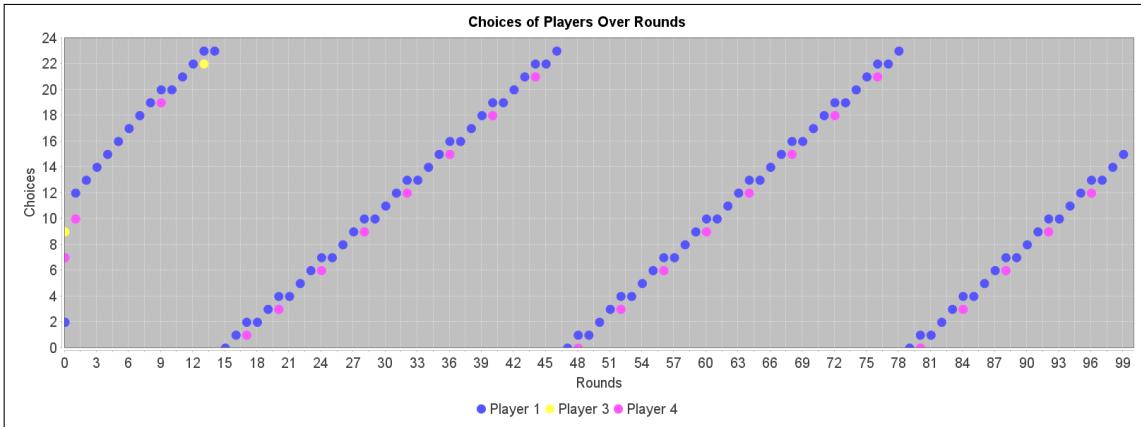


Figure 4.17: Behavior patterns of three ToM₀ agents (agent₁, agent₃, agent₄) over 100 rounds of the game. These agents had adjacent learning speeds of: $\lambda_1 = 0.9$, $\lambda_3 = 0.8$, $\lambda_4 = 0.6$, and their actions show strong alignment, indicating that agents with the most adjacent LS tend to behave similarly.

c. Low-Mid-High Configuration

This configuration explores how agents with low, medium, and high learning speeds interact in a competitive environment. Unlike the previous two configurations, this setup introduces a broader range of learning speed values, leading to more varied behaviors and outcomes. Table[4.4] presents the experiments' results.

ToM0 - Third Configuration Experiments											
Number of Agents	Learning Speed					Number of Rounds	Payoff				
	λ_0	λ_1	λ_2	λ_3	λ_4		P_0	P_1	P_2	P_3	P_4
2	0.1	0.5				50	0	14			
						100	0	32			
						150	0	50			
3	0.1	0.5	0.8			50	0	14	36		
						100	0	39	77		
						150	0	38	110		
4	0.1	0.5	0.8	0.3		50	0	15	60	15	
						100	0	33	127	33	
						150	2	55	189	49	
5	0.1	0.5	0.8	0.3	0.7	50	0	32	53	20	32
						100	0	69	104	43	68
						150	0	108	172	66	108

Table 4.4: Experimental Results for Zero-Order ToM agents with Low-Mid-High Configuration. This table presents the outcomes of the zero-order ToM agents across different group sizes (2, 3, 4, and 5 agents) and various learning speed combinations. The Low-Mid-High configuration is characterized by assigning two agents with learning speeds of (0.1 and 0.5) and adding more agents with variant learning speeds (0.3, 0.8, 0.7). The table details the agents' learning speeds, the number of rounds played (50, 100, 150), and the final payoffs each agent in the group achieved.

The experiments began with a two-agent group, where one agent was assigned a low learning speed ($\lambda_0 = 0.1$) and the other a medium learning speed ($\lambda_1 = 0.5$). The results show that the agent with a higher learning speed consistently outperforms the slower agent in all three game lengths (50, 100, and 150 rounds). In this two-agent game, we see non-zero results, contrary to what we saw in the previous two configurations. What is the reason behind this? In fact, after trying all the possible values on two-agent game, it was found that assigning to one of the agents (agent_{k1}) a LS value of ($\lambda_{k1} = 0.1$), with a value to the other agent (agent_{k2}) in the range of ($\lambda_{k2} \geq 0.2$), will result in a payoff > 0 for agent_{k2}, with a zero payoff, on the other hand, to agent_{k1}. Therefore, that is why we see non-zero results for the game of two ToM₀ agents in this game.

Introducing a third agent with a high learning speed ($\lambda_2 = 0.8$) significantly altered the dynamics. In this case, agent₂ consistently secured the highest payoffs, leveraging its superior learning speed. Agent₁, with a moderate learning speed, maintained a middle-ground position in the ranking. This hierarchical distribution of scores (highest for agent₂, followed by agent₁, and the lowest for agent₀) persisted across all game lengths.

When the group was expanded to four agents by adding another slow learner $\lambda_3 = 0.3$, the results revealed a more complex interaction. Agent₂ (with $\lambda_2 = 0.8$) maintained its superior position, consistently achieving the highest payoffs. However, agent₁ (with $\lambda_1 = 0.5$) tied with agent₃ (with $\lambda_3 = 0.3$) when playing the game for 50 and 100 rounds, while it slightly underperformed when repeating the game for 150 rounds. However, this did not happen when the adjacent learning speed values were high (i.e., the first configuration[a.]); agents that had adjacent high learning speed values ($\lambda_2 = 0.7$ and $\lambda_3 = 0.9$) were highly competitive when playing the game in a group of 4-agents, and kept on getting the same score regardless how many times the game was repeated.

Other experiments on a four-agent group that are worth mentioning which is the case when assigning a high LS value ($\lambda \geq 0.5$) to the 4th added agent instead of having a low learning speed value, as was the case in all previous configurations. In this case, agent₀ will get a payoff of $P_0 = 0$, and all other three agents will get a payoff of no more than 8 points playing the game for any number of rounds (see Table[4.5]). Therefore, the learning speed value alone does not always fully determine the behavior of a ToM₀ agent, especially an agent with an intermediate value; other factors like the interaction among agents with varying learning speeds and playing the game for an increasing number of rounds also play crucial roles.

Number of Agents	Learning Speed				Number of Rounds	Payoff			
	λ_0	λ_1	λ_2	λ_3		P_0	P_1	P_2	P_3
4	0.1	0.5	0.8	0.7	50	0	5	7	6
					100	0	5	7	7
					150	0	4	4	5

Table 4.5: Extra experiment on a four-agent ToM₀ group

Finally, the hierarchy became even more apparent in the five-agent configuration, where another agent ($\lambda_4 = 0.7$) was added. Agent₂ continued to dominate, followed by agent₄ and agent₁ equally, the agent₃ (with $\lambda_3 = 0.3$), and finally agent₀ (with $\lambda_0 = 0.1$). The results demonstrate a clear pattern where agents with higher learning speeds consistently outperform those with lower values. One significant observation is that agent₁ started to get more identical with agent₄ instead of agent₃, as was the case in the four-agent group, although both agents have evenly adjacent LS values from the agent₁ LS value. This shows some

deviation towards a specific agent under specific conditions. In other words, agent₁ with a LS of $\lambda_1 = 0.5$ showed a better performance in a five-agent group and matched the behavior and payoff of agent₄ with a LS of $\lambda_4 = 0.7$ (see Figure[4.18]). This performance was only revealed when a competitor with a higher LS value joined the group.

According to the results of all configuration experiments on ToM₀ agents, the performance difference between agents becomes more evident as the range of learning speeds widens, highlighting the impact of this parameter on the agents' competitiveness and their ability to adapt.

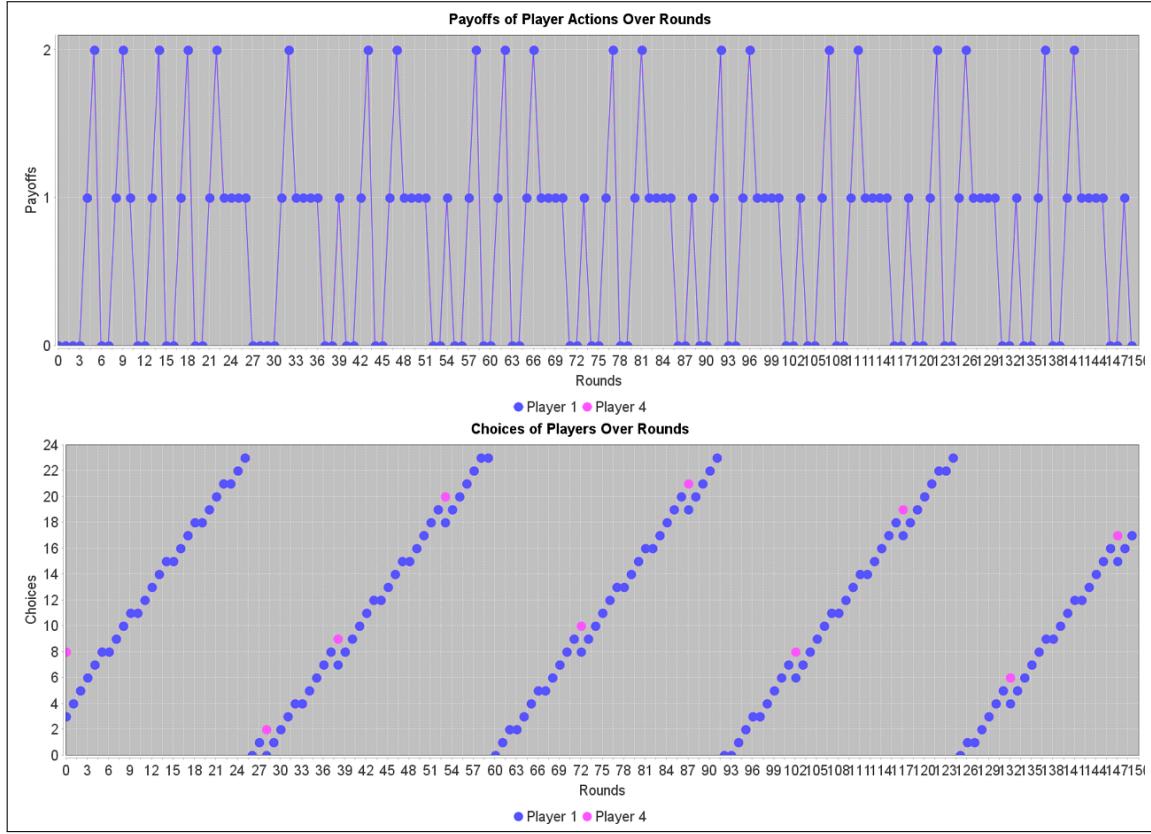


Figure 4.18: Comparative behavior and performance of agent₁ and agent₄ in a five-agent ToM₀ game. The top subplot displays the payoff trends of both agents over 150 rounds, while the bottom subplot illustrates their corresponding choices. Agent₁ with a learning speed of $\lambda_1 = 0.5$ exhibits alignment in both behavior and payoff outcomes with agent₄, which has a higher learning speed of $\lambda_4 = 0.7$. This convergence only became evident after agent₄ participation, suggesting that learning dynamics can shift based on the presence of agents with specific adaptation rates.

ToM₀ Results Conclusion:

- (a) An agent that plays the game using a learning speed of $\lambda = 0.1$ always gets zero payoff among other agents of higher learning speeds.
- (b) Agents that play the game using the same learning speed values gain zero points by the end of the game, due to identical behavior.
- (c) When a group of two ToM₀ agents plays the game, the result is always zero unless one of the agents has a learning speed value of $\lambda > 0.1$, with any value that is $\lambda > 0.2$ for the other one.
- (d) In three-agent groups, the agents' payoff depends more on the learning speed margin of the agents.
- (e) For four-agent groups, the agents with adjacent learning speed values keep behaving identically and getting similar scores for all repeated game rounds.
- (f) In a group of 5 agents, the playing hierarchy becomes even clearer; the agents with high LS values outperform the others and get higher payoffs.
- (g) Although the value of the learning speed plays a crucial role for ToM₀ groups, sometimes the learning speed alone does not dictate the outcome; instead, it is the relative difference in learning speeds within the group, the number of agents, and sometimes the number of repeated rounds, all together shape competitive interactions and scoring patterns.

4.3.2 Groups of ToM₁ Agents

A ToM₁ agent is an agent that uses a single reasoning level in questioning their opponent's mental state and intentions (see section[2]). Remember that the ToM₁ agent generates two models of his opponent, one acts as a ToM₀ and one as a ToM₁, and through playing the game for multiple rounds, the agent determines which model best describes his opponent's behavior and behaves according to it. In this section, you will find the experiments in a similar structure to the ones executed on ToM₀.

1. Part 1: Groups of identical learning speeds:

For these groups of agents that are playing using the same learning speed values ($\lambda_i = \lambda_j$).

Similarly to ToM₀, examining ToM₁ agents under the same game's configuration (group sizes, number of rounds, and learning speed values) used for ToM₀, the payoffs of agents were zero for all settings. For example, in a game of two (see Figure[4.19]) and three agent (see Figure[4.20]) groups, you can see that ToM₁ agents of the same LS values make identical choices, which is causing the zero payoff for all agents since each has the same adapting speed to the other's changes.

