Journal of Artificial Intelligence Research 82 (2025) 285-311 Submitted 04/2024; published 01/2025

ToMA: Computational Theory of Mind with Abstractions for Hybrid Intelligence

Emre Erdogan E.ERDOGAN1@UU.NL
Utrecht University, Utrecht, Netherlands

Frank Dignum DIGNUM@CS.UMU.SE
Umeå University, Umeå, Sweden

Rineke Verbrugge L.C.VERBRUGGE@RUG.NL
University of Groningen, Groningen, Netherlands

Pınar Yolum P.YOLUM@UU.NL
Utrecht University, Utrecht, Netherlands

Abstract

Theory of mind refers to the human ability to reason about the mental content of other people, such as their beliefs, desires, and goals. People use their theory of mind to understand, reason about, and explain the behaviour of others. Having a theory of mind is especially useful when people collaborate, since individuals can then reason on what the other individual knows as well as what reasoning they might do. Similarly, hybrid intelligence systems, where AI agents collaborate with humans, necessitate that the agents reason about the humans using computational theory of mind. However, to try to keep track of all individual mental attitudes of all other individuals becomes (computationally) very difficult. Accordingly, this paper provides a mechanism for computational theory of mind based on abstractions of single beliefs into higher-level concepts. These abstractions can be triggered by social norms and roles. Their use in decision making serves as a heuristic to choose among interactions, thus facilitating collaboration. We provide a formalization based on epistemic logic to explain how various inferences enable such a computational theory of mind. Using examples from the medical domain, we demonstrate how having such a theory of mind enables an agent to interact with humans effectively and can increase the quality of the decisions humans make.

1. Introduction

Hybrid intelligence requires human-agent collaboration, where a human and a computational agent complement each other in the tasks that they achieve. Many times their interactions require a mixed initiative. Computational agents are excellent at processing large amounts of data quickly and accurately (High, 2012), as well as performing repetitive tasks with precision (Van der Aalst et al., 2018). On the other hand, humans have creativity, intuition, and the ability to reason in complex, non-linear ways. By working together, computational agents can enhance human decision-making and problem-solving abilities, while humans can provide context, judgment, and critical thinking skills that machines lack, paving the way for potentially revolutionizing many industries and improving the quality of life for people around the world. In addition to performing their individual tasks, agents and humans need to interact often and effectively so that they can create successful collaborations. To realize these interactions, agents need to be empowered further with capabilities that humans use on an everyday basis. One of these crucial capabilities is the modeling of Theory of Mind (ToM). Put simply, this capability enables a human to reason about other humans, making it possible to understand and predict their behaviour (Premack & Woodruff, 1978; Carruthers & Smith, 1996; Bamicha & Drigas, 2022; Ho et al., 2022). It is even possible for humans to use higher-order ToM reasoning to infer how others employ ToM (e.g., Alice believes that I do not know that she is an expert on this topic). This capability of ToM in humans is crucial in order to develop and employ social skills, such as coordination, negotiation, persuasion, etc. These social skills allow humans to carry out tasks effectively and efficiently, thereby allowing human social interactions to create added value to all parties.

To understand how ToM works, various computational models have been developed. An important line of research analyzed its use in game settings where the rules of the game are well-defined and possible behaviours are limited (de Weerd et al., 2013, 2014b, 2015; Kröhling & Martínez, 2019; de Weerd et al., 2014a; Osten et al., 2017; de Weerd et al., 2017). Experiments in competitive, cooperative, as well as mixed-motive settings show that agents equipped with ToM reasoning achieve better results compared to agents without them. Various techniques to model ToM exist. For example, Baker *et al.* (2011) model ToM within a Bayesian framework using partially observable Markov decision processes. Their evaluation in a simple spatial setting is promising. Winfield (2018) shows how robots use a ToM model by imitating other robots' actions. Using simple ethical rules, they show that ToM helps to improve robots' safety.

An important area where computational Theory of Mind could be of particular use is hybrid intelligence (Akata et al., 2020), where an agent can collaborate with a human towards a particular goal, where the agent would have varying capabilities that could complement those of the human to yield the goal. As an example, consider a computational agent doctor that is designed to collaborate with a human doctor. Such an agent doctor's capabilities can include cooperating with surgeons in operations (Shademan et al., 2016) as well as providing assistance to improve medical diagnosis processes (Gargeya & Leng, 2017). For a more complete human-agent collaboration to take place, an ideal agent doctor should not only function as a medical support tool, but also be able to understand the doctor's behaviour, communicate well with her, and continuously learn from their shared experience. Thus, we argue that the agent doctor would benefit from having a functional computational ToM for the human doctor in achieving their collective goals in such hybrid settings.

There has been a lot of research on human-machine collaboration in various domains such as negotiation (Hindriks et al., 2008), planning (Sycara et al., 2010), and behavioral support systems (Shamekhi et al., 2017). However, the use of computational ToM in human-machine collaboration is relatively new. Hiatt *et al.* (2011) describe a ToM robot model based on the ACT-R cognitive architecture (Anderson, 2009) to account for human behavioral variability in human-robot teams. Devin and Alami (2016) develop a ToM-based agent framework for collaborative task achievement. Their system takes mental states regarding the goals, plans, and actions of humans into account when executing human-robot shared plans. Buehler and Weisswange (2020) propose a ToM-based communication framework for human-agent cooperation. They combine Bayesian inference with planning under uncertainty to evaluate the effect of ToM-based communication on joint performance in an illustrative scenario. Lim *et al.* (2020) design a Bayesian ToM-inspired (Baker et al., 2017) agent model and investigate the performance of humans with agents with and without a ToM in a collaborative setting. The results of these studies around computational ToM models are generally promising and collectively suggest that the use of ToM can have positive impacts on human-agent collaboration.

Realizing such a ToM model for effective human-agent collaboration is useful but difficult. We argue that for a computational ToM model to gain widespread adoption as a versatile tool applicable across various settings, it needs to adhere to the following three important criteria:

Formal: Imbuing ToM with logical rigor helps us navigate the intricacies of human interaction. At the same time, formal reasoning can provide the logical foundation necessary for a ToM-using agent to formally interpret and anticipate the actions of others. By adhering to formal logical principles, such an agent should be able to make explainable inferences, enhancing its overall reliability in social reasoning tasks.

Human-inspired: Incorporating human-inspired social decision-making heuristics (e.g., trust (Castelfranchi & Falcone, 2010)) is essential for bridging the gap between computational ToM and human cognition. Being able to reason with these concepts enriches a ToM-using agent's understanding of social dynamics and enables it to interpret and respond to human behavior in a more nuanced manner, improving the quality of human-machine interactions.

Effective: In complex social settings characterized by continuous interaction between humans and machines, a ToM-using agent will accumulate a diverse array of beliefs about others over time, where some of these will only be applicable in certain situations, and others will be useful in other situations. To continue its effectiveness in engaging with human partners over time, the agent should also be effective in storing and maintaining (i.e., creating and updating) these beliefs, as well as using them for a variety of interactive purposes.

The contribution of our work in this paper is a novel computational ToM model that is designed to meet these three criteria. Specifically, we propose a computational ToM mechanism based on abstracting agents' beliefs and knowledge into higher-level, abstract concepts, namely, *abstractions*. These abstractions, similar to those that guide human interactions, can correspond to and be triggered by various social roles, norms, human values as well as emotions among individuals. Collectively, they serve as human-inspired, practical approximations for the computational agent to make effective decisions when interacting with humans. To provide the foundation necessary to formally describe how to create, update, and employ abstractions in the context of human-agent collaboration, we use epistemic logic (Meyer & Van Der Hoek, 2004). We computationally model several human decision-making heuristics and show how ToM reasoning can be efficiently used within our abstraction procedure. We subsequently indicate the importance of social roles and norms with respect to the interaction context and illustrate how these can be integrated naturally into our framework. Integrating roles and norms helps the agent to choose among different actions to yield a result that would fit the current situation better.

The rest of this paper is organized as follows. Section 2 discusses abstractions. Section 3 sets up our working example, featuring a human-agent collaboration scenario. Section 4 describes our computational ToM mechanism and shows how epistemic logic can be used for abstracting beliefs and knowledge of agents. Section 5 explains the dynamics of abstractions, such as their creation and updates. Section 6 shows how the abstractions with ToM reasoning can be used to enable effective interactions between humans and computational agents. Section 7 discusses our work, addresses related research on computational ToM in the literature, and finally points to future research directions.

2. Understanding Abstractions in Hybrid Intelligence

We propose abstractions as enablers for agents to reason effectively in hybrid settings by capturing essential characteristics of beliefs and knowledge. Rather than focusing on individual pieces of belief and knowledge, the agent can reason about the abstractions, which can enhance the efficiency of human-agent collaboration. Here, we examine the concept of abstractions, their importance in collaborative settings, and their application in hybrid intelligence scenarios where agents and humans work together to achieve shared goals.

2.1 The Concept and Importance of Abstractions

In general terms, *abstracting* is the process of reducing complex or concrete concepts, ideas, or objects to their essential characteristics, in order to simplify them and make them more manageable (Falguera et al., 2022). It involves filtering out details and focusing on the most important aspects or features of a subject which are relevant for a particular purpose. Thinking in abstractions is viewed as a important characteristic of modern human behavior (McBrearty & Brooks, 2000), and the development of this trait is believed to be closely linked to the advancement of human language, as both the spoken and written forms of language seem to require and enable abstract thinking. Since this cognitive process enables us to represent a vast collection of information in a summary with a few words or sentences, whether to succinctly synthesize a general theory about a topic or to convey a message in a methodically efficient manner, we argue that a software agent, which is designed to work with humans in a skillful and efficient manner, can also benefit from computationally capturing humans' abstraction mechanisms when interacting with them.

