Alma Mater Studiorum · Università di Bologna

Scuola di scienze · Sede di Bologna
Corso di Laurea in Informatica
Dipartimento di Informatica - Scienza e Ingegneria
- DISI -

Theory of Mind
in
Large Language Models

Bachelor Thesis in
Computer Science

Supervisor:
Prof.
Andrea Asperti

Co-supervisor:
Dr.
Emanuele La Malfa

Presented by:
Xuanqiang Huang

I session
Academic Year 2023/2024

This page is intentionally left blank.

To my brother
Chen

Contents

1 Introduction

Constructing intelligent machines has been a driving beacon of information science since its birth. In modern days, we possess machines capable of surpassing human performance in solving a wide range of specialized tasks: classical board games like chess [1] and go [2], complex real-time environments like StarCraft II [3], image recognition [4] and segmentation [5], protein folding [6], are some non-exhaustive yet remarkable examples where machine learning models were able to learn and perform these superbly.
However, the long-sought ability to generalize and adapt to diverse environments has not yet been reached [7].
In this work, we focus on a specific aspect of general intelligence: Theory of Mind (ToM). Inspired by World Models [8], we develop a system which we call Discrete World Models (DWM), which explicitly builds and maintains interpretable world states. We query a Large Language Model (LLM) to give a description of the state of the world given a partial description of a given task. We then use this enhanced problem description to get a solution. We discover this simple prompting technique is effective in improving LLMs' reasoning abilities in a broad range of ToM tasks, spanning from classical false belief, to commonsense and social reasoning.
We show this technique across 5 different datasets and report a 9.9% increase in accuracy over baseline methods, such as Chain of Thought and Tree of Thought in LLama-3-8B.
Alongside this contribution, we develop a formal framework to evaluate the hardness of a specific ToM problem. Recently, many benchmarks have been proposed [9–11], but the tasks are diverse and do not uniquely define abilities that LLMs can solve. Our framework uniquely defines a measure of complexity and shows that it correlates with the performance of CoT of the LLMs.
In summary, we propose and analyze a theoretical framework that analyzes the complexity of a ToM problem, and inspired by this work, we create a prompting method that enhances LLMs' reasoning.

Why Theory of Mind? Theory of Mind, a field born among developmental psychologists, has gained considerable traction in recent years. Humans develop this ability naturally and employ it in every situation where communication among members of the group happens, sometimes also cross-racially [12] e.g. human to dogs and vice versa. Then, imbuing this ability into machines could easily enhance human-computer-assistant interactions, as the model could *understand* the needs of the human user.
Computer scientists' surge of interest in this field has been driven by the objective of building artificial intelligence that is capable of understanding states of mind, such as beliefs, desires and intentions.

This problem has been historically tied to the problem of imbuing computers with commonsense capabilities.
Traditionally, logical systems were the typical approach [13]. However, these models frequently struggle to generalize beyond simplified toy scenarios, making them impractical for real-world applications.
Recently, thanks to the advances in the field of Natural Language Processing with GPTs [14–16], a broad area of research that actually makes this problem seem feasible has flourished.

Why Large Language Models? The abundance of data, advancements in deep learning, and rapid developments in recent years have given rise to new techniques aimed at addressing this problem.
Large language models began to display emergent capabilities [17]. These models were unexpectedly able to deal with complex problems with which they were not initially trained. Planning [18, 19], text summarization, text understanding, arithmetic abilities [17], and Theory of Mind [7, 20] are some not comprehensive examples of claimed emergent properties.
Although it is unclear how these capabilities arise, it is important to understand first the limits of these emergent properties.
Normally, in scientific inquiries, one would try to create theories and models to explain the observed behaviour. This would require starting from a few general laws and working from the ground until everything, or most of it, is explained. This approach has been quite successful in physics and allows to create systems that work *by design*, and not just with long *trial and errors*.
Even with the analysis of the behaviour of language models, this method is not employable due to the model's enormous size of parameters and the difficulty of analyzing the effects of a single input or weight change. In principle, a general learning theory that explains the current behaviour of those systems is possible, but more than the current apparatus is required. Therefore, current approaches operate from a different level of abstraction, not from first principles but from assessing the models' behaviour. In the case of large language models, one approach could be to analyze the *algorithmic behaviour* (see mechanistic interpretability) which entails analyzing output logits, or internal attention weights; the main drawback is the challenge in interpretability of these values. Another approach could be assessing the *linguistic behaviour* which analyzes the output in natural language produced by conditioning on the prompt given by the examiner.

Following this thought, I will keep this line of analysis and try to develop insights based on the observation of the model's behaviour. In this work, the performance of Large Language Models (LLM) will be assessed through prompting techniques. Most of the content of this thesis is published as a paper submitted to EMNLP 2024

in [21]. The submitted paper should be the main source of reference regarding ideas about the complexity measure and the prompting method. This thesis reproposes those ideas and broadens the view on history (see 2) and benchmarking methods discussed in section 5.
Code and data are available here.