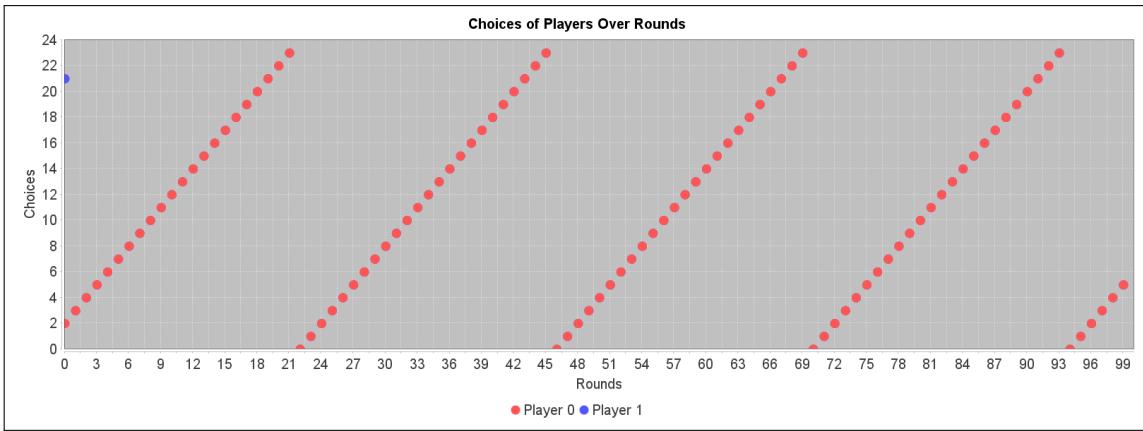


Figure 4.19: Choices distribution of two ToM₁ agents playing the game for 100 rounds. The agents were assigned the same LS value: $\lambda_0 = \lambda_1 = 0.2$

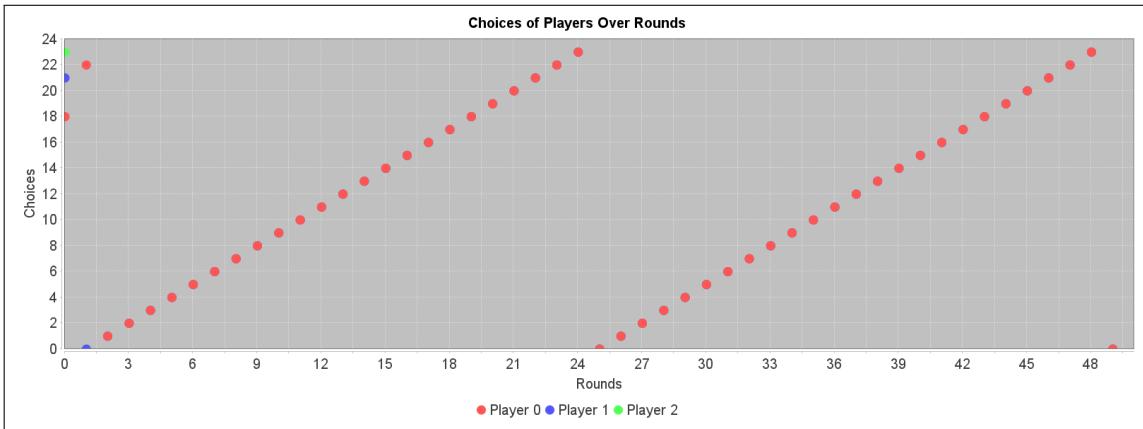


Figure 4.20: Choices distribution of three ToM₁ agents playing the game for 50 rounds. The agents were assigned the same LS value: $\lambda_0 = \lambda_1 = 0.9$

2. Part 2: Groups of different random learning speed values.

Different combinations of players were assigned random learning speeds to test all possible outcomes and get a full overview of agents' behaviors under different conditions. These combinations were also tested for 50, 100, and 150 rounds, in groups of 2, 3, 4, and 5 players. However, the same three configurations used in ToM₀ (high-high, low-high, and low-mid-high) will also be used for this order to show the outcomes and compare with ToM₀ results.

The experiments on ToM₁ agents showed an interesting observation; the behavior of ToM₁ agents is 'almost' the same as ToM₀ agents in terms of behavior and the score range. This could be due to the actual extent of ToM abilities used mainly by the agent. Since a ToM₁ agent has the option to play using its zero or first order, it could be a possible theory that a ToM_l agent mostly uses its ToM_{l-1} abilities more than its actual ToM level. Through the analysis of the experiments, we can check the validity of this idea for ToM₁ and higher-order agents. Although the performance between ToM₀ and ToM₁ is very similar, a difference in the behavior was noticed; the repetition of the choices by ToM₁ agents through the game is less than that of ToM₀ agents under the same game conditions. For example, in the following figures[4.21][4.22], you can see the choices of two groups, each consisting of three agents, one of ToM₀ and another of ToM₁ agents, repeating the game over 50 rounds using learning speed values of ($\lambda_0 = 0.8, \lambda_1 = 0.1, \lambda_2 = 0.5$). You can see that ToM₀ agents tend to repeat the same choice for three rounds in a row before moving forward to the next choice, while in ToM₁ agents, the repetition happens for a maximum of two rounds in a row.

Otherwise, everything else about ToM₁ behavior is identical to ToM₀ under all different game settings. In the following, some results on the same 3 configurations that were tested on ToM₀ agents will be presented to compare and see the similarity with ToM₀ performance.

In every game configuration, you will directly find the results' analyses; a list of similarities and differences with ToM₀ results will be mentioned. However, the explanation of each one is the same from the ToM₀ experiments section, thus it will not be repeated here again.

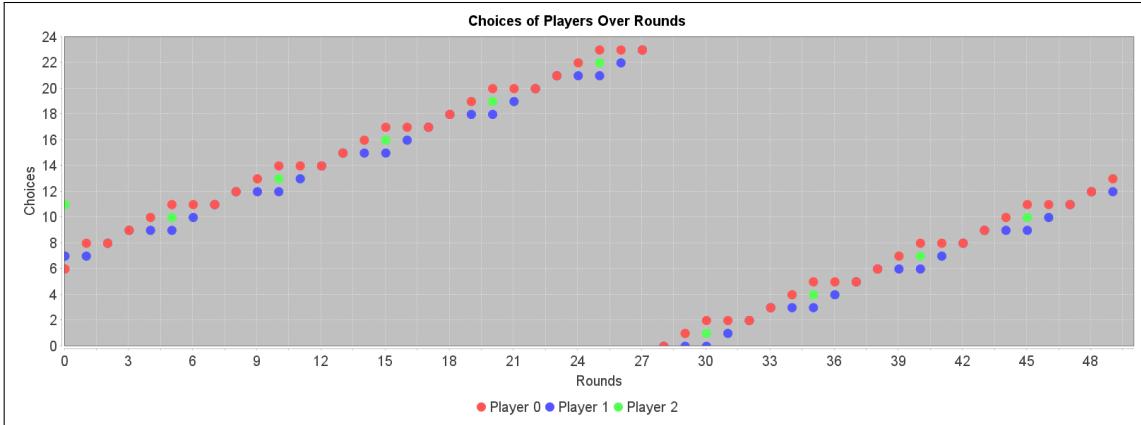


Figure 4.21: Choices pattern of a three-agent ToM₀ group playing the game for 50 rounds with learning speed values of: $\lambda_0 = 0.8$, $\lambda_1 = 0.1$, and $\lambda_2 = 0.5$. The agents demonstrate repetitive decision behavior, often selecting the same number for up to three consecutive rounds before progressing.

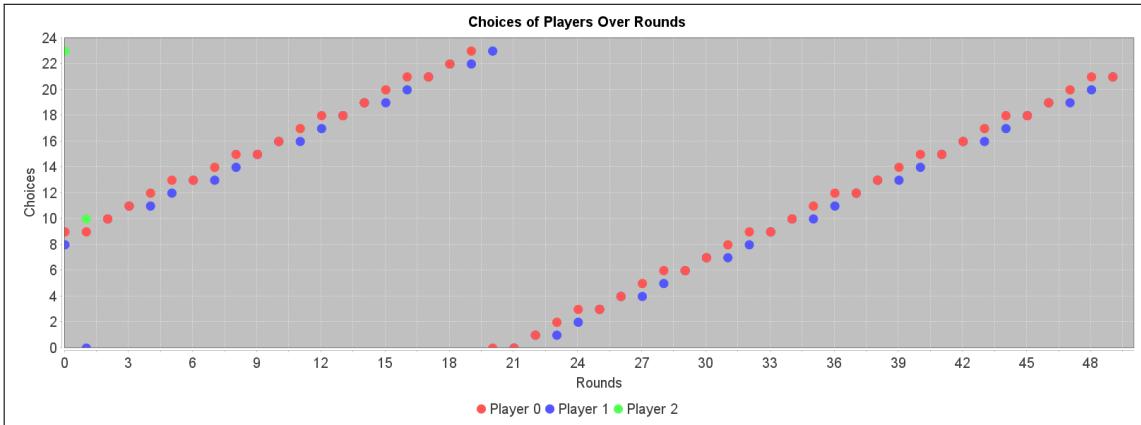


Figure 4.22: Choices pattern of a three-agent ToM₁ group playing the game for 50 rounds using the same learning speed values: $\lambda_0 = 0.8$, $\lambda_1 = 0.1$, and $\lambda_2 = 0.5$. Compared to ToM₀ agents, these agents show slightly less repetition in their choices, typically repeating a number for no more than two rounds before advancing.

a. High-High Configuration

Results Analysis:

Looking at the results table of results and comparing it with the results from ToM₀ (Configuration[a.]), the similarity with ToM₀ results for all numbers of rounds and the different numbers of players in the group is validated as follows:

- Two ToM₁ agents playing the game $\xrightarrow{\text{score}}$ zero's for all agents.

- Three ToM₁ agents playing the game $\xrightarrow{\text{score}}$ a few points for agents of high learning speed values due to random choices at the beginning of the game, and zero points for the agent of $\lambda_2 = 0.1$.
- Four ToM₁ agents playing the game $\xrightarrow{\text{score}}$ same range of score for each agent. For instance, in the same game settings, and for 50 rounds, the first ToM₀ agent of $\lambda_0 = 0.7$ got a score of 33, the ToM₁ agent got 39 points. The same ToM₁ agent for 150 rounds gets 100 points, and it was 102 points for the ToM₀ agent under the same game settings.
- Five ToM₁ agents playing the game $\xrightarrow{\text{score}}$ same performance for each agent.

ToM1 - First Configuration Experiments										
Number of Agents	Learning Speed					Number of Rounds	Payoff			
	λ_0	λ_1	λ_2	λ_3	λ_4		P_0	P_1	P_2	P_3
2	0.7	0.9				50	0	0		
						100	0	0		
						150	0	0		
3	0.7	0.9	0.1			50	2	2	0	
						100	5	3	0	
						150	2	2	0	
4	0.7	0.9	0.1	0.2		50	39	39	0	7
						100	64	64	2	1
						150	100	101	0	3
5	0.7	0.9	0.1	0.2	0.4	50	38	67	1	9
						100	73	121	0	38
						150	126	210	1	23
										65

Table 4.6: Experimental Results for First-Order ToM Agents with High-High Configuration. This table presents the outcomes of the first-order ToM agents across different group sizes (2, 3, 4, and 5 agents) and various learning speed combinations. The high-high configuration is characterized by assigning two agents with high learning speeds (0.7 and 0.9) and adding more agents with lower learning speeds (0.1, 0.2, 0.4). The table details the agents' learning speeds, the number of rounds played (50, 100, 150), and the final payoffs each agent in the group achieved.

b. Low-High Configuration

Results Analysis:

Similarly to ToM₀ agents results, ToM₁ agents satisfy the following observations:

- Two ToM₁ agents playing the game with a big gap between their learning speed values came back with zero payoff for each.
- In this game's configuration, we start to see results in the groups of 3 agents instead

of 4 agents compared to the first configuration (High-High) and only when adding a new agent with a learning speed value of 0.1, any other LS value for the added agent would result zero for all agents (same as in ToM_0).

- In four-agent groups, of which two agents have high LS and two low LS, the agents that have adjacent LS values started to get almost the same scores in all numbers of repeated rounds due to identical behavior.
- In the five-agent game, the performance hierarchy becomes even more evident. The high-speed agents continue to dominate, while the low-speed agents compete among each other for lower scores. Thus, the 'number of agents' parameter plays an effective role in stabilizing the agents' behavior.

ToM1 - Second Configuration Experiments											
Number of Agents	Learning Speed					Number of Rounds	Payoff				
	λ_0	λ_1	λ_2	λ_3	λ_4		P_0	P_1	P_2	P_3	P_4
2	0.2	0.9				50	0	0			
						100	0	0			
						150	0	0			
3	0.2	0.9	0.1			50	16	50	0		
						100	37	111	0		
						150	50	170	0		
4	0.2	0.9	0.1	0.8		50	2	29	0	30	
						100	3	70	0	71	
						150	3	102	2	101	
5	0.2	0.9	0.1	0.8	0.6	50	9	70	0	67	37
						100	25	151	1	151	82
						150	37	222	4	222	113

Table 4.7: Experimental Results for First-Order ToM Agents with Low-High Configuration. This table presents the outcomes of the first-order ToM agents across different group sizes (2, 3, 4, and 5 agents) and various learning speed combinations. The low-high configuration is characterized by assigning two agents with learning speeds of (0.2 and 0.9) and adding more agents with variant learning speeds (0.1, 0.8, 0.6). The table details the agents' learning speeds, the number of rounds played (50, 100, 150), and the final payoffs each agent in the group achieved.

c. Low-Mid-High Configuration

Results Analysis:

In this game's configuration, the agents' choices are more varied than in the previous games. All conclusions, observations, and explanations provided about this game's configuration for ToM_0 agents (Configuration[c.]) also apply here to ToM_1 groups.

- In this game's settings, we start to see results for two-agent ToM₁ groups.
- Agents of high LS values outperform those of low values.
- The agent's behavior of a moderate LS value ($\lambda = 0.5$) might get affected by other factors (i.e., the LS value of the 4th agent added to the group).
- In the five-agent groups, the hierarchy became more apparent. The results demonstrate a clear pattern where agents with higher learning speeds consistently outperform those with lower values.

ToM1 - Third Configuration Experiments											
Number of Agents	Learning Speed					Number of Rounds	Payoff				
	λ_0	λ_1	λ_2	λ_3	λ_4		P_0	P_1	P_2	P_3	P_4
2	0.1	0.5				50	0	16			
						100	0	33			
						150	0	45			
3	0.1	0.5	0.8			50	0	17	35		
						100	0	24	70		
						150	0	40	111		
4	0.1	0.5	0.8	0.3		50	1	20	56	17	
						100	0	33	123	31	
						150	0	55	186	49	
5	0.1	0.5	0.8	0.3	0.7	50	4	36	68	16	35
						100	1	67	109	40	67
						150	1	110	174	67	110

Table 4.8: Experimental Results for First-Order ToM Agents with Low-Mid-High Configuration. This table presents the outcomes of the first-order ToM agents across different group sizes (2, 3, 4, and 5 agents) and various learning speed combinations. The Low-Mid-High configuration is characterized by assigning two agents with learning speeds of (0.1 and 0.5) and adding more agents with variant learning speeds (0.3, 0.8, 0.7). The table details the agents' learning speeds, the number of rounds played (50, 100, 150), and the final payoffs each agent in the group achieved.

After noticing that ToM₁ agents' simulations are resulting in payoffs that are almost the same ones we had in ToM₀ agents under the same simulation conditions, a deeper look into the agents' choice patterns clarifies the possible reason. Taking one of the simulations as an example(see Figure[4.23]) you can see that ToM₁ agents play using their ToM_{l-1} capabilities; repeating their previous choice or choosing the one that is +1 higher than the last round throughout game instead of using their higher order extent (the first order). This is also the case for all the other game simulations; some agents rarely use their first level of reasoning during the game. However, checking the exis-

tence of this observation for other ToM order agents will clarify whether this behavior is exclusive to ToM_1 agents or is a common theory across all ToM levels.

