UNIVERSITY OF SARAJEVO
FACULTY OF ELECTRICAL ENGINEERING
DEPARTMENT OF COMPUTING AND INFORMATICS

Machine Theory of Mind: How to make robots learn about the minds of other robots or humans?

BACHELOR'S THESIS

- FIRST CYCLE OF STUDIES -

Student:
Džana Krnjić

Supervisor:
Doc. dr Senka Krivić

Sarajevo,
July 2025

Sažetak

Ovaj rad istražuje uticaj različitih nivoa sofisticiranosti teorije uma (Theory of Mind – ToM) kod agenata u rekurzivnim modelima, koristeći Python paket tomsup. Korištene su igre sa društvenom interakcijom – Matching Pennies i Prisoner's Dilemma – kako bi se analizirala performansa agenata različitih nivoa k-složenosti ($0 \leq k \leq 5$). Eksperimenti su sprovedeni u simuliranom okruženju koristeći Python platformu Google Colab, gdje su agenti igrali međusobno po round-robin principu uz detaljno praćenje ponašanja i vremena izvođenja. Performanse su procjenjivane putem metrika kao što su prosječni rezultat, brzina učenja i trošak izvođenja. Ključni zaključci pokazuju da povećanje nivoa ToM iznad k=2 donosi minimalne dobitke u performansi, dok značajno povećava trošak računanja. Najviši nivoi sofisticiranosti, kao što su 4-ToM i 5-ToM, pokazuju ograničenu primjenjivost zbog visokih zahtjeva za procesiranje. Posebna pažnja je posvećena upotrebi Bajesove inferencije kao kognitivnog mehanizma kroz koji agenti simuliraju vjerovatnoće ponašanja protivnika u realnom vremenu. Na osnovu rezultata, rad potvrđuje hipotezu da visoki nivoi rekurzivne sofisticiranosti nisu nužno korisni u svim društvenim okruženjima, te predlaže 3-ToM model kao optimalan kompromis između efikasnosti i računske složenosti.

Ključne riječi: teorija uma, rekurzivno rezonovanje, Bajesovo zaključivanje, k-ToM agenti, tomsup paket

Abstract

This study investigates the impact of varying levels of Theory of Mind (ToM) sophistication in agents using a recursive model-based approach within the Python package tomsup. Social games such as Matching Pennies and the Prisoner's Dilemma were used to evaluate agents with recursive reasoning levels k ($0 \leq k \leq 5$). The experiment was conducted in a simulated environment via Google Colab using round-robin interactions, with performance tracked across metrics like average score, learning speed and computational cost. The findings confirm the hypothesis that increasing ToM sophistication beyond level 2 yields marginal performance gains at the expense of significant computational overhead. Highly sophisticated agents, such as 4-ToM and 5-ToM, show limited practical value due to the excessive runtime required. A central feature of the modeling approach is the use of Bayesian inference, which enables agents to simulate their opponents' beliefs and predict behaviors in real time. Results highlight that ToM effectiveness is context-dependent and influenced by environmental dynamics and game objectives, suggesting that 3-ToM may represent the optimal trade-off between cognitive depth and efficiency.

Keywords: theory of mind, recursive reasoning, Bayesian inference, k-ToM agents, tomsup package

Elektrotehnički fakultet, Univerzitet u Sarajevu
Odsjek za Računarstvo i informatiku
Doc. dr Senka Krivić, dipl. ing. el.
Sarajevo, Juli 2025

Postavka zadatka završnog rada I ciklusa:
Mašinska teorija uma: Kako omogućiti robotima da uče o umovima drugih robota ili ljudi?

Cilj završnog rada je istražiti kako različiti nivoi rekurzivnog razmišljanja u modelima teorije uma (Theory of Mind – ToM) utiču na performanse agenata u ponavljanim društvenim igrama. Korištenjem tomsup biblioteke i Bayesove inferencije, simulira se ponašanje agenata sa različitim nivoima rekurzivnog razmišljanja (0–5) u okruženjima kao što su Matching Pennies i Prisoner's Dilemma. Rad uključuje implementaciju eksperimenta, prikupljanje podataka o ponašanju agenata, analizu metrika performansi i računske složenosti, te evaluaciju koristi viših ToM nivoa. Na osnovu rezultata, donosi se zaključak o optimalnom nivou sofisticiranosti u odnosu na efikasnost i trošak računanja.

Polazna literatura:

[1] Abrini, M., Abend, O., Acklin, D., Admoni, H., Aichinger, G., et al., "Proceedings of 1st Workshop on Advancing Artificial Intelligence through Theory of Mind", arXiv preprint arXiv:2505.03770, 2025.

[2] Baker, C., Saxe, R., Tenenbaum, J., "Bayesian theory of mind: Modeling joint belief-desire attribution", Proceedings of the Annual Meeting of the Cognitive Science Society, Vol. 33, No. 33, 2011.

Doc. dr Senka Krivić, dipl. ing. el.

Univerzitet u Sarajevu
Elektrotehnički fakultet
Odsjek za Računarstvo i informatiku

Izjava o autentičnosti radova

Završni rad
I ciklusa studija

Ime i prezime: Džana Krnjić
Naslov rada: Mašinska teorija uma: Kako omogućiti robotima da uče o umovima drugih robota ili ljudi?
Vrsta rada: Završni rad Prvog ciklusa studija
Broj stranica: 32

Potvrđujem:

- da sam pročitao dokumente koji se odnose na plagijarizam, kako je to definirano Statutom Univerziteta u Sarajevu, Etičkim kodeksom Univerziteta u Sarajevu i pravilima studiranja koja se odnose na I i II ciklus studija, integrirani studijski program I i II ciklusa i III ciklus studija na Univerzitetu u Sarajevu, kao i uputama o plagijarizmu navedenim na web stranici Univerziteta u Sarajevu;
- da sam svjestan univerzitetskih disciplinskih pravila koja se tiču plagijarizma;
- da je rad koji predajem potpuno moj, samostalni rad, osim u dijelovima gdje je to naznačeno;
- da rad nije predat, u cjelini ili djelimično, za stjecanje zvanja na Univerzitetu u Sarajevu ili nekoj drugoj visokoškolskoj ustanovi;
- da sam jasno naznačio prisustvo citiranog ili parafraziranog materijala i da sam se referirao na sve izvore;
- da sam dosljedno naveo korištene i citirane izvore ili bibliografiju po nekom od preporučenih stilova citiranja, sa navođenjem potpune reference koja obuhvata potpuni bibliografski opis korištenog i citiranog izvora;
- da sam odgovarajuće naznačio svaku pomoć koju sam dobio pored pomoći mentora i akademskih tutora/ica.

Sarajevo, Juli 2025

Potpis:

Džana Krnjić

Contents

List of Figures

Chapter 1

Introduction

1.1 Background and motivation

Collective intelligence is a foundational concept across numerous disciplines—including economics, evolutionary theory, neuroscience, and studies of eusocial organisms and serves as a key element in understanding emergence and self-organization within complex systems. In recent years, interest has been focused on developing socially aware agents capable of interacting effectively with humans in real world settings. To achieve this goal agents are required to infer and interpret the intentions, goals, and behaviors of their human counterparts which is an ability that comes naturally to humans[3]. From an early age, people develop an intuitive form of psychology that enables them to reason about the mental states of others based on observable behavior. This cognitive capacity, known as Theory of Mind, has inspired advancements in artificial intelligence, particularly within the domain of multi-agent systems, where agents must model and respond to the intentions of other autonomous entities.

Designing effective collaboration strategies in multi-agent systems (MAS) is a complex and persistent challenge, particularly when multiple agents are simultaneously learning and adapting. One approach involves modeling the joint action space to apply single agent reinforcement learning (RL) algorithms within multi-agent contexts, this method quickly becomes computationally infeasible as the number of agents increases, due to the exponential growth of the action space.

Although recent efforts in multi-agent collaboration modeling have made progress, the integration of ToM into scalable, general-purpose multi-agent systems remains a significant obstacle. Some methods try to model nested beliefs, but when this goes beyond three levels of recursion, the computational cost becomes too high and a model like this exibits slow and complex use in real situations. This “belief over belief” modeling introduces exponential complexity, often referred to as the curse of recursive intractability.

Despite the theoretical appeal of recursive ToM models, there remains uncertainty about their practical utility in real-world or simulated social environments. While higher order ToM agents (e.g., 4-ToM, 5-ToM) can model increasingly complex mental hierarchies, it is unclear whether such sophistication leads to significantly better performance, or whether it results in diminishing returns, computational inefficiency, or reduced robustness.

Understanding the limits and trade-offs of recursive ToM in artificial agents is crucial for designing socially intelligent AI systems. If lower-order ToM is sufficient for effective interaction, simpler and more efficient models could be favored in robotics, games, and multi-agent platforms. This study aims to explore the relationship between ToM depth and agent performance in repeated social games, using simulated environments and controlled agent competitions.

1.2 Hypothesis

This study hypothesizes that increasing the recursive depth of Theory of Mind (ToM) beyond level 3 yields diminishing returns in agent performance within repeated social games.

To evaluate this hypothesis, the study employs simulated environments and competitions between artificial agents exhibiting varying levels of recursive ToM reasoning, ranging from 0 to 5. The aim is to determine whether the additional complexity introduced by higher-order ToM models translates into measurable performance gains, or whether simpler models are sufficient for effective interaction in multi-agent systems. This analysis seeks to inform both theoretical understanding and practical implementation of ToM in artificial agents, particularly in systems where efficiency, scalability, and real-time interaction are crucial.

1.3 Research objectives

The aim of this study is to evaluate the impact of recursive ToM sophistication on the performance of artificial agents in repeated social interactions where sophistication is defined as the depth of recursive thinking of the sort ‘I think that you think that I think, etc.’.

The primary objective of this study is to investigate the relationship between the depth of recursive Theory of Mind (ToM) reasoning and the performance of artificial agents in repeated game theoretic environments. The study aims to:

- Evaluate agent performance across varying ToM levels ($0 \leq k \leq 5$) in repeated two-player strategic games.
- Determine the point at which additional ToM depth no longer yields significant performance gains, identifying a trade-off between reasoning complexity and outcome effectiveness.
- Compare agent performance across different types of social games (competitive vs. cooperative) to assess whether ToM utility depends on the interaction context.
- Measure the computational cost associated with increasing ToM sophistication and assess whether such cost is justified by corresponding performance improvements.
- Provide insight into the practical applications of recursive ToM models in scalable multi-agent AI systems, particularly in environments requiring real time, adaptive decision-making.

The complete code is publicly accessible and contained in the GitHub repository which can be accessed via the following link: `https://github.com/dkrnjic1/BToM-How-to-infer-what-an-agent-knows-about-an-environment`.

1.4 Expected results

This study examines whether increasing the depth of recursive Theory of Mind reasoning in agents leads to meaningful improvements in strategic performance.

It is expected that agent performance will improve from 0-ToM to 2-ToM, but that higher levels with k > 3 will show diminishing returns in most game setups. It is generally anticipated that agents with higher ToM sophistication will exhibit better overall performance. It is anticipated that the results will vary depending on the structure of the game (competitive vs. cooperative), and that increasing ToM levels will be associated with higher computational cost.

These findings are expected to contribute to understanding the trade-offs between reasoning complexity and performance in multi-agent systems, offering guidance for designing socially intelligent, computationally efficient agents.

1.5 Outline

Chapter 2 provides an overview of recent developments in Theory of Mind research, with a focus on probabilistic modeling approaches, multi-agent reasoning and practical applications across collaborative learning and multimodal environments. In Chapter 3 the conceptual foundation of the work is presented. It discusses the notion of Theory of Mind, the role of recursive reasoning in artificial agents, and the importance of theory of mind in human interaction. It also provides limitations on wheather high sophistication makes sense when compared to extent to which humans are able to perform this type of reasoning also called mind reading. into python package used to conduct the experiment. Chapter 4 describes the design of the experiment and a model that has been used to conduct the experiment. It includes the agent configurations, simulation setup, types of games used, and the metrics applied to evaluate agent performance. Chapter 5 presents the experimental evaluations, including quantitative findings such as agent performance and learning trends after which a computational cost analysis is provided. The results are interpreted in relation to the original hypothesis, with attention to both expected and unexpected patterns. The final chapter, Chapter 6, reflects on the findings and their implications for multi-agent system design. It proposes directions for future work, and concludes with a summary of the contributions made through this research.

Chapter 2

Literature review

In recent years, theory of mind (ToM) has emerged as a crucial component in advancing collective intelligence within both human and artificial systems. Although traditional models of collective intelligence often overlooked individual psychological processes, modern research shows that ToM is important in helping people work together and adapt in social situations. The integration of ToM into AI systems has the potential to enhance the development of effective human-AI hybrid social ecologies, where machines can interpret, predict, and adapt to human mental states in complex, dynamic environments.

The ongoing surge in ToM research interest corresponds to significant and dynamic advances and innovation. We will provide a review of some of the most notable recent developments.

One significant approach conceptualizes the theory of mind as a problem of probabilistic inference established in planning under uncertainty. The core idea is to model the agent's decision making as a partially observable Markov decision process (POMDP), which accounts for both the agent's subjective perception of the environment and its goal-directed behavior. The inverse problem inferring the agent's beliefs and preferences from its actions is then approached using Bayesian inference. The experiment showed that human inferences about the mental states of others align closely with the predictions of this rational model. This model outperforms simplified alternatives that omit belief update or complete observability and shows that to simulate human inferences, it is necessary to perform joint inference about agents' beliefs and desires and to explicitly model the agent's observational process[2].