Structure of the work. This work concerns an exploration of the Theory of Mind (ToM) as applied to language models.
In the first section, we develop psychological 2.1.1 and computational 2.1.2 backgrounds concerning the development of ToM, *Sally-Anne* tests 2.1.1, and approaches of commonsense in computer science 2.1.2.
Afterwards, I introduce a theoretical framework of analysis of ToM problems and discuss how it correlates with performance. Then, I discuss the creation and analysis of the prompting method developed for this study 3 and analyze the experimental results 4.
We conclude with discussions about different methods to evaluate ToM abilities in LLMs in 5.

2 Background

2.1 Theory of Mind

This section attempts to give an overview of the original ideas born among psychologists and philosophers (see 2.1.3) about Theory of Mind, and shows how early contributors in artificial intelligence and robotics already have considered the social aspect to be a fundamental property in intelligent systems.

2.1.1 The Roots in Cognitive Psychology

The Origins. While ideas regarding ToM abilities in humans have long been present in human culture, for example, in Intentional Systems [22], this term has been popularized by [23] in the study of Chimpanzees. In his seminal article in 1978, Premack describes the theory of mind as the ability of individuals to ascribe mental states of themselves and other individuals, such as sensory perceptions, beliefs, and desires, and the ability to use this knowledge about other individuals to predict their behaviour. Initially, this idea was explored in the study of the cognitive abilities of primates, suggesting their ability to use information to model knowledge of other group members in competitive situations [24].

Some Conceptual Frameworks for ToM. Later, the concept was exploited to study human beings' cognitive abilities in their early years of life. This path of exploration gave rise to frameworks for studying Theory of Mind.
For example, one approach is "Theory-Theory" [25], which presupposes the existence of an individual's implicit theory regarding phenomena around their environment. This initial theory is then refined over time based on the individual's observations and experiences.
Another approach presupposes the existence of innate thought modules in which ToM develops [26], justifying how this ability seems to develop similarly regardless of the culture in which the individual grows up.

Sally-Anne False Belief Tests. The tests studied by [27] are of particular interest for understanding deception in children and [28] in the analysis of autistic children. Disabilities such as autism or deafness usually delay the acquisition of theory of mind skills as defined above. In Baron-Cohen's work, a series of tests, the *Sally Anne Tests*, were created and used to probe children's abilities. These tests subsequently inspired well-known benchmarks in the literature for evaluating language models, such as [29].

The *Sally Anne* false belief test is a commonly used experimental paradigm involving two individuals. Typically, it follows a narrative structured as follows:

1. Sally and Anne are situated in a room.
2. Sally places an object, such as an apple, in a basket within the room and then exits.
3. During Sally's absence, Anne relocates the apple to a box.
4. Sally returns and seeks the apple.

The question posed to participants is: given these events, where will Sally search for the apple?
Typically, three-year-old children with typical cognitive development correctly infer that Sally will search for the apple in the basket.

2.1.2 Influences in Computer Science

Given the success of the conceptual framework derived from developing ToM in psychology, its influences have extended to external fields, influencing philosophy and neuroscience.
In this section, I will report mainly historical notes about ideas that have a similar meaning to those developed in psychology.

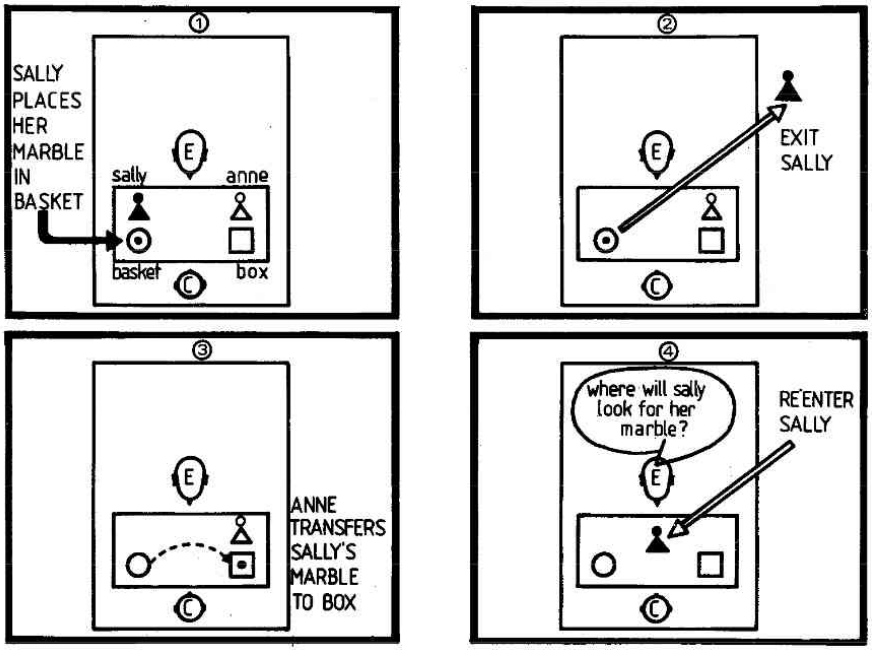

Figure 1: Sally Anne False Belief Test from (Baron-Cohen 1985). 1) Sally and Anne are both in the room, and Sally knows that the marble is in the basket. 2) Sally exits the room, leaving Anne alone. 3) Anne moves the object, unbeknownst to Sally, who is away. 4) Sally goes back into the room, and the observer needs to infer where is Sally going to look.