In the context of human-agent collaboration, we are particularly interested in computationally capturing how humans use their abstraction ability in complex social situations to effectively interact with others. For example, consider the abstraction *trust*, which serves as a backbone in collaboration and captures one's confidence in others' abilities, reliability, and commitment (Mattessich & Monsey, 1992). During collaboration, a human can use the abstraction technique and reason about the relevant information about her partner (e.g., "able?", "reliable?", "committed?", etc.) in order to decide whether to trust the partner or not. This serves as a reliable shortcut in making decisions, capturing essential aspects of a situation while discarding irrelevant noise. We envision a computational agent that can mimic such an ability to simplify its beliefs and knowledge about its partner into abstractions that can serve for behavioral heuristics to use in its decision-making processes, just like humans do (Tversky & Kahneman, 1974).

An interesting extension to capturing abstractions is to embed them in ToM reasoning. Thus, the agent would not only capture how much it trusts the human it is interacting with but also how much the human trusts it back. Capturing trust in the human has the pre-mentioned benefit of easing interactions (e.g., the agent can leave certain tasks to a human whom it trusts). Thanks to its ToM, the agent can monitor the human to better understand their decisions. For example, the agent can model that the human does not trust it enough and thus does not delegate certain tasks. This would inspire the agent to perform additional actions to engender trust on the human side.

As we perceive them, abstractions can further offer computational agents flexibility in their interactions by helping them adapt to changing circumstances. In dynamic environments, abstractions enable individuals to make swift adjustments to their behavior and decisions based on updated information. For instance, in human-agent collaboration, an agent equipped with abstractions can adjust its strategies in real-time to accommodate changes in its partner's behavior or the task at hand. That being said, abstractions may not always be suitable, particularly in situations where circumstances rarely change or where precision is paramount. In static environments with stable conditions in which agents do not need to change their decisions much, maintaining abstractions can impose an additional computational burden. Furthermore, relying too heavily on abstractions may lead to oversimplification and overlooking nuanced details. Similarly, in rapidly changing environments where accuracy is critical, abstractions may fail to capture in a timely manner how situations evolve, leading to sub-optimal decisions.

2.2 Application of Abstractions in Hybrid Settings

Abstractions can manifest (computationally) in various ways. Again, we look at trust as an example. Extensive literature explores how a computational agent can learn whether to trust another agent using machine learning methods (Teacy et al., 2006; Granatyr et al., 2015). Employing machine learning techniques often requires agents to have numerous interactions with others in order to properly learn how to trust others. However, in real-world scenarios, one often must make trust decisions with a limited number of interactions. That being said, humans benefit from other contextually relevant information in order to make quick decisions, such as social cues and organizational constructs. For example, humans might trust someone because they are a doctor in a reputable hospital, even without prior interactions. It is also crucial to understand the underlying reasons for trust (Castelfranchi & Falcone, 2010), which is difficult to accomplish with machine learning techniques. Given our goal of facilitating computational agents to effectively create, update, and reason about abstractions, we opt for formalizing abstractions through predefined rules of formal logic rather than data-driven methods. However, whenever data necessary for data-driven methods are available, then the abstractions can also so be created through them. The formal framework that we propose in Section 4 is generic and can accommodate abstractions that are derived through different methods.

A specific area of related work involves use of POMDPs and interactive POMDPs (Gmytrasiewicz & Doshi, 2005; Rathnasabapathy et al., 2006; Baker et al., 2011, 2017). Markov decision processes offer a formal mathematical framework for modeling decision-making processes in dynamic environments, allowing for abstract representation of belief states and explicitly capturing uncertainty and state transitions (Dearden & Boutilier, 1997; Abel et al., 2016; Congeduti & Oliehoek, 2022). This approach allows for precise probabilistic reasoning and optimal decision making under uncertainty and has been shown to be useful for computational ToM modeling (albeit in simpler settings (Rathnasabapathy et al., 2006; Baker et al., 2017)). However, POMDPs can be computationally demanding and may require extensive data and computational resources to implement effectively in complex social settings that feature continuous interaction (e.g., human-agent collaboration). In contrast, the way that we define abstractions focuses on simplifying complex information into intuitive concepts, emphasizing human-like reasoning and decision making. While our approach may lack the precision of Markov decision processes, it offers a more interpretable and human-centric approach, which can be advantageous in contexts where transparency and ease of understanding are prioritized.

Our interpretation of abstractions needs more precision. In the remainder of this text, we will refer to the abstract behavioral heuristics of humans, such as trust, which we aim to formally capture for use by computational agents in their interactions, as *abstractions* – the “what” aspect of our framework. Furthermore, we will refer to the formal methods by which agents create and update these abstractions as *abstraction rules* – the “how” aspect of our framework. To clarify, our intention of introducing abstraction rules *is not only* to group abstraction-related information and aggregate them into a label that passes as an abstraction definition. Instead, we want to show how the reasoning processes behind creating and updating abstractions can be clarified via use of epistemic logic. Moreover, these rules are more than mere deduction/induction tools. When considered together with an agent’s (current) set of beliefs and knowledge, abstraction rules can act as an action-interpreting device to support why agents perform certain actions depending on their (current) set of abstractions. We will formalize these notions in Section 4.

3. Collaboration Essentials of Hybrid Intelligence

In order to give a sense of the types of interactions required for hybrid intelligence, we now present a working example. Following this example, we will highlight the challenges related to the maintenance and application of abstractions in such collaborative settings.

3.1 Working Example: Human-Agent Collaboration in Medicine

Our working example is inspired from a medical diagnostic process (National Academies of Sciences, and Engineering, and Medicine and others, 2015), where different collaborators share the workload according to their strengths during the diagnostic process. Without loss of generality, we consider a computational agent doctor A and a human doctor D that work together towards the diagnosis of a patient C’s health problem. In this setting, the core objective of A is to use its capabilities to complement those of D. For example, D can perform the patient interview and the physical examination processes, while A can work on the diagnostic testing (e.g., analyzing MRI scan results (Hazlett et al., 2017)).

Although Artificial Intelligence (AI) research in healthcare continues to progress (Loh, 2018; Briganti & Le Moine, 2020), the usual paradigm suggests that AI agents as well as robots and software applications are treated as decision support systems that doctors can use (Sutton et al., 2020). Doctors have the final say in the medical procedure and can neglect the information that such agents may provide altogether. However, within our working example, we give equal stance to both agents and humans in the diagnostic process; thus, we have both an “agent doctor” and a “human doctor” (Coeckelbergh, 2010; De Graaf et al., 2021). Essentially, our example provides a collective decision-making process in which A and D can share their findings with each other, assess each other’s work, and agree on the diagnosis together in an interactive manner. Although we do not explicitly discuss this point, the interaction can include that the human doctor explains her decision to the agent doctor. This will in itself also be a good check for the human doctor on the correctness of that decision.

Now, suppose that a difference of opinion has arisen between A and D during their discussion for the diagnosis of C’s health problem. For instance, D may say that the clinical interview R_1 and the physical examination results R_2 (provided by D) together point to a specific disease S_1 but A may say that it can be another disease S_2 according to the diagnostic testing results R_3 (provided by A). D may further add that they should discount the diagnostic testing results R_3 because the disease is nearly always S_1 when similar physical examination results are observed. In this case, A can simply check whether it should insist on its own diagnosis decision and elaborate on its findings or simply accept D’s decision, say, because of time constraints.

Compared to a simple medical decision support tool that is designed to register and retrieve patient data and diagnostic testing results, one can see that A can utilize its beliefs and knowledge interactively in different ways. In our scenario, the set of possible actions that A can do includes (but is not limited to) *doing interactive reasoning* to check whether a diagnostic result is of good quality, *warning* D about poor quality results, *advising* D to put more emphasis on one result rather than another, *consulting* another doctor E, *telling* D its beliefs and knowledge, and *asking* D's opinion on a subject that is relevant to C's health problem. Figure 1 outlines the interaction that takes place among the agent doctor A, the human doctor D, and the patient C during the diagnostic process. In the remainder of this paper, we refer to A as "it", D as "her", and C as "him" for practical purposes.

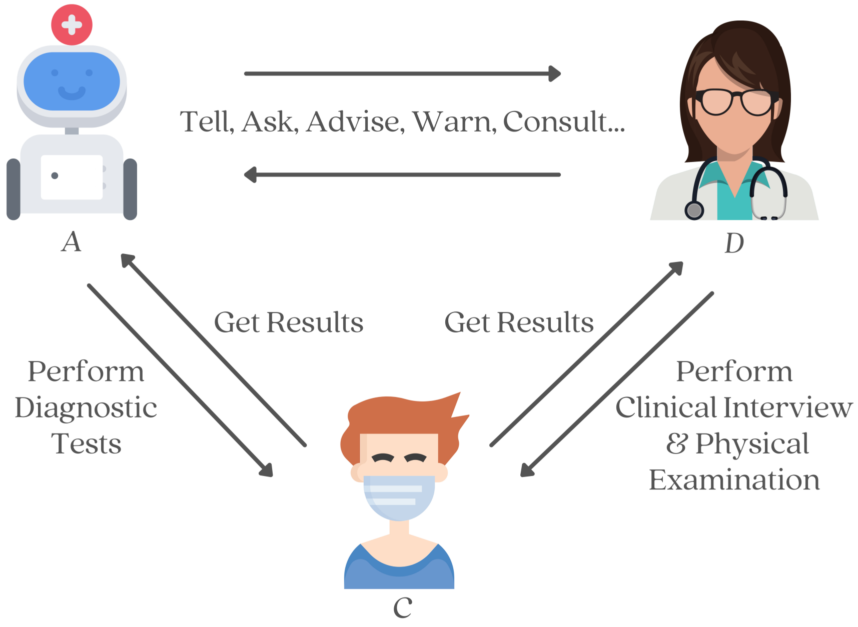
Tell, Ask, Advise, Warn, Consult...
A
D
Get Results
Get Results
Perform Diagnostic Tests
Perform Clinical Interview & Physical Examination
C

Figure 1: Hybrid Collaboration in Medicine: A computational agent doctor A and a human doctor D are working together towards the diagnosis of a patient C's health problem. Each doctor has different set of capabilities that would be useful for the diagnosis.