In order to evaluate the performance of rational models such as the one mentioned above, an article [3] has been published whose purpose was to make a benchmark for core psychology reasoning, consisting of a large-scale dataset of cognitively inspired tasks based on psychology studies with infants. This benchmark called AGENT is designed to probe the understanding of key concepts of machine agents of intuitive psychology in four scenarios: Goal Preferences, Action Efficiency, Unobserved Constraints, and Cost-Reward Trade-offs. To test this, the authors built two models ToMnet-G and BIPaCK and found that BIPaCK generalizes better across varied physical and goal-directed situations due to its structured reasoning and physical simulation capabilities. Although ToMnet-G shows promising results in tasks it has seen during training, it struggles to generalize across new scenarios, highlighting a key limitation in its current architecture. This performance gap suggests that improvements such as integrating more advanced model architectures or pre-training on a broader set of physical environments could enhance its general reasoning capabilities. The fact that AGENT reveals these limitations clearly demonstrates its value as a diagnostic benchmark, providing a solid foundation for future advancements in modeling intuitive psychology in artificial agents.

This need for improved generalization naturally leads to a broader discussion on explainable AI and interpretable agent reasoning, where understanding not only what a model predicts but also how and with what certainty becomes essential.

Researchers from the paper [4] explore how language models (LMs) can be used to model not only the beliefs of others in conversation but also the uncertainty behind those beliefs. The area of uncertainty is often oversimplified in traditional ToM research. Typically, beliefs are treated in binary terms (held or not held), but in reality, people often experience different degrees of certainty. This paper introduces a novel suite of tasks aimed at evaluating how well LMs can forecast such uncertainty in dialogue, treating interlocutors as forecasters themselves. The authors base their approach on "conversation forecasting", which moves beyond predicting factual uncertainty (inherent randomness in data) and instead focuses on epistemic uncertainty. To formalize this, they define a set of regression tasks in which eight LMs are asked to predict continuous probabilities reflecting the level of certainty of the interlocutor. To tackle this, the authors experiment with several methods, such as rescaling techniques, ensemble strategies, and chain-of-thought prompting to reduce variance in predictions. They also investigate the impact of contextual factors such as demographic information, goals, and type interaction. The results show that while current LMs can capture up to 7% of the variance in interlocutors' uncertainty, the task remains very challenging, even for humans. Interestingly, people in social interactions often obscure their true mental states, which complicates prediction even further. Ultimately, this work opens new directions for research in uncertainty-aware ToM and provides a foundation for future studies aiming to improve how AI systems interpret, reason about and respond to human uncertainty in conversation.

In multi-agent systems (MAS), especially when multiple learning agents are involved, establishing effective collaboration is a complex challenge. One common method is to model the joint actions of all agents so that traditional single agent reinforcement learning (RL) algorithms can be applied directly. However, as the number of agents increases, the joint action space becomes exponentially large, making this approach impractical. To address this, the researchers of the article [5] use centralized training with decentralized execution. In this setup, each agent learns a policy based on its own observations, while a centralized critic helps guide learning during training. A new solution that has been proposed allows agents to develop collaboration skills by learning only a first-level belief over belief (i.e., what an agent thinks another agent believes about its own state). Instead of explicitly modeling all levels of reasoning, the algorithm trains agents to approximate this belief and react accordingly. This process happens through adaptive training where one agent learns while the other remains fixed, creating a stable environment and solving the problem of non-stationarity. The researchers tested this adaptive ToM approach in games in which each agent has access to public information and also holds private information. Each agent tries to infer both the private state of the other agent and the belief that the other agent has about its own private state. This belief-over-belief is learned by using centralized training, where agents can share information for better supervision. Two test environments were used: a kitchen task inspired by robot-human collaboration and a meeting scheduling game where agents with different private calendars must agree on a meeting time. Compared to other popular multi-agent algorithms like Q-learning and policy gradients, this adaptive ToM method performed consistently well. It achieved results close to state-of-the-art models that rely on centralized training and modules like shared Q-functions or meta-agents. In particular, ToM-enabled agents were better at adapting to new partners, forming more generalized strategies than simply memorizing interaction patterns with specific agents.

Previous work has evaluated ToM reasoning in AI through benchmarks that tend to focus on just one type of data at a time, either language or visual input. In contrast, humans integrate information from multiple sources to infer what someone is thinking or intending to do. This study [6] addresses that gap by introducing a new benchmark called MMToM-QA (Multimodal Theory of Mind – Question Answering), which combines both video and text to more closely mimic the way humans interpret mental states. The MMToM-QA benchmark presents short household activity scenarios in both video and text form, along with multiple-choice questions that ask about the mental states (beliefs or goals) of the person in the scenario. To tackle this benchmark, the authors developed a new model called BIP-ALM (Bayesian Inverse Planning Accelerated by Language Models). This approach builds on earlier work in Bayesian inverse planning, which models how people might infer others' goals and beliefs by observing their actions. BIP-ALM extends this method to handle both video and text data by first converting them into symbolic representations. These symbols capture the state of the environment and the person's actions. The model then merges the representations of both modalities to form a unified understanding of the situation. In the tests, BIP-ALM outperformed existing models, including GPT-4 and its visual version, GPT-4V, especially on questions that required reasoning about false beliefs or changing mental states. Although LLMs were able to retrieve factual information well, they often struggled to infer mental states or goals.

Among the various practical applications of Theory of Mind in artificial intelligence, one particularly interesting case is its use in teaching agents that adapt their instruction based on the inferred internal state of the learner. The article [7] explores how equipping a teacher agent with a Bayesian ToM model enables it to infer the learner's goals and sensory capabilities from limited observations and adapt its teaching strategy accordingly. The key method introduced involves Bayesian inference, allowing the teacher to estimate the learner's internal state by observing its behavior in a simpler environment. Using this inferred model, the teacher selects the most useful demonstration to help the learner succeed in a more complex task, while also minimizing teaching costs. Experimental results show that ToM-based teacher agents significantly outperform learner-agnostic baselines, particularly when learners have limited sensory capacities. These findings highlight the importance and effectiveness of incorporating ToM in AI systems to facilitate personalized and efficient teaching, representing a valuable step towards socially intelligent and adaptive machines.

Chapter 3

Theoretical Framework

3.1 Theory of mind in cognitive science

Theory of Mind (ToM) is widely recognized in cognitive science as a component of human intelligence, referring to the capacity to infer and attribute mental states such as beliefs, desires, and intentions to oneself and others. This cognitive ability underlies our capacity for social interaction, as it enables individuals to understand that others may have perspectives, knowledge, and motivations different from their own. ToM is not a singular process but rather involves an interplay of both emotional ("hot") and non-emotional ("cold") cognitive mechanisms [8]. While cold cognition involves more detached, logical reasoning, hot cognition encompasses social and emotional interpretation. These dual processes contribute to how individuals predict and respond to the behaviors of others, especially in socially complex or emotion involved settings.

Beyond its conceptual role in social cognition, Theory of Mind has been the subject of extensive developmental research aimed at understanding how it emerges and evolves in early childhood. While many core ToM abilities are believed to be in place by the age of five, studies show that different aspects of mental state attribution develop gradually and in stages. Baron-Cohen [9] proposed one of the earliest developmental models of ToM, outlining how infants begin by distinguishing animate from inanimate motion around six months of age. By twelve months, they typically demonstrate joint attention and begin to form representations of both their own and others' perceptual states. More advanced reasoning abilities such as understanding false beliefs become stable closer to age five or six. Research in developmental psychology has also shown that even at one year of age, infants can recognize goal-directed behavior and respond to others' intentions. These findings illustrate that ToM is not a single, but rather a layered set of skills that is being developed throughout early childhood. This developmental trajectory offers important insights into how cognitive models of ToM might be structured in artificial agents that aim to replicate or simulate the reasoning that humans possess.

3.2 Importance of theory of mind in human interaction

Although artificial intelligence has been applied across numerous domains, many current systems still operate on relatively static and deterministic architectures, often resembling expert systems. These systems rely on predefined rule-based logic commonly structured through programmed "if-then" statements rather than exhibiting flexible, adaptive behavior that resembles human intelligence [10]. In response, recent technological developments have focused on integrating cognitive models that emulate human-like reasoning. These newer approaches are typically grounded in complex black-box methods, including machine learning, deep neural networks, and model-based reinforcement learning, which aim to capture patterns and decision-making processes in ways that mirror aspects of human cognition [11] [12].

Despite their powerful predictive capabilities, the methods used by machine learning models to process data and recognize patterns are often opaque, offering limited insight into the internal mechanisms that explain their decisions. As a result, the rationale behind an AI system's behavior is frequently difficult to interpret. Unlike humans, who can typically explain the reasoning behind their actions, machine learning models operate through processes that are not easily accessible or replicable by human understanding. This lack of interpretability rises a field of explainable artificial intelligence (XAI), which seeks to address this challenge by either improving the transparency of model decision-making or by developing tools that translate machine outputs into explanations comprehensible to human users.

A key consequence of the opacity in many AI systems is a reduction in user trust. When individuals are unable to interpret or understand an agent's decisions, their willingness to rely on the system diminishes. This poses a challenge, as trust is widely acknowledged as a crucial element in effective human-agent collaboration. Building trust requires that users develop a clear understanding of the agent's behavior, intentions, and limitations. As AI systems become more autonomous, ensuring that users can accurately assess the agent's reliability will be increasingly essential for successful deployment.

Fostering trust in artificial agents will require the integration of artificial social intelligence (ASI), which is the capacity to express internal model outputs in ways that are transparent and understandable to humans. Successful human-agent collaboration depends on equipping agents with social and communicative competencies that support meaningful and context-sensitive exchanges of information.

In essence, Theory of Mind can be viewed as the core cognitive mechanism people use to deal with others in social situations. The integration of an artificial form of Theory of Mind (AToM) is seen as an essential step toward enabling agents to not only understand human mental states, but also express their own internal reasoning in ways that humans can understand [13]. ToM supports the identification and correction of false or incomplete mental representations through targeted interventions. This mutual transparency is especially important in collaborative environments, where misalignment between human expectations and agent behavior can lead to errors, too much reliance, or even system rejection. By incorporating ToM into artificial social intelligence, agents can contribute to more fluid, reliable, and human-like interactions supporting the development of systems that are not only capable, but also trusted and accepted.

3.3 Recursive Theory Of Mind

While first-order Theory of Mind involves attributing mental states to others (e.g., I believe that you want...), second-order Theory of Mind also known as recursive mind reading refers to the capacity to represent mental states nested within other mental states, for example, forming beliefs about what someone else believes or intends.This recursive capacity, also referred to as mental metarepresentation, is central to many complex forms of human interaction, including communication, strategic planning, religion, and story-telling. Although earlier cognitive theories have suggested that humans struggle to reason beyond the second or third level of recursion, more recent research [14] highlights that such limitations may be partially due to the fact that those experiments were too direct and not natural enough. The study now suggests that humans may perform better on recursive mindreading tasks when scenarios are presented implicitly, in more ecologically valid social contexts. In such environments, where collaboration, deception, and social evaluation are frequent, deeper recursive reasoning may emerge more naturally.

These insights are particularly relevant to the development of artificial agents, as recursive ToM models (such as the k-ToM model used in this thesis) aim to simulate varying levels of social reasoning depth. Understanding how recursion operates in human cognition offers not only theoretical insight, but also practical guidance for scaling the complexity of artificial mindreading in multi-agent environments.

3.4 Limitations and Open Questions

Despite the widely held view that recursive mindreading is a defining feature of human cognition, evidence suggests that human performance in deeply nested reasoning tasks is far more limited than the theoretical potential of such systems. While recursive structures allow for unlimited depth in social reasoning and strategy, empirical studies show that people typically perform well only up to the first or second level of recursion. Higher-order recursive tasks, such as understanding what person A thinks that person B believes that person C wants, become increasingly difficult due to the exponential growth in mental state combinations. Although a small percentage of individuals may achieve high level of reasoning under specific conditions, such as financial incentives [15], most adults exhibit a sharp decline in performance beyond two to three levels of reasoning. This limitation is particularly evident in language processing, economic games, and ToM reasoning tasks, where the cognitive load increases rapidly with each added level of recursion. The complexity involved in managing such mental state combinations highlights that recursive ToM, while theoretically limitless, faces significant performance constraints in practice.

These findings raise important questions about how deeply agents, whether human or artificial, can and should reason about others. In the context of this thesis, where simulated k-ToM agents are evaluated in structured environments, such limitations point to a potential trade-off between increased recursive depth and practical performance. If human reasoning rarely exceeds two or three levels, and even artificial agents incur computational costs with each recursive step, it becomes essential to conclude whether increasing sophistication beyond a certain point yields meaningful benefits. In the following chapters, we explore this question by examining the performance of agents with varying levels of recursive Theory of Mind in repeated social games. This offers new insights into the computational and strategic value of recursive reasoning in artificial systems.

Chapter 4

Methodology

4.1 Experimental Environment and Tools

This experiment was implemented using the **tomsup** Python package, an open source simulation framework for studying theory of mind reasoning. The explanation of the model used was made according to the article that proposed the tomsup package, which can be found in this article [16]. This package enables researchers to investigate the behavioral implications of computational ToM models by integrating them into artificial agents and observing their interactions across various game theoretic environments. The framework supports a range of games including the Prisoner's Dilemma and Matching Pennies, allowing systematic comparison of agent performance under different strategic conditions.

The tomsup framework allows for the implementation of agents equipped with a variety of cognitive strategies, enabling comparative analysis of their behavior across different interaction environments. One of its key features is the ability to simulate interactions not only between artificial agents but also between agents and human participants in real time. This functionality allows for the dynamic generation of experimental stimuli, providing a basis to observe how humans respond to distinct agent strategies. The system can then be used to infer which levels of recursive reasoning or other cognitive parameters may underlie human behavior in these interactions.

The tomsup package includes a range of games designed to simulate repeated interactions between two agents. In each game, both agents simultaneously choose between two possible actions, with their resulting payoffs which are determined by the combination of choices they have made. These interactions are modeled using a 2 by 2 payoff matrix, a standard representation in game theory that abstracts real world scenarios into formal structures of strategic decision making. The matrix captures the outcomes of agent interactions by associating each pair of decisions with corresponding rewards or penalties, allowing consistent and systematic analysis across different games. One of the games that are included in the tomsup package is a game "battle of the sexes" whose payoff matrix is shown below.