McCarthy's work on "Mental Qualities". John McCarthy, a founder of the field since its inception in 1956 [30], displayed an early interest in modelling common sense and understanding "Mental Qualities," akin to ToM. In [31] he writes:

> To ascribe certain beliefs, knowledge, free will, intentions, consciousness, abilities or wants to a machine or computer program is legitimate when such an ascription expresses the same information about the machine that it expresses about a person.

McCarthy defined the usefulness of ascribing mental qualities to the information extractable from this presupposition. He already knew that ToM, as defined above, is a fundamental ability that needs to be investigated and implemented in machines in order to instil general intelligence, even before the birth of the field in psychology.

Most importantly, McCarthy highlighted that supposing a program possessed mental states was:
1. Beneficial in understanding the program's behaviour when internal states are inaccessible and reason about the belief and goal structures of the program imagined by the designer.
2. Helpful in the ability to generalize despite the limited knowledge of the world, similar to Common Sense.
3. A necessary ability to obtain information and cooperate with other machines and humans.

Clearly, the main difference with the modern direction of analysis of ToM was the first point, the focus on program analysis, alongside the distancing from logic-based models.
Nonetheless, the other two points are still relevant in recent years' research.

The MIT Cog Project. As early as 2002, ToM was identified as a major component in the construction of robots that could effectively interact with and understand humans [32]. This work was part of the Cog project [33] at the Massachusetts Institute of Technology, which attempted to build and analyze embodied intelligence. Central to this endeavour was the philosophical argument that general intelligence necessitated embodiment to develop higher-order thought processes, commonly known as the "situated approach", pioneered by Rodney Brooks.

Why has ToM risen in interest today? ToM has not garnered significant interest until recent years in Computer Science.
Today, the presence of enormous amounts of raw text data, the progress in the deep learning approach to intelligence, and the fast-paced developments of recent

years have enabled a paradigm shift, allowing for an empirical analysis of these abilities [34].

This led to works that tried to assess the abilities of Language Models in typical theory of mind tasks. In February of 2023, an article by Kosinsky [20] garnered lots of attention following the claim that language models have developed ToM as an emergent property. Further research [9, 35–38] indicated that the claim was probably false, and often driven by spurious correlations the models have learned during training.
Furthermore, Shapira in [37] highlights the need for better benchmarks to help test various ToM aspects directly on LLMs. Clinical tests designed for humans do not assure their validity for machines as the subject of examination changed.

2.1.3 Some Ideas from Philosophy

One of the theories proposed by Daniel C. Dennett has been quite influential in developing ToM. In [22], he explores three conceptual frameworks to interpret and explain complex systems. These are called "stances". They are methods observers could adopt to give explanations about a given system. As we will see, they give rise to very different explanations.

Physical Stance. The explanation of systems following the physical stance develops an explanation that follows the laws of physics. For example, if the subject of our study is human behaviour, the physical stance would try to reduce it to the brain and then the physical properties of the brain, coming to biology and eventually to chemistry. In this specific case, the explanations are less valuable than a higher-level explanation because observing the physical action potential of the brain is directly transferable to explanations of behaviour.
Nonetheless, the physical stance is useful for explaining many natural phenomena, such as lightning, levitation, electromagnetic radiation, etc.

Design Stance. The design stance follows a *functionalist* point of view. It assumes that a system was designed by some intelligent agent, e.g. a human, to have a certain function or complete a certain task. Therefore, the system is usually assumed to have a clear design structure that has a goal to accomplish, often similar to what the creator desired to instil. For example, in computer science many systems are modular by design. It is often more useful to understand the behaviour of a system given a specific level of abstraction than trying to observe the electric pulses at the physical level and infer the operation from this information. An electronic circuit is a simple example: all we need to know is the logical function it was designed to compute, not how exactly the current flows in it.

Intentional Stance. This stance is most similar to ToM. It is a precursor of the idea developed in [23] and has influenced McCarthy's system in [31].
With Intentional Stance, we assume that a system has certain beliefs, intentions and desires and then try to assess how well this explains its behaviour. The programs' behaviour in artificial systems often aligns with the creator's intentions. For example, if a system is built to play Go, we can safely assume that its "desire" is to win the game, and its "intention" on every move is to gain the advantage of the other player during the game. There is no need to delve down into the physical layer or the system's design to explain the system's general behaviour.

2.2 Computer Science

Although recurrent neural networks for language modelling are regaining popularity [39,40], this thesis will primarily use language models based on Transformers [16]. Recent systems based on transformers like GPT have shown surprising abilities [15], which are difficult to understand given their pretraining procedure (which can be reduced to just next-word prediction). This has given rise to analysis of emergent properties, discussed in 2.2.1, and prompting techniques 2.2.2.
Finally, we end the section with some formal definitions and properties of Language Models in section 2.2.3.