3.2 Challenges Regarding Maintenance and Usage of Abstractions

To fully harness the potential of our agent A for hybrid intelligence, it is imperative that A's capabilities are not limited to these actions alone. Specifically, we argue that making A capable of tracking mental contents of both D and C and interacting via computational ToM reasoning (when necessary) can bring added value to human-agent collaboration. Below, we address two challenges regarding the effective maintenance and usage of abstractions within the context of our disagreement resolution scenario and provide six examples for clarification.

Effective Maintenance of Abstraction-based ToM: A needs to use its ToM of D and C to decide on the actions to perform when interacting with them. Over time, the number of its beliefs and knowledge about both *D* and *C* can increase dramatically. Even then, *A* should be capable of making its decisions efficiently. What would be a practical way for *A* to do this?

Example 1 Some actions of *A* can be dependent on its trust in *D*. For instance, when a disagreement happens between *A* and *D*, *A* can decide that they should consult another doctor if it does not trust *D*. Thus, *A* needs to decide whether it should trust *D* or not. How can *A* estimate its trust in *D* in an efficient manner?

One way of addressing the situation in Example 1 could be to collect a lot of data to create an accurate trust model for *D* as is customary in computational trust literature (Granatyr et al., 2015). However, many times, collaborations might emerge on the spot without a long history; hence it would not be possible to have a large amount of historical interaction data. Hence, it would be required to derive abstractions from a small amount of data.

Example 2 Other actions of *A* can be dependent on its perception of *D*'s trust in itself. For example, when another disagreement happens between *A* and *D*, *A* can decide that they should converse with each other in a collaborative manner (instead of seeking an additional opinion) if it believes that *D* trusts *A*. How can *A* practically check whether *D* trusts *A* or not?

Generally, *A* needs to be flexible in its trust modeling: Others' reasons for trusting *A* can be different than its own reasons for trusting others. In this situation, *A* needs to employ computational ToM reasoning to take *D*'s perspective first and accordingly assess *D*'s trust in itself. To do this, it can benefit from available context-relevant information that *D* deems important for trust (e.g., *D* believes that *A* has good medical capabilities) in the form of knowledge and beliefs (e.g., *A* believes that *D* believes that *A* has good medical capabilities).

Example 3 As interactions between *A* and *D* progress, *D*'s initial trust in *A* to change or even disappear over time. Thus, *A* should be able to update its belief about *D*'s trust in *A* to stay consistent with the actual situation. Furthermore, *A*'s trust in *D* may also change due to other reasons. How can *A* capture these changes effectively?

Addressing the situation in Example 3 requires the agent to have an update mechanism for its abstractions. For efficiency concerns, this update mechanism should not run after every change in the environment but should quickly handle the major updates (e.g., *D* starts to avoid collaborating with *A*). Hence, it might be acceptable to miss slight changes in trust evaluation because the updates are not frequent, but major changes should be reflected in a timely manner.

Effective Usage of Abstraction-based ToM: Even though *A* works with humans collaboratively, it does not need to reason exactly like humans; but when it needs to interpret why humans perform certain actions that involve ToM reasoning, it can benefit from the decision-making heuristics from which humans also benefit.

Example 4 Suppose *A* decides that *D* is reluctant to trust *A*. After checking possible reasons, *A* infers that this is due to *D*'s lack of knowledge about *A*'s medical capabilities. How can *A* use this information to perform actions to positively change *D*'s trust?

Simply observing the environment and the others passively is not enough for computational agents to interact effectively with humans. The agents need to decide which actions to perform to make a desired change in others' ToM. For Example 4, one interesting approach for *A* to positively influence *D*'s trust is through proactive communication and demonstration of its medical expertise.

Example 5 Suppose *A* observes that the patient *C* does not seem to trust *D*, which can have negative effects in the diagnostic process. *A* further observes that *D* is not aware of this, exhibiting an inconsistency in their respective beliefs about the situation. How can *A* identify and act in order to resolve such inconsistencies effectively?

To address this, *A* first needs to explicitly point out the contradicting beliefs (i.e., its own and *D*'s beliefs about *C*'s trust), as well as the reasons behind the inconsistency. Then, *A* can engage in direct communication with *D* to make the necessary warnings about the diagnostic process. Using this information, *A* can further advise *D* to perform other actions that can help them resolve the diagnostic disagreement (e.g., consulting another doctor).

Example 6 Before deciding how to interact with *D* during a conflict resolution moment, *A* may need to consider multiple pieces of information that it has about the situation (e.g., trust between *A* and *D* and trust between *D* and *C*). On their own, such pieces of information may indicate conflicting courses of action for *A* to follow (e.g., "*A* trusts *D*, so agree with *A*'s diagnosis" vs. "*C* does not trust *D* (which can negatively affect the diagnostic process), so consult another doctor's opinion"). How can *A* resolve such inconsistencies effectively?

Selecting the appropriate course of action poses a challenge for *A* when it needs to consider multiple abstractions before making a decision. Depending on the context, *A* may need to adopt different strategies. For instance, if one abstraction holds greater significance than another, the action aligned with the former may supersede that of the latter.

4. Computational Theory of Mind with Abstractions (ToMA)

We propose TOMA to accommodate the collaboration requirements for hybrid intelligence. TOMA has at its core a formal model of Theory of Mind, where abstractions play a key role. We explain this formal core and its usage next.

4.1 Formal Design of TOMA

Representing the above conceptualization requires taking into account two important aspects. First, the agent's abstractions about the human and its perception of what the human thinks of the agent can vary in time. For example, the agent might trust the human now but the trust might decline over time. Similarly, the agent might perceive that the human trust it, only to find out later that this is not the case. Hence, it is necessary to be able to capture the fluid nature of the state of the abstractions. Second, since in many settings multiple agents and humans are present, it is necessary to be able to differentiate individual beliefs and knowledge from each other and enable each agent to do its own reasoning based on its own information. To accommodate these requirements, we base our formalization on epistemic logic (Meyer & Van Der Hoek, 2004), a subfield of epistemology concerned with logical approaches to knowledge, belief, and related notions.

Agents: We define an *agent* as an entity that can hold beliefs and knowledge about other agents, maintain its beliefs and knowledge over time, and use its beliefs and knowledge when interacting with other agents. We denote the finite set of agents as $\mathscr{X}$ where X,Y are agents in $\mathscr{X}$.

Knowledge and Beliefs: Our agents can form *beliefs* through interactions with other agents, observations, or other means. These beliefs do not need to be true. In contrast, agents only have true *knowledge*. Moreover, if an agent knows something, then the agent also believes it. This distinction provides flexibility for our agents, allowing them to make decisions based on both uncertain beliefs and certain knowledge.

To formally represent *knowledge and beliefs* of a set of agents $\mathscr{X}$, we use the modal operators K_X and B_X for all $X \in \mathscr{X}$ and the following language $\mathscr{L}_{KB}^{\mathscr{X}}$ given by the *Backus-Naur* form:

$$\varphi := p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi \mid K_X \varphi \mid B_X \varphi$$

Here, p are propositional atoms that represent atoms in a fragment of first-order logic and $X \in \mathscr{X}$. For example, given $p_1 = Doctor(Y)$ and $X,Y \in \mathscr{X}$, $K_X p_1$ and $B_X p_1$ are read as "X knows that Y is a doctor" and "X believes that Y is a doctor", respectively. Notice that $B_Y K_X p_1$, which is read as "Y believes that X knows that Y is a doctor", is also a member of $\mathscr{L}_{KB}^{\mathscr{X}}$. Such formulas with nested epistemic operators allow us to represent agents' higher-order beliefs and knowledge succinctly.

The semantics of our language are defined in terms of possible worlds. Given the set of agents $\mathscr{X}$, let $M = (W, \bigcup\limits_{X \in \mathscr{X}} R_{K_X}, \bigcup\limits_{X \in \mathscr{X}} R_{B_X}, \pi)$ be a Kripke structure where:

- W is a non-empty set of possible worlds,
- $R_{K_X} \subseteq W \times W$ is binary relation on W representing knowledge of agent X, such that $R_{K_X}(w, w')$ means that world w' is accessible from world w according to X's knowledge,
- $R_{B_X} \subseteq W \times W$ is binary relation on W representing beliefs of agent X, such that $R_{B_X}(w, w')$ means that world w' is accessible from world w according to X's beliefs, and
- π is valuation function that assigns truth values to propositional atoms in each world.

For the knowledge and belief operators, we use the standard modal systems $S5_n$ and $KD45_n$ for n agents, respectively (Meyer & Van Der Hoek, 2004; Dignum et al., 2001), where n equals to the number of agents in $\mathscr{X}$. Formulas are evaluated with respect to pairs (M, w) of a model M and a world $w \in W$, using binary relations R_{K_X} and R_{B_X} corresponding to each agent X's knowledge and beliefs. The relations R_{K_X} and R_{B_X} are serial, transitive, and euclidean. The relations R_{K_X} are also reflexive (i.e., knowledge is always true) yet the relations R_{B_X} may not be (i.e., a belief may not be true).

Abstractions: An *abstract concept* is a human-inspired, abstract decision-making heuristic such as *trust* or *respect*, which can guide agents in their interaction decisions. We formally define a finite set of abstract concepts as $\mathscr{A} = \{Abs_1, Abs_2, ..., Abs_m\}$ such that Abs_i denotes an abstract concept where $1 \leq i \leq m$. For example, $\mathscr{A} = \{Trust, Respect\}$ denotes the set of abstract concepts *Trust* and *Respect*. Essentially, these abstract concepts are meaningful only when defined in a relational manner. Thus, we formally define an *abstraction* as an atom of first-order logic structured as $Abs(X,Y)$, where $Abs \in \mathscr{A}$ and $X,Y \in \mathscr{X}$. For example, $Trust(X,Y)$ can be read as "X trusts Y".

Our formalization also allows agents to hold (higher-order) knowledge and beliefs about abstractions. For instance, given $p_2 = \mathit{Trust}(X,Y)$, $K_Y p_2$ and $B_X B_Y p_2$ can be read as "Y knows that X trusts Y" and "X believes that Y believes that X trusts Y", respectively. We refer to such (higher-order) knowledge and beliefs about abstractions simply as abstractions.