Table 4.1: Payoff matrix for Agent 0 in the "battle of the sexes" game. Each cell denotes the reward for Agent 0 depending on their choice and the choice of Agent 1. For example, if both agents choose action 0, Agent 0 receives 10 points. If Agent 0 chooses 1 and Agent 1 chooses 1, Agent 0 receives 5 points.

		Agent 1's Choice	
		0	**1**
Agent 0's Choice	**0**	10	0
	1	0	5

Table 4.2: Payoff matrix for Agent 1 in the "battle of the sexes" game. Each cell denotes the reward for Agent 1 depending on the joint actions. For example, if both agents choose action 1, Agent 1 receives 10 points. If Agent 1 chooses 0 and Agent 0 chooses 0, Agent 1 receives 5 points.

		Agent 1's Choice	
		0	**1**
Agent 0's Choice	**0**	5	0
	1	0	10

In research on Theory of Mind, such as in the work proposed in an article [17], game theory is often used as a helpful framework to explore how agents understand and predict others' behavior. This is because game theory focuses on strategic situations where a small number of individuals must make decisions that depend on what others might do. Although it may not fully explain behavior in large open-ended social situations, it is still very useful when studying controlled interactions. The key idea is that if agents try to understand the thoughts or limits of others, especially things they cannot directly observe, it helps them narrow down what the other agents might do. This makes the interaction easier to handle. This kind of thinking, where one imagines what others believe or intend, is what we call Theory of Mind. Game theory helps capture that kind of reasoning and is useful both for understanding human social behavior and for building smart artificial agents.

Simulations within the tomsup framework can be structured in a variety of configurations. In the most basic setup, two agents repeatedly interact with one another. However, the framework also supports more complex arrangements involving multiple agents, such as round-robin tournaments in which each agent competes against every other agent in the population. These more nuanced configurations allow researchers to simulate socially relevant dynamics, making established approaches used to study phenomena such as the emergence of cooperation.

To carry out the experiment, the simulation based on Python was developed and executed within Google Colab platform. This environment was selected because of its ease of use, support for Python libraries, and the ability to leverage computational resources that are based on the cloud. The simulations are heavily based on the `tomsup` package and the entire experimental pipeline which includes agent creation, their simulation, and result collection executed within this platform. For complete reproducibility, the complete code is publicly accessible via the following Colab notebook link: `https://github.com/dkrnjic1/BToM-How-to-infer-what-an-agent-knows-about-an-environment`.

4.2 The k-ToM Model

This section explains the conceptual overview of the recursive Theory of Mind (ToM) model, followed by a more detailed explanation of its mathematical implementation. Each ToM agent operates through two main components: a learning process and a decision process. During learning, the agent infers the internal parameters of its opponent and estimates the likelihood that the opponent will choose each available action (as detailed in the sections on 0-ToM and k-ToM learning processes). In the decision phase, these predictions are used to guide the agent's own behavior by determining the probability of its choice based on the expected action of the opponent. Based on these predictions, the agent calculates the expected reward for each of its own possible actions and selects the one with the highest expected payoff. The key difference between ToM agents lies in how they represent their opponents. A 0-ToM agent (without recursion) assumes its opponent has fixed preferences or biases toward certain actions. In contrast, k-ToM agents (where k > 0) model their opponents as agents with internal mental states. These agents simulate how their opponents learn and make decisions,allowing them to estimate what their opponent believes about them in return. The value k in a k-ToM agent defines the depth of recursive reasoning it can perform, which is how many layers of "thinking about the other's thinking" it can represent. For example, a 1-ToM agent assumes that its opponent is a 0-ToM agent, while a 2-ToM agent can model its opponent as either a 0-ToM or a 1-ToM agent, and so on. Importantly, a k-ToM agent must also estimate the opponent's level of ToM (from 0 to k1) using a process similar to reinforcement learning.

Each k-ToM agent is defined by a set of four parameters, collectively denoted as θ, which represents its behavior and learning dynamics. The **behavioral temperature** ($\beta \in [0,\infty]$) controls the level of randomness in the decision making process where lower values make agent choices more deterministic, while higher values introduce more noise. The **volatility** parameter ($\sigma \in [0,\infty]$) reflects the agent's belief about how likely the opponent is to change their internal parameters over time, influencing how quickly the agent updates its beliefs. The **bias** term ($b \in [-\infty,\infty]$) shows if the agent prefers one option more than the other, even without learning anything (like choosing something out of habit or because of its position). Lastly, the **dilution** parameter ($d \in [0,1]$) determines how quickly the agent forgets previous estimates about the opponent's ToM level, effectively weighting recent interactions more heavily and modeling assumptions about the opponent's potential for change.

4.2.1 0-ToM's learning process

The 0-ToM model, as illustrated in Figure 4.1, operates under the assumption that the opponent follows a simple, biased random strategy. In the learning process, 0-ToM agent analyzes outcomes from previous trial to estimate the bias of the opponent which is denoted as p_t^{op}. The agent tracks this bias using a probabilistic approach, where both the estimated mean μ and variance Σ of the bias are computed. These values are then combined into a single point estimate to guide the agent's decision-making in a simplified manner.

The variance Σ of the estimate of the opponent's bias is updated using the following equation:

$$\Sigma_t \approx \frac{1}{\left(\frac{1}{\Sigma_{t-1}+\sigma}\right) + s(\mu_{t-1})\,(1-s(\mu_{t-1}))} \tag{4.1}$$

where Σ_t denotes the variance or uncertainty of the 0-ToM bias estimate of the opponent in trial t, and μ_{t-1} is the mean in log-odds of the estimate from the previous trial. s is the sigmoid function used to convert the mean parameter estimate into a probability between 0 and 1. σ denotes the volatility parameter of the 0-ToM agent, which represents the assumption on how much the opponent's parameters deviate over time. σ defines a lower bound for how certain the 0-ToM agent can be of their opponent's bias estimate.

The updated Σ_t is then used when updating the mean estimate μ of the opponent's bias, as shown below:

$$\mu_t \approx \mu_{t-1} + \sum_t \left(c_{t-1}^{\mathrm{op}} - s(\mu_{t-1})\right) \tag{4.2}$$

At each time step t, the previous action of the opponent, denoted as c_{t-1}^{op}, serves as the input to update the estimate of their choice bias. The mean estimate μ is adjusted considering the difference between the observed choice c_{t-1}^{op} and the previously predicted probability of that choice, given by the sigmoid function s($\mu_{\mathrm{t-1}}$). This difference described as a prediction error,

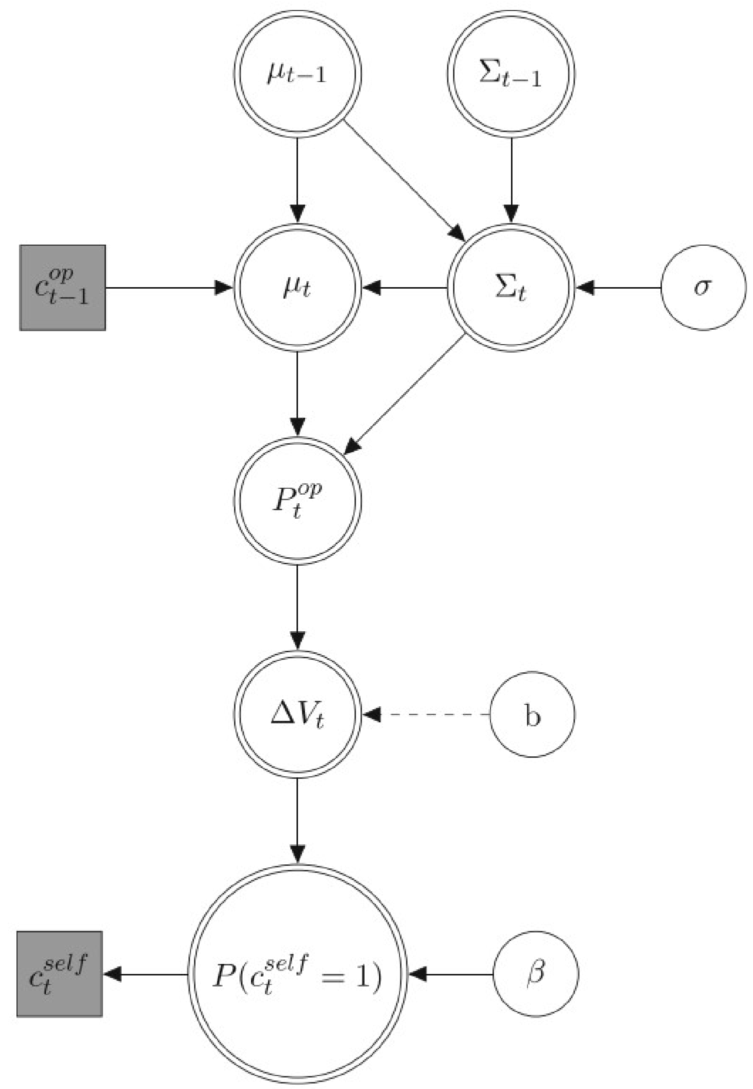

Figure 4.1: A graphical model of the 0-ToM model's learning and decision processes, which is repeated on each trial. Variables that are observed (data) are shaded. Unobserved deterministic variables are represented with a double border. Discrete variables are represented as squares, while continuous ones as circles.

is scaled by the current level of uncertainty Σ_t. Higher uncertainty leads to a greater update, making the model more responsive to new evidence. Together, the update rules (as formalized in Equations 4.1 and 4.2) resemble a Kalman filter, allowing the agent to iteratively refine its estimate of the opponent's choice probability.

To derive a single-point estimate of the opponent's bias, which is the probability that the opponent will select action 1 in the current trial, the model integrates both the mean μ and the variance Σ of the bias estimate using a probabilistic adjustment:

$$p_t^{\text{op}} \approx s\left(\frac{\mu_t}{\sqrt{1+(\Sigma_t+\sigma)3/\pi^2}})\right) \tag{4.3}$$

This estimate, denoted as p_t^{op}, reflects the agent's current belief about the likelihood that the opponent would choose option 1. The volatility parameter σ, which represents how much the agent expects the opponent's behavior to vary over time, also plays a role in this calculation.

Intuitively, the greater the uncertainty in the estimate, the closer the predicted probability shifts toward the chance level, rather than strictly aligning with the mean μ. To simplify computation and reduce identification problems, an approximate version of this expression is proposed:

$$p_t^{\mathrm{op}} \approx s\left(\frac{\mu_t}{\sqrt{1+0.36\cdot\Sigma_t}}\right) \tag{4.4}$$

The variant with the approximation is set as a defaut in tomsup package, p_t^{op} is then used in the decision process as it will be decribed in "The decision process".

4.2.2 k-ToM's learning process

When agents operate with a higher level of Theory of Mind where $k > 0$, the learning mechanism becomes increasingly complex. As illustrated in Figure 4.2, a k-ToM agent no longer treats the opponent as following a simple bias, but instead actively simulates the opponent's internal learning and decision-making processes to estimate the probability p^{op} of them selecting action 1. In doing so, the agent assumes that the opponent operates at a lower ToM level $\kappa < k$. The agent must continuously estimate with each round both the likelihood λ^{κ} that the opponent is using a specific sophistication level κ, and the corresponding model parameters θ that define the opponent's behavior.

Given the recursive structure of the model where the agent represents the opponent, who in turn represents the agent and so on, the estimation of the opponent's parameters necessarily involves modeling how the opponent estimates the agent's own internal parameters. This is achieved through a non-linear variational Bayes approach, specifically using a Laplace approximation. As a result, the model produces a mean estimate μ^{θ} and variance Σ^{θ} for each of the opponent's parameters included in the set θ (which typically includes the behavioral temperature β and volatility σ, and optionally the bias b and dilution d). Based on these estimates, the k-ToM agent can simulate how the opponent perceives its own probability of choosing action 1, thereby approximating the opponent's decision-making behavior.

In this process, the gradient W^{θ} represents the sensitivity of the opponent's estimated choice probability μ^{κ} to changes in the parameter estimates $\mu^{\kappa,\theta}$, calculated separately for each parameter within the set θ. This gradient determines the extent to which each parameter estimate $\mu^{\kappa,\theta}$ should be updated ensuring that parameters with less influence on the choice probability receive smaller adjustments. It also informs the weighting of uncertainties $\Sigma^{\kappa,\theta}$, such that parameters deemed more influential contribute more significantly to the belief formed about the opponent's choice probability $p^{\mathrm{op},\kappa}$.

For each possible opponent sophistication level $\kappa < k$, the model estimates both the parameters and the corresponding choice probability. The overall predicted probability that the opponent will select action 1, denoted as p^{op}, is computed as a weighted average of the individual probabilities $p^{\mathrm{op},\kappa}$ associated with each level κ. These weights are determined by the estimated likelihood λ^{κ} that the opponent is operating at that specific level of recursion. This final estimate is then passed to the agent's decision process, just like how the 0-ToM agent utilizes its simpler prediction model, as proposed in the section on the decision process.