2.2.1 Emergence

What is emergence? *Emergence* is informally defined as qualitative changes in a system's behaviour arising from quantitative changes [17, 41]. This phenomenon often arises naturally in complex systems studied in physics. The two qualitatively different layers seem to be separated by a layer of abstraction governed by a different set of rules. It is similar to the example provided by Hofstadter in [42]. He imagines a frictionless pool populated by myriads of small and slow balls that bounce off the walls, hit each other and clump together due to magnetic properties. He calls the pool "Careenium", and the ball clumps "simmballs", a joyful word-play for "cranium" and "symbols".

This system reacts to external stimuli, such as a person hitting the pool's border, which influences the movements of the balls. These balls then react to these stimuli and "encode" the information through their interactions, becoming effectively *symbols*. The interesting part of the thought experiment is the series of patterns that could arise when we zoom out and speed up the pool so that single balls are not visible anymore, but only the clumps. We now can observe symbols and general patterns that did not exist at the lower level of single balls.

Language Models, like our brain and every giant neural network, seem to be similar to this pool of balls. The movement of the activation signal at a single neuron influences each other and creates novel patterns not present at the level of the single computational entity. Yet, by observing the general behaviour of the system, one can observe new complex patterns.

Probing Emergence in Large Language Models. Large Language Models seem to possess emergent capabilities as their model size grows [17]. In this work, the authors train many different models of different sizes evaluated on different problems through few-shot prompting (*linguistic behaviour*) and observe there is a sharp increase in performance after the model uses about 10^{24} FLOPs of training. Surprisingly, as the model size grows and given enough compute time and data,

the model seems able to solve tasks that smaller models cannot, even though loss, data, and compute are kept constant.

The emergent capabilities are, probably, precisely what enabled LLMs to attain widespread acceptance rapidly. If a language model could only complete a sentence or solve a limited set of classification tasks, its scope would be limited and not be the helpful assistant that attracted the media coverage. The internet, the corpus the model was trained on, is not designed to aid helpful conversational assistants. Conceptually, if the training stage stopped to fine-tune over text completion, it would not be used as today's general problem solver [15], an advanced retrieval engine, akin to search engines [43].
These abilities are quite remarkable. It is difficult to grasp their inner working mechanism.
Further research tried to oppose the view of emergence by asserting that emergence is just In-Context Learning [44] or that it is caused by choice of the metrics [45]. However, it still displays emergence on a narrower set of tasks.

2.2.2 Prompting methods

Prompting methods can be generally viewed as cheaper methods compared to fine-tuning to achieve high performances on downstream tasks [46].
A prompting function is a function $f : V^* \to V^*$ where V is the token vocabulary. It often takes the input string and uses templates to create a final prompt that is then given to a language model.
An example from [47]:

$$f_{prompt}(X) = [X] \text{ Overall, it was a [Z] movie}$$

In this example, the prompting function maps the input text to a template with a Z still to be filled.
We expect X to be a preamble, a set of instructions followed by a task.

Other similar methods have been developed. For instance, *soft prompting*, also known as *prompt-tuning*, operates at the embedding layer, the model-dependent continuous representation of the token, by training some special, usually prefixed, token embedding specific to a task [48, 49]. A similar method, *prefix-tuning* [50], trains all prefixes across all attention blocks. It can be seen as an extension of *soft prompting*.
In this work, we will only focus on *hard prompting* methods, which do not need to access to underlying weights and are simpler to implement.

Chain of Thought. Chain of Thought attempts to allow large language models more time to reason before answering [51]. Empirically, we observe that chain of

Thought prompting produced more tokens compared to standard prompting.
Tokens have different information densities: some are more difficult to predict than others.
For example, in the sentence "I am a student that works on Large Language ____", it is easy to predict that "Models" follows, but it is not the same for the sentence "My favourite Italian meal is ____", which, for instance, is "pizza".

The informal idea that justifies why Chain of Thought works is the following way: by asking step-by-step reasoning on the task, the model has more computing, which in turn helps for more complex tasks.
This idea aligns with the intuitive notion that complex problems require more computation than easy problems. But we acknowledge this claim is just a not-tested hypothesis.