Abstraction Rules: An *abstraction rule* is a derivation rule in the form of $\varphi \to \psi$ such that ψ is an abstraction. For instance, both $p_1 \to p_2$ (i.e., "Y is a doctor" implies that "X trusts Y") and $K_X p_1 \to B_X K_Y p_2$ (i.e., "X knows that Y is a doctor" implies that "X believes that Y knows that X trusts Y") are abstraction rules. Note that the φ can pertain to multiple pieces of knowledge and beliefs as well as other abstractions – the abstraction rules are not only used for creating abstractions but also for updating them (e.g., from "trust" to "no trust").

Epistemic logic enables us to formally exploit epistemic principles. In our framework, we will use the following prominent epistemic principles P_K, P_B, and P_I to create and update abstractions:

P_K: $K_X(\varphi \to \psi) \to (K_X\varphi \to K_X\psi)$ (i.e., knowledge is closed under implication)

P_B: $B_X(\varphi \to \psi) \to (B_X\varphi \to B_X\psi)$ (i.e., belief is closed under implication)

P_I: $K_X\varphi \to B_X\varphi$ (i.e., knowledge implies belief)

These epistemic principles are useful for deriving new beliefs and knowledge – and especially abstractions – from already existing ones. For example, if we have that $K_X p_1$ and $K_X(p_1 \to p_2)$, by using P_K and modus ponens, we can derive that $K_X p_1 \to K_X p_2$ and hence, $K_X p_2$. Similarly, if we have that $K_X(p_1)$ and $B_X(p_1 \to p_2)$, we can first use P_I to get $B_X(p_1)$, and then use P_B and modus ponens to derive that $B_X p_1 \to B_X p_2$ and hence, $B_X p_2$. Here, one may notice the utility of using P_I, as it allows an agent to derive beliefs from knowledge. Note that it is useful to be able to reason with both knowledge and belief in an interaction context. For instance, it may be preferable for an agent to *know* the conditions under which it should *not trust* another agent, while merely *believing* when to *trust* that agent may be sufficient.

4.2 ToMA in Use

Actions and Action Decision Rules: An agent uses its knowledge and beliefs for two purposes. The first one is creating and updating abstractions through abstraction rules. The second purpose is deciding which *actions* to perform next according to the *action decision rules* that the agent has. In our framework, the action decision rules are based on conditional reasoning, where actions depend on the agent's specific knowledge or beliefs, which may correspond to abstractions as well.

While we acknowledge that actions and action decision rules could be introduced into the formal language φ as explicit operators, we choose not to add them directly within the language. In line with our contribution ToMA, a computational ToM mechanism based on abstractions, we want to have a clear separation between epistemic reasoning and procedural decision-making and focus on knowledge, beliefs, and abstractions without complicating the framework with action dynamics.

To illustrate how agents decide which actions to perform based on their knowledge, beliefs, and abstractions, we use the following notation for action decision rules: $\varphi \to \mathbf{U}$ where $\varphi \in \mathscr{L}_{KB}^{\mathscr{X}}$ and $\mathbf{U}$ is an action. For example, $K_X p_1 \to \mathbf{Accept}(S_1)$ can be read as "'X knows that Y is a doctor' implies that X performs the action '**Accept** diagnosis S_1'". To clearly distinguish them, actions are represented in **bold** and for specificity reasons, they are combined with elements from $\mathscr{L}_{KB}^{\mathscr{X}}$.

In the following sections, representation of an agent's abstractions and action decisions will be done via derivation tables which consists of three parts (separated by two horizontal lines): Action decision rules (1^{st} part), beliefs and knowledge (2^{nd} part), and derived abstractions and action decisions (3^{rd} part). Figure 2 explains the different parts of a derivation table. Figure 2 also illustrates how we actualize abstraction by using epistemic logic. By using $K_X(p \rightarrow q)$ and P_K, the agent can logically derive $K_X p \rightarrow K_X q$ via modus ponens. By using $K_X p$ and $K_X p \rightarrow K_X q$, the agent can further derive $K_X q$, which is presented in the third part of the derivation table. One can see that the derivation of $B_X s$ is also made in a similar manner as well as the action decisions. All of the abstraction examples given in Sections 5 and 6 feature this type of epistemological flow.

$$\begin{array}{cll}
 & K_X q \rightarrow \mathbf{U} & \text{Agent's action decision rules are shown in the first part (until the first line).} \\
 & B_X s \rightarrow \mathbf{V} & \\ \cline{2-2}
 & K_X p & \text{Agent's beliefs and knowledge are in the second part (until the second line).} \\
 & K_X(p \rightarrow q) & \text{They are used to create abstractions with the help of } P_K, P_B\text{, and modus ponens.} \\
 & B_X r & \\
 & B_X(r \rightarrow s) & \\ \cline{2-2}
\therefore & K_X q & \text{Abstractions come first in the third part.} \\
 & B_X s & \\
 & \mathbf{U} & \text{Action decisions (in \textbf{bold}) come second.} \\
 & \mathbf{V} &
\end{array}$$

Figure 2: Visual representation of an agent's abstraction and action decision mechanisms.

Instantiated atom	Meaning
$Doctor(A)$	A is a doctor.
$Doctor(D)$	D is a doctor.
$Doctor(E)$	E is a doctor.
$Patient(C)$	C is a patient.
$Diagnosis(S_1)$	S_1 is a diagnosis.
$Diagnosis(S_2)$	S_2 is a diagnosis.
$MedicalCollaboration(A,D)$	A and D collaborate in a medical setting.
$GoodCommunication(A,D)$	A communicates well with D.
$Capabilities(A)$	A has medical capabilities.
$Result(D,C,R_1)$	R_1 is a result that D gets after examining C.
$Lie(C,D)$	C lies to D.
$Disagreement(A,S_2,D,S_1)$	A's diagnosis S_2 conflicts with D's diagnosis S_1.
$Arguable(A,S_2,D,S_1)$	A's diagnosis S_2 can be argued against D's diagnosis S_1.
$Trust(A,D)$	A trusts D.
$Trust(D,A)$	D trusts A.
$Trust(C,D)$	C trusts D.
$TrustHigh(A,D)$	A has a high level of trust in D.

Table 1: List of instantiated atoms of first-order logic that are used in the examples.

For our working example that features a total of three agents (human and computational), we specifically use $\mathscr{L}_{KB}^{\mathcal{X}}$ where the set of agents consists of A, D, and C. Table 1 provides a list of things that agents may know or believe, represented in the form of instantiated atoms of first-order logic together with their meanings. Table 2 provides the list of actions that A can perform.

Action	Meaning
$\mathbf{Demonstrate}(D, Capabilities(A))$	Demonstrate A's medical capabilities to D.
$\mathbf{Warn}(D, \neg Trust(C,D))$	Warn D about the lack of C's trust towards D.
$\mathbf{Consult}(A,D,E)$	Advise a consultation meeting with A, D, and E.
$\mathbf{Accept}(S_1)$	Accept diagnosis S_1.
$\mathbf{Argue}(S_2,D,S_1)$	Argue diagnosis S_2 against D's diagnosis S_1.

Table 2: List of actions that are used in the examples along with their meanings.

5. Dynamics of ToMA

An important part of ToMA is the management of abstractions that one has of others as well as the believed abstractions others have. Managing these abstractions requires creating and revising individual abstractions as well as addressing different perspectives.

5.1 Abstractions and Abstraction Rules

Creating abstractions and formulating abstraction rules are not always easy. If the agent has access to a large data set to learn from, one possibility could be to incorporate machine learning techniques to learn existing abstractions as well as to learn rules that apply in different situations. However, such a dataset is generally not available in many settings. Humans, on the other hand, derive these abstractions from a few interactions and accept that they might not be accurate and can be updated as the interactions progress. We follow the same reasoning here to show how an agent can derive abstractions from roles and abstraction rules from norms of the society.

Roles serve as socially expected sets of behaviors based on an individual's status or position within society (Solomon et al., 1985). Humans, as well as agents, can have multiple social roles in the groups to which they belong (Dastani et al., 2003). Understanding these roles requires a sophisticated ToM to discern how role-governance dynamics, concerning attitudes such as beliefs, goals, emotions, etc., correspond to specific roles. *Norms*, which encompass commonly accepted standards of social behavior, represent integral components of these dynamics. Ranging from basic customs like politeness to complex rules governing attire and conduct, norms foster social order and cohesion by establishing behavioral expectations and guiding interactions. Using ToM reasoning in tandem with social norms and roles, humans can correctly interact with others in a very practical manner. For instance, an individual equipped with a functional ToM can anticipate and adhere to cultural norms, such as removing shoes before entering someone's home, thereby demonstrating respect for cultural practices and avoiding potential offense. Within this context, the roles are "host" and "guest" and the norm is to be aware of these roles as well as the context in order to show respect. In the following examples, we will show how roles can facilitate the creation and updating of abstractions and how norms can contribute to the formulation of rules that govern these abstractions within social contexts.

5.2 Creating Abstractions

We have stated in Example 1 that capturing and interpreting trust computationally can be helpful in case of a disagreement between the agent doctor A and the human doctor D. To capture trust in a practical manner, A can directly benefit from D's role. Specifically, A can begin with a state of trust towards D, just because D is a *doctor* and they are working together in a *medical setting*. If, for example, they were in a completely different setting (e.g., competitors in an auction), A would not need to trust D due to her profession. Table 3 illustrates this short-cutting abstraction approach. Note that the abstraction rule "$B_A(Doctor(D) \wedge MedicalCollaboration(A,D) \rightarrow Trust(A,D))$" is expressed as a belief rather than knowledge, reflecting the inherent uncertainty that A must account for in this situation.

$$\begin{array}{rl}
\hline
 & K_A(Doctor(D)) \\
 & K_A(MedicalCollaboration(A,D)) \\
 & B_A(Doctor(D) \wedge MedicalCollaboration(A,D) \rightarrow Trust(A,D)) \\
\hline
\therefore & B_A(Doctor(D)) \\
 & B_A(MedicalCollaboration(A,D)) \\
 & B_A(Trust(A,D))
\end{array}$$

Table 3: An abstraction can be directly induced by a role: Capturing A's trust towards D.