The mathematical formulation begins with estimating the probability λ^{κ} that the opponent operates at each recursion level κ. When the dilution parameter d is employed, it increases the uncertainty of previous λ^{κ} estimates by partially "forgetting" past values. This mechanism enables the agent to remain adaptive to potential changes in the opponent's reasoning strategy. The effect of dilution is incorporated using the following equation:

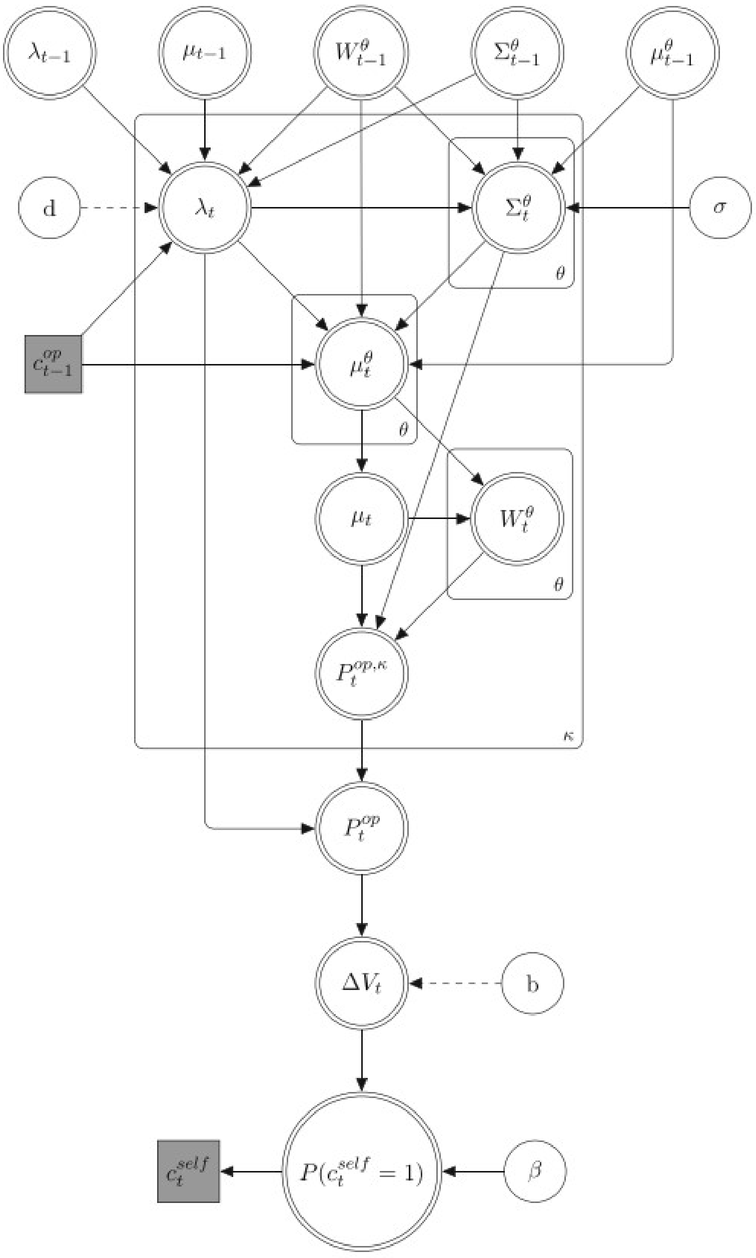

Figure 4.2: A graphical model of the k-ToM model's learning and decision processes, which is repeated on each trial. Variables that are observed (data) are shaded. Unobserved deterministic variables are represented with a double border. Discrete variables are represented as squares, while continuous ones as circles.

$$\lambda^{\kappa}_{t-1} = (1-d) \cdot \lambda'^{\kappa}_{t-1} + \frac{d}{k} \tag{4.5}$$

where λ'^{κ}_{t-1} denotes the pre-update estimate of $\lambda^{\kappa}_{t-1}a$ from the previous trial. The dilution parameter d ranges between 0 and 1 and is passed through a sigmoid transformation. The value of k reflects both the current agent's ToM sophistication and the number of possible recursion levels it considers for the opponent.

The agent updates the probability estimates λ^{κ}_t for each potential opponent recursion level κ by comparing the expected behavior, given each κ, to the actual observed action taken by the opponent. This update is computed as follows:

$$\lambda_t^{\kappa} \approx \left(\frac{\lambda_{t-1}^{\kappa} \cdot p_{t-1}^{\mathrm{op},\kappa}}{p_{t-1}^{\mathrm{op},\kappa} \cdot \sum_{\kappa' < \kappa} \lambda_{t-1}^{\kappa'}} \right)^{c_{t-1}^{op}} \left(\frac{\lambda_{t-1}^{\kappa} \cdot p_{t-1}^{\mathrm{op},\kappa}}{p_{t-1}^{\mathrm{op},\kappa} \cdot \sum_{\kappa' < \kappa} \lambda_{t-1}^{\kappa'}} \right)^{1-c_{t-1}^{op}} \tag{4.6}$$

In this context, λ_{t-1}^{κ} refers to the previous estimate of the likelihood that the opponent is using recursion level κ at trial t. The term $p_{t-1}^{\mathrm{op},\kappa}$ represents the predicted probability that the opponent would select action 1 under the assumption that it operates at level κ on trial $t-1$. Note that the symbol $\sum$ is used here to denote summation, not the uncertainty in parameter estimates. $p_{t-1}^{\mathrm{op},\kappa}$ is approximated using the following equation:

$$p_{t-1}^{\mathrm{op},\kappa} \approx s\left(\frac{\mu_{t-1}^{\kappa} - 0.319 \cdot (\Sigma_{t-1}^{\kappa})^{0.781}}{\sqrt{1 + 0.205 \cdot (\Sigma_{t-1}^{\kappa})^{0.870}}} \right) \tag{4.7}$$

In this formulation, μ_{t-1}^{κ} represents the predicted log-odds of the opponent selecting action 1 at trial t, based on information from the previous trial $t-1$, for each potential opponent level κ. The term Σ_{t-1}^{κ} denotes the uncertainty of the agent with respect to this prediction. This uncertainty is computed as a weighted average of the variances associated with the agent's parameter estimates, where the contribution of each parameter is determined by its influence on the predicted opponent behavior:

$$\Sigma_{t-1}^{\kappa} \approx \sum_{\theta} \Sigma_{t-1}^{\kappa,\theta} \cdot \left(W_{t-1}^{\kappa,\theta} \right)^2 \tag{4.8}$$

In this context, $\Sigma_{t-1}^{\kappa,\theta}$ indicates the agent's estimate of uncertainty at trial $t-1$ for each parameter θ associated with a given recursion level κ. The term $W_{t-1}^{\kappa,\theta}$ represents the gradient that captures how changes in parameter estimates influence the predicted choice probability. It is important to note that this formulation holds under the assumption of a mean-field approximation across parameters.

At this stage, the k-ToM agent proceeds to refine its estimations for each of the opponent's parameters θ, namely, the behavioral temperature β, the volatility σ, and optionally the bias b and dilution factor d. The update begins with the computation of uncertainty associated with these parameter estimates:

$$\Sigma_t^{\kappa,\theta} \approx \frac{1}{\frac{1}{\Sigma_{t-1}^{\kappa,\theta} + \sigma} + s\left(\mu_{t-1}^{\kappa}\right)\left(1 - s\left(\mu_{t-1}^{\kappa}\right)\right) \lambda_t^{\kappa} \left(W_{t-1}^{\kappa,\theta} \right)^2} \tag{4.9}$$

In this context, μ_{t-1}^{κ} represents the agent's belief, based on the previous trial, about the probability that the opponent, assumed to be at sophistication level κ, will select option 1.

The mean estimates of each parameter in θ are now updated:

$$\mu_t^{\kappa,\theta} \approx \mu_{t-1}^{\kappa,\theta} + W_{t-1}^{\kappa,\theta} \Sigma_t^{\kappa,\theta} \lambda_t^{\kappa} \left(c_{t-1}^{op} - s\left(\mu_{t-1}^{\kappa}\right) \right) \tag{4.10}$$

The estimate $\mu^{\kappa,\theta}$ is updated based on the prediction error, defined as the difference between the opponent's actual choice c_{t-1}^{op} and the expected choice probability $s(\mu_{t-1}^{\kappa})$. This adjustment is scaled by several factors: the gradient $W^{\kappa,\theta}$ indicating how sensitive the predicted choice probability is to changes in parameter estimates, the agent's belief λ^{κ} regarding the opponent's level of sophistication and the associated uncertainty $\Sigma^{\kappa,\theta}$ in the parameter estimation.

The agent proceeds to calculate the probability of the expected choice μ^{κ} for each possible level of sophistication of the opponent κ by simulating the corresponding learning dynamics of the opponent. This requires maintaining and continuously updating internal representations of hypothetical opponents at different levels κ.

To support this process, the agent estimates the gradient W that reflects how changes in the estimates of internal parameters μ^{θ} influence probabilities of the predicted choice μ for each κ.

$$W_t^{\kappa,\theta} \approx \frac{d\mu_t^{\kappa}}{d\mu_t^{\kappa,\theta}} \tag{4.11}$$

This gradient is approximated using local linearization, where small perturbations are applied to each parameter in θ (such as behavioral temperature β, volatility σ, and optionally bias b and dilution d), and the resulting impact on the choice probability is measured. Calculating W^{θ} is essential for appropriately weighting and updating parameter estimates, given their non-linear relationship to the opponent's observed behavior.

Next, for each possible level of opponent sophistication κ, the agent estimates the opponent's probability of selecting action 1, applying an approximation to avoid identification concerns.

$$p_t^{op,\kappa} \approx s\left(\frac{\mu_t^{\kappa}}{\sqrt{1+0.36\cdot\Sigma_t^{\kappa}}}\right) \tag{4.12}$$

where μ_t^{κ} is the mean estimate of the opponent's probability of choosing 1 on trial t. The uncertainty term Σ_t^{κ} is derived from the variances of the individual parameters Σ_{θ}, as previously defined, and the corresponding gradients $W_t^{\kappa,\theta}$ of the current trial.

The choice probability estimates of opponents $p^{op,\kappa}$ for each sophistication level κ are then combined into a final aggregate estimate for the opponent's probability of choosing action 1 which is done by computing a probability weighted average:

$$P_t^{op} = \sum_{\kappa} \lambda_t^{\kappa} \cdot P_t^{op,\kappa} \tag{4.13}$$

This final aggregated estimate of the opponent's behavior is subsequently used within the agent's decision-making process to determine its own action, as described in the following subsection.

4.2.3 The decision process

Based on the internal inferred model of the opponent that is outlined in detail in the section *k-ToM's learning process* the agent computes, at each trial t, an estimate of the probability that the opponent will select action 1, denoted as p_{op}^{t}. Using this probability, the agent calculates the expected relative payoff of choosing action 1 instead of action 0. This expected value, represented as ΔV_t, is derived by weighting the potential payoffs associated with each choice by the opponent's predicted action:

$$\Delta V_t' = p_t^{op}\left(U(1,1)-U(0,1)\right)+(1-p_t^{op})(U(1,0)-U(0,0)) \tag{4.14}$$

The notation $U(c^{self}, c^{op})$ represents the utility function, which produces the reward R based on the specified payoff matrix and the hypothetical actions of the k-ToM agent (c^{self}) and its opponent (c^{op}). Following this, bias parameter b can be added to influence the expected utility toward a specific choice:

$$\Delta V_t = \Delta V_t' + b \tag{4.15}$$

where $\Delta V_t'$ is the value of ΔV_t before the optional update.

The k-ToM agent then computes the probability $P(c_t^{self} = 1)$ of selecting action 1 by applying a softmax function to the expected utility difference ΔV_t between the two available choices, as expressed in the following equation:

$$P(c_t^{self} = 1) = \frac{1}{1 + exp\left(-\frac{\Delta V_t}{\beta}\right)}) \tag{4.16}$$

where $P(c_t^{self} = 1)$ is the probability of k-ToM's for choosing 1 in the current trial t. β represents 0-ToM's behavioral temperature parameter, where agent shows more random behavior for higher values.

4.3 Experimental Setup

4.3.1 Agent Configuration

The experiment includes a set of simulated agents designed to reflect different levels of Theory of Mind (ToM) sophistication. Six recursive ToM agents were configured, each corresponding to a ToM depth k ranging from 0 to 5. These agents represent increasing levels of recursive mentalizing, from simple estimations based on the bias of actions of the opponent (0-ToM), to higher-order recursive reasoning where agents infer what the opponent believes about their own beliefs.

4.3.2 Agent Parameters

Each agent model used in the experiment is concluded of a set of cognitive and behavioral parameters that define its decision-making and learning characteristics. The recursive ToM agents (k-ToM), implemented using the `tomsup` package, are parameterized by four key components: *volatility*, *behavioral temperature*, *bias*, and *dilution*. The **volatility** parameter (σ) captures how much an agent expects its opponent's behavior or internal parameters to oscillate over time. It is expressed on a logarithmic scale, with a default value of -2, which corresponds to a moderate belief in the variability of the opponent. The **behavioral temperature** (β) controls the stochasticity of agent's decisions: lower values result in more deterministic actions while higher values introduce noise. This is also log-scaled, with a default of -1. The **bias** term shows if the agent prefers one option more than the other, independent of the opponent's behavior or expected reward. Finally, the **dilution** parameter models how much weight is given to recent evidence to update beliefs about the opponent's sophistication level; values closer to 1 imply greater memory decay. In this experiment, agents with ToM levels $k \in \{0,1,2,3,4,5\}$ were instantiated using the default parameter values provided by the package, to maintain consistency with the previous benchmark setups.

4.3.3 Game Scenarios

To evaluate the performance and adaptability of agents in different social dynamics, the experiment was conducted using three classical two-player game scenarios: *Matching Pennies* and *Prisoner's Dilemma*. These games were selected because of their well-established theoretical significance and their ability to capture a wide range of strategic interactions. **Matching Pennies** represents a purely competitive zero-sum setting, where one agent's gain is the other's loss, emphasizing adversarial reasoning and prediction. In contrast, the **Prisoner's Dilemma** shows a conflict between individual and collective interest, where working together gives better results, but people are tempted to act selfishly. This makes it a key example for studying trust and how people cooperate. These games range from competitive to cooperative, helping us see how different levels of Theory of Mind affect behavior and success in different social situations.

Matching Pennies

The Matching Pennies game is a competitive two-player interaction. Each player simultaneously chooses between two actions, traditionally labeled as "Heads" or "Tails." If the choices match, one player wins; if they differ, the other player wins. The game has no pure strategy and requires agents to adopt mixed strategies to remain unpredictable. Agents have thus to predict their opponents' behavior, which is ideal for investigating ToM. The corresponding payoff matrix is shown in Tables 4.3 and 4.4.

Table 4.3: Payoff matrix for Agent 0 in the Matching Pennies game.

		Agent 1's Choice	
		0	**1**
Agent 0's Choice	**0**	-1	1
	1	1	-1

Table 4.4: Payoff matrix for Agent 1 in the Matching Pennies game.