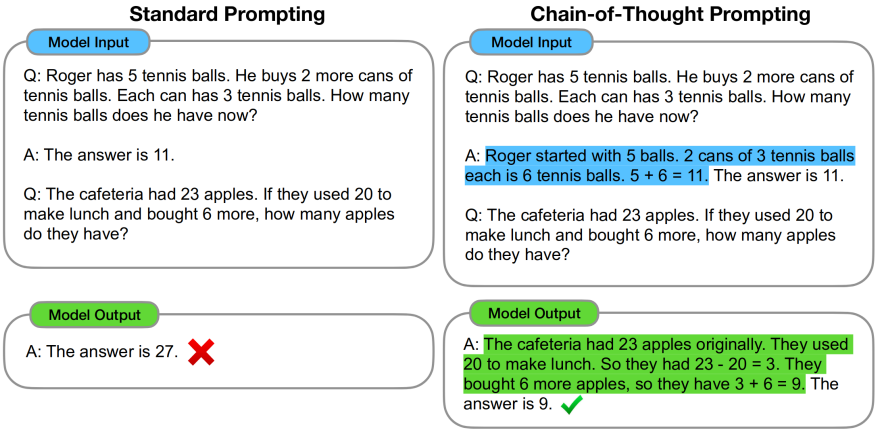

Figure 2: Example of Chain-of-Thought prompting from [51]

Later, it was discovered that adding the simple line "Let's think step by step" was enough to trigger the chain of thought behaviour [52].

Tree of Thought. Tree of Thought is a recent method of prompting proposed almost at the same time in [53] and in [54].
With this technique, the language model proposes possible lines of reasoning at each step and automatically evaluates and discards the unfavourable paths while retaining and exploring promising ones.
While Chain of Thought prompting can explore only a single reasoning path, this technique can explore more paths, using itself as a heuristic to decide which path looks favourable.

Other methods. Many prompting methods have been proposed in the last few years [55–57]. The above two are just some famous examples of prompting

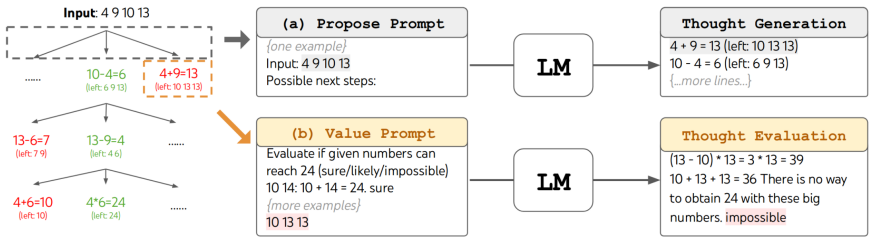

Figure 3: Example of Tree-of-Thought from [53], the model proposes "thoughts" and evaluates each thought by prompting itself to output a textual judgement, which is then converted to an ad-hoc numerical value.

techniques developed. Even though these methods provide compelling empirical evidence of superior performance over standard prompting, their theoretical underpinning are still not clearly understood [46].
The methods above can be catalogued in a relatively new field, *prompt engineering*, which tries to study how large language models behave when conditioned on different prompting styles. This resource collects many common prompting techniques and concise and short explanations of the methods.

2.2.3 Language models

Motivation. A fundamental observation in natural language processing is the following: the occurrence of a specific word in a sentence depends on the context, defined as the words that precede and succeed it [58].
This idea motivates models that define a probability distribution over a set of possible tokens Σ^* *conditioned* over the previous set of tokens, using this distribution to sample more tokens.
It makes sense to define the probability of a sentence as

$$P(w_1 w_2 \dots w_i) = P(w_1) \cdot P(w_2|w_1) \dots P(w_i|w_{1:i-1}) = \prod_{k=1}^{n} P(w_k|w_{1:k-1})$$

Following Jurafsky's development.

Bi-gram models. The model above is usually approximated using only part of the preceding sequence. In this specific case, we care only about the preceding token.

$$P(w_n \mid w_{1:n-1}) \approx P(w_n \mid w_{n-1})$$

This helps to simplify the model but requires assumptions that are often not true,

such as the Markov assumption (when a sequence of random variables depends only on the preceding one: $P(X_n \mid X_1, \ldots, X_{n-1}) = P(X_n \mid X_{n-1})$).
While not applicable in practical applications, this model can be viewed as a toy model for understanding n-gram language models.

The notion of approximating the context could be easily extended to n preceding tokens, creating the **n-gram** models in this way:

$$P(w_n \mid w_{1:n-1}) = P(w_n \mid w_{n-N-1:n-1})$$

This probabilistic model takes into account only the N preceding tokens

Language Models. Following a similar approach in [59], we say that a Language Model is a probability distribution, namely p_{LM}, over Σ^* the set of all finite-length sequence of tokens.
This is useful if we can generate sequences that have a finite generation (referred to as *locally normalized*, in the reference), so we introduce a new token EOS that symbolizes the end of the string and we define a probability of a sequence of tokens $w_1, w_2, \ldots, w_n, EOS$ as

$$p_{LM}(w_1 w_2, \ldots, w_n) = p(EOS \mid w_1 w_2 \ldots w_n) \prod_{t=1}^{T} p(w_t \mid w_{<t})$$

where p is a conditional probability distribution of a single token conditioned on the **context**.

In practice, most common language models in recent years like GPT-3 [15] are *causal language models*, meaning they learn to predict only the next token in the vocabulary Σ. We define a language model to be a function $\varphi : \Sigma^* \to X$ where Σ is the vocabulary, and X is the uniform distribution over tokens with the same

Training Large Language Models. Training language models like GPT-3 [15], usually involves 3 phases: pre-training, fine-tuning, and reinforcement learning with human feedback.