By using its contextually relevant knowledge and the principles P_B and P_I, A makes the abstraction of trust. In Table 3, $K_A(Doctor(D))$ and $K_A(MedicalCollaboration(A,D))$ correspond to the knowledge that A uses to create the beliefs $B_A(Doctor(D))$ and $B_A(MedicalCollaboration(A,D))$ first (via P_I and modus ponens) and then, the abstraction $B_A(Trust(A,D))$ in the form of a derived belief (via P_B and modus ponens). Let us revisit Example 1 where A needs to decide whether it should trust D or not. In case of a disagreement with D, A can now decide not to consult another doctor thanks to its trust in D. Instead, A can simply agree with D's diagnostic decisions as long as it continues to trust D.

5.3 Capturing Others' Abstractions

For the computational agent, in addition to creating abstractions about others, it is also important to create abstraction on how others see the agent. That is, the agent would monitor what others (including humans) think of it. One way of doing this is to assume that others would have the same abstraction rules to infer the same abstractions. This would mean that the agent assumes others would trust it in the same way the agent would trust them. However, it is possible that different participants model this abstraction differently; hence, one agent develops trust in a different way than another.

Recall Example 2 in which A wants to check whether D trusts A. Here, D does not need to trust A in the same way that A trusts D. Taking a more critical approach, D can demand more from A to comfortably trust A's decisions when working together. For instance, D may further want to see whether A is designed to *communicate well* with doctors and can demonstrate the benefits of its *medical capabilities* that are valuable to D. Table 4 shows how computational ToM reasoning can help A to model D's trust towards it in such a setting. The formulas $B_AB_D(Doctor(A))$,

$B_A B_D(GoodCommunication(A,D))$, and $B_A B_D(Capabilities(A))$ represent the beliefs that are used to produce the abstraction $B_A(Trust(D,A))$. Notice that instead of K_A, we use B_A in Table 4 to represent the uncertainty that A may have in this case: A may not be completely sure about D's beliefs about A. Regardless, thanks to being capable of having higher-order beliefs about others' beliefs and how it may impact their behaviour (both essential for effective ToM reasoning), A can utilize this basis as a checkpoint for capturing D's trust. Furthermore, A can build on this abstraction to decide its action in case of a disagreement with D (e.g., conversing with D collaboratively for a mutually agreed diagnosis instead of agreeing with D's diagnosis without a conversation).

$$
\begin{array}{ll}
\cline{2-2}
 & B_A B_D(Doctor(A)) \\
 & B_A B_D(GoodCommunication(A,D)) \\
 & B_A B_D(Capabilities(A)) \\
 & B_A(B_D(Doctor(A)) \wedge B_D(GoodCommunication(A,D)) \wedge B_D(Capabilities(A)) \rightarrow Trust(D,A)) \\
\cline{2-2}
\therefore & B_A(Trust(D,A))
\end{array}
$$

Table 4: The role of a role in abstraction can differ between agents: Capturing D's trust towards A.

5.4 Updating Abstractions

Updating abstractions like trust in response to evolving trust dynamics between humans and computational agents is crucial for maintaining alignment in collaborative interactions. Let us re-visit Example 3 where A needs to reassess the trust dynamics between itself and D. Suppose A has observed D making critical mistakes during their collaborations. Suppose also that A has observed that D has recently started to interact with A in such a way that A formed the belief that D does not believe that A has good capabilities anymore (e.g., D wants to do diagnoses on her own or interrogates A's every decision/action). To act correctly in its interactions with D in the future, it is crucial for A to update its trust in D and D's perceived trust in A depending on these changes. Table 5 below describes how A can handle this situation: A updates its abstractions according to the changes that happen in its beliefs and knowledge about D. Notice that in the table, $K_A(\neg Trust(A,D))$ and $B_A(\neg Trust(D,A))$ are created via P_K and P_B, respectively. It is important to explicitly state what kind of beliefs and knowledge should be taken into account when revising abstractions. This way, A does not need to consider changes that are neither relevant nor significant when updating its abstractions.

$$
\begin{array}{ll}
\cline{2-2}
 & K_A(CriticalMistakes(D)) \\
 & B_A B_D(\neg Capabilities(A)) \\
 & K_A(CriticalMistakes(D) \rightarrow \neg Trust(A,D)) \\
 & B_A(B_D(\neg Capabilities(A)) \rightarrow \neg Trust(D,A)) \\
\cline{2-2}
\therefore & K_A(\neg Trust(A,D)) \\
 & B_A(\neg Trust(D,A))
\end{array}
$$

Table 5: Revising abstractions: Detecting changes in the trust dynamics between A and D.

While we employ modus ponens and the epistemic principles P_K, P_B, and P_I for basic logical derivations within our framework, updating abstractions involves more than these methods. We acknowledge that this challenge is also related to the well-studied problem of belief revision (Gärdenfors, 2003). One potential approach for managing conflicting abstractions could use principles from AGM belief revision theory (Alchourrón et al., 1985). For instance, A might revise its abstractions (e.g., changing $B_A(\mathit{Trust}(D,A))$ to $B_A(\neg \mathit{Trust}(D,A)))$ by considering which individual beliefs to retain and which to adjust (e.g., removing the old belief $B_A B_D(\mathit{Capabilities}(A))$ from its set of beliefs and keeping the new one $B_A B_D(\neg \mathit{Capabilities}(A))$ instead), ensuring consistency with the new information. Such an approach enables systematic updates of abstractions while maintaining the flexibility to adapt to the specific needs of the interaction context. As detailed mechanisms for managing conflicting abstractions are beyond the scope of this work, we do not explore them further here.

6. TOMA for Hybrid Intelligence Interactions

As already mentioned in Section 3, hybrid intelligence requires effective interactions between agents and humans. We explain more what these interactions are and demonstrate how TOMA handles them in several examples.

6.1 Realizing Missing Abstractions Proactively

Trust plays an important role for cooperative behavior, and agents may take proactive measures to foster trust within human-agent collaborations. Recall Example 4 in which A decides that D is reluctant to trust A. In this situation, A should first look for the reasons why D is reluctant to trust it. Suppose that after ruling out the other potential reasons, A learns that this is due to D's lack of knowledge about A's medical capabilities. Building on this reasoning, A can then proactively *demonstrate its capabilities* in an attempt to change D's stance towards it. Table 6 below describes the situation and A's corresponding action: A demonstrates its capabilities if it finds out that D's lack of knowledge is the root cause of her lack of trust in A.

$$\begin{array}{rl}
 & B_A(\neg \mathit{Trust}(D,A)) \wedge B_A \neg K_D(\mathit{Capabilities}(A)) \rightarrow \mathbf{Demonstrate}(D, \mathit{Capabilities}(A)) \\
\hline
 & B_A(\neg \mathit{Trust}(D,A)) \\
 & B_A(\neg \mathit{Trust}(D,A) \rightarrow \neg K_D(\mathit{Capabilities}(A))) \\
\hline
\therefore & B_A \neg K_D(\mathit{Capabilities}(A)) \\
 & \mathbf{Demonstrate}(D, \mathit{Capabilities}(A))
\end{array}$$

Table 6: Realizing abstractions: A proactively taking action.

The agent uses P_B and modus ponens to make the inference $B_A \neg K_D(\mathit{Capabilities}(A))$ and then acts accordingly. On the other hand, if A observed that D indeed trusted it, then the inference would not be triggered and A would not need to demonstrate its capabilities to D.

6.2 Monitoring Inconsistencies

Normally, a doctor-patient relationship is expected to be built on trust, communication, and a common understanding of both sides' needs (National Institutes of Health (U.S.), 2016). Founding a good relationship is deemed important since it can affect the quality of the patient interviewing process and hence, the determination of the diagnosis (National Institutes of Health (U.S.), 2016). In our collaborative diagnosis scenario, D needs C to share all relevant information, whereas C trusts D to keep this information to herself and not disclose it to others. Although adhering to these medico-social norms is expected from both parties, they may choose to not follow them.

Consider Example 5 where A observes that C does not seem to trust D, which can have negative effects in the diagnostic process. Now, suppose that C chooses to keep some sensitive information about himself and *lie* about his health conditions out of mistrust, shame, or other personal reasons and that A observes this. With this information, A can infer that C does not trust D. This is an important piece of information for A to capture since the lack of trust could have a negative effect on the accurate determination of D's diagnosis. Now suppose also that A further learns that D still thinks that C trusts her (i.e., "A believes that D believes that C trusts D."). Building on this inconsistency in their beliefs, A can then warn D about the inconsistency during their discussion and advise her to use the interview information cautiously. A can further support its argument by providing accompanying reasons (e.g., shortness of the duration of the interview, lack of detailed questions/answers, etc.) and suggest putting more emphasis on the diagnostic testing results R_3 rather than the interview R_1 and seeking consultation with another doctor E. In Table 7, we give an example reasoning mechanism and an accompanying action scheme that A can use in this case. A uses its beliefs about D and C and the principle P_B to infer C's lack of trust in D. Since A is capable of doing ToM reasoning, it further checks if there is an inconsistency between its own belief and D's belief about C's trust. Correspondingly, A can warn D about C's lack of trust in D and the patient interview R_1 and advises D to also consult doctor E.

$$
\begin{array}{ll}
 & B_A B_D(\mathit{Trust}(C,D)) \land B_A(\neg \mathit{Trust}(C,D)) \rightarrow \mathbf{Warn}(D, \neg \mathit{Trust}(C,D)) \\
 & B_A B_D(\mathit{Trust}(C,D)) \land B_A(\neg \mathit{Trust}(C,D)) \land K_A(\mathit{Result}(D,C,R_1)) \rightarrow \mathbf{Warn}(D,R_1) \\
 & B_A B_D(\mathit{Trust}(C,D)) \land B_A(\neg \mathit{Trust}(C,D)) \land K_A(\mathit{Doctor}(E)) \rightarrow \mathbf{Consult}(A,D,E) \\
\cline{2-2}
 & B_A(\mathit{Lie}(C,D)) \\
 & B_A(\mathit{Lie}(C,D) \rightarrow \neg \mathit{Trust}(C,D)) \\
 & B_A B_D(\mathit{Trust}(C,D)) \\
 & K_A(\mathit{Result}(D,C,R_1)) \\
 & K_A(\mathit{Doctor}(E)) \\
\cline{2-2}
\therefore & B_A(\neg \mathit{Trust}(C,D)) \\
 & \mathbf{Warn}(D, \neg \mathit{Trust}(C,D)) \\
 & \mathbf{Warn}(D,R_1) \\
 & \mathbf{Consult}(A,D,E)
\end{array}
$$

Table 7: Monitoring inconsistencies: A's actions after inferring lack of trust in doctor-patient relationship.