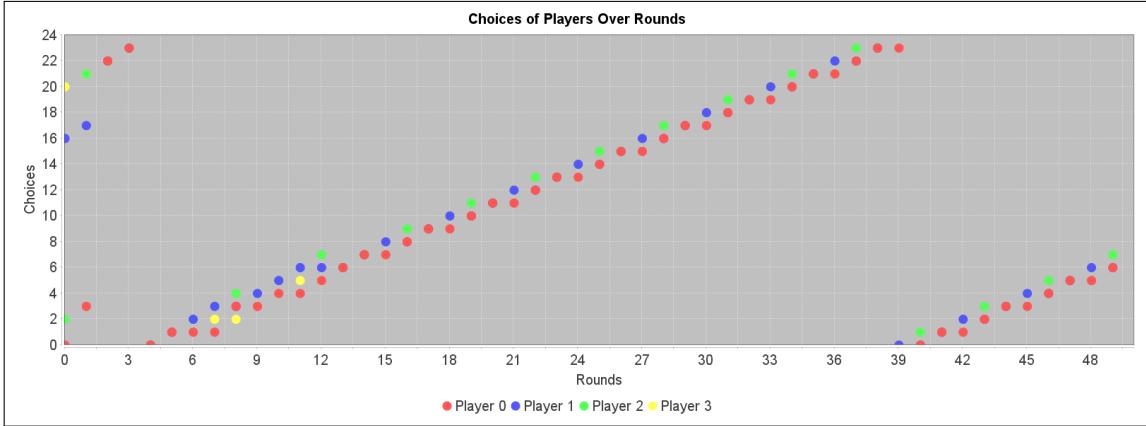


Figure 4.23: Choices distribution of a four-agent ToM_1 group playing the game for 50 rounds with randomly assigned learning speed values: $\lambda_0 = 0.1$, $\lambda_1 = 0.5$, $\lambda_2 = 0.8$, $\lambda_3 = 0.3$. This pattern reveals that agents predominantly rely on ToM_0 -like behavior rather than consistently utilizing their first-order reasoning abilities.

ToM₁ Results Conclusion:

1. All observations from ToM_0 experiments (see ToM_0 results conclusion[2]) for the different game configurations also apply to ToM_1 agents.
2. This match suggests that when playing the game, each ToM agent of a specific level tends to use its ToM_{l-1} abilities much more than the extent of its actual ToM level.

4.3.3 High-Order ToM Agents

4.3.3.1 Groups of ToM₂ Agents

For ToM_2 agents experiments, unexpected behavior or variant patterns were more likely to be seen here than in other lower orders. This can be attributed to the fact that this order represents an intersection point between the first and higher ToM levels, since it is the first level of higher orders. Contrary to the case in lower orders, no fixed rule applies to all groups of a specific number of players. It is more a combination of more than one factor that explains agents' performance in a specific group. Moreover, playing the game for 50 rounds in this order is not enough to show the real performance of an agent; that is why, many times in 50-round experiments, the results are

different from what we can get after playing the game for more rounds.

In this order, examining the agents of identical learning speed values also resulted in approximately zero outcomes for all agents. Therefore, only the groups of different learning speed values will be discussed. The results presentation here will be divided depending on the group's size, instead of the learning speed value, since it was noticed that for higher orders, the agent's performance is not related mainly to the agent's learning speed value, or at least, not related *only* to that value.

1. Groups of two agents

Starting with groups of two-agent ToM₂, it was noticed that the agent with the lower learning speed value usually performs better than the one with a higher learning speed for all tested repeated game rounds. However, the gap between the agents' scores is related to the margin between their learning speed values; the gap in the payoff between both agents decreases directly with the decrease of the margin of the agents' learning speed values. For instance, when playing the game for 100 rounds (see Table[4.9]), the difference in the payoff between agents of the first group of ($\lambda_0 = 0.2, \lambda_1 = 0.9$) is 21 points, while for the second group of ($\lambda_0 = 0.1, \lambda_1 = 0.5$), where the learning speed margin is less, it is 12 points. The performance of agents of adjacent learning speed values ($\lambda_0 = 0.7, \lambda_1 = 0.9$) is almost the same, either a few points are gained due to random choices at the beginning, or both get zero.

Number of Agents	Learning Speed		Number of Rounds		Payoff	
	λ_0	λ_1			P_0	P_1
2	0.2	0.9	50		13	15
			100		33	12
			150		50	29
2	0.1	0.5	50		14	11
			100		28	16
			150		48	30
2	0.7	0.9	50		2	3
			100		0	0
			150		5	1

Table 4.9: Results of two-agent ToM₂ simulations

2. Groups of three agents

Three-agent ToM₂ groups presented diverse performance outcomes due to the unique interactions between agents with different learning speeds. The key challenge was identifying

whether there was a consistent pattern related to the combination of speed values, especially when the group included extreme speed differences (e.g., a high and a low value with an intermediate one).

For example (see Table[4.10]), when the intermediate speed agent was positioned between two agents of extreme LS values (i.e., $\lambda_0 = 0.1$, $\lambda_1 = 0.5$, $\lambda_2 = 0.8$), a clear performance hierarchy emerged, regardless of the number of rounds: the intermediate-speed agent consistently outperformed the others (see Figure[4.25]). However, in other cases (i.e., $\lambda_0 = 0.2$, $\lambda_1 = 0.9$, $\lambda_2 = 0.1$) when the agent of the intermediate speed had a low value ($\lambda_0 = 0.2$), this combination produced different patterns (see Figure[4.26]); the agents of two extreme speed values ($\lambda_1 = 0.9$, $\lambda_2 = 0.1$) outperformed the agent of the intermediate speed with a big difference in the score.

Extended Investigation:

Since the observation that the performance of three-agent groups was highly dependent on the specific configuration, to resolve the ambiguity observed, an extended experiment was conducted with varied speed values ranging from 0 to 1 (see Table[4.11]). The groups were categorized into five distinct sets based on speed range results (see Figure [4.24]).

- **Group 1: Extremes values ($\lambda_0 = 0.9$ or $\lambda_0 = 0.1$)**

When the third agent had a similar speed to one of the extremes, the agent's performance aligned with that speed agent, and the agent of the different value won over both.

- **Group 2: Intermediate-Low values ($\lambda_0 = 0.2$ or $\lambda_0 = 0.3$)**

When the third agent had a low LS value, the two extreme values ($\lambda_1 = 0.9$ or $\lambda_2 = 0.1$) almost matched each other's behavior, and both outperformed the third agent.

- **Group 3: Intermediate values ($\lambda_0 = 0.4$ or $\lambda_0 = 0.5$)**

Introducing intermediate values effectively balanced the group. Neither high-speed nor low-speed dominance was observed. However, the agent with the highest speed had the weakest performance among the other two agents.

- **Group 4: Intermediate-High values ($\lambda_0 = 0.6$ or $\lambda_0 = 0.7$)**

In this group, the agent of the intermediate value outperformed the other two agents with a clear difference in the payoff, while those two agents performed almost identically.

- **Group 5: High values** ($\lambda_0 = 0.8$ or $\lambda_0 = 1$)

In this group, the performance of all agents remained uniform.

Number of Agents	Learning Speed			Number of Rounds	Payoff		
	λ_0	λ_1	λ_2		P_0	P_1	P_2
3	0.2	0.9	0.1	50	17	27	31
				100	22	61	62
				150	32	100	106
3	0.1	0.5	0.8	50	18	30	14
				100	36	67	26
				150	56	105	36
3	0.7	0.9	0.1	50	26	8	10
				100	55	21	27
				150	89	24	47

Table 4.10: Results of games played by three ToM₂ agents of different learning speed values

Group Number	λ_0	λ_1	λ_2	Number of Rounds	P_0	P_1	P_2
Group1	0.1	0.9	0.1	100	25	40	27
	0.9	0.9	0.1		35	36	44
Group2	0.2	0.9	0.1	100	22	61	62
	0.3	0.9	0.1		29	55	45
Group3	0.4	0.9	0.1	100	61	32	44
	0.5	0.9	0.1		59	22	39
Group4	0.6	0.9	0.1	100	54	27	23
	0.7	0.9	0.1		55	21	27
Group5	0.8	0.9	0.1	100	38	36	42
	1	0.9	0.1		39	36	41

Table 4.11: Results of extended experiments on three-agent ToM₂ group

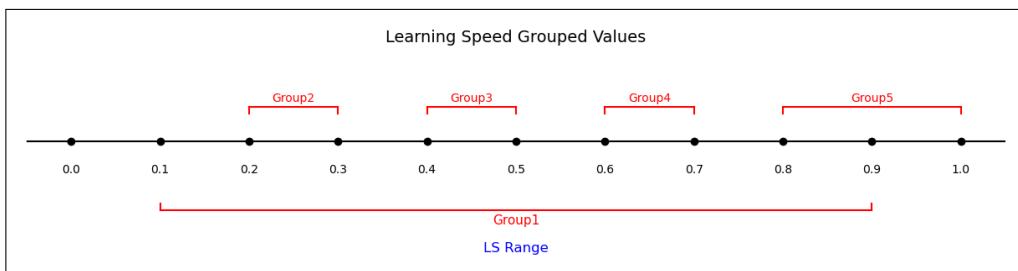


Figure 4.24: Categorization of learning speed values into five distinct groups based on their range intervals. This division resulted from the extended investigation of ToM₂ agents' behavior, showing how different configurations of learning speeds affect agents' interactions and performance. Each group represents a specific span of learning speeds, allowing a structured evaluation of the agents' reasoning dynamics.

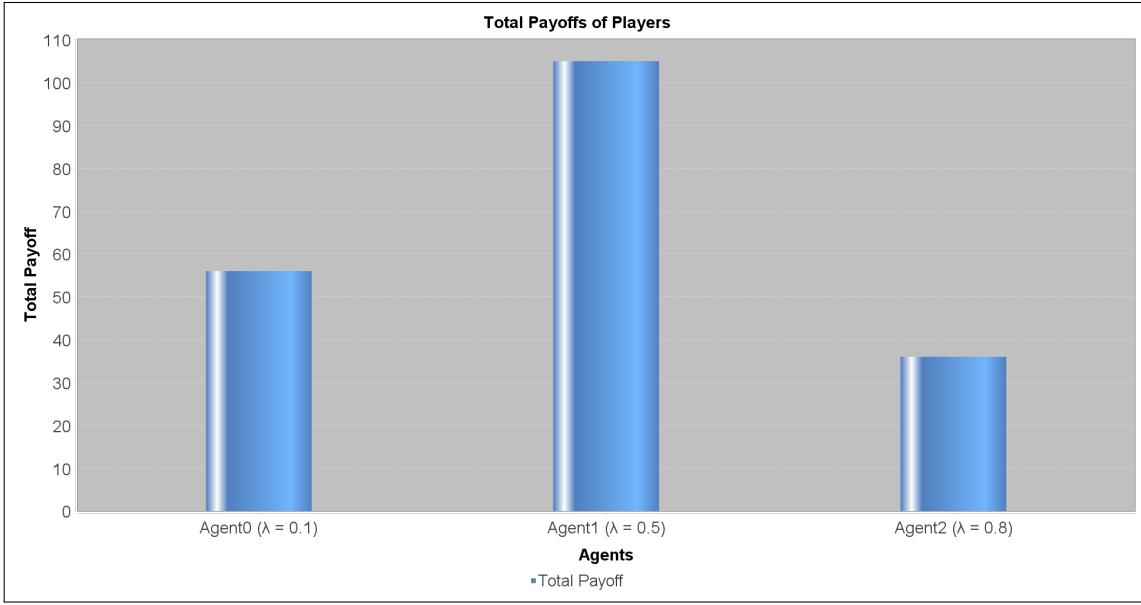


Figure 4.25: Total payoff scores of three ToM₂ agents playing the game for 150 rounds using learning speed values of: $\lambda_0 = 0.1$, $\lambda_1 = 0.5$, $\lambda_2 = 0.8$. The agent with the intermediate learning speed $\lambda_1 = 0.5$ achieves the highest total payoff.

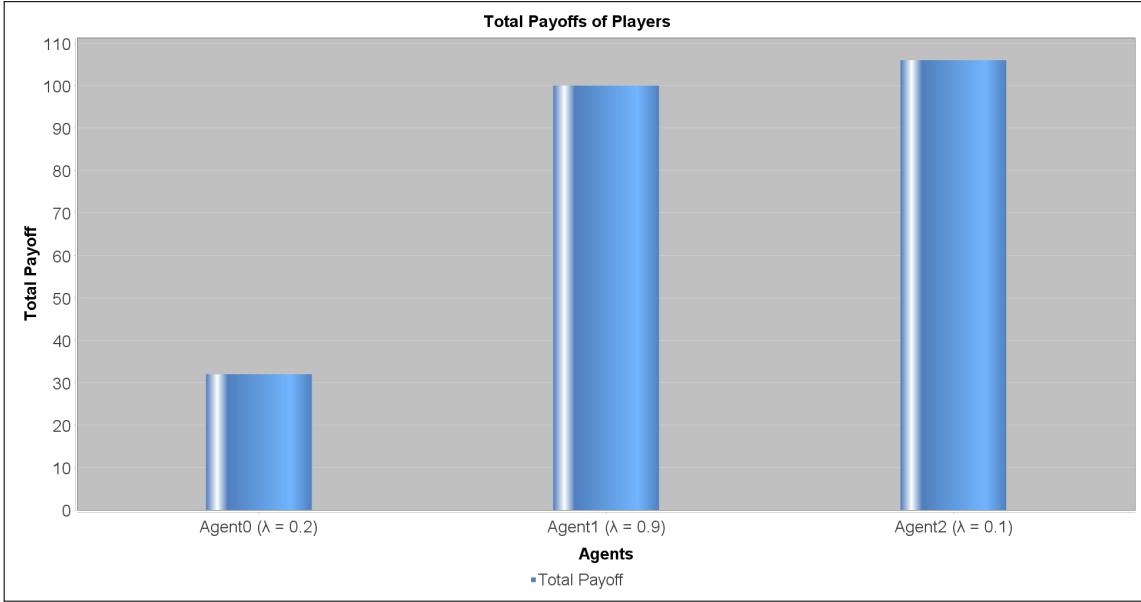


Figure 4.26: Total payoff scores of three ToM₂ agents playing the game for 150 rounds using learning speed values of: $\lambda_0 = 0.2$, $\lambda_1 = 0.9$, $\lambda_2 = 0.1$. The two agents with the most extreme learning speeds outperform the intermediate agent, highlighting a reversed pattern in performance compared to the previous configuration illustrated in Figure 4.25.

3. Groups of four & five agents

The performance of agents in four and five-agent ToM₂ groups exhibited a notable shift

toward more consistent and predictable outcomes compared to smaller groups. This consistency is primarily due to the larger group size, which stabilizes the agents' interactions, leading to clearer performance hierarchies.

In both four-agent and five-agent groups, the agents of lower learning speeds consistently outperform higher learning speed agents, and their performance becomes more stable as the number of rounds increases. This is because slower learners maintain their strategies, while faster learners tend to over-adjust, making them vulnerable to predictable patterns. For example, when a four-agent group ($\lambda_0 = 0.2, \lambda_1 = 0.9, \lambda_2 = 0.1, \lambda_3 = 0.8$) plays the game for 50 rounds, the scores of all four agents are relatively close, suggesting that the game duration is too short for significant performance differences to emerge. When the game is repeated for 100 and 150 rounds, the agents with lower learning speeds ($\text{agent}_0, \text{agent}_2$, and agent_3) gradually outperform the high-speed one (agent_1).