During the first phase, the model is trained on a large corpus of text, usually created through web scraping [14, 60, 61]. The model thus created is optimized for text-completion. Although remarkable, this ability is not very useful for creating helpful artificial assistants. Interacting with a computer assistant needs a conversational style, which LLMs often don't possess at this stage.
In order to solve this problem, the model is further trained on other datasets that allow conversational use of the model. This stage is sometimes called instruction-tuning because the model is fine-tuned to follow the instructions of the user.

Finally, responses are ranked according to human preferences. Different answers are proposed to the user, who chooses the answers he/she prefers. Then, through reinforcement learning algorithms like PPO [62], or more recent methods [63], the model is further tuned to produce content more likeable by a standard human.

3 Methods

In this section we first describe the proposed complexity framework and then delve into the prompting method.

3.1 A Notion of Complexity

This section mainly presents ideas and methods developed in [21].

Entity tracking concerns about correct representations of objects through time. It is considered a *prerequisite* for long-context understanding [64] and central in ToM false-belief tasks. In this work, we characterise the complexity of a ToM problem in terms of **sufficient elements to track** to output the correct result (See section 3 of [21]).
We define an environment object **obj** to be a query about some available information about the state of the environment at a precise timestamp, which could be the position of the apple as well as the k^{th}-order belief of an agent about the apple position.
We assume this object has T unique configurations during the development of the text to be represented by a prompt p expressed in natural language. We define these T unique configurations to be expressed as $E_{\mathbf{obj}} = \{e_1, .., e_T\}$
As specified in [21], to correctly solve a ToM task where p is complemented by a query about **obj**, a model should distinguish between the interactions that modify the configuration of **obj**, i.e., the **stateful** states, from those that modify any other **stateless** object $Obj \setminus \mathbf{obj}$, i.e., those one does not need to track.

We define here, following my work in [21], the cost of tracking a task's *stateful* states, which we complement with that of the *stateless*. Both definitions concur in defining the **complexity** of a ToM task.

3.1.1 Stateful and Stateless Complexity

Consider a ToM task expressed as p, describing an environment's evolution where an unknown number of atomic interactions T modify **obj** or its perception. Each environment state $e_t \in E_{\mathbf{obj}}$ can be coupled with the prompt prefix $p_{\leq t}$ s.t. $p_{\leq t} \oplus p_{>t} = p$, describing that configuration. We denote $(e_t, p_{\leq t})$ as a generic *state description*.

Definition 3.1 (State event) *A* state event *for an object* **obj** *is an event that links adjacent* state descriptions *that involve, for both the environment state e_t and the sub-prompt $p_{\leq t}$, a state change of* **obj**. *Formally, we define a relation, $F_{\mathbf{obj}}$, to specify which pairs of state descriptions form a state event:*
$F_{\mathbf{obj}}((e_t, p_{\leq t}), (e_{t+1}, p_{\leq t+1})) \equiv e_t \neq e_{t+1} \ \wedge \ p_{\leq t+1} = p_{\leq t} \oplus p_{t+1}$ *where* $1 \leq t \leq |p|$.
($|p|$ denotes the number of atomic prompts.)

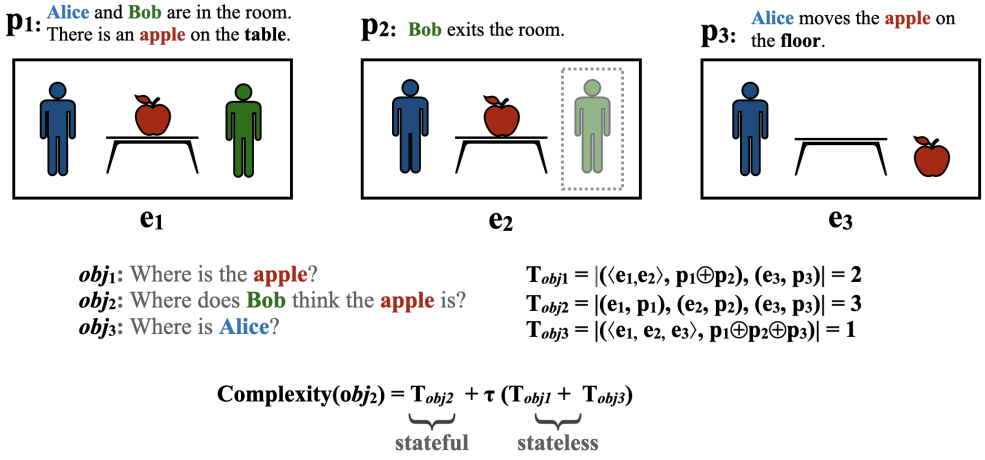

Figure 4: For obj_1, an optimal split to track the **apple** merges the first two states and chunks of the input prompt. For obj_2, which involves the 1st-order belief of **Bob**, the statefulness is higher, with e_2 that cannot be merged with e_3 as it introduces partial observability. The complexity of the task (bottom) is computed as per Eq. 3.1.2, with the complexity of stateless objects that is discounted as not directly relevant to the question/answer. Image and caption from my work in [21]

.