6.3 Managing Multiple Abstractions

Consider Example 6 where A needs to take multiple abstractions into account before deciding how to engage with the doctor about the case. On one hand, the medico-social norm can dictate that the agent should argue for a more detailed resolution of the disagreement, especially if the patient-doctor relationship is *known* to be problematic (i.e., the patient does not trust the doctor). On the other hand, the agent can be designed in such a way that having a *high* level of trust in D (for instance, due to her experience and specialization in the medical area of concern) can override the norm to not take an argumentative approach in certain contexts. Then, A may need to simply accept D's diagnosis S_1 over its own diagnosis S_2. Below, Table 8 illustrates A's decision-making process in this case: By following the norm (i.e., "argue when there is a diagnostic disagreement"), A finds out that its own diagnosis can be argued against that of D's; however, its high trust in D restrains it from actually arguing with D and instead, it accepts D's diagnosis.

$$\begin{array}{ll}
 & K_A(\mathit{Arguable}(A,S_2,D,S_1)) \land K_A(\neg\mathit{Trust}(C,D)) \land K_A(\mathit{TrustHigh}(A,D)) \rightarrow \mathbf{Accept}(S_1) \\
 & K_A(\mathit{Arguable}(A,S_2,D,S_1)) \land K_A(\neg\mathit{Trust}(C,D)) \land \neg K_A(\mathit{TrustHigh}(A,D)) \rightarrow \mathbf{Argue}(S_2,D,S_1) \\ \hline
 & K_A(\mathit{TrustHigh}(A,D)) \\
 & K_A(\neg\mathit{Trust}(D,C)) \\
 & K_A(\mathit{Disagreement}(A,S_2,D,S_1)) \\
 & K_A(\mathit{Disagreement}(A,S_2,D,S_1) \rightarrow \mathit{Arguable}(A,S_2,D,S_1)) \\ \hline
\therefore & K_A(\mathit{Arguable}(A,S_2,D,S_1)) \\
 & \mathbf{Accept}(S_1)
\end{array}$$

Table 8: Managing multiple abstractions: Role-induced abstraction overrides norm.

The reasoning process in Table 8 is one way to deal with clashing abstractions, namely, one overriding another. There are other alternatives. For instance, A can ask D for more information before deciding on the action to perform next. Such information may be about the diagnosis itself and/or include D's own beliefs and knowledge about the case (e.g., her stance towards the diagnosis with more details, her confidence in her decision, etc.). With a little more reasoning, A can then make a more informed decision about its next actions regarding the diagnostic disagreement. Alternatively, when managing multiple abstractions, A can also choose to perform a (slightly) different action rather than "accepting" and "arguing". For example, A can still accept D's decision (due to high level of trust) but also provide the information on why it chooses to do so along with the abstract beliefs and knowledge that are relevant to the context and used in its reasoning process (e.g., "although the norm dictates that I should argue with you in case of a disagreement, I decided to accept your diagnosis instead due to my trust in you and your experience in the field").

7. Conclusion

Computationally modeling ToM ability with the abstraction heuristics that we defined in Section 4 is a first step towards our long-term goal of designing social agents that are capable of collaborating efficiently with human partners. With examples from the medical domain, we illustrated how abstracting beliefs and knowledge into higher-level concepts can be useful for an agent doctor in dealing with disagreements that can happen when doing collective decision-making with a human doctor towards the diagnosis of a patient's health problem. By explicitly taking into account the interaction context that the agent is in, we emphasized how social dynamics shaped by roles and norms can play important parts in such hybrid settings. Furthermore, we sketched several ways with the help of epistemic logic to demonstrate how the agent doctor can employ these social dynamics computationally to create various abstractions and use them to resolve disagreements efficiently, suggesting the power and versatility of the proposed abstraction framework.

7.1 Discussion of TOMA

We further explore how TOMA meets the three criteria outlined in Section 1. TOMA is a **formal** framework, as its elements, including abstractions and their dynamics, have been formalized in epistemic logic. This enables the model to be enacted formally. The way abstractions and their dynamics work, such as inducing an abstraction from a social role or an abstraction rule from a social norm, mimic how humans derive such heuristics. An agent that implements TOMA will be able to formally execute operations on abstractions and has the potential to interact with humans seamlessly as it accounts for **human-inspired** notions. As mentioned in Section 3, the effectiveness of such a framework comes in two parts. The first part is to effectively maintain the information it contains (e.g., update only when needed). Examples 1–3 demonstrate the dynamics expected from such frameworks. Section 5 demonstrates how TOMA handles these cases. The second part to effectiveness is in handling interactions with humans as expected in a hybrid intelligence setting, as demonstrated by Examples 4–6. Section 6 shows how TOMA addresses these interactions; thus, yielding an **effective** framework.

In general, computational or human, agents use their (contextually relevant) beliefs and knowledge to make decisions about the actions to take when interacting with others. We argue that the computational agents that benefit from TOMA can effectively simplify the same beliefs and knowledge (via abstraction rules) into more compact information in the form of abstractions that they can use for the same purpose. It is important to recognize that this capability may not always be useful for agents, as its utility depends heavily on the interaction setting. For instance, in scenarios where decisions are infrequent, abstractions may introduce unnecessary computational overhead. On the other hand, if the setting features continuous interaction, which often demands agents' participation, then abstractions will prove valuable to such agents.

The environment also plays a role in determining the utility of abstractions. For example, in simpler cases (e.g., games like Rock-Paper-Scissors as used widely in the literature) where a computational agent needs to maintain only a few of beliefs about others, abstractions may not be needed at all. We provide abstraction rules mainly for creating abstractions from individual beliefs and knowledge, so the overall setting should be complex enough to necessitate abstracting. Furthermore, the frequency and magnitude of changes in the environment are also important. In a highly static environment, an agent can create abstractions at the beginning and may not need to use abstraction rules ever again. Since our formalization enables an agent to employ abstraction rules *also* for updating its abstractions when its beliefs and knowledge change, a more dynamic environment provides more potential for our design to thrive. That being said, if the environment is *excessively* dynamic and requires the agent to take many actions, abstractions and abstraction rules may become cumbersome for the agent.

The research on computational ToM models suggests that agents can benefit in various ways from ToM reasoning, especially at higher orders. De Weerd *et al.*'s research (2013) shows that agents can benefit from first-order and second-order ToM reasoning in competitive game-theoretic situations, although with diminishing returns beyond third-order ToM. They also explore how higher-order ToM reasoning can aid agents in a strictly cooperative game (de Weerd et al., 2015), demonstrating that agents with beyond zero-order ToM can establish communication more quickly. De

Weerd *et al.* (2017) investigate the extent to which agents can benefit from higher-order ToM reasoning in a mixed-motive scenario called Colored Trails, finding that second-order ToM provides considerable advantages, while first-order ToM has limited effectiveness. Kröhling and Martínez (2019) examine the role of ToM in single-issue negotiations between context-aware agents, where the negotiation context is modeled by two variables: necessity and risk. Görür *et al.* (2017) propose a ToM-based agent model for estimating human intentions in a shared human-robot task while Brooks and Szafir (2019) demonstrate how robots can create second-order ToM models by observing human actions in spatial settings. Montes *et al.* (2023) introduce an agent model that integrates ToM reasoning with abductive reasoning capabilities, testing it within the framework of an incomplete-information, cooperative card game called Hanabi (Bard et al., 2020).

Although these research results are generally promising and demonstrate that the use of ToM leads to better outcomes for the studied tasks, the existing models have not been widely adopted as a computational tool in many real-life settings. This is mainly because the computational ToM models are not meant to deal with all of the three criteria that we address in Section 1 (i.e., formal, human-inspired, and effective). Most of these formal models are tailored for a specific, restricted setting in which the agents' action space and the information that they can use for ToM reasoning are limited (e.g., moves in a spatial setting, tactics in a simple game, and so on). This creates a crucial drawback for a ToM-based agent to succeed in real-life scenarios that demands more complex ways of (social) interaction because it is not possible for such an agent to use high-level decision-making heuristics, like abstractions, to properly realize the full potential of ToM reasoning and hence, accomplish their tasks more effectively. In this paper, we introduce a medical domain setting, which is rich in information and reflective of real-life situations (e.g., conflict resolution) to illustrate how a computational agent capable of doing ToM reasoning in a variety of ways can thrive in collaboration, particularly when leveraging abstractions.

When considering the potential implementation of our abstraction-based computational ToM design, it is crucial to carefully consider the characteristics of deductive and inductive reasoning that would guide agents in managing abstractions. In the context of creating abstractions, deductive reasoning may be utilized to derive general conclusions from given premises. The "role-induced trust" example formalized in Section 5 illustrates this point. Inductive reasoning, on the other hand, involves inferring general principles or patterns from specific observations. In the context of updating abstractions, inductive reasoning may be used to generalize from new beliefs and knowledge to revise existing abstractions (e.g., when "trust" transitions into "no trust", as illustrated in the same section). Of course, the scope of both of these reasoning mechanisms extends beyond these specific considerations. A comprehensive understanding of their roles in managing abstractions is crucial for a robust implementation, which should result in a more systematic formalization of the abstraction dynamics and the effective integration of both types of reasoning mechanisms into a unified system.