Number of Agents	Learning Speed				Number of Rounds	Payoff			
	λ_0	λ_1	λ_2	λ_3		P_0	P_1	P_2	P_3
4	0.2	0.9	0.1	0.8	50	42	34	37	29
					100	61	46	62	57
					150	100	79	101	97

Table 4.12: Results of games played by four-agent ToM₂ agents of different learning speed values

The five-agent groups further reinforce the same trend, with the performance gap between the agents becoming clearer as the number of rounds increases. This clarifies the effect of the number of agent factor on the agents' performance; in four-agent groups, even when the agent of a low LS value wins over agents of higher learning speeds, the margin between their scores was not as clear as in five-agent groups. For example, playing the game by a group of five agents (see Table[4.13]) with learning speed values of ($\lambda_0 = 0.1, \lambda_1 = 0.5, \lambda_2 = 0.8, \lambda_3 = 0.3, \lambda_4 = 0.7$), Lower learning speed agents consistently and obviously achieve higher payoffs (see Figure[4.27]).

These outcomes suggest that high-speed, high-reasoning-level agents tend to over-adapt, making them more predictable and thus easier to defeat.

Number of Agents	Learning Speed					Number of Rounds	Payoff				
	λ_0	λ_1	λ_2	λ_3	λ_4		P_0	P_1	P_2	P_3	P_4
5	0.1	0.5	0.8	0.3	0.7	50	23	37	32	69	30
		100				100	85	103	43	104	57
		150				150	130	151	88	141	96

Table 4.13: Results of playing the game by five ToM₂ agents of different learning speed values

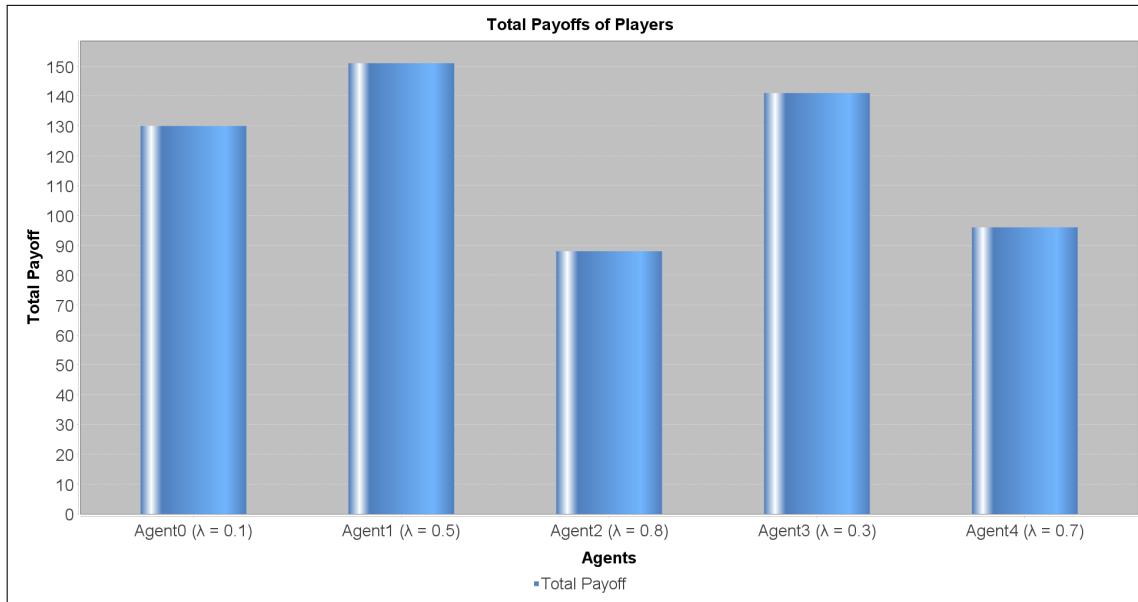


Figure 4.27: Total payoff comparison among five ToM₂ agents playing the game for 150 rounds using random LS values of: $\lambda_0 = 0.1$, $\lambda_1 = 0.5$, $\lambda_2 = 0.8$, $\lambda_3 = 0.3$, $\lambda_4 = 0.7$. The results illustrate a clear performance hierarchy where agents with lower learning speed values consistently achieve higher scores.

ToM₂ Results Conclusion:

1. ToM₂ agents with low learning speed value perform better than those playing using high values, contrary to the case in ToM₀ and ToM₁.
2. For two-agent groups, although agents with low learning speed value outperform, the gap in performance between both agents decreases directly with the decrease of the agents' learning speed value margin.
3. Starting from group size ≥ 3 in ToM₂, playing the game for 50 rounds is not enough to illustrate the agents' performance.

4. Other factors alongside the Learning Speed parameter influence the agents' performance in groups of ≥ 3 agents.
5. As the number of agents increases, the performance patterns become more consistent, making the effects of learning speed differences more recognizable.
6. In three-agent groups, an extended investigation dividing learning speeds into categorized groups clarified the understanding of how agents with different LS combinations affect the performance.
7. In four and five-agent groups, lower learning speeds provide a significant advantage for the agents in maintaining stable strategies, which is beneficial in competitive environments where high-speed, high-reasoning-level agents are more likely to overfit.

4.3.3.2 Groups of ToM₃ Agents

The analysis of ToM₃ agent groups closely aligns with the findings from the ToM₂ groups, with a few different observations becoming apparent. Notably, while third-order reasoning theoretically allows for better decision-making, the results demonstrate that this increased reasoning level does not consistently lead to better performance.

Similarly to ToM₂ agents:

- Identical learning speed values result in approximately zero scores for all agents.
- ToM₃ agents' performance depends on the combination of several factors instead of depending only on the Learning speed factor.
- 50 repeated rounds are not enough to illustrate the behavior of ToM₃ agents.
- The two-agent ToM₃ configuration revealed an important improvement over the lower ToM orders, which will be presented below.
- The results of three-agent ToM groups mirrored the ones we saw for ToM₂ agents, even the extended experiment conducted on three-agent ToM₂ groups[2] was also applied to ToM₃ agents and returned similar outcomes.
- The four-agent and five-agent ToM₃ groups also followed the same general observation from ToM₂ experiments. For instance, playing the game by five ToM₃ agents that have learning

speed values of $[0.2, 0.9, 0.1, 0.8, 0.6]$ for 100 rounds (see Figure[4.29]) shows that lower-speed agents outperform high-speed agents. Comparing results of the same game settings simulated on ToM_2 agents clarifies the adjacent outcomes of ToM_2 and ToM_3 agents, as illustrated in the following figure[4.28].

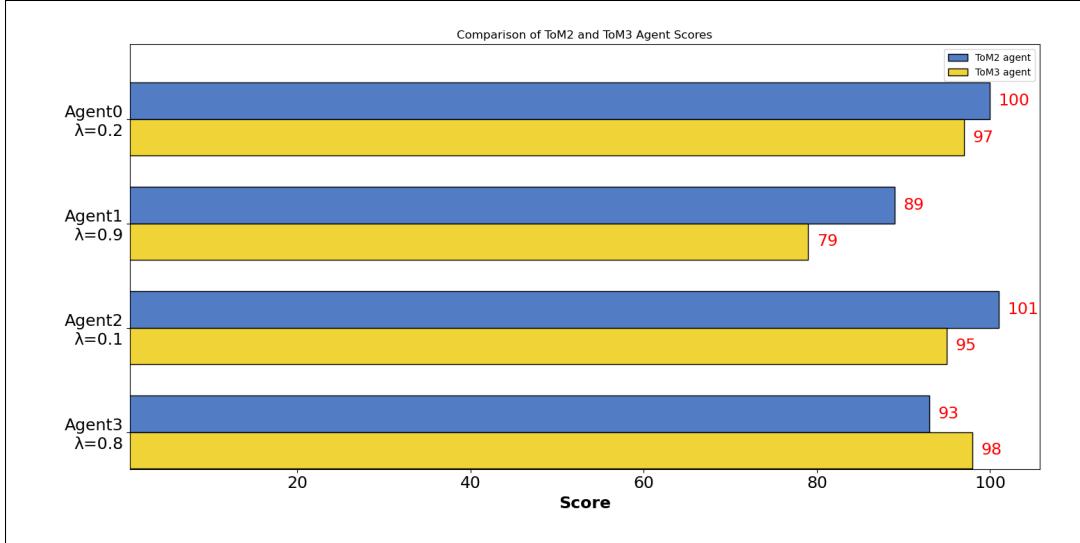


Figure 4.28: Comparative performance of ToM_2 & ToM_3 agents under identical simulation settings. Both groups consist of five agents assigned LS values of: $\lambda_0 = 0.2, \lambda_1 = 0.9, \lambda_2 = 0.1, \lambda_3 = 0.8, \lambda_4 = 0.6$, and the game was played for 100 rounds. The bar chart shows each agent's total score, highlighting the adjacent outcomes of both groups' agents.

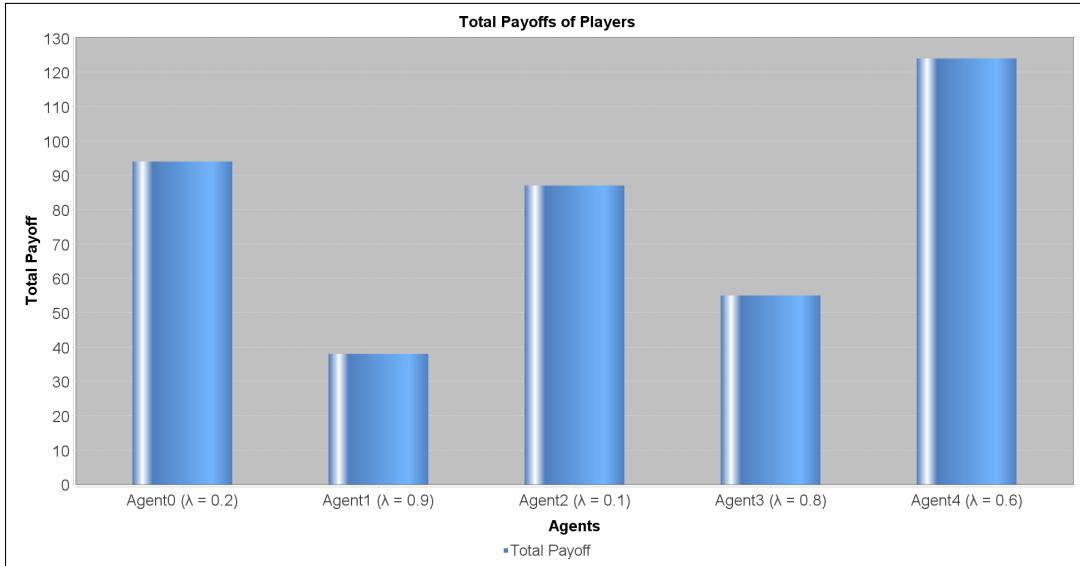


Figure 4.29: Total scores achieved by the same five-agent ToM_3 group playing the game for 100 rounds. The figure reflects how individual lower-speed agents generally outperform those with higher values.

Groups of two agents:

Although most simulations in two-agent ToM₃ groups were similar to those in ToM₂, notably, for high learning speed groups (i.e., $\lambda_0 = 0.7$, $\lambda_1 = 0.9$), the results are much better than those in ToM₂. Playing the game by two ToM₂ agents of adjacent high learning speed values resulted in almost zero for both (see Table[4.9]), while looking at the results of the same game by ToM₃ agents (see Table[4.14]), you can notice that the agent with the lower learning speed value outperform the other high-speed agent, following the general behavior of high-order agents. This outcome confirms that higher levels of reasoning enable agents of adjacent high learning speed in competitive environments to win the game, depending on their adaptation speed advantage (the lower outperforms the higher).

Number of Agents	Learning Speed		Number of Rounds	Payoff	
	λ_0	λ_1		P_0	P_1
2	0.2	0.9	50	8	5
			100	32	16
			150	37	23
2	0.1	0.5	50	16	7
			100	32	22
			150	55	25
2	0.7	0.9	50	16	2
			100	38	4
			150	48	3

Table 4.14: Results of two-agent ToM₃ simulations

ToM₃ Results Conclusion:

1. ToM₃ findings mirror the observations of ToM₂ agents in all game settings and different group configurations.
2. One new observation in two-agent ToM₃ groups shows the advantage of an additional reasoning level for agents of adjacent high learning speed values.
3. In general, a third ToM order does not show any extra benefits on the agents' performance that do not exist in the second order.

4.3.4 Groups of Different ToM Orders

To analyze the performance of different ToM order agents, a series of simulations of the Mod Game were conducted involving four agents per game, one agent for each ToM order: ToM₀, ToM₁, ToM₂, and ToM₃. Each game ran for 100 rounds. Tested agents were using different sets of learning speed $\lambda \in [0, 1]$, sometimes uniformly (all group's agents share the same LS value) and randomly (each agent of the group has a different LS value). In the following, you will find two sections, one for each group of experiments.

1. Uniform Learning Speed

In this scenario, all agents learn at the same speed. The goal is to observe how agents with different ToM levels compete, adapt, and perform when they play under the same parameter configurations.

a. Approach 1.1: Uniform Learning Speed $\lambda = 0.2$

Number of Agents	λ_0	λ_1	λ_2	λ_3	Rounds	P_0	P_1	P_2	P_3
4	0.2	0.2	0.2	0.2	100	1	0	195	148

The beginning was with testing the agents using a low learning speed value. Looking at the result in the table below, we can see that ToM₂ outperforms all agents from other orders with a payoff of 195, followed by ToM₃. ToM₀ and ToM₁ agents almost got no payoff, indicating that deeper reasoning is crucial when adaptation is slow in a group of different ToM orders. However, when looking at the pattern ToM₂ agent showed while playing this game, ToM₀ and ToM₁ agents showed similar choices' patterns, but the additional reasoning level of ToM₂ enabled it to outperform the other two agents and make the correct decisions for the proper actions (see Figure[4.30]). Although ToM₂ uses its zero-order behavior for most of the game, adapting to its ToM₀ and ToM₁ opponents, ToM₂ outperformed ToM₃, which mostly transferred between using its zero and first-order behaviors (see Figure[4.31]), trying to outperform other agents.

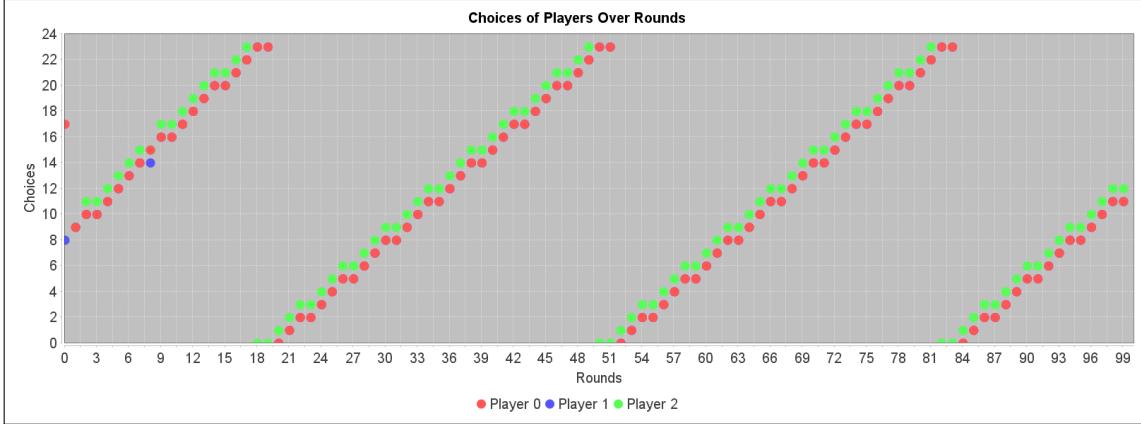


Figure 4.30: Comparison of choices made by ToM_0 , ToM_1 , and ToM_2 agents over 100 repeated rounds using the same learning speed value ($\lambda = 0.2$). Despite being assigned identical LS values, ToM_2 agents more effectively adapt to the game's strategic environment than other ToM agents.