A *state event* $F_{\mathbf{obj}}$ identifies those *state descriptions* $(e_t, p_{\leq t})$ which have a successor $(e_{t+1}, p_{\leq t+1})$ where **obj** has changed its configuration.

In the context of ToM tasks, a *state event* could involve a person moving an object, exiting (introducing partial observability), or witnessing an environmental change. Our prompting technique, namely DWM (see 3.2), aims to make implicit observations about objects explicit.

3.1.2 The Complexity of a ToM Task

We define the **statefulness** of a ToM task specified as a prompt p as the size of the , namely $T_{\mathbf{obj}} = |E_{\mathbf{obj}}|$. Fig. 4 illustrates computing an object's statefulness or the belief associated about it.

For a ToM task where the question to solve relates to an object **obj**, one must ensure that changes to any other object, namely $Obj \setminus \mathbf{obj}$, do not affect **obj**. Tracking the evolution of irrelevant objects is unnecessary, but a computational model must assess if a particular environmental change affects **obj**.
We thus introduce the notion of **statelessness**, i.e., the cost of discerning whether a change in the environment affects **obj**.

The complexity of a ToM task regarding an object **obj** is the complexity of the

stateful states plus the (discounted) sum of the stateless states, mathematically formalized as:

$$T_{\mathbf{obj}} + \tau \sum_{obj \in Obj \setminus \mathbf{obj}} T_{obj}$$

The process of computing the complexity of a ToM task is illustrated in Figure 4.

3.2 An elicited discrete world model

Language models are empirically able to rephrase sentences while retaining the semantic information, a similar ability to summarization [65].
The main idea of this prompting method is to guide the model in describing how the environment evolves sentence by sentence and using this additional information to answer the final question. We claim this prompting method allows the language model to build a *discrete* world model about the task he is trying to solve. Having world models allows us to predict the next states of the current task the agent is engaged with.
When splitting the problem into multiple sub-parts, the language model is explicitly asked to make a description of each state. This is similar to explicitly describing a hidden state of the world or environment. Then, with multiple iterations, the state is updated by the model itself. In this way, the model produces a series of *states* (description of the environment), *state-updates* (the new information given by the text), and *new-states* (new description of the environment), in a self-supervised way allowing itself learn how to make these transitions.

3.2.1 DWM prompting

For the reason above, we prompt the model with partial information about the task, asking for a description.
Given a task s we divide it into n equal parts, referred as $x_1, \ldots, x_n$ such that their concatenation is the original task $s = x_1 x_2 \ldots x_n$. Then we apply the Discrete World Model (DWM) to produce the description tokens $z_{i+1} \sim p_\theta^{DWM}(z_{i+1}|x_1 z_1 x_2 z_2 \ldots x_i z_i)$ sampling from the language model conditioned on each description of every context. The final string $x_1 z_1 \ldots x_n z_n$ is then given to the model to produce an answer, usually formatted as <answer>YOUR ANSWER</answer>.

Instead of describing the evolution of the environment sentence per sentence, we split the whole sample case into n parts with $n \in \{1, 2, 3, 4, 5\}$, this value will be referred to as the splitted-context variable from now on. The splitted-context variable allowed us to explore how the performance varied by adding more descriptions than a few.

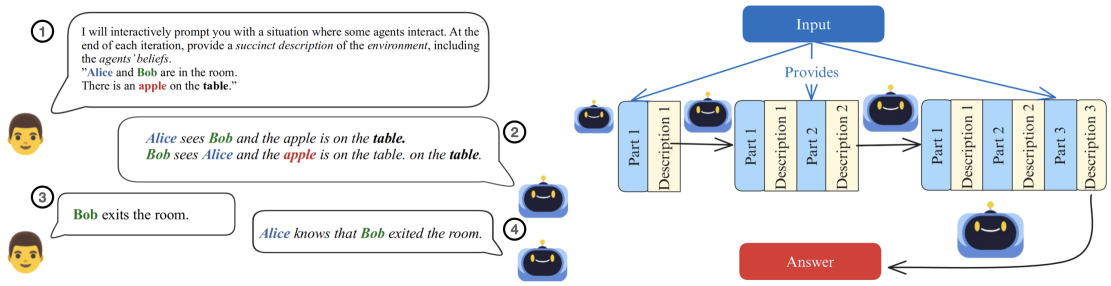

Figure 5: Left: illustration of DWM prompting. We interactively prompt an LLM with a ToM problem, asking to provide a succinct representation of each agent's beliefs. Right: schematic presentation of the DWM method. We first break the input string into T *state descriptions*. Then, for each part, we ask the LLM to provide the **state event** of the environment and how it changes. In the last step, every part of the input and description is fed to the LLM with another prompt to get the answer for the task. Image and caption from my work in [21]

Another consideration for the splitting context variable concerns the budget. Assuming that each description had k tokens and that the original text is divided in n contexts with length l, then the number of tokens requested will be

$$l + (2l + k) + (3l + 2k) + \cdots + (nl + (n-1)k) = \frac{n(n+1)}{2}l + \frac{n(n-1)}{2}k = O(n^2)$$

Adding a quadratic increase of the cost per single case.