Epistemic reasoning tools like eclingo (Cabalar et al., 2020), EP-ASP (Son et al., 2017), and other epistemic extensions of Answer Set Programming (Brewka et al., 2011), can be useful for handling basic epistemic reasoning in our framework. These tools can represent agents' knowledge and beliefs and evaluate action decision rules accordingly. However, such systems may face limitations in managing the dynamic aspects of our framework, particularly the updating of complex abstractions such as trust, which may evolve over time. Additionally, handling continuous, real-time updates during agent interactions can be difficult for current epistemic reasoning tools, as they often require manual adjustments. Machine learning models offer potential here, as they provide flexibility for real-time adaptation, by either creating new beliefs or updating existing ones with new information in a timely manner, thus complementing logic-based reasoning approaches. Consequently, a hybrid approach that combines the strengths of both methods may provide an effective solution.

7.2 Future Research Directions

It is important to keep in mind that the hypothetical agent-human collaboration example that we have worked with throughout the paper is primarily meant to *demonstrate the potential functionality* of our formal, abstraction-based, computational ToM design. The applicability of our approach is based on the assumption that healthcare professionals will be open to embracing a computational agent as their collaborator. Given the increasing demand for more efficient clinical decision-support systems in which the design arises from a collaborative and multidisciplinary perspective (Sarkar & Samal, 2020), our model's potential application in the future seems promising. However, its success will depend on its ability to address other issues that medical professionals may face when closely and continuously working together with computational agents (e.g., alert fatigue, physician burnout, etc.).

Note that the abstractions that we work with in this body of work can emerge through different means. Argumentation schemes (Walton et al., 2008), which are stereotypical patterns of reasoning with a corresponding set of critical questions, represent one of these sophisticated ways that can be used to reason about abstractions such as trust (Parsons et al., 2012). Employing argumentation schemes can also be advantageous for the agent doctor in assessing a patient's trust level by analyzing responses to critical questions associated with these schemes. Such an approach can offer additional support for the agent's subsequent actions.

In addition to social norms and roles, there are other human concepts that hold relevance for our abstraction-based computational ToM design. Within the social sciences, *values* denote a person's set of preferences that determine appropriate courses of action in their lives. These values exert a considerable influence on social behavior (Bardi & Schwartz, 2003) and can serve as guiding principles in our lives (Schwartz et al., 2012), shaping our choices and influencing how we interact with the world around us. While human values can vary across cultures, religions, and individuals, certain core values such as honesty, kindness, fairness, and responsibility are universally recognized and prized. Similar to roles and norms, these values can also be seen as valuable abstract concepts that could be incorporated into our framework. Although not elaborated on in this paper, we propose that they could offer additional guidance to ToM-using computational agents in interpreting human behavioral patterns.

Since 2022, large language models (LLMs) such as chatGPT and GPT4 have received a lot of attention. One research question is to which extent LLMs can perform standardized ToM tasks that are usually used to test children, such as first-order ("Where will Sally look for her marble") and second-order ("Where does Anne think that Sally will look for her marble?") false belief tasks. Initial results appeared to be positive, with LLMs passing the standard first-order false belief tasks as they appear in the psychological literature, as well as slight reformulations (Kosinski, 2023). However, it soon appeared that the LLMs had a relatively low accuracy when they were not tested on the standard tests but on new variations of first-order false belief tasks; the LLMs did even worse when solving second-order false belief tasks (Ullman, 2023; van Duijn et al., 2023). Performance of LLMs on large benchmark suites of ToM tasks was shown to be mixed at best (Sap et al., 2022; Shapira et al., 2023).

Recently, there have been interesting reflections that propose to view the question "Do LLMs have ToM?" with more nuance than as a "yes—no" question, namely, as a continuum (van Dijk et al., 2023). For example, they propose to look at an LLM's intermediate reasoning steps and to check whether sophisticated prompting may improve the LLM's accuracy on ToM tasks. Another line of work accepts the fact that LLMs are not yet capable of fully autonomous ToM in a range of applications, and propose to combine an existing ToM system with LLMs' impressive grasp of language use, for example, Bayesian Inverse Planning Accelerated by Language Models (BIP-ALM) (Jin et al., 2024). This appears to point to an appealing way forward for our proposal as well, namely, to combine the symbolic ToM reasoning and abstraction capabilities of TOMA with the subsymbolic language capabilities of a LLM in order to create a system that can reason about human mental states and communicate well with human users.

As a follow-up work, we aim for a more complete abstraction model that captures the ways humans abstract their beliefs and knowledge. Since we intend to build an interactive reasoning system which should be well-versed in the ways of social cognition, we plan to benefit from various methods and tools in logic, artificial intelligence, and cognitive sciences (e.g., ontologies, argumentation schemes, belief-desire-intention (BDI) models (Rao & Georgeff, 1998), etc.). Another research direction can be to further investigate the role of human-agent communication in recursive ToM reasoning. For that purpose, "mind perception theory" (Gray et al., 2007; Lee et al., 2021) can be beneficial when designing higher-order ToM-using agents that can accurately infer how their own artificial minds are perceived and modeled by humans. With a more comprehensive computational ToM model, which is also equipped with mind abstraction abilities, we will further test our agents in human-agent settings in order to evaluate their collaborative skills in dynamic environments.

Acknowledgments

This paper significantly extends "Abstracting Minds: Computational Theory of Mind for Human-Agent Collaboration" that appeared in the First Hybrid Human-Agent Intelligence Conference. This research was funded by the Hybrid Intelligence Center, a 10-year programme funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research, https://hybrid-intelligence-centre.nl, grant number 024.004.022. We thank Davide Dell'Anna for useful discussions.

References

Abel, D., Hershkowitz, D., & Littman, M. (2016). Near optimal behavior via approximate state abstraction. In *International Conference on Machine Learning*, pp. 2915–2923. PMLR.

Akata, Z., Balliet, D., De Rijke, M., Dignum, F., Dignum, V., Eiben, G., Fokkens, A., Grossi, D., Hindriks, K., Hoos, H., et al. (2020). A research agenda for hybrid intelligence: augmenting human intellect with collaborative, adaptive, responsible, and explainable artificial intelligence. *Computer*, *53*(08), 18–28.

Alchourrón, C. E., Gärdenfors, P., & Makinson, D. (1985). On the logic of theory change: Partial meet contraction and revision functions. *The journal of symbolic logic*, *50*(2), 510–530.

Anderson, J. R. (2009). *How can the human mind occur in the physical universe?* Oxford University Press.

Baker, C. L., Jara-Ettinger, J., Saxe, R., & Tenenbaum, J. B. (2017). Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nature Human Behaviour*, *1*(4), 1–10.

Baker, C. L., Saxe, R. R., & Tenenbaum, J. B. (2011). Bayesian theory of mind: Modeling joint belief-desire attribution. *Proceedings of the Thirty-Third Annual Conference of the Cognitive Science Society*, *33*(33).

Bamicha, V., & Drigas, A. (2022). The evolutionary course of theory of mind-factors that facilitate or inhibit its operation & the role of icts. *Technium Soc. Sci. J.*, *30*, 138.

Bard, N., Foerster, J. N., Chandar, S., Burch, N., Lanctot, M., Song, H. F., Parisotto, E., Dumoulin, V., Moitra, S., Hughes, E., et al. (2020). The hanabi challenge: A new frontier for ai research. *Artificial Intelligence*, *280*, 103216.

Bardi, A., & Schwartz, S. H. (2003). Values and behavior: Strength and structure of relations. *Personality and social psychology bulletin*, *29*(10), 1207–1220.

Brewka, G., Eiter, T., & Truszczyński, M. (2011). Answer set programming at a glance. *Communications of the ACM*, *54*(12), 92–103.

Briganti, G., & Le Moine, O. (2020). Artificial intelligence in medicine: today and tomorrow. *Frontiers in medicine*, *7*, 27.

Brooks, C., & Szafir, D. (2019). Building second-order mental models for human-robot interaction. *CoRR*, *abs/1909.06508*.

Buehler, M. C., & Weisswange, T. H. (2020). Theory of mind based communication for human agent cooperation. In *2020 IEEE International Conference on Human-Machine Systems (ICHMS)*, pp. 1–6. IEEE.

Cabalar, P., Fandinno, J., Garea, J., Romero, J., & Schaub, T. (2020). eclingo: A solver for epistemic logic programs. *Theory and Practice of Logic Programming*, *20*(6), 834–847.

Carruthers, P., & Smith, P. K. (1996). *Theories of theories of mind*. Cambridge University Press.

Castelfranchi, C., & Falcone, R. (2010). *Trust theory: A socio-cognitive and computational model*. John Wiley & Sons.

Coeckelbergh, M. (2010). Robot rights? towards a social-relational justification of moral consideration. *Ethics and information technology*, *12*, 209–221.

Congeduti, E., & Oliehoek, F. A. (2022). A cross-field review of state abstraction for markov decision processes. In *34th Benelux Conference on Artificial Intelligence (BNAIC) and the 30th Belgian Dutch Conference on Machine Learning (Benelearn)*.

Dastani, M., Dignum, V., & Dignum, F. (2003). Role-assignment in open agent societies. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pp. 489–496.

De Graaf, M. M., Hindriks, F. A., & Hindriks, K. V. (2021). Who wants to grant robots rights?. In *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 38–46.

de Weerd, H., Verbrugge, R., & Verheij, B. (2013). How much does it help to know what she knows you know? An agent-based simulation study. *Artificial Intelligence*, *199-200*, 67–92.

de Weerd, H., Verbrugge, R., & Verheij, B. (2014a). Agent-based models for higher-order theory of mind. In *Advances in Social Simulation, Proceedings of the 9th Conference of the European Social Simulation Association*, Vol. 229, pp. 213–224.

de Weerd, H., Verbrugge, R., & Verheij, B. (2014b). Theory of mind in the Mod game: An agent-based model of strategic reasoning. In *European Conference on Social Intelligence*, pp. 128–136. Springer.

de Weerd, H., Verbrugge, R., & Verheij, B. (2015). Higher-order theory of mind in the tacit communication game. *Biologically Inspired Cognitive Architectures*, *11*, 10–21.

de Weerd, H., Verbrugge, R., & Verheij, B. (2017). Negotiating with other minds: The role of recursive theory of mind in negotiation with incomplete information. *Autonomous Agents and Multi-Agent Systems*, *31*(2), 250–287.