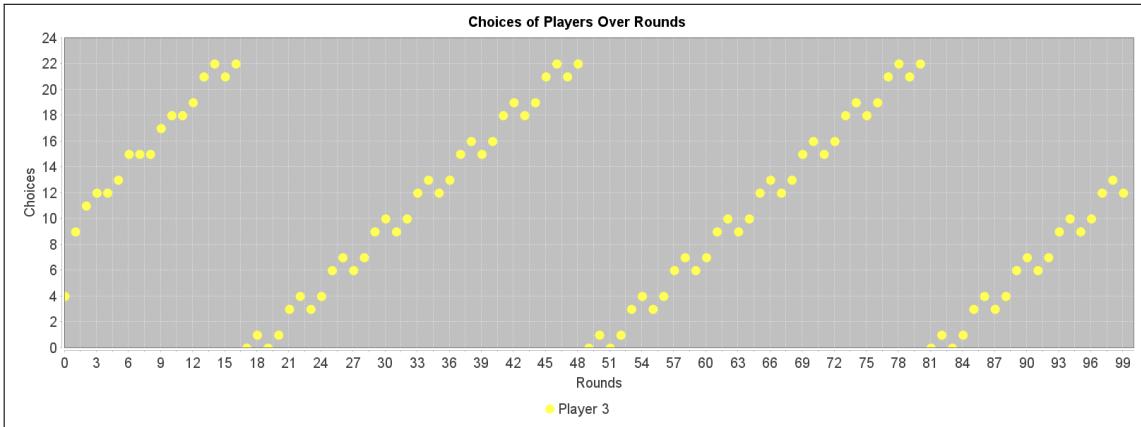


Figure 4.31: Isolated behavior of the ToM_3 agent from the same simulation shown in Figure 4.30. Its decision pattern fluctuates between zero and first-order reasoning.

b. Approach 1.2: Uniform Learning Speed $\lambda = 0.5$

Number of Agents	λ_0	λ_1	λ_2	λ_3	Rounds	P_0	P_1	P_2	P_3
4	0.5	0.5	0.5	0.5	100	2	2	180	148

Although the results of this game, shown in the table above, are similar to the previous approach's results, where ToM_2 agent wins with the highest score (180), ToM_3 follows with 148 points, while ToM_0 and ToM_1 remained far behind, but there is a difference in ToM_2 and ToM_3 agents chosen actions. An intermediate LS value > 0.2 enables

the agents' higher-order behaviors more. The ToM_2 agent shows much more of its first-order behavior (see Figure[4.32]) than it did in the previous approach experiments. Same for ToM_3 agent (see Figure[4.33]) that uses its second order in this game, which did not happen in the previous game when agents were playing using a very low learning speed.

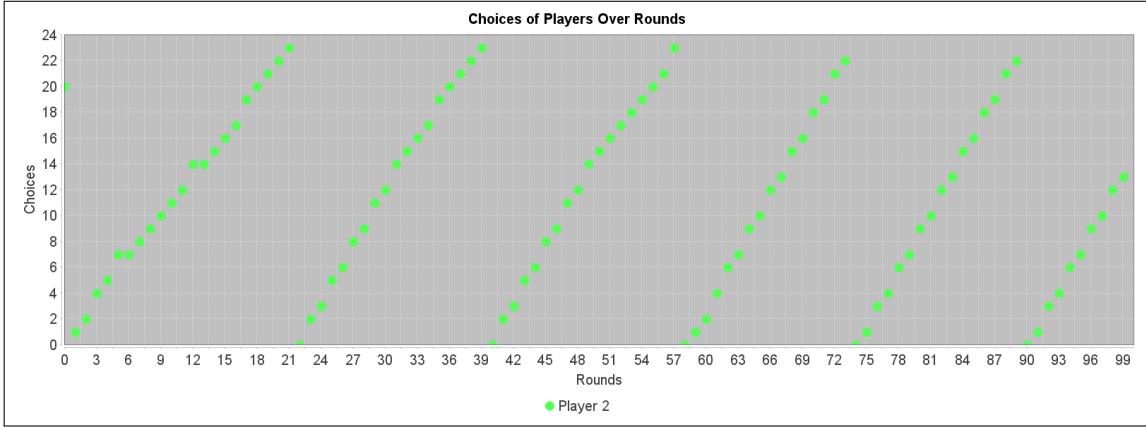


Figure 4.32: Action choices of a ToM_2 agent when participating in a group of four agents, each assigned a different ToM level. All agents were assigned the same learning speed value ($\lambda = 0.5$), and the game was played for 100 rounds. The ToM_2 agent consistently uses ToM_1 reasoning.

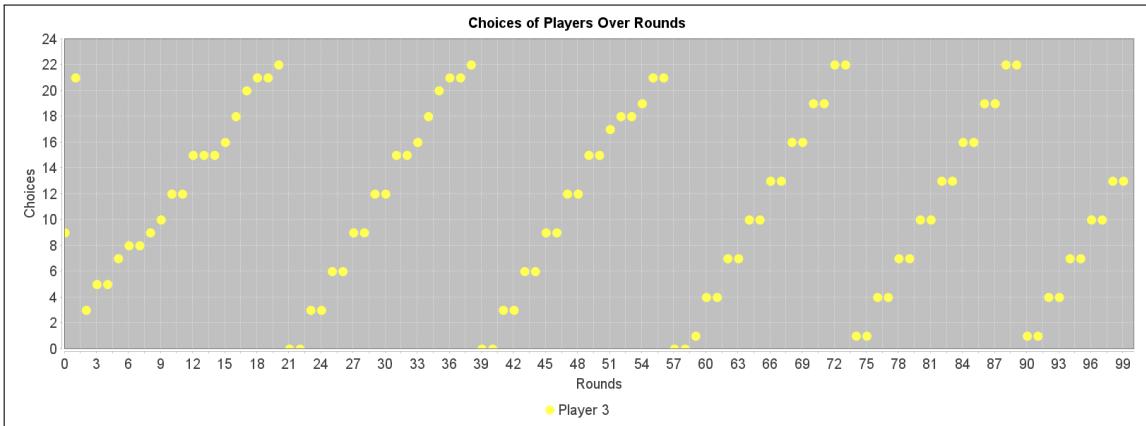


Figure 4.33: Action choices of a ToM_3 agent under the same experimental setup as in Figure 4.32. The agent also exhibits a ToM_2 reasoning behavior in this game, which was not observed in previous simulations with lower learning speed values. This change highlights the role of learning speed in enabling higher-order reasoning.

c. Approach 1.3: Uniform Learning Speed $\lambda = 0.8$

Number of Agents	λ_0	λ_1	λ_2	λ_3	Rounds	P_0	P_1	P_2	P_3
4	0.8	0.8	0.8	0.8	100	0	0	193	188

Even with a high learning speed, lower-order agents (ToM_0 , ToM_1) scored 0, while ToM_2 (193) and ToM_3 (188) continued to dominate. However, with higher learning speed, the second and third order agents are getting more adjacent scores than before. Looking at the agents' behavior (see Figure[4.34]), a higher value is not causing the agents to play using their extent abilities as expected; instead, even high-order agents are playing using their zero-order abilities, choosing the action that is 1 higher than the one selected in the last round, without repeating any action in consecutive rounds. Throughout the whole game, agents of zero and first order are acting the same, and agents of second and third order are acting the same as well as outperforming the other lower-order agents at the same rate, besides a few differences at the beginning where the agents' choices are still random.

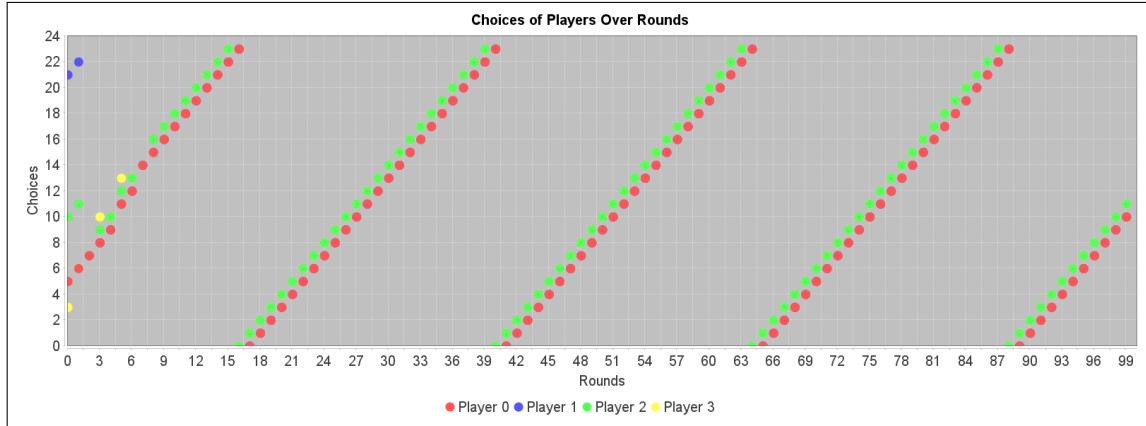


Figure 4.34: All agents actions pattern for playing the game over 100 rounds ($\lambda = 0.8$). Note that the choices of ToM_0 and ToM_1 agents are overlapping, as well as the choices of ToM_2 and ToM_3 agents, indicating similar behavior.

d. Approach 1.4: Zero Learning Speed ($\lambda = 0$)

With no learning speed, how will each agent perform?

Agents fixed their choices after round 1 on the same choice until the end of the game.

There was no specific winner; it was determined randomly based on the initial decision; the agent who makes the random correct guess that is +1 higher than another agent's,

wins the whole game. For example, the figure below [4.35] shows the agents' choices for one of the simulations, where you can see that the ToM₂ agent made a correct random action (the choices in green) that was +1 higher than ToM₃ agent (the choices in yellow) enabled the agent to get 1 point each round and win the game (see Figure[4.36]). Otherwise, all agents scored zero. This reflects a total lack of adaptability and confirms the necessity of learning for competitive advantage.

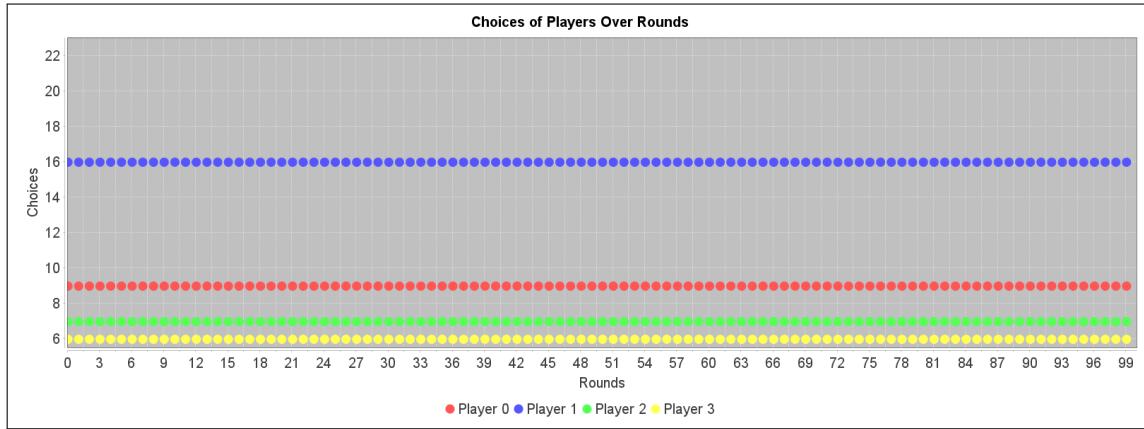


Figure 4.35: Agents' choices for playing the game with no learning speed over 100 rounds. The ToM₂ agent randomly chose number 7, which is +1 higher than the ToM₃ agent's choice.

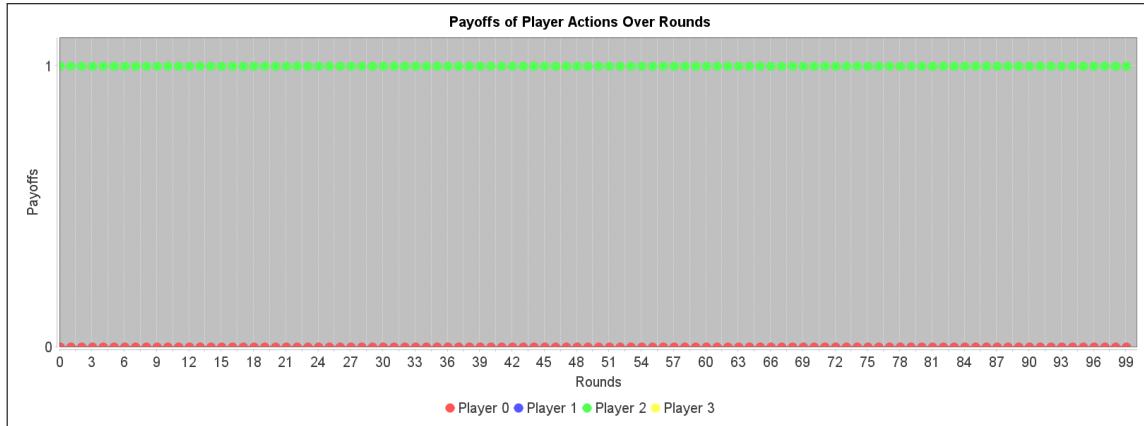


Figure 4.36: Agents' payoffs. Note that the payoff of ToM₀, ToM₁, and ToM₂ agents is overlapping at the value zero, while the ToM₂ agent gets a payoff of 1 in all rounds.

e. Approach 1.5: Maximum Learning Speed ($\lambda = 1$)

A maximum learning speed means that the agent adapts only depending on the most recent actions (see section[2]), ignoring its memory about its opponents. In this case,

all agents learned maximally quickly. While ToM_2 and ToM_3 agents still led, ToM_0 and ToM_1 improved considerably (see Figure[4.37]), and they did not earn these points through random choices in the first few rounds. ToM_0 and ToM_1 agents gained those points by outperforming other agents through the game in some rounds.