		Agent 1's Choice	
		0	**1**
Agent 0's Choice	**0**	1	-1
	1	-1	1

Prisoner's Dilemma

The Prisoner's Dilemma is a foundational game in game theory that models the tension between cooperation and defection. Two agents must independently decide whether to cooperate (choice 0) or defect (choice 1). Although mutual cooperation yields moderate rewards for both, unilateral defection provides a high payoff for the defector and a poor outcome for the cooperator. Mutual defection leads to suboptimal outcomes for both. This game is particularly relevant for studying social dilemmas, reciprocity, and the emergence of trust, making it ideal for evaluating whether Theory of Mind models can facilitate cooperation under selfish incentives.The corresponding payoff matrix is shown in Table 4.5 and Table 4.6.

Table 4.5: Payoff matrix for Agent 0 in the Prisoner's Dilemma game.

		Agent 1's Choice	
		0	**1**
Agent 0's Choice	**0**	1	5
	1	0	3

Table 4.6: Payoff matrix for Agent 1 in the Prisoner's Dilemma game.

		Agent 1's Choice	
		0	**1**
Agent 0's Choice	**0**	1	0
	1	5	3

4.3.4 Interaction Dynamics

The experiment was structured as a round-robin tournament in which each unique pair of agents interacted in two game environments. Each match consisted of 100 rounds and, to ensure statistical reliability, 30 simulations were run for each agent pairing. This setup enabled a comprehensive comparison of agent behavior across different levels of Theory of Mind sophistication and across different game types.

Although the `tomsup` package provides a built-in environment to manage round-robin matches, a custom simulation loop was implemented instead. This approach offered greater flexibility in storing and accessing interaction data, particularly to build runtime visualizations and compute performance metrics for each round. By manually managing the interactions, it was possible to collect granular data on the dynamics of decision making and adapt the data to the needs of the analysis.

To maintain consistency with the recursive nature of ToM agents, symmetric matches (e.g., 1-ToM vs. 1-ToM) were excluded, as agents are designed to model opponents of lower sophistication. This ensures that each interaction aligns with the intended modeling assumptions of the k-ToM framework.

4.3.5 Data Collection

In order to analyze the behavioral dynamics and performance of agents in different pairings and games, a structured data collection approach was employed. The `tomsup` framework provides built-in functionality to track agent behavior throughout the simulation via the `save_history=True` flag. This parameter was set for all participating agents, enabling each of them to record interaction histories during matches. The data frame stores the choice of each agent as well as their resulting payoff with additional columns reporting simulation number and competing agent pair (agent0 and agent1). Summing the payoff columns would determine the winner.

Following the simulation phase, the data was organized into structured `DataFrame` objects using the `pandas` library. This allowed efficient post-processing and exporting of results into CSV format for further analysis. The complete interaction histories and the time required to execute 30 simulations of 100 rounds per agent pair were kept to support reproducibility and enable visualization. Visualizations themselves were presented using graphs such as runtime graphs and behavioral performance graphs. This allowed identifying which agent configurations were the most computationally intensive, particularly in relation to higher-order k-ToM models.

This setup ensured that every critical component of agent behavior was systematically recorded, thereby supporting robust comparative analysis of different levels of Theory of Mind under varied game-theoretic conditions.

4.3.6 Metrics

In order to systematically evaluate the behavior and computational demands of the k-ToM agents in different game environments, several key performance metrics were used. These metrics were selected to provide both behavioral insight, regarding how well agents perform and adapt, and technical understanding of the computational trade-offs associated with increasing model sophistication. The metrics used for the evaluation are shown below:

- **Average Score per Agent**
  This metric captures the mean payoff achieved by each agent in all simulations. It serves as a direct indicator of the agent's strategic success in different game settings. The results are visualized using heatmaps and bar plots to facilitate the comparison of agent types and opponent configurations.

- **Learning Speed and Convergence Behavior**
  This measure reflects how quickly agents reach stable performance during repeated interactions. It is calculated as the mean score for all simulation rounds and plotted over time to observe adaptation dynamics and convergence trends.

- **Computational Cost**
  This metric evaluates the time required to simulate 30 iterations of 100 rounds for each pair of agents. It highlights the scalability and efficiency of the agents, particularly with respect to increasing levels of ToM sophistication. The results are presented in the form of a runtime plot, allowing for a clear comparison of processing demands.

Together, these metrics offer a comprehensive evaluation for analyzing the relationship between Theory of Mind sophistication, strategic performance, and computational feasibility.

Chapter 5

Experimental Evaluation

The purpose of the conducted experiment was to investigate how different levels of Theory of Mind (ToM) sophistication affect agent performance in different strategic environments. More specifically, the experiment aimed to evaluate whether increasing recursive reasoning abilities lead to more advantageous outcomes in multi-agent interactions. By simulating agents with k-ToM levels ranging from 0 to 5 the study explored how these agents behave and compete within two distinct game-theoretic scenarios: Matching Pennies and Prisoner's Dilemma.

The evaluation presented in this chapter is based on several key performance metrics: average score per agent, learning speed and convergence, and computational cost. These metrics were chosen to provide a comprehensive understanding of agent behavior from multiple perspectives, including efficiency, stability, and adaptability.

5.1 Agent performance across games

Firstly, we will investigate how the increase in ToM sophistication affects agents' ability to adapt, anticipate, and respond to different opponents. The outcomes were analyzed through several key performance indicators, including pairwise performance scores using heatmaps, overall average scores per agent via bar graphs and learning dynamics which were conducted using mean scores across simulations. The performance results were mainly based on the average score per agent pair, and different behavior is shown through different game environments, which are analyzed in the next sections.

5.1.1 Matching Pennies

Matching Pennies is a classic competitive game in which one agent's gain is precisely the other's loss, making it a perfect zero-sum scenario to test adaptive strategic behavior.

Figure 5.1 presents a heatmap visualization of the average scores achieved by agents in pairs at all ToM levels (0 to 5). Heatmap analysis reveals a clear trend: increased ToM sophistication tends to improve performance. The asymmetric nature of the results (where higher-order agents consistently outperform lower-order agents) validates the theoretical framework that more sophisticated ToM reasoning should show strategic advantages in competitive environments. In particular, this trend appears to be more distinct in the lower ToM range. For example, both the 0-ToM and 1-ToM agents exhibit a greater performance deficit when facing their immediate k+1 opponent (e.g., 1-ToM vs. 2-ToM) than when competing against the most complex model (e.g., 1-ToM vs. 5-ToM). This pattern changes for agents with $k \geq 2$, where the negative impact of facing more sophisticated opponents becomes increasingly linear and more pronounced. For example, the 2-ToM agent performs nearly evenly against 3-ToM (average score: -0.019), but its performance drops further against 4-ToM and 5-ToM (average scores: -0.051 and -0.053, respectively).

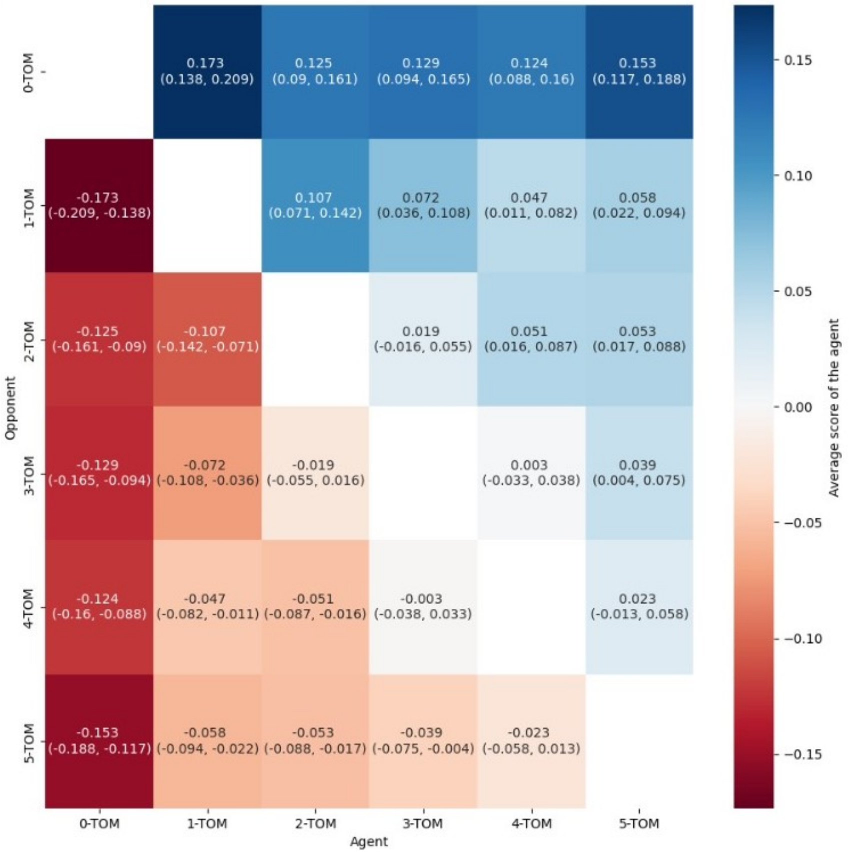

Figure 5.1: Heatmap of average scores per agent pair across Theory of Mind levels in Matching pennies game. The matrix displays performance outcomes when agents with different ToM sophistication (0 to 5) compete against each other. Color intensity represents score magnitude, with blue indicating positive scores and red indicating negative scores. Values in parentheses show confidence intervals.

Additionally, comparisons among higher-level ToM agents reveal nuanced dynamics. The 3-ToM and 4-ToM agents demonstrate balanced outcomes in direct competition, with a near-zero average score (-0.003), indicating mutual adaptation. However, 3-ToM performs significantly worse when faced with 5-ToM, while 4-ToM shows marginally better performance in the same matchup, outperforming 3-ToM by 0.017 points. These results suggest that although 3-ToM and 4-ToM behave similarly in middle-tier encounters, the marginal gain of increased sophistication becomes evident when challenged by the most complex agents. In particular, the performance difference between the 4-ToM and 5-ToM agents appears minimal in most matches, which suggests that going beyond level 4 may not bring much extra benefit. The only notable exception is when facing 3-ToM agents, where 5-ToM outperforms 4-ToM. This indicates that beyond a certain point ($k \geq 4$), the added sophistication does not translate into significant performance gains between all opponents.

The general trend observed in the heatmap is further supported by the aggregated average score per agent shown in the bar graph (Figure 5.2). As illustrated, average performance increases with higher ToM sophistication, confirming that agents equipped with deeper recursive models are better suited for this competitive setting.

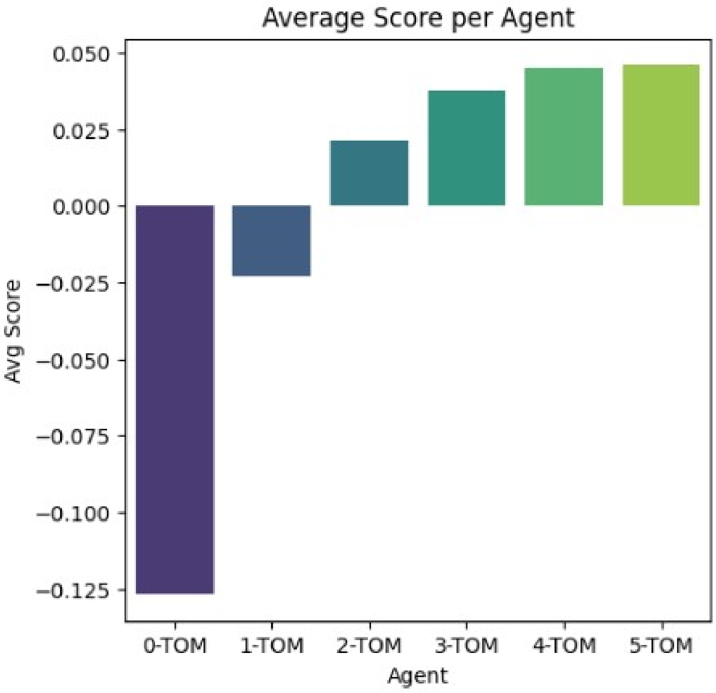

Figure 5.2: Average score per agent in the Matching Pennies game. Higher ToM sophistication is associated with improved overall performance, confirming the trends observed in the heatmap.

Learning Speed and Convergence Behavior. To gain further insight into agent behavior beyond cumulative performance, we analyzed the progression of average scores in 100 interaction rounds, repeated over 30 simulations. This metric provides a view into how quickly and effectively each agent adapts its strategy over time, as a proxy for learning speed and convergence stability.

As presented in Figures 5.3 and 5.4, lower-level agents **0-ToM** and **1-ToM** exhibit faster convergence early in the game. This is somewhat intuitive, as these agents operate with relatively simple internal models; for example, 0-ToM assumes a static bias in its opponent, which can be estimated quickly. Likewise, 1-ToM models its opponent as a 0-ToM agent, making its learning dynamics simple, but fast. Their limited depth results in rapid stabilization, though not necessarily in optimal outcomes.