Dearden, R., & Boutilier, C. (1997). Abstraction and approximate decision-theoretic planning. *Artificial Intelligence*, *89*(1-2), 219–283.

Devin, S., & Alami, R. (2016). An implemented theory of mind to improve human-robot shared plans execution. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 319–326. IEEE.

Dignum, F., Dunin-Kęplicz, B., & Verbrugge, R. (2001). Agent theory for team formation by dialogue. In *Intelligent Agents VII Agent Theories Architectures and Languages: 7th International Workshop, ATAL 2000 Boston, MA, USA, July 7–9, 2000 Proceedings 7*, pp. 150–166. Springer.

Falguera, J. L., Martínez-Vidal, C., & Rosen, G. (2022). Abstract Objects. In Zalta, E. N. (Ed.), *The Stanford Encyclopedia of Philosophy* (Summer 2022 edition). Metaphysics Research Lab, Stanford University.

Gärdenfors, P. (2003). *Belief revision*. No. 29. Cambridge University Press.

Gargeya, R., & Leng, T. (2017). Automated identification of diabetic retinopathy using deep learning. *Ophthalmology*, *124*(7), 962–969.

Gmytrasiewicz, P. J., & Doshi, P. (2005). A framework for sequential planning in multi-agent settings. *Journal of Artificial Intelligence Research*, *24*, 49–79.

Görür, O. C., Rosman, B. S., Hoffman, G., & Albayrak, S. (2017). Toward integrating theory of mind into adaptive decision-making of social robots to understand human intention. In *International Conference on Human-Robot Interaction*, Workshop on the Role of Intentions in Human-Robot Interaction.

Granatyr, J., Botelho, V., Lessing, O. R., Scalabrin, E. E., Barthès, J.-P., & Enembreck, F. (2015). Trust and reputation models for multiagent systems. *ACM Computing Surveys (CSUR)*, *48*(2), 1–42.

Gray, H. M., Gray, K., & Wegner, D. M. (2007). Dimensions of mind perception. *Science*, *315*(5812), 619–619.

Hazlett, H. C., Gu, H., Munsell, B. C., Kim, S. H., Styner, M., Wolff, J. J., Elison, J. T., Swanson, M. R., Zhu, H., Botteron, K. N., et al. (2017). Early brain development in infants at high risk for autism spectrum disorder. *Nature*, *542*(7641), 348–351.

Hiatt, L. M., Harrison, A. M., & Trafton, J. G. (2011). Accommodating human variability in human-robot teams through theory of mind. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

High, R. (2012). The era of cognitive systems: An inside look at ibm watson and how it works. *IBM Corporation, Redbooks*, *1*, 16.

Hindriks, K. V., Jonker, C., & Tykhonov, D. (2008). Towards an open negotiation architecture for heterogeneous agents. In *International Workshop on Cooperative Information Agents*, pp. 264–279. Springer.

Ho, M. K., Saxe, R., & Cushman, F. (2022). Planning with theory of mind. *Trends in Cognitive Sciences*, *26*(11), 959–971.

Jin, C., Wu, Y., Cao, J., Xiang, J., Kuo, Y.-L., Hu, Z., Ullman, T., Torralba, A., Tenenbaum, J. B., & Shu, T. (2024). MMToM-QA: Multimodal theory of mind question answering..

Kosinski, M. (2023). Theory of mind may have spontaneously emerged in large language models. *arXiv preprint arXiv:2302.02083*, *4*, 169.

Kröhling, D., & Martínez, E. (2019). On integrating theory of mind in context-aware negotiation agents. In *XX Simposio Argentino de Inteligencia Artificial (ASAI 2019)-JAIIO 48 (Salta)*, pp. 180–193.

Lee, M., Lucas, G., & Gratch, J. (2021). Comparing mind perception in strategic exchanges: Human-agent negotiation, dictator and ultimatum games. *Journal on Multimodal User Interfaces*, *15*(2), 201–214.

Lim, T. X., Tio, S., & Ong, D. C. (2020). Improving multi-agent cooperation using theory of mind..

Loh, E. (2018). Medicine and the rise of the robots: A qualitative review of recent advances of artificial intelligence in health. *BMJ Leader*, *2*(2), 59–63.

Mattessich, P. W., & Monsey, B. R. (1992). *Collaboration: what makes it work. A review of research literature on factors influencing successful collaboration.* ERIC.

McBrearty, S., & Brooks, A. S. (2000). The revolution that wasn't: a new interpretation of the origin of modern human behavior. *Journal of human evolution*, *39*(5), 453–563.

Meyer, J.-J. C., & Van Der Hoek, W. (2004). *Epistemic logic for AI and computer science*. No. 41. Cambridge University Press.

Montes, N., Luck, M., Osman, N., Rodrigues, O., & Sierra, C. (2023). Combining theory of mind and abductive reasoning in agent-oriented programming. *Autonomous Agents and Multi-Agent Systems*, *37*(2), 36.

National Academies of Sciences, and Engineering, and Medicine and others (2015). The diagnostic process. In Balogh, E. P., Miller, B. T., & Ball, J. R. (Eds.), *Improving Diagnosis in Health Care*, chap. 2, pp. 31–80. National Academies Press.

National Institutes of Health (U.S.) (2016). *Talking with Your Older Patient: A Clinician's Handbook.* NIH publication. Department of Health & Human Services, NIH, National Institute on Aging.

Osten, F. B. V. D., Kirley, M., & Miller, T. (2017). The minds of many: Opponent modeling in a stochastic game. In *IJCAI*, pp. 3845–3851. AAAI Press.

Parsons, S., Atkinson, K., Haigh, K. Z., Levitt, K. N., McBurney, P., Rowe, J., Singh, M. P., & Sklar, E. (2012). Argument schemes for reasoning about trust.. *COMMA*, *245*, 430.

Premack, D., & Woodruff, G. (1978). Does the chimpanzee have a theory of mind?. *Behavioral and Brain Sciences*, *1*(4), 515–526.

Rao, A. S., & Georgeff, M. P. (1998). Decision procedures for BDI logics. *Journal of Logic and Computation*, *8*(3), 293–343.

Rathnasabapathy, B., Doshi, P., & Gmytrasiewicz, P. (2006). Exact solutions of interactive POMDPs using behavioral equivalence. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 1025–1032.

Sap, M., LeBras, R., Fried, D., & Choi, Y. (2022). Neural theory-of-mind? On the limits of social intelligence in Large LMs. *arXiv preprint arXiv:2210.13312*.

Sarkar, U., & Samal, L. (2020). How effective are clinical decision support systems?. *British Medical Journal*, *370*.

Schwartz, S. H., Cieciuch, J., Vecchione, M., Davidov, E., Fischer, R., Beierlein, C., Ramos, A., Verkasalo, M., Lönnqvist, J.-E., Demirutku, K., et al. (2012). Refining the theory of basic individual values.. *Journal of personality and social psychology*, *103*(4), 663.

Shademan, A., Decker, R. S., Opfermann, J. D., Leonard, S., Krieger, A., & Kim, P. C. (2016). Supervised autonomous robotic soft tissue surgery. *Science translational medicine*, *8*(337), 337ra64–337ra64.

Shamekhi, A., Bickmore, T., Lestoquoy, A., & Gardiner, P. (2017). Augmenting group medical visits with conversational agents for stress management behavior change. In *International Conference on Persuasive Technology*, pp. 55–67. Springer.

Shapira, N., Levy, M., Alavi, S. H., Zhou, X., Choi, Y., Goldberg, Y., Sap, M., & Shwartz, V. (2023). Clever Hans or neural theory of mind? Stress testing social reasoning in large language models. *arXiv preprint arXiv:2305.14763*.

Solomon, M. R., Surprenant, C., Czepiel, J. A., & Gutman, E. G. (1985). A role theory perspective on dyadic interactions: the service encounter. *Journal of marketing*, *49*(1), 99–111.

Son, T. C., Le, T., Kahl, P. T., & Leclerc, A. P. (2017). On computing world views of epistemic logic programs.. In *IJCAI*, pp. 1269–1275.

Sutton, R. T., Pincock, D., Baumgart, D. C., Sadowski, D. C., Fedorak, R. N., & Kroeker, K. I. (2020). An overview of clinical decision support systems: benefits, risks, and strategies for success. *NPJ digital medicine*, *3*(1), 17.

Sycara, K., Norman, T. J., Giampapa, J. A., Kollingbaum, M. J., Burnett, C., Masato, D., McCallum, M., & Strub, M. H. (2010). Agent support for policy-driven collaborative mission planning. *The Computer Journal*, *53*(5), 528–540.

Teacy, W. L., Patel, J., Jennings, N. R., & Luck, M. (2006). Travos: Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multi-Agent Systems*, *12*, 183–198.

Tversky, A., & Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases. *Science*, *185*(4157), 1124–1131.

Ullman, T. (2023). Large language models fail on trivial alterations to theory-of-mind tasks. *arXiv preprint arXiv:2302.08399*.

Van der Aalst, W. M., Bichler, M., & Heinzl, A. (2018). Robotic process automation..

van Dijk, B., Kouwenhoven, T., Spruit, M. R., & van Duijn, M. J. (2023). Large language models: The need for nuance in current debates and a pragmatic perspective on understanding. In Bouamor, H., Pino, J., & Bali, K. (Eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12641–12654.

van Duijn, M. J., van Dijk, B., Kouwenhoven, T., de Valk, W., Spruit, M. R., & van der Putten, P. (2023). Theory of mind in large language models: Examining performance of 11 state-of-the-art models vs. children aged 7-10 on advanced tests. In Jiang, J., Reitter, D., & Deng, S. (Eds.), *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pp. 389–402.

Walton, D., Reed, C., & Macagno, F. (2008). *Argumentation Schemes*. Cambridge University Press.

Winfield, A. F. T. (2018). Experiments in artificial theory of mind: From safety to story-telling. *Frontiers in Robotics and AI*, *5*, 75.