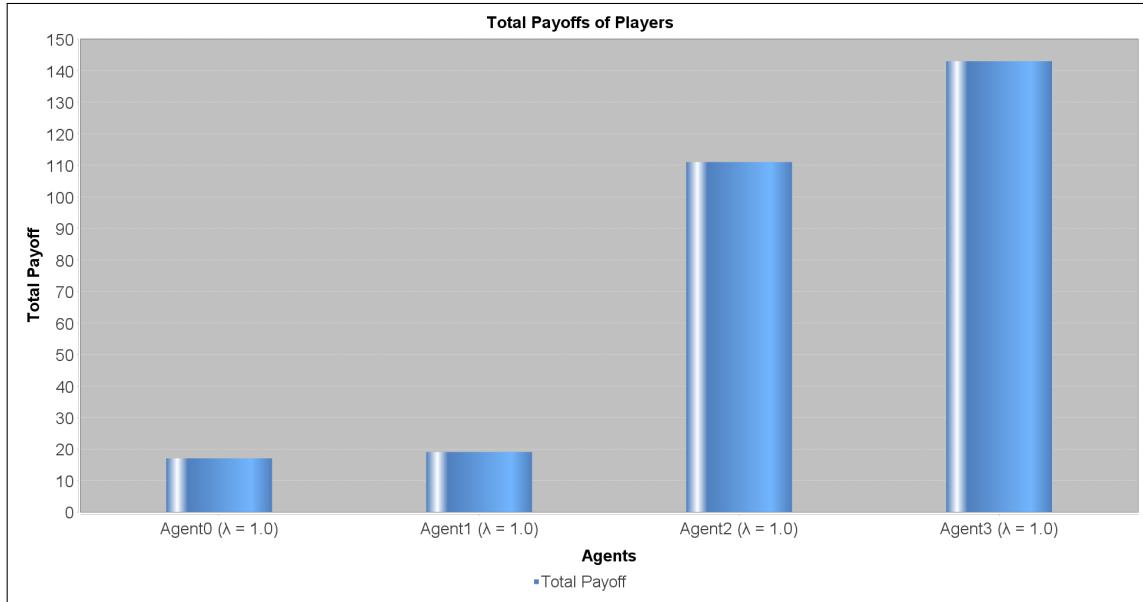


Figure 4.37: Total payoff comparison among four agents with different ToM levels, all playing the game with a maximum learning speed ($\lambda = 1.0$). ToM_2 and ToM_3 agents significantly outperformed their lower-order counterparts. The final scores were: $\text{ToM}_0 = 17$, $\text{ToM}_1 = 19$, $\text{ToM}_2 = 111$, $\text{ToM}_3 = 143$, illustrating the advantage of higher-order reasoning even under fast-adapting conditions.

2. Random Learning Speeds

In this scenario, each agent was assigned a different learning speed. The goal was to examine how variant learning speed values could affect the competition among different ToM orders, especially the performance of lower ToM orders; will it make any difference if higher/lower values are assigned to them under the same configurations of the previous experiments; rounds' number = 100 and groups of four different ToM agents (ToM_0 , ToM_1 , ToM_2 , and ToM_3).

- a) **Approach 2.1: Random LS = [0.9, 0.7, 0.3, 0.4]** (High for ToM_0 & ToM_1 , low for ToM_2 & ToM_3 agents).

Number of Agents	λ_0	λ_1	λ_2	λ_3	Rounds	P_0	P_1	P_2	P_3
4	0.9	0.7	0.3	0.4	100	6	9	153	145

Despite slower adaptation, ToM_2 and ToM_3 still outperformed the faster-learning lower-order agents with a huge difference. What can be noticed here is the behavior itself of higher-order agents. For example, looking at ToM_3 agent behavior, we can see (see Figure[4.38]) the affection of low learning speed value, where for many times during the game, the agent tends to repeat its choices multiple times before taking a new action, which is the opposite of what happens when it is assigned a high learning speed value as you will see in the next approach. However, in both cases, high-order agents outperform lower ones, but with a different behavior pattern.

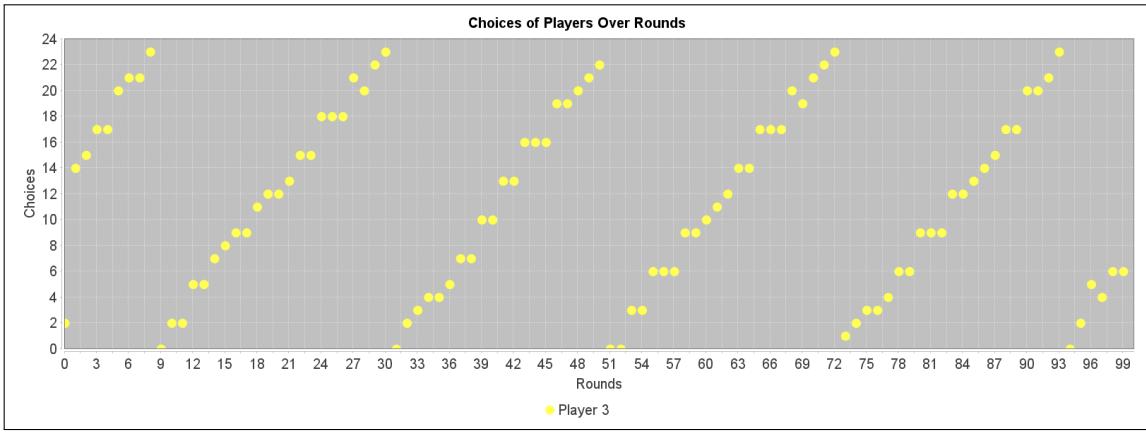


Figure 4.38: Choices distribution of a ToM_3 agent over 100 rounds when assigned a low learning speed ($\lambda_3 = 0.4$). The agent demonstrates a repetitive behavior pattern, frequently selecting the same action for multiple consecutive rounds before transitioning to a new one. Despite the slow adaptation, the higher-order reasoning of the ToM_3 agent still leads to superior performance over lower-order agents.

b) **Approach 2.2: Random LS = [0.3, 0.4, 0.7, 0.9]** (High for ToM_2 & ToM_3).

Number of Agents	λ_0	λ_1	λ_2	λ_3	Rounds	P_0	P_1	P_2	P_3
4	0.3	0.4	0.7	0.9	100	3	4	161	144

Increasing ToM_2 & ToM_3 learning speeds improved their scores a bit, but their behavior is notable. Continuing with ToM_3 as an example, the agent here repeats its choices much less than in the previous approach (2.1) when it had a low LS, and at round 48, it recognizes the pattern that will guarantee its winning (see Figure[4.39]) with a fixed payoff until the end of the game. The same goes for ToM_2 regarding the payoff, but the difference is that the ToM_2 agent does not repeat any of its choices for two rounds in a row (see Figure[4.40]).

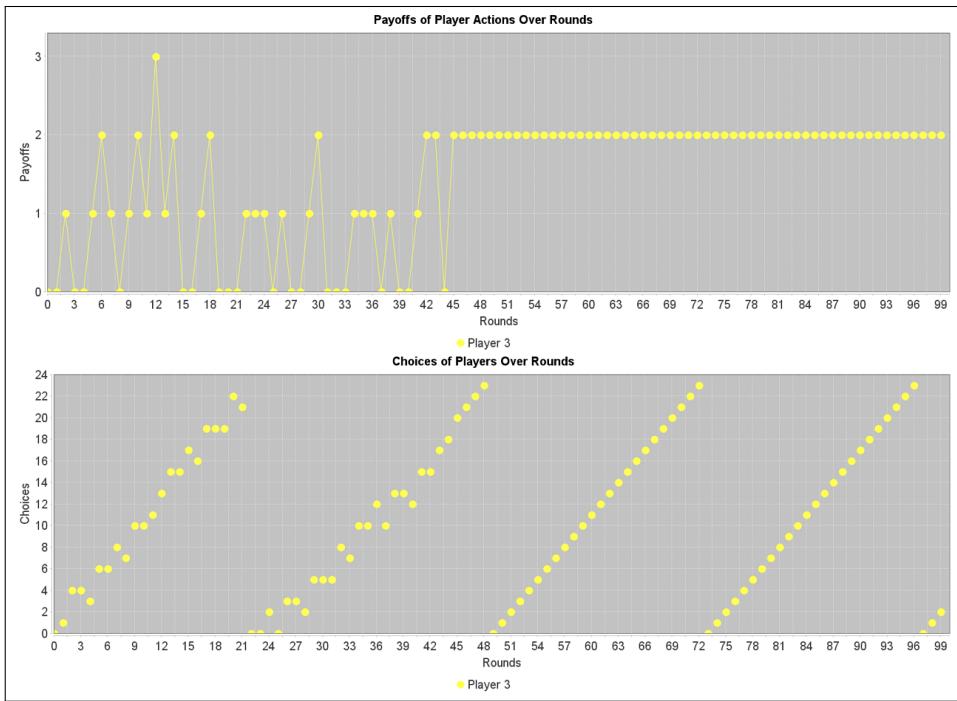


Figure 4.39: At the bottom display, you can see the ToM₃ agent choices' pattern when assigned a high learning speed (0.9). At the top display, you can see the fixed payoff the agent gets from each round (2 points each round, resulting from ToM₃ outperforming ToM₀ & ToM₁)

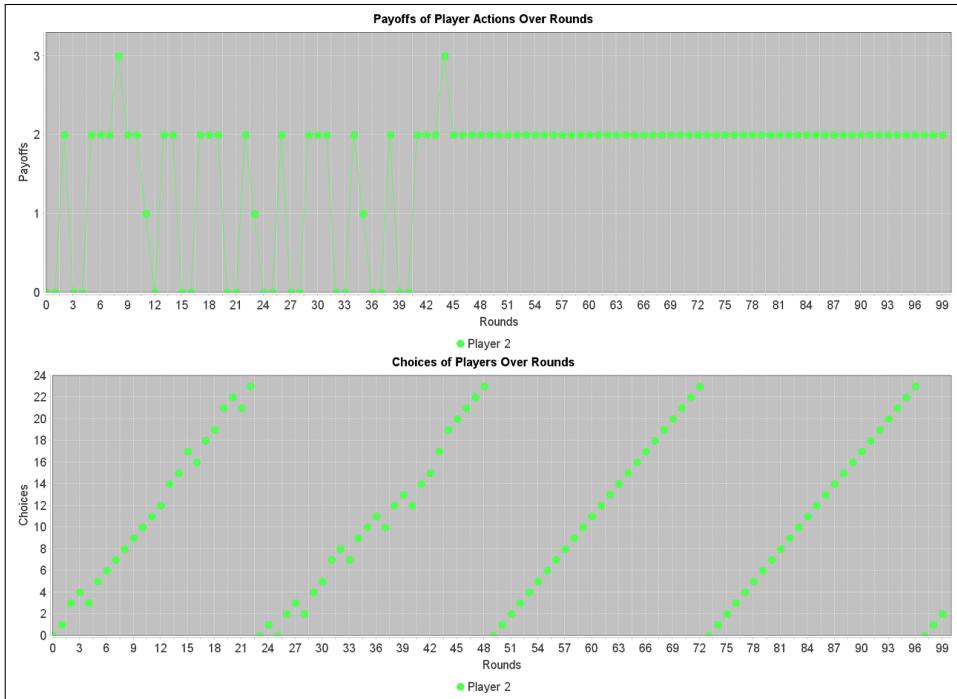


Figure 4.40: At the bottom display, you can see the ToM₂ agent choices' pattern when assigned a high learning speed (0.7). At the top display, you can see the agent's fixed payoff from each round (2 points each round, resulting from ToM₂ outperforming ToM₀ & ToM₁).

c) **Approach 2.3: Random LS = [0.7, 0.4, 0.9, 0.3]**

Number of Agents	λ_0	λ_1	λ_2	λ_3	Rounds	P_0	P_1	P_2	P_3
4	0.7	0.4	0.9	0.3	100	1	0	191	147

In this approach, the big difference between learning speed values of ToM₂ & ToM₃ agents enabled ToM₂ to outperform ToM₃ with a significant margin (191 vs. 147). However, this does not mean that ToM₂ had better reasoning abilities than ToM₃, because when looking at both agents' behavior (see Figure[4.41]), you can notice that during the game, ToM₃ either chose a choice that is +1 higher than ToM₂ or the same choice, which means that ToM₃ won a point over ToM₂ multiple times even though ToM₂ has a higher learning speed, while all the points ToM₂ got were over ToM₀ & ToM₁ only.

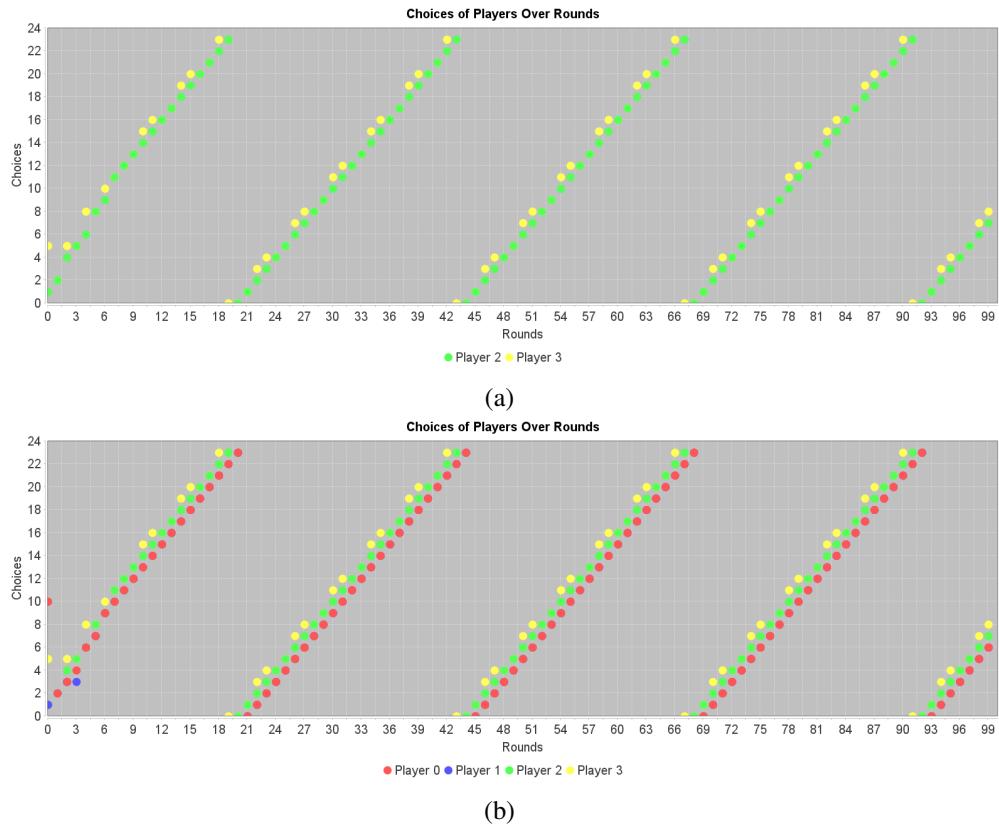


Figure 4.41: The top image shows ToM₂ & ToM₃ behavior, you can see that ToM₃ choices (the yellow dots) either overlap with ToM₂ agent's choices (the green dots) or increase it by 1. In the bottom image, you see all agents' behavior, which shows how ToM₂ wins over ToM₀ & ToM₁ agents only, while ToM₃ wins over all agents throughout the game.