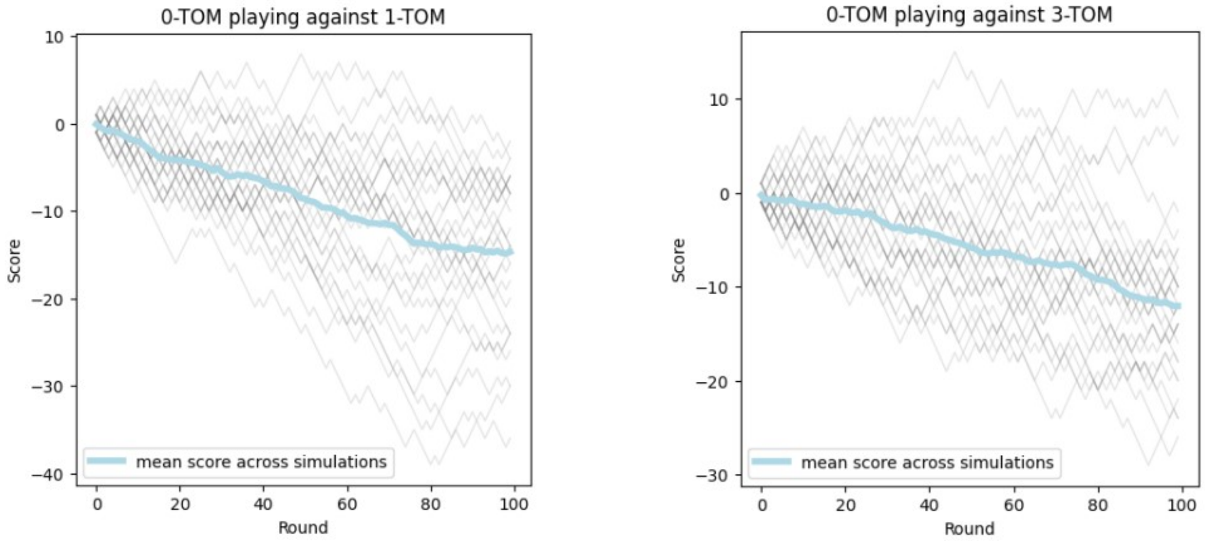

Figure 5.3: Mean score across simulations for 0-ToM agent

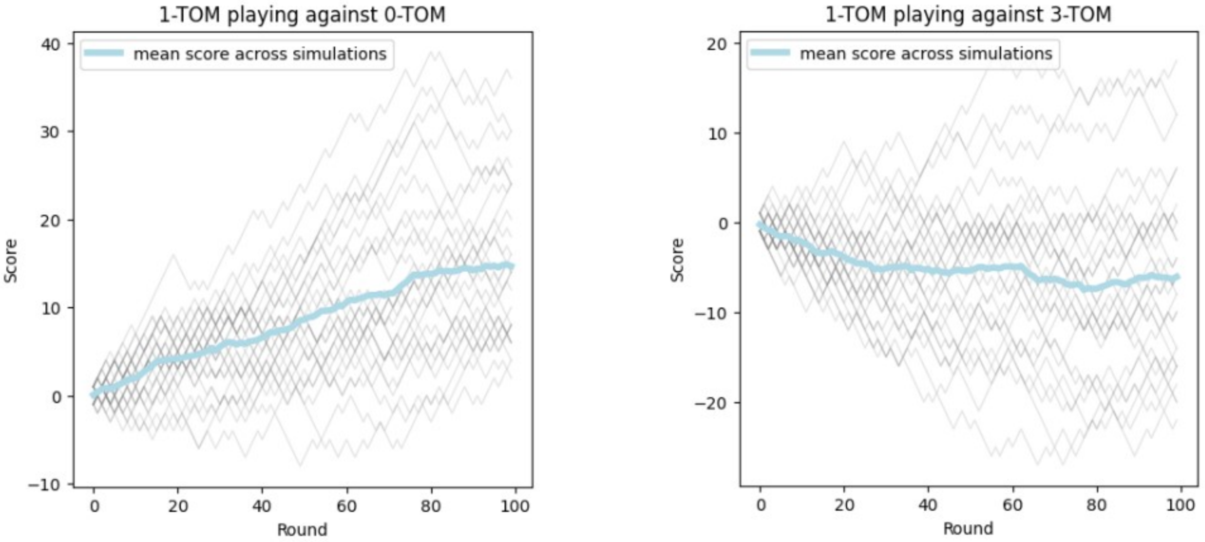

Figure 5.4: Mean score across simulations for 1-ToM agent

On the other hand, higher-order agents (**2-ToM** and above) show slower convergence. In particular, **3-ToM**, **4-ToM**, and **5-ToM** require more rounds to approach stable performance, reflecting the increased computational demands and recursive inference involved in simulating both the opponent's beliefs and their beliefs about the agent's own model. This extended adaptation process allows for more nuanced strategic reasoning, but also introduces volatility, especially in early rounds.

Interestingly, the dynamics between agents of similar sophistication levels with $k > 2$ tend to produce oscillations around the average score, as illustrated in Figure 5.5. This suggests that mutual recursive modeling leads to strategic ambiguity: each agent continuously updates its beliefs in response to an opponent that is doing the same. This reinforces the earlier observation that increased sophistication beyond a certain point may yield diminishing returns, except in selective pairings such as **5-ToM vs 3-ToM**, where the deeper recursion of 5-ToM allows it to outperform slightly (as seen in the average score heatmap).

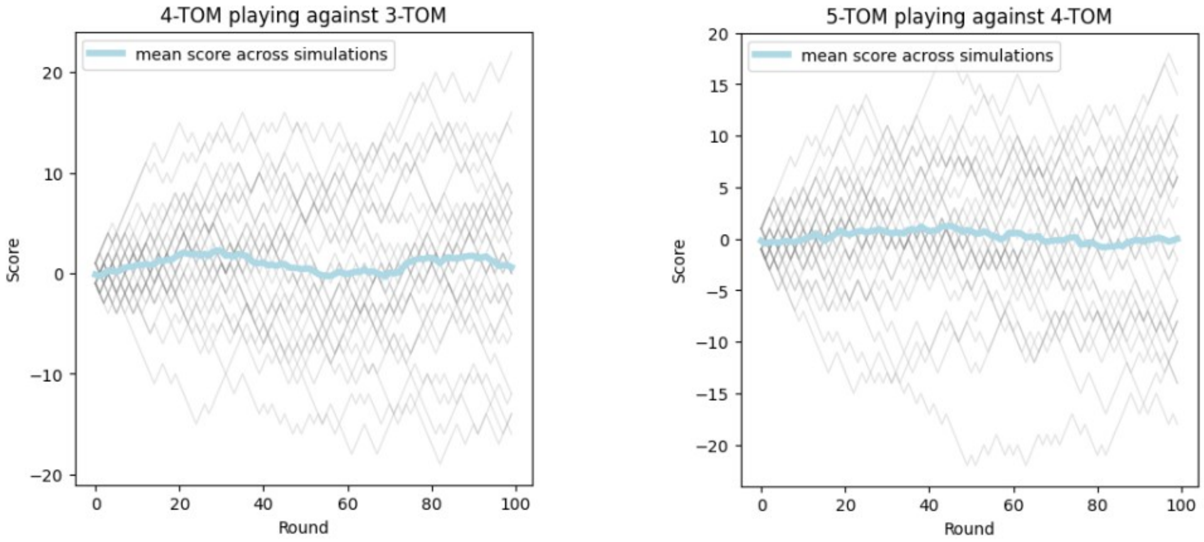

Figure 5.5: Mean score across simulations for agents with similar sophistication level with $k > 2$

5.1.2 Prisoner's Dilemma

The Prisoner's Dilemma is a classical game-theoretic scenario that models the tension between cooperation and defection. In this setting, each of the two players simultaneously chooses to cooperate or to defect. Mutual cooperation yields moderate rewards for both agents, while mutual defection results in lower payoffs. However, if one agent defects while the other cooperates, the defector receives a high payoff, and the cooperator gets the lowest possible outcome. This creates a dilemma: Although mutual cooperation is collectively optimal, defection is the dominant strategy for self-interested agents. The game thereby provides a useful environment to evaluate the strategic depth of recursive Theory of Mind (ToM) agents, particularly in testing whether increased sophistication leads to more cooperative or exploitative behavior.

The heatmap summarizing the average scores per agent pair in the simulations in the Prisoner's Dilemma reveals several interesting patterns. (see Figure 5.6) Notably, when lower-k agents (e.g., 0-ToM or 1-ToM) are paired against more sophisticated agents (e.g., 2-ToM to 5-ToM), they tend to achieve significantly higher average scores, often ranging between 4.981 and 4.991. In contrast, higher-k agents in those same matches average much lower scores, typically between 0.013 and 0.022. This implies that, contrary to what one might expect from more complex mentalizing, the higher-k agents are at a disadvantage when paired with more naive agents. The reason for this is rooted in the behavior of sophisticated ToM agents: they often lean toward cooperation by modeling their opponent's internal state, assuming a certain level of strategic reasoning. However, when faced with lower-k agents that behave more greedily or defect more frequently, this cooperative intent is not reciprocated, resulting in poorer outcomes for the high-k agents.

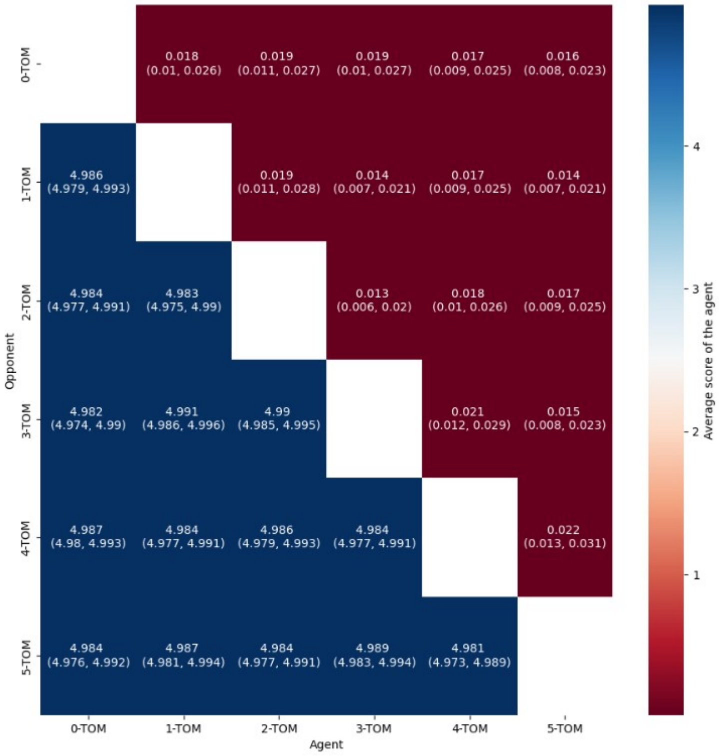

Figure 5.6: Heatmap displaying the average scores obtained by each agent in pairwise competitions within the Prisoner's Dilemma setting. Rows represent the agent indexed as Agent 0, while columns represent Agent 1. Cooler colors (blue) indicate higher average scores, while warmer colors (red) indicate lower scores. The diagonal represents twin-pair matches.

This finding suggests that increased ToM sophistication does not necessarily offer a strategic advantage in cooperative contexts. Being more sophisticated than a 2-ToM agent often yields no added benefit in cooperative interactions. In our heatmap, this is reflected in the fact that agents 2-ToM, 3-ToM, and 4-ToM show relatively similar average results, and the marginal improvement between 4-ToM and 5-ToM is insignificant. Although increased sophistication equips agents to infer the intentions of more complex opponents, it also makes them more susceptible to being exploited by simpler agents that do not reciprocate mentalizing or cooperative reasoning.

These heatmap results show that, in the Prisoner's Dilemma setting, having a higher sophistication level is only valuable if the opponent can also think and act cooperatively. Otherwise, over-mentalizing may lead to systematic losses against selfish, simplistic agents.

In line with the trends observed in the heatmap, the bar graph shown in Figure 5.7) describing the average score per agent further supports the interpretation that lower ToM sophistication may be more advantageous in cooperative environments such as the Prisoner's Dilemma. The 0-ToM agent achieves the highest overall average score, with scores gradually decreasing as the sophistication level increases. The 5-ToM agent records the lowest average score among all, indicating that excessive modeling of the opponent may lead to overthinking or misaligned expectations in cooperative settings. This outcome highlights that in games where mutual cooperation yields the best individual results, simpler agents may behave in a more predictable and mutually beneficial manner, while highly recursive agents may misinterpret the intentions of the opponent or act overly defensively.

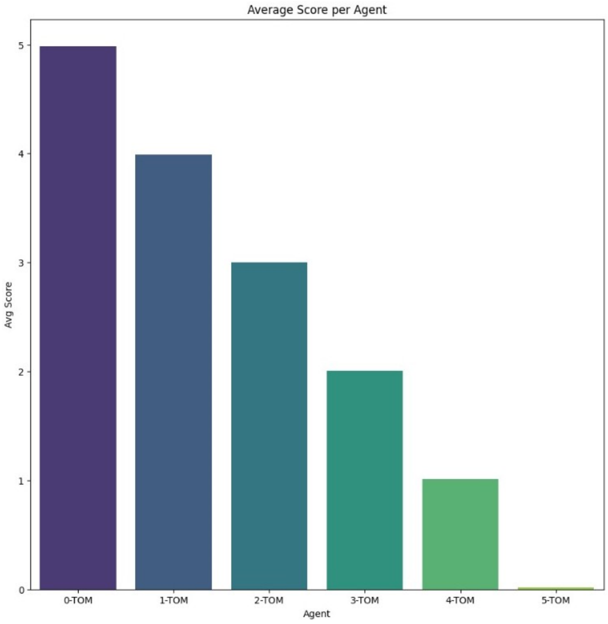

Figure 5.7: Bar graph showing the average scores obtained by each agent in the Prisoner's Dilemma environment. Simpler agents (e.g., 0-ToM) achieve higher average scores, while scores steadily decrease as the sophistication level increases.

Learning Speed and Convergence Behavior. The learning dynamics observed through average score progression across simulations further reinforce the performance trends highlighted by the heatmap and bar graph. In this case, the learning speed is inversely correlated with the level of sophistication of the agent. Lower k agents consistently demonstrate faster convergence compared to their more complex counterparts. For example, agent 0-ToM exhibits a stable and linear improvement when competing against a 1-ToM opponent, gaining approximately 5 points per round, culminating in a score of around 500 over 100 rounds (see Figure 5.8). This indicates a high level of adaptability and responsiveness to the opponent's strategy.

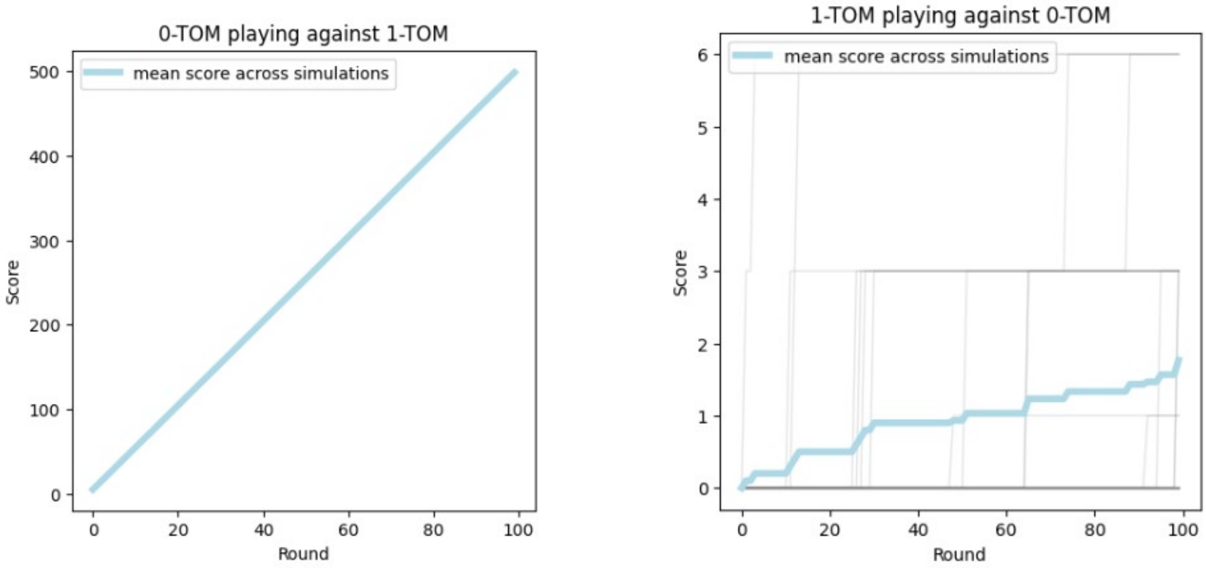

Figure 5.8: Learning trajectory of a 0-ToM agent against a 1-ToM agent in the Prisoner's Dilemma environment. The consistent linear growth illustrates rapid and stable adaptation.

This linear convergence pattern is a recurring trait when a lower k agent competes against a more sophisticated opponent. Across these pairings, the learning curve follows a trajectory close to a 45-degree line, representing steady learning. Meanwhile, differences begin to emerge among higher-k agents based on their opponent's sophistication. For example, the 2-ToM agent learns faster against a 1-ToM opponent than against a 0-ToM opponent. The 3-ToM agent exhibits very similar learning rates in all less sophisticated agents, maintaining consistent performance. In particular, the 4-ToM agent demonstrates rapid progress only during the first 40 rounds against a 0-ToM opponent, after which the learning slows down and resembles a logarithmic trend. When facing an agent with $1 \leq k \leq 3$, the learning curve of the 4-ToM agent closely parallels that of the 3-ToM agent. 5-ToM agent mostly mirrors the learning curve of the 4-ToM agent, with slightly improved pace and convergence.

An interesting observation across all pairings where higher k agents play against lower k agents is the convergence toward a similar score, approximately two points on the plot scale. This suggests that in a cooperative game, better results come from working together and staying consistent, not from increasing sophistication level.

5.1.3 Computational Cost Analysis

Purpose and Scope

As artificial agents become more cognitively sophisticated, understanding the computational implications of this sophistication becomes crucial. Although higher-order Theory of Mind (ToM) agents are capable of modeling deeper levels of recursive reasoning, this comes with higher computing costs, so it is important to balance the benefits with the extra effort needed.

This subsection aims to assess the computational efficiency of ToM agents with different sophistication levels, focusing on the trade-off between performance and run-time complexity. In real-world use, intelligent agents require real-time responsiveness and limited resources, so it is important to know when adding more thinking power is no longer worth the extra cost. The analysis presented here is intended to inform decisions about the practical feasibility of using high-level ToM agents in different applied contexts.

Method of Measuring Cost

To assess the computational cost associated with agents of different levels of sophistication, a manual simulation loop was implemented in Google Colab. For each unique agent pair, the time required to complete 30 simulations of 100 rounds was recorded using Python's `time` module. These timing values were aggregated and used to build a runtime plot that serves as the central reference point for evaluating computational efficiency across agent configurations. The measured durations reflect the time needed to execute the full interaction between two agents.

Figure 5.9 presents the results of this analysis. The horizontal axis represents the agent configuration (agent A vs. agent B), while the vertical axis denotes the time required to complete the simulations, expressed in seconds. Each color-coded bar on the graph corresponds to a different opponent agent.

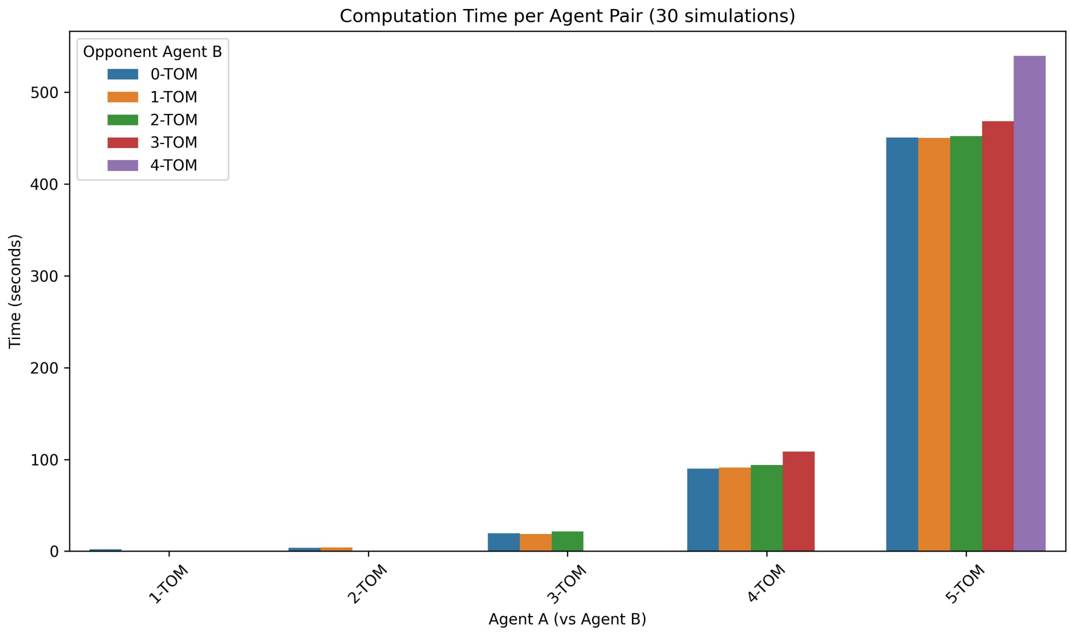

Figure 5.9: Computation Time per Agent Pair across 30 simulations.

Key Observations:

- There is a clear and rapid increase in the computation time as the level of sophistication increases. The progression from 1-ToM to 5-ToM shows a non-linear escalation in cost, especially beyond level 3.
- The runtime for 1-ToM and 2-ToM agents remains relatively minimal and similar, averaging less than 10 seconds regardless of the opponent.
- Starting from 3-ToM, runtime increases noticeably (up to 25 seconds), while 4-ToM agents take more than 100 seconds for the same simulation workload.
- The most significant leap is observed at the 5-ToM level, where the simulation time exceeds 500 seconds when playing against other high-level agents.

Trade-Off Analysis:

The computational demands of recursive ToM models scale sharply with an increasing depth of recursion. Although performance gains (as evidenced by the average score and behavioral adaptiveness) justify the use of agents up to 3-ToM or 4-ToM, the computational cost of employing a 5-ToM agent raises serious questions about diminishing returns. The results of the previous sections suggest that 4-ToM and 5-ToM agents show marginal or insignificant improvements in behavioral metrics over 3-ToM agents in most game scenarios. However, the runtime plot clearly shows that 5-ToM agents take up to 5–10 times longer to compute their responses compared to 3-ToM agents. This implies that the marginal benefits in performance may not warrant the exponential increase in computational effort. For most practical applications, especially those that require real-time responsiveness or have hardware constraints, agents up to level 3 or 4 may strike an optimal balance between cognitive sophistication and computational efficiency. Agent configurations beyond this level should be reserved for research or environments where computational resources are plentiful and strategic depth is paramount.

Chapter 6

Conclusion

This research aimed to find out if agents with higher levels of recursive thinking perform better in social games and if that improvement is worth the extra time and resources needed to run them. By implementing agents with sophistication levels ranging from 0 to 5 using the `tomsup` framework, and evaluating their behavior in both competitive (Matching Pennies) and cooperative (Prisoner's Dilemma) settings, this study aimed to identify the trade-off between cognitive complexity and practical utility in artificial social agents.

The results revealed nuanced dynamics in how different levels of sophistication impact agent behavior, depending significantly on the nature of the social environment. In the Matching Pennies game, which reflects a zero-sum, adversarial setting, increasing depth of recursion clearly exhibited a strategic advantage. Higher-k agents consistently outperformed their less sophisticated counterparts, with 3-ToM, 4-ToM, and 5-ToM agents showing improved adaptability and more accurate modeling of their opponents' behaviors.

In the Prisoner's Dilemma, a game grounded in cooperation, the benefits of deeper recursion levels were less pronounced. The agents with the highest performance, in terms of average payoff, were k-tom agents with $k < 3$, particularly when playing against opponents of similar level. This suggests that in cooperative scenarios, hyper-sophistication may not yield better outcomes and can even disrupt mutual cooperation because of the over-mentalizing or excessive modeling. Higher-k agents try to infer beliefs and potential actions of their opponents, which can make them expect defection or strategic manipulation even when cooperation would be better for both. In contrast, lower-order agents such as 1-ToM or 2-ToM tend to behave in a more straightforward and self-interested manner, often cooperating simply because it yields favorable rewards without overanalyzing the opponent's intent. Ironically, this simpler heuristic can result in more stable cooperation, especially in settings where both agents follow similar logic. The sophisticated modeling of higher k agents may cause them to "overthink" the interaction, undermining trust in environments that reward cooperation.

These contrasting results underscore an important point: the utility of recursive ToM is not universal but context-dependent. The current modeling framework does not account for social preferences or norms such as fairness, reciprocity, or aversion to inequity, which are often central to human decision making in social dilemmas. For example, agents do not infer fairness preferences or adjust to inequity, which limits their realism in modeling complex human-like cooperation.

An additional layer of analysis focused on the computational feasibility of deploying ToM agents. A runtime analysis demonstrated that the computational cost scales exponentially with the ToM level. Although 1-ToM and 2-ToM agents ran efficiently, 5-ToM agents routinely required more than 500 seconds per pair for 30 simulations, making them impractical for real-time or scalable applications. Even 4-ToM agents exhibited significant computational demand, with runtimes exceeding 100 seconds in some cases.

In particular, while 4-ToM outperformed 3-ToM in certain edge scenarios, such as when facing a 5-ToM opponent in Matching Pennies, these advantages were marginal and did not appear consistently across other contexts. Thus, from a computational and practical standpoint, 3-ToM emerged as the optimal compromise. It provided competitive performance in all games while maintaining a reasonable execution time (under 25 seconds per full simulation round), balancing sophistication with feasibility.

In contrast, 0-ToM agents were shown to consistently outperform in a competitive environment. If viewed as a phenotype in an evolutionary setting, such agents would likely be phased out. Although 1-ToM and 2-ToM agents remain viable options for simple or highly cooperative environments, they lack the adaptability of higher *k* agents. However, their lower computational burden and cooperative alignment make them valuable in specific low-conflict scenarios.

This work confirms the central thesis: Higher sophistication level improves agent performance, but the benefits depend heavily on the environment and must be considered alongside computational limitations. Although deeper recursion levels offer theoretical advantages, they are not always practically justified. In most cases, the 3-ToM agent appears to offer the best trade-off between strategic power and computational cost.

6.1 Future work

These findings open several directions for future research. Incorporating human-like social preferences into the modeling process, such as fairness, reciprocity, or aversion to inequality, could enhance realism. Additionally, adaptive agents capable of dynamically adjusting their ToM level based on environmental cues or past performance might further optimize performance without unnecessary computational expense. Finally, extending simulations to include human participants would provide a richer evaluation of artificial ToM models and their real-world potential.