4.3.4.1 Comparison and Interpretation

In all tested settings, high-order ToM agents (ToM_2 & ToM_3 agents) outperformed the low-order ToM agents when playing the game under similar and variant settings. ToM_3 agents consistently performed well but did not surpass ToM_2 . Since the experiments done for this section of playing the game by different ToM order agents did not fully consider all possible configurations, theories or findings cannot be 100% validated. However, keep in mind the fact that high-order agents, as shown earlier through the experiments, play using their $(l - 1)$ order behavior for most of the game, with some actions using their max reasoning abilities.

The existing results suggest that the second-order ToM performs best among other reasoning orders' agents and can achieve strong predictive accuracy. While ToM_3 can simulate more deeply, it may overfit or be misled by lower-order agents.

These results comply with the finding from Prof. Harmen's paper [21] that, similarly to human participants, agents give better results by increasing their reasoning depth up to the second level when playing a competitive game for a large number of rounds, contrary to what has been proved by other searches on playing the rock-paper-scissors game where the game's is only repeated for limited times [36], where participants reason at lower level.

4.3.5 Common Observations Among Different ToM Orders

- **Random Behavior - High ToM orders**

Across multiple simulations, an unexpected behavioral pattern has been observed among higher-order ToM agents, such as those of ToM_2 or beyond. In specific scenarios, agents with low learning speed (e.g., $\lambda \leq 0.3$) show random patterns of their behavior that suggest incomprehensible, inconsistent reasoning, deviating from the typical trajectories of their order. This phenomenon was notably visible in multiple simulations of ToM_2 & ToM_3 agents' games. For instance, a game of a four-agent ToM_3 group was played for 150 rounds. The agent of the low learning speed values (specifically $\lambda = 0.1$) displayed irregular choices across rounds (see Figure[4.42]). The same observation appeared in a similar experiment involving five-agent ToM_2 ($\lambda_0 = 0.1, \lambda_1 = 0.5, \lambda_2 = 0.8, \lambda_3 = 0.3, \lambda_4 = 0.7$). Here again, the agents with ($\lambda_0 = 0.1$ & $\lambda_3 = 0.3$) demonstrated comparable volatility in their choice

patterns (see Figures[4.43][4.44]).

These findings suggest that this noise may be a compound effect of multiple factors combined: the number of agents in the game, the levels of the agents' reasoning being simulated during the game, and the large number of repeated game rounds. Together, these parameters can overwhelm a slow learner's belief creation process, introducing randomness into their behavior that would blur the integrity of the game's outcome. According to that, we can conclude that even when agents are equipped with advanced reasoning models, low learning speed may act as a bottleneck, impairing their ability to respond meaningfully to dynamic environments with overloaded requirements.

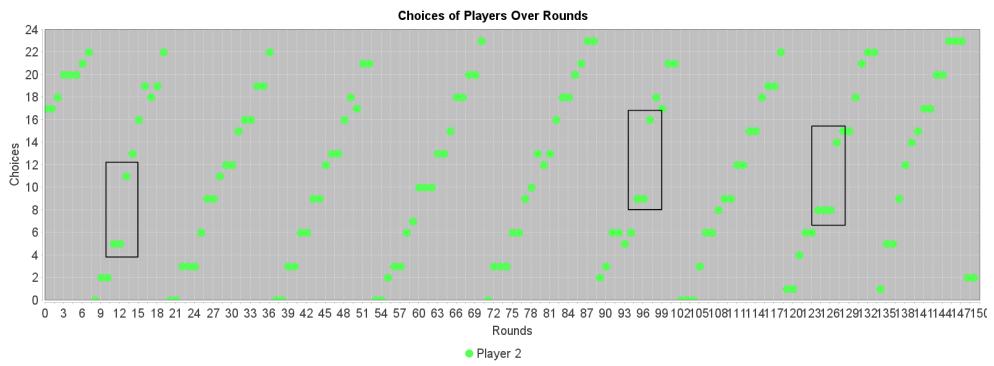


Figure 4.42: Illustration of irregular behavior exhibited by a ToM_3 agent assigned a low learning speed value of $\lambda = 0.1$ while playing in a group of four agents over 150 rounds. The agent's choices display unpredictable and inconsistent variations, diverging from the expected pattern of reasoning.

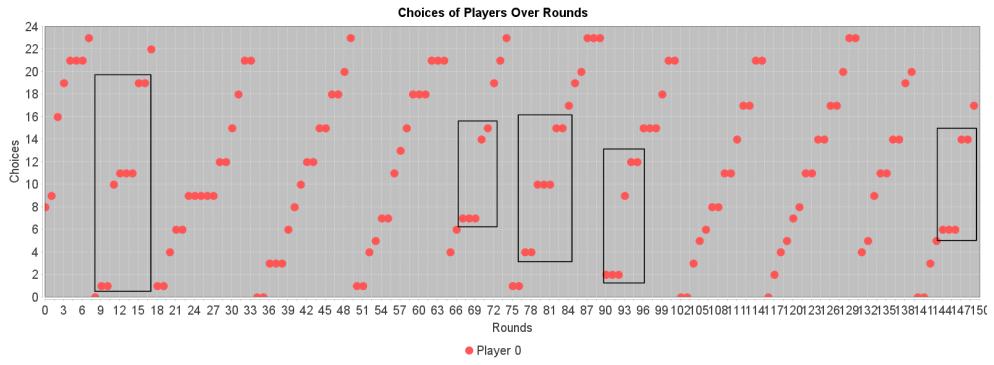


Figure 4.43: Illustration of irregular behavior exhibited by a ToM_2 agent with a learning speed of $\lambda_0 = 0.1$ in a five-agent game over 150 rounds. The agent exhibits similarly unpredictable behavior patterns as seen in Figure 4.42, indicating that low learning speeds can lead to deviations from logical strategic reasoning.

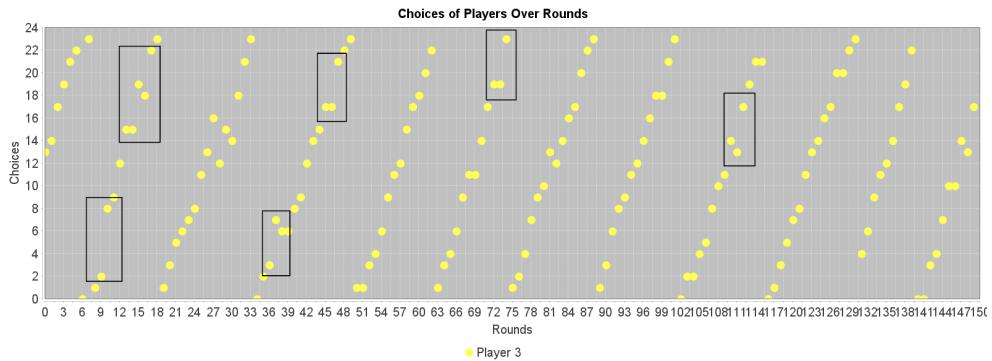


Figure 4.44: Illustration of irregular behavior exhibited by a ToM₂ agent with a learning speed of $\lambda_0 = 0.3$ in a five-agent game over 150 rounds. The agent exhibits similarly unpredictable behavior patterns as seen in Figures 4.42 and 4.43, indicating that even moderately low learning speeds can compromise the agent's decision stability.

- **Actual used level extent - All ToM orders**

One consistent pattern observed in the behavior of all ToM agents is their tendency to operate primarily using lower-order reasoning levels, while making very limited use of their full order capabilities. Therefore, in a simulation for ToM₁ groups, they tend to use their zero-order capabilities more than their first-order capabilities. For ToM₂ groups, they use their zero and first capabilities more than using their max order extension of the second reasoning level, and so on for higher orders. For example, but not limited to, a simulation of the game involving four ToM₃ agents over 100 rounds, where each agent had a different learning speed: $\lambda_0 = 0.7$, $\lambda_1 = 0.9$, $\lambda_2 = 0.1$, and $\lambda_3 = 0.2$. As illustrated in the visualizations (see Figure[4.45]), agent₀ (subfigure[4.45a]) utilized its third-order ToM only twice, while agent₁ (subfigure[4.45b]) never used it at all. Despite having the slowest learning speed, agent₂ (subfigure[4.45c]) used third-order reasoning once. Similarly, agent₃ (subfigure[4.45d]) used its highest reasoning extent only twice.

The reasons behind such behavior might vary, and they are unclear to deduce. It could be due to multiple reasons related to the environment, the game's complexity, or uncertainty in modeling. However, this finding clarifies that access to deeper reasoning does not necessarily translate into its frequent use. Even when the agents can simulate others' reasoning up to two or three levels deep, they tend to use shallower mental capabilities.

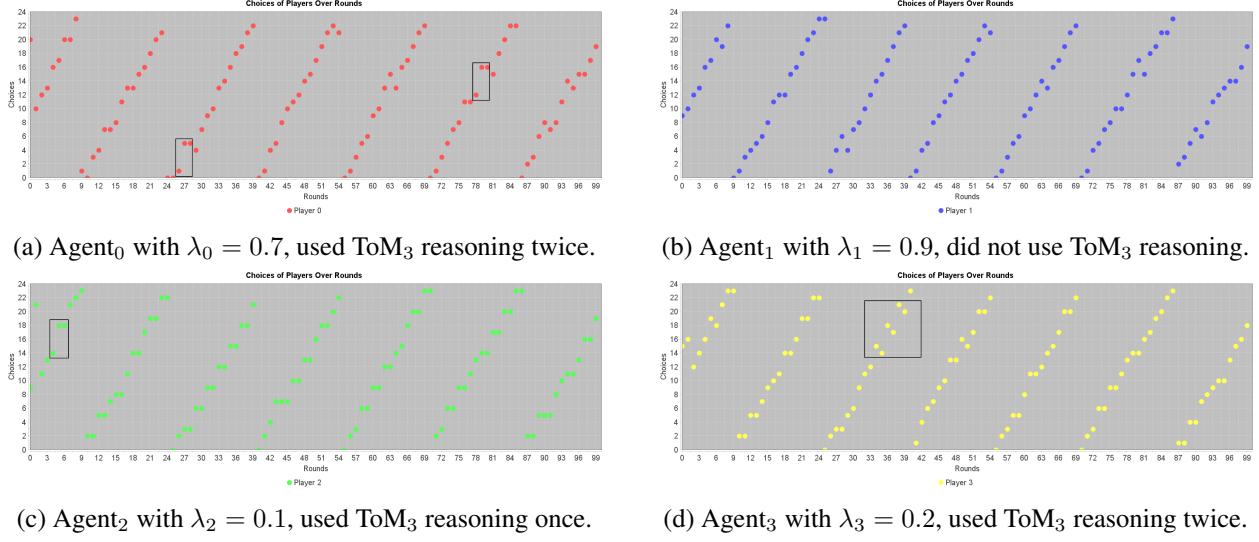


Figure 4.45: Individual choice patterns of four ToM₃ agents with different learning speeds over 100 rounds. Each subplot shows the behavior of one agent. This visualization highlights the tendency of ToM₃ agents to rely more on lower-order capabilities while occasionally utilizing higher-order reasoning.

4.3.6 Summary

To analyze how agents of each specific ToM order perform in defined environments and to compare these performances in order to answer this thesis's questions, a bunch of experiments were conducted to explore the influence of various factors on the agents' performance within each ToM group. Specifically, the number of rounds, the number of agents per group, and the learning speed values.

Groups of the same ToM order:

ToM₀ and ToM₁ agents showed identical behavior in all game configurations. The learning speed most notably affected the agents' performance among the other tested parameters. Groups of agents (**low and high ToM orders**) that are assigned identical learning speed values always result in zero payoff for all agents. These behaviors can be attributed to the fact that agents of the same reasoning level and who learn at the same rate would behave identically and make the same decisions, due to the correct predictions by all about each other. Having one of the agents with the lowest values in the learning speed range $\lambda = [0, 0.1]$ makes the agents vary in their predictions, trying to outperform the weak agent that can be trounced, while also competing with other higher-value agents to get the highest score. In general, in ToM₀ and ToM₁ orders, agents with high learning speed

value outperform those with low learning speed value. The number of agents per group was another factor that significantly impacted the agents' performance. In groups of 5 agents, increasing the number of competitors enabled the agents to maintain stable strategies, each according to its learning speed capabilities, erasing the ambiguity that appeared in smaller-sized groups. This suggests that, maybe, for lower-order reasoning groups, increasing environment complexity benefits the agents in stabilizing their behavior and generating better-quality beliefs about their opponents' actions.

Experiments on agents of higher ToM orders, specifically ToM_2 & ToM_3 , showed that the agents' performance is not *only* related to the learning speed factor; other parameters (number of agents and number of repeated rounds) combined with the learning speed influence the agents' behavior. Contrary to what the simulations revealed in lower orders, agents with lower learning speed values outperform agents of higher learning speeds. This outcome can be translated as follows: the deeper the reasoning level is, the better to process the information at a lower speed, which means depending more on the memory than the most recent information of the game. Also, the number of rounds as well as the number of agents per group parameters significantly affected the stabilization of the agents' performance, according to their adaptation speed. For example, in most simulations, playing the game for 50 rounds was never enough to show the exact performance of a ToM_2 or ToM_3 agent; repeating the game for 100 and 150 rounds in groups of four or five agents enabled a clear vision of the gap between the agents' performance. Therefore, the performance of agents that use higher reasoning levels depends on the combination of factors in the game configuration and cannot be determined due to a single parameter value.

Groups of different ToM order:

To check how agents of different reasoning levels perform under the same environment configurations, experiments considering different requirements were carried out on groups of four agents, each of a different order of ToM (ToM_0 , ToM_1 , ToM_2 , ToM_3). The findings showed that high-order ToM agents outperformed the low-order ToM agents when playing the game under all tested settings. The best player was the ToM_2 agent, which in most cases achieved accurate predictions over all other agents, including the ToM_3 agent. In specific scenarios, the ToM_3 agent might tend to overfit or get distracted by other orders' agents.

Finally, some **common observations** that apply to all ToM agents were presented and discussed, including the actual extent to which a ToM agent uses while playing a competitive game like the Mod Game. It was observed that an agent of the level ToM_l rarely uses its max capabilities and tends to behave more according to its $[0, \dots, l - 1]$ orders. Also, higher orders are more exposed to the risk of random behavior, especially when playing using a very low learning speed and reasoning at a high level of ToM. The explanation of this finding was attributed to the noise that could be caused by the complexity of the environment that contains multiple competitors, reasoning at high levels, and for a large number of rounds.

Chapter 5

Discussion and Concluding Remarks

5.1 Discussion

Experimental studies on the ToM in competitive games using agent-based models primarily focused on computing the averages of the outcomes of the agents' groups. This work aimed to see how the individual agents of these groups react under different conditions, either when the agent plays against others of the same or different reasoning level. Since the studies in this field aim to understand and enable machines to think in a similar way to humans in order to involve them more in various life aspects [10] [37], from daily activities to executing complex tasks, it is necessary to know how these agents' behavior gets impacted by different factors in the environment.