Chapter 7

Appendix

The following is the Python code used to run simulations in the Google Colab environment.

```
# -*- coding: utf-8 -*-
"""ThesisExperiment.ipynb

Automatically generated by Colab.

Original file is located at
    https://colab.research.google.com/drive/1
       pFCSmoRzVL1b9hoEsUk3GVBda76h_UXh

#Install dependecies
"""

!pip install tomsup

"""#Import dependecies"""

import tomsup as ts
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import scipy.stats as stats
import time

"""#Define agents and environment"""

# create a list of agents
agents = ["0-TOM", "1-TOM", "2-TOM", "3-TOM", "4-TOM", "5-TOM"]

# create a list of their starting parameters (an empty dictionary {} simply
    assumes defaults)
start_params = [
    {"level": 1, "save_history":True},
    {"level": 2, "save_history":True},
    {"level": 3, "save_history":True},
    {"level": 4, "save_history":True},
    {"level": 5, "save_history":True},
    {"level": 6, "save_history":True}
]

# create a group of agents
group = ts.create_agents(agents, start_params)
```

```
print(group)
#set environment
group.set_env(
    env="round_robin"
)

"""#Define payoff matrices"""

penny = ts.PayoffMatrix(name=’penny_competitive’)
print(penny)

prisoners_dilemma = ts.PayoffMatrix(name=’prisoners_dilemma’)
print(prisoners_dilemma)

"""#Matching Pennies game

##Save results from the competition
"""

#results_penny = group.compete(p_matrix=penny, n_rounds=100, n_sim=30,
   verbose=True)

agent_names = group.get_names()

timing_data = []
results_all = []

for i, name_a in enumerate(agent_names):
    for j, name_b in enumerate(agent_names):
        if i <= j:
            continue

        agent_a = group.get_agent(name_a)
        agent_b = group.get_agent(name_b)

        print(f"\n␣Simulating␣pair:␣({name_a},␣{name_b})")

        start_time = time.time()
        df = ts.compete(agent_a, agent_b, p_matrix=penny, n_rounds=100,
           n_sim=30)
        elapsed = time.time() - start_time

        timing_data.append({
            "agent_a": name_a,
            "agent_b": name_b,
            "elapsed_sec": elapsed
        })

        df[’agent_a’] = name_a
        df[’agent_b’] = name_b

        results_all.append(df)

timing_df = pd.DataFrame(timing_data)

df_all = pd.concat(results_all, ignore_index=True)
```

```
df_all.to_csv("MP_results.csv", index=False)
timing_df.to_csv("MP_timing_results.csv", index=False)

group.compete(p_matrix=penny, n_rounds=100, n_sim=30, verbose=True)
results = group.get_results()
results.head()

"""##Plot the results

Main performance metrics:

1. Average Score per Agent => heatmaps and bar graphs
2. Learning Speed / Convergence Time => mean score across simulations plot
3. Performance Variability / Robustness => confidence intervals

Computation Cost metric: Computation Cost => runtime plot

###Heatmap
"""

plt.rcParams["figure.figsize"] = [11, 11]

fig = group.plot_heatmap(cmap="RdBu")

"""###Bar Graphs for Average Score per Agent"""

# First reshape to get all agent performances into one column
agent0_df = results[['agent0', 'payoff_agent0']].rename(columns={'agent0':
    'agent', 'payoff_agent0': 'payoff'})
agent1_df = results[['agent1', 'payoff_agent1']].rename(columns={'agent1':
    'agent', 'payoff_agent1': 'payoff'})
all_agents = pd.concat([agent0_df, agent1_df])

# Group and average
avg_scores = all_agents.groupby('agent')['payoff'].mean().reset_index()

# Plot
sns.barplot(data=avg_scores, x='agent', y='payoff', palette='viridis')
plt.title('Average␣Score␣per␣Agent')
plt.ylabel('Avg␣Score')
plt.xlabel('Agent')
plt.rcParams["figure.figsize"] = [7, 7]
plt.savefig("MP_average_score_bar.jpg", dpi=300)
plt.show()

"""###Learning Speed: Performance Over Time"""

plt.rcParams["figure.figsize"] = [5, 5]
group.plot_score(agent0="0-TOM", agent1="1-TOM", agent=1)
group.plot_score(agent0="0-TOM", agent1="1-TOM", agent=0)

group.plot_score(agent0="1-TOM", agent1="2-TOM", agent=1)
group.plot_score(agent0="1-TOM", agent1="2-TOM", agent=0)

group.plot_score(agent0="0-TOM", agent1="3-TOM", agent=1)
group.plot_score(agent0="0-TOM", agent1="3-TOM", agent=0)

group.plot_score(agent0="1-TOM", agent1="3-TOM", agent=1)
group.plot_score(agent0="1-TOM", agent1="3-TOM", agent=0)

group.plot_score(agent0="2-TOM", agent1="3-TOM", agent=1)
group.plot_score(agent0="2-TOM", agent1="3-TOM", agent=0)

group.plot_score(agent0="0-TOM", agent1="4-TOM", agent=1)

group.plot_score(agent0="1-TOM", agent1="4-TOM", agent=1)

group.plot_score(agent0="2-TOM", agent1="4-TOM", agent=1)

group.plot_score(agent0="3-TOM", agent1="4-TOM", agent=1)
group.plot_score(agent0="3-TOM", agent1="4-TOM", agent=0)

group.plot_score(agent0="3-TOM", agent1="5-TOM", agent=1)

group.plot_score(agent0="2-TOM", agent1="5-TOM", agent=1)

group.plot_score(agent0="1-TOM", agent1="5-TOM", agent=1)
group.plot_score(agent0="1-TOM", agent1="5-TOM", agent=0)

plt.rcParams["figure.figsize"] = [5, 5]
group.plot_score(agent0="0-TOM", agent1="5-TOM", agent=1)
group.plot_score(agent0="0-TOM", agent1="5-TOM", agent=0)

group.plot_score(agent0="4-TOM", agent1="5-TOM", agent=1)
group.plot_score(agent0="4-TOM", agent1="5-TOM", agent=0)

"""###Runtime plot"""

plt.figure(figsize=(10, 6))
sns.barplot(data=timing_df, x="agent_a", y="elapsed_sec", hue="agent_b")
plt.title("Computation␣Time␣per␣Agent␣Pair␣(30␣simulations)")
plt.ylabel("Time␣(seconds)")
plt.xlabel("Agent␣A␣(vs␣Agent␣B)")
plt.xticks(rotation=45)
plt.legend(title="Opponent␣Agent␣B")
plt.tight_layout()
plt.savefig("MP_runtime_plt.jpg", dpi=300)
plt.show()

"""#Prisoners dilemma game

##Save results from the competition
"""

group.compete(p_matrix=prisoners_dilemma, n_rounds=100, n_sim=30, verbose=
    True)
results_pd = group.get_results()

results_pd.to_csv("PD_results.csv", index=False)

"""##Plot the results

###Heatmap
"""

plt.rcParams["figure.figsize"] = [11, 11]
```

```
group.plot_heatmap(cmap="RdBu")

"""###Bar Graphs for Average Score per Agent

"""

agent0_df = results_pd[['agent0', 'payoff_agent0']].rename(columns={'agent0
    ': 'agent', 'payoff_agent0': 'payoff'})
agent1_df = results_pd[['agent1', 'payoff_agent1']].rename(columns={'agent1
    ': 'agent', 'payoff_agent1': 'payoff'})
all_agents = pd.concat([agent0_df, agent1_df])

avg_scores = all_agents.groupby('agent')['payoff'].mean().reset_index()

sns.barplot(data=avg_scores, x='agent', y='payoff', palette='viridis')
plt.title('Average␣Score␣per␣Agent')
plt.ylabel('Avg␣Score')
plt.xlabel('Agent')
plt.show()

"""###Learning Speed: Performance Over Time"""

plt.rcParams["figure.figsize"] = [5, 5]
group.plot_score(agent0="0-TOM", agent1="1-TOM", agent=1)
group.plot_score(agent0="0-TOM", agent1="1-TOM", agent=0)

group.plot_score(agent0="0-TOM", agent1="2-TOM", agent=1)
group.plot_score(agent0="0-TOM", agent1="2-TOM", agent=0)

group.plot_score(agent0="1-TOM", agent1="2-TOM", agent=1)

group.plot_score(agent0="1-TOM", agent1="2-TOM", agent=0)

group.plot_score(agent0="0-TOM", agent1="3-TOM", agent=1)

group.plot_score(agent0="1-TOM", agent1="3-TOM", agent=1)

group.plot_score(agent0="2-TOM", agent1="3-TOM", agent=1)
group.plot_score(agent0="2-TOM", agent1="3-TOM", agent=0)

group.plot_score(agent0="0-TOM", agent1="4-TOM", agent=1)

group.plot_score(agent0="1-TOM", agent1="4-TOM", agent=1)

group.plot_score(agent0="2-TOM", agent1="4-TOM", agent=1)

group.plot_score(agent0="3-TOM", agent1="4-TOM", agent=1)
group.plot_score(agent0="3-TOM", agent1="4-TOM", agent=0)

group.plot_score(agent0="0-TOM", agent1="5-TOM", agent=1)

group.plot_score(agent0="1-TOM", agent1="5-TOM", agent=1)

group.plot_score(agent0="2-TOM", agent1="5-TOM", agent=1)

group.plot_score(agent0="3-TOM", agent1="5-TOM", agent=1)
group.plot_score(agent0="4-TOM", agent1="5-TOM", agent=1)
group.plot_score(agent0="4-TOM", agent1="5-TOM", agent=0)
```

Listing 7.1: Simulation Loop for Agent Pairs

Bibliography

[1] Abrini, M., Abend, O., Acklin, D., Admoni, H., Aichinger, G., Alon, N., Ashktorab, Z., Atreja, A., Auron, M., Aufreiter, A., Awasthi, R., Banerjee, S., Barnby, J. M., Basappa, R., Bergsmann, S., Bouneffouf, D., Callaghan, P., Cavazza, M., Chaminade, T., Chernova, S., Chetouan, M., Choudhury, M., Cleeremans, A., Cywinski, J. B., Cuzzolin, F., Deng, H., Diamond, N., Pasquasio, C. D., Dumas, G., van Duijn, M., Dwarikanath, M., Gao, Q., Goel, A., Goldstein, R., Gombolay, M., Gonzalez, G. E., Halilovic, A., Halmdienst, T., Islam, M., Jara-Ettinger, J., Kastel, N., Keydar, R., Khanna, A. K., Khoramshahi, M., Kim, J., Kim, M., Kim, Y., Krivic, S., Krasnytskyi, N., Kumar, A., Kwon, J., Lee, E., Lee, S., Lewis, P. R., Li, X., Li, Y., Lewandowski, M., Lloyd, N., Luebbers, M. B., Luo, D., Lyu, H., Mahapatra, D., Maheshwari, K., Mainali, M., Mathur, P., Mederitsch, P., Miura, S., de Miranda, M. P., Mirsky, R., Mishra, S., Moorman, N., Morrison, K., Muchovej, J., Nessler, B., Nessler, F., Nguyen, H. M. J., Ortego, A., Papay, F. A., Pasquali, A., Rahimi, H., Raghu, C., Royka, A., Sarkadi, S., Scheuerman, J., Schmid, S., Schrater, P., Sen, A., Sheikhbahaee, Z., Shi, K., Simmons, R., Singh, N., Smith, M. O., van der Meulen, R., Solaki, A., Sun, H., Szolga, V., Taylor, M. E., Taylor, T., Waveren, S. V., Vargas, J. D., Verbrugge, R., Wagner, E., Weisz, J. D., Wen, X., Yeoh, W., Zhang, W., Zhao, M., Zilberstein, S., “Proceedings of 1st workshop on advancing artificial intelligence through theory of mind”, dostupno na: https://arxiv.org/abs/2505.03770 2025.

[2] Baker, C., Saxe, R., Tenenbaum, J., “Bayesian theory of mind: Modeling joint belief-desire attribution”, in Proceedings of the annual meeting of the cognitive science society, Vol. 33, No. 33, 2011.

[3] Shu, T., Bhandwaldar, A., Gan, C., Smith, K. A., Liu, S., Gutfreund, D., Spelke, E., Tenenbaum, J. B., Ullman, T. D., “Agent: A benchmark for core psychological reasoning”, dostupno na: https://arxiv.org/abs/2102.12321 2021.

[4] Sicilia, A., Alikhani, M., “Evaluating theory of (an uncertain) mind: Predicting the uncertain beliefs of others in conversation forecasting”, dostupno na: https://arxiv.org/abs/2409.14986 2024.

[5] Yuan, L., Fu, Z., Zhou, L., Yang, K., Zhu, S.-C., “Emergence of theory of mind collaboration in multiagent systems”, dostupno na: https://arxiv.org/abs/2110.00121 2021.

[6] Jin, C., Wu, Y., Cao, J., Xiang, J., Kuo, Y.-L., Hu, Z., Ullman, T., Torralba, A., Tenenbaum, J. B., Shu, T., “Mmtom-qa: Multimodal theory of mind question answering”, dostupno na: https://arxiv.org/abs/2401.08743 2024.

[7] Grislain, C., Caselles-Dupr’e, H., Sigaud, O., Chetouani, M., “Utility-based adaptive teaching strategies using bayesian theory of mind”, ArXiv, Vol. abs/2309.17275, 2023, dostupno na: https://api.semanticscholar.org/CorpusID:263310739

[8] Langley, C., Cirstea, B. I., Cuzzolin, F., Sahakian, B. J., “Theory of mind and preference learning at the interface of cognitive science, neuroscience, and ai: A review”, Frontiers in artificial intelligence, Vol. 5, 2022, str. 778852.

[9] Baron-Cohen, S., Mindblindness: An Essay on Autism and Theory of Mind. The MIT Press, 02 1995, dostupno na: https://doi.org/10.7551/mitpress/4635.001.0001

[10] Kaplan, A., Haenlein, M., “Siri, siri, in my hand: Who’s the fairest in the land? on the interpretations, illustrations, and implications of artificial intelligence”, Business horizons, Vol. 62, No. 1, 2019, str. 15–25.

[11] LeCun, Y., Bengio, Y., Hinton, G., “Deep learning”, nature, Vol. 521, No. 7553, 2015, str. 436–444.

[12] Gao, Y., Yang, F., Frisk, M., Hemandez, D., Peters, C., Castellano, G., “Learning socially appropriate robot approaching behavior toward groups using deep reinforcement learning”, in 2019 28th IEEE international conference on robot and human interactive communication (RO-MAN). IEEE, 2019, str. 1–8.

[13] Williams, J., Fiore, S. M., Jentsch, F., “Supporting artificial social intelligence with theory of mind”, Frontiers in artificial intelligence, Vol. 5, 2022, str. 750763.

[14] O’Grady, C., Kliesch, C., Smith, K., Scott-Phillips, T. C., “The ease and extent of recursive mindreading, across implicit and explicit tasks”, Evolution and Human Behavior, Vol. 36, No. 4, 2015, str. 313-322, dostupno na: https://www.sciencedirect.com/science/article/pii/S1090513815000148

[15] Wilson, R., Hruby, A., Perez-Zapata, D., van der Kleij, S. W., Apperly, I. A., “Is recursive “mindreading” really an exception to limitations on recursive thinking?”, Journal of Experimental Psychology: General, Vol. 152, No. 5, 2023, str. 1454.

[16] Waade, P. T., Enevoldsen, K. C., Vermillet, A.-Q., Simonsen, A., Fusaroli, R., “Introducing tomsup: Theory of mind simulations using python”, Behavior Research Methods, Vol. 55, No. 5, 2023, str. 2197–2231.

[17] Harré, M. S., “What can game theory tell us about an ai ‘theory of mind’?”, Games, Vol. 13, No. 3, 2022, str. 46.