It was shown by Camerer [38] that iterated reasoning supports learning models that follow an adaptive learning approach to converge to the Nash equilibrium at the same rates as humans. In later research [21] [35], experiments were conducted using agent-based models as well as human participants to explore whether humans and agents adopt the same ToM theories regarding iterated reasoning in competitive games with mixed-strategies equilibrium. A specific simulated game was the Mod Game, which we have delved into investigating its agent-based model's results in this thesis.

In previous work [39], the humans' depth of iterated reasoning was determined to be 1.5 steps, approximately equal to the First-order reasoning level. However, in games repeated for many rounds, participants showed deeper reasoning levels (up to the second-order) [35]. Simulations of those experiments on agents showed similar behavior to that observed in humans [21]. Our findings aligned

with the general theories of each ToM order, adding multiple essential observations. Examining the agents under varied environmental conditions that can surround the agent, the advantage of using the second-order ToM was clear. However, reasoning on higher levels than the second did not show a critical progress that can be beneficial; on the contrary, it showed random behaviors due to the complexity of the environment. In cooperative games, agents most frequently reason at the first-order ToM (ToM_1) to achieve effective collaboration [40] [31], while in competitive games like the Mod Game, reasoning at the first level was not mostly effective. This finding can be explained by the observation that agents of ToM_l play using their lower abilities for most of the game; thus, a ToM_1 agent reacts most of the time as a ToM_0 agent. Moreover, for all ToM order agents, having agents of the same reasoning level learning at the same speed rate in a competitive environment resulted in a 'no-win' situation for any of the players. When the competitors reason at different ToM levels, agents of high-order ToM (specifically, ToM_2 agents) completely defeat others of lower orders, who get approximately zero payoffs, and perform better than ToM_3 agents. This finding does not align with what was demonstrated in a previous study on cooperative settings; pairs of agents from different ToM orders in one group perform better than those of the same order, where ToM_1 agents showed the best performance among other ToM order agents [40]. Nevertheless, Interesting outcomes of the various factors' impact on the agents of all ToM orders were revealed, illustrating the observations from the agents' behaviors. For instance, agents of lower ToM orders showed a better performance when they played using high learning speed values. In contrast, high-order agents performed much better with low learning speed values. However, other factors, alongside the learning speed, affected the agent's performance (i.e., number of repeated rounds and number of agents per group), which clarifies that the combination of multiple elements of the environments plays a crucial role in determining the performance's quality that agents can adopt while playing. Therefore, studying the environmental elements' impact on the agents is a vital aspect to consider when developing agent-based models to simulate a real-life experience involving humans, who are also affected by many external influences.

5.2 Future Work

Deeper and more complex examinations can be conducted to discover more about the agents' behavior in different environments and check how it can align with real-life examples. Below you can find some of the future potential ideas related to this work:

1. A suggestion is to conduct the experiments with human participants, following the approach used for each order, to determine whether the results would be identical or different from what we had with agents. This will give a wider idea about similarities and differences between both and prove whether this agent-based model simulates real-life environments.
2. Although some experiments on groups of different ToM levels were held in this research, this was not the primary focus of the work, and deeper experiments with more varying parameters would provide a beneficial vision of the behavior of different agents when playing the game under various conditions.
3. Extending this work by including noise or imperfect information, to examine how agents with incomplete or incorrect beliefs perform compared to idealized models.
4. A version of the Mod Game; contains signaling by agents on the choice they tend to select, considers cheating and cooperative behaviors, called 'The Mod-Signal Game', is also an exciting experiment to be examined on an agent-based model.

5.3 Conclusion

The performance of ToM agents in competitive game environments is shaped by the intersection of multiple factors: learning speed, group size, number of repeated rounds, and the agent's reasoning level. For low-order ToM agents (ToM_0 and ToM_1), learning speed is the most influential factor; faster learners consistently outperform slower ones. However, increasing the number of agents and repeating the game for more rounds reduces ambiguity and allows all agents to better stabilize their performance.

In contrast, for higher-order ToM agents (ToM_2 and ToM_3), slower learners usually outperform

faster ones, but their performance does not rely solely on learning speed. ToM₂ agents, in particular, achieve the best performance overall when competing with agents of different ToM orders, effectively balancing prediction accuracy and adaptation speed. This aligns with the depth of reasoning humans showed in similar game settings (up to the second order) [35]. ToM₃ agents, while capable of a higher level of reasoning, often perform less well than expected when overusing their full reasoning extent, especially in complex game settings.

Finally, all ToM agents tended to rely more on their lower-order reasoning, using their higher-order strategies only occasionally. Therefore, optimal performance is not achieved through a single factor (i.e., higher reasoning level), but through the right combination of reasoning depth, learning speed, group dynamics, and environmental settings.

Appendix A

ChatGPT interventions in this thesis

For the sake of transparency and academic integrity, I would like to acknowledge the use of OpenAI's ChatGPT as a formatting and writing enhancement assistant while preparing this thesis to improve the clarity, structure, and readability of complex ideas. ChatGPT served as an effective tool in rephrasing dense or informal segments into more academically appropriate language. For more details, please find the following examples:

1. Original version:

“Although the same settings mentioned in the main paper were used to regenerate the results, the values of one of the parameters were not determined exactly, which is the agents’ learning speeds. It was only mentioned that the agents are playing using ‘random learning speed values’. However, random learning speed values were also used to regenerate the results, but we cannot validate whether they are the same ones that were used by the authors or not.”

ChatGPT’s version:

“Although the same experimental settings described in the main paper were used to reproduce the results, one parameter—the agents’ learning speeds—was not explicitly specified. The original work only noted that agents were assigned “random learning speed values.” While random values were similarly applied in our replication, it is not possible to confirm whether these match those originally used by the authors.”

2. Original version:

“In the experiments, multiple values for the number of agents per group were used. For each

ToM order, the game was run for groups of two, three, and four agents. As well as validating the main paper results [21] for groups of five agents. ”

ChatGPT's version:

“In the experiments, various group sizes were tested by using different numbers of agents per group. For each ToM level, the game was run with groups of two, three, and four agents. Additionally, groups of five agents were used to replicate and validate the findings reported in the main study by de Weerd et al [21].”

In addition to using it as a writing assistant, ChatGPT was especially helpful with LaTeX-related tasks. For example, writing captions for figures and tables that would not appear in the List of Figures/Tables, using the \caption [short] {long} structure. It also provided practical guidance on resolving LaTeX errors that I encountered but could not fix independently.

Advantages:

Although all structural decisions, research interpretations, and content judgments were made solely by me, I believe it is important to acknowledge the role this tool played in improving the quality of the thesis writing and saving the time needed in resolving issues during the writing process.

Disadvantages/Concerns:

Using ChatGPT or any other chatbot must be done with extreme caution, and be highly aware of its risks. Even if the task was as simple as rephrasing, sometimes it might confuse information, provide incorrect or misleading information, or even make the idea more complicated to read than the original version. If you provide big paragraphs to the tool, essential details in your content might be ignored and neglected in its refined version. Therefore, re-reading and modifying its response to suit your desired outcome and overall writing style would be the best approach when working with Chatbots. Moreover, I am still concerned about the security level of these tools, even though of the founders' claims that using a paid version would guarantee your provided chats security and that the data will not be used in training the model, I still recommend keeping the highly sensitive details (personal data, original research findings, etc) off the chatting space.

Bibliography

- [1] David Premack and Guy Woodruff. Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences*, 1(4):515–526, 1978.
- [2] Heinz Wimmer and Josef Perner. Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children’s understanding of deception. *Cognition*, 13(1):103–128, 1983.
- [3] Josef Perner and Heinz Wimmer. “john thinks that mary thinks that ...” attribution of second-order beliefs by 5- to 10-year-old children. *Journal of Experimental Child Psychology*, 39(3):437–471, 1985.
- [4] Beate Sodian, Christian Hülsken, and Claudia Thoermer. The self and action in theory of mind research. *Consciousness and Cognition*, 12(4):777–782, 2003.
- [5] Simon Baron-Cohen, Alan M Leslie, and Uta Frith. Does the autistic child have a “theory of mind”? *Cognition*, 21(1):37–46, 1985.
- [6] Yoav Shoham. Agent-oriented programming. *Artificial Intelligence*, 60(1):51–92, 1993.
- [7] Arjun Chandrasekaran, Deshraj Yadav, Prithvijit Chattopadhyay, Viraj Prabhu, and Devi Parikh. It takes two to tango: Towards theory of ai’s mind, 2017.
- [8] Neil C. Rabinowitz, Frank Perbet, H. Francis Song, Chiyuan Zhang, S. M. Ali Eslami, and Matthew Botvinick. Machine theory of mind, 2018.
- [9] Jessica Williams, Stephen M. Fiore, and Florian Jentsch. Supporting artificial social intelligence with theory of mind. *Frontiers in Artificial Intelligence*, Volume 5 - 2022, 2022.

- [10] Rohan Chandra, Aniket Bera, and Dinesh Manocha. Stylepredict: Machine theory of mind for human driver behavior from trajectories, 2020.
- [11] Tomislav Petković, Ivan Marković, and Ivan Petrović. Human intention recognition in flexible robotized warehouses based on markov decision processes, 2018.
- [12] Qiaosi Wang and Ashok K. Goel. Mutual theory of mind for human-ai communication, 2024.
- [13] Terence X. Lim, Sidney Tio, and Desmond C. Ong. Improving multi-agent cooperation using theory of mind. *CoRR*, abs/2007.15703, 2020.
- [14] Harmen De Weerd, Rineke Verbrugge, and Bart Verheij. Higher-order social cognition in rock-paper-scissors: A simulation study. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 1115–1116, 2012.
- [15] Ismael T. Freire, Xerxes D. Arsiwalla, Jordi-Ysard Puigbò, and Paul F. M. J. Verschure. Modeling theory of mind in multi-agent games using adaptive feedback control. *CoRR*, abs/1905.13225, 2019.
- [16] Chris Baker, Rebecca Saxe, and Joshua Tenenbaum. Bayesian theory of mind: Modeling joint belief-desire attribution. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, 2011.
- [17] Dung Nguyen, Svetha Venkatesh, Phuoc Nguyen, and Truyen Tran. Theory of mind with guilt aversion facilitates cooperative reinforcement learning. In Sinno Jialin Pan and Masashi Sugiyama, editors, *Proceedings of The 12th Asian Conference on Machine Learning*, volume 129 of *Proceedings of Machine Learning Research*, pages 33–48. PMLR, 18–20 Nov 2020.
- [18] Luis A. Rivera-Batiz and Danyang Xie. Gatt, trade, and growth. *The American Economic Review*, 82(2):422–427, 1992.
- [19] Harmen A. de Weerd. *If you know what I mean: Agent-based models for understanding the function of higher-order theory of mind*. PhD thesis, University of Groningen, 2015.
- [20] Seth Frey and Robert L. Goldstone. Cyclic game dynamics driven by iterated reasoning. *PLoS ONE*, 8(2):e56416, 2013.

- [21] Harmen de Weerd, Rineke Verbrugge, and Bart Verheij. Theory of mind in the mod game: An agent-based model of strategic reasoning. In Andreas Herzig and Emiliano Lorini, editors, *Proceedings of the European Conference on Social Intelligence (ECSI-2014)*, volume 1283 of *CEUR Workshop Proceedings*, pages 128–136. CEUR-WS.org, 2014.
- [22] Harmen de Weerd, Rineke Verbrugge, and Bart Verheij. How much does it help to know what she knows you know? an agent-based simulation study. *Artificial Intelligence*, 199-200:67–92, 2013.
- [23] Christelle Langley, Bogdan-Ionut Cirstea, Fabio Cuzzolin, and Barbara J. Sahakian. Theory of mind and preference learning at the interface of cognitive science, neuroscience, and AI: A review. *Frontiers Artif. Intell.*, 5:77852, 2022.
- [24] Tessa Rusch, Saurabh Steixner-Kumar, Prashant Doshi, Michael Spezio, and Jan Gläscher. Theory of mind and decision science: Towards a typology of tasks and computational models. *Neuropsychologia*, 146:107488, 2020.
- [25] Maaike Harbers, Karel Van Den Bosch, and John-Jules Meyer. Modeling agents with a theory of mind. In *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, volume 2, pages 217–224. IEEE, 2009.
- [26] Harmen de Weerd, Rineke Verbrugge, and Bart Verheij. Negotiating with other minds: the role of recursive theory of mind in negotiation with incomplete information. *Autonomous Agents and Multi-Agent Systems*, 31(2):250–287, 2017.
- [27] William P Fox. Teaching the applications of optimisation in game theory’s zero sum and non-zero sum games. *International Journal of Data Analysis Techniques and Strategies*, 2(3):258–284, 2010.
- [28] Kim Veltman, Harmen de Weerd, and Rineke Verbrugge. Training the use of theory of mind using artificial agents. *Journal on Multimodal User Interfaces*, 13:3–18, 2019.
- [29] Matthew O Jackson. A brief introduction to the basics of game theory. *Available at SSRN 1968579*, 2011.

- [30] Soleiman Mohammadi Limaei. Mixed strategy game theory, application in forest industry. *Forest Policy and Economics*, 12(7):527–531, 2010.
- [31] Harmen de Weerd, Rineke Verbrugge, and Bart Verheij. Higher-order theory of mind in the tacit communication game. *Biologically Inspired Cognitive Architectures*, 11:10–21, 2015.
- [32] Harmen de Weerd, Rineke Verbrugge, and Bart Verheij. Higher-order theory of mind is especially useful in unpredictable negotiations. *Autonomous Agents and Multi-Agent Systems*, 36(1):30, 2022.
- [33] Vaigarai Sathi, Sabahat Shaik, and Jaswanth Nidamanuri. Cooperation dynamics in multi-agent systems: Exploring game-theoretic scenarios with mean-field equilibria. *arXiv preprint arXiv:2309.16263*, 2023.
- [34] Daniel B Neill. An evolutionary resolution to the finitely repeated prisoner’s dilemma paradox. 2003.
- [35] Seth Frey. *Complex collective dynamics in human higher-level reasoning; A study over multiple methods*. PhD thesis, Indiana University, 2013.
- [36] Dimitris Batzilis, Sonia Jaffe, Steven Levitt, John A. List, and Jeffrey Picel. Behavior in strategic settings: Evidence from a million rock-paper-scissors games. *Games*, 10(2), 2019.
- [37] Garret Ridinger and Michael McBride. Theory of mind ability and cooperation. *Manuscript, Univ. California, Irvine*, 2017.
- [38] Colin F Camerer, Teck-Hua Ho, and Juin-Kuan Chong. Sophisticated experience-weighted attraction learning and strategic teaching in repeated games. *Journal of Economic theory*, 104(1):137–188, 2002.
- [39] Colin F Camerer, Teck-Hua Ho, and Juin-Kuan Chong. A cognitive hierarchy model of games. *The Quarterly Journal of Economics*, 119(3):861–898, 2004.
- [40] Marie Devaine, Guillaume Hollard, and Jean Daunizeau. Theory of mind: did evolution fool us? *PloS One*, 9(2):e87619, 2014.