3.2.2 Limitation for single sentence cases

In classical Sally-Anne Tests, the environment changes due to the intervention of some agents unbeknownst to others. This asymmetry in information renders possible all the interesting mind-games possible.
The prompting method tries to solve this problem explicitly by describing the change in the environment.
Nonetheless, some tasks like socialIQa [66] do not have a stateful description of how an environment is, but just rely on some absolute common sense knowledge natural for humans.
With these tasks, we limited the splitted-context variable to 1 because there was no state change to be tracked.

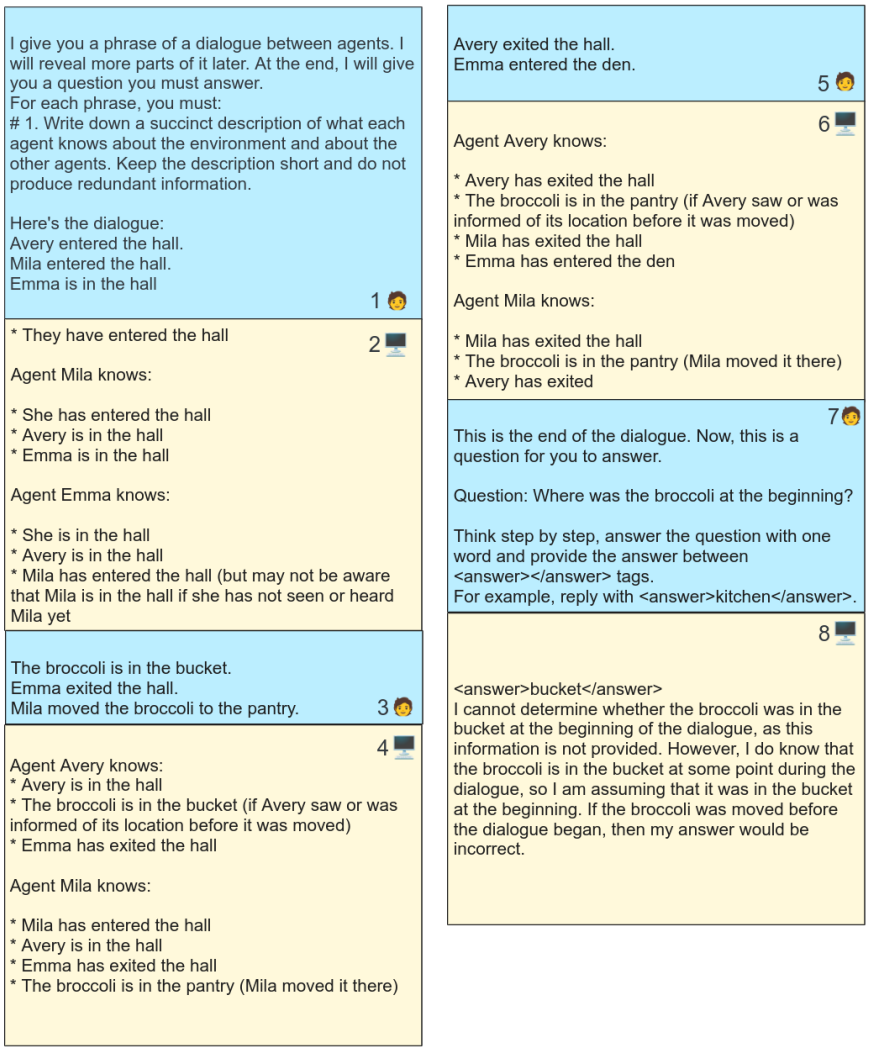

Figure 6: This image schematically presents how DWM works. We use Mixtral 8x7B and the ToMi dataset [29] in this short demo. We begin with a set of instructions, then the Language Model gives the descriptions. This will continue back and forth until the problem is fully described. At that point, we ask the question. We observe in this specific problem the model was able to give his best guess on the question, resolving the dubious setting of the task inherent to some questions of the ToMi datasets.

3.2.3 Structured prompting with YAML and JSON

Language Models displayed great sensitivity to the structure of the query [67], the order of the question [68]. Recent work argues that providing a structured representation of the problem aids the model in reasoning effectively [19, 69].

Following this idea, we develop a structured prompting approach. in this method, we prompt the model to generate a structured representation of the most important information in the text and chain this information with the question that follows.

A short introduction to the formats. We ask the language model to produce a representation in YAML and JSON formats. JSON [70] (JavaScript Object Notation) is a common format used in web-based interchanges. It stores the data in key-value pairs which make the file human-readable, in stark contrast with a binary file format like BSON. YAML [71] (yet another markup language) is a superset of JSON that adheres to the Python philosophy to use spaces instead of curly braces to define the key-value relations.

Why this format should be better? Normal text has lots of redundancies. A structured format should be able to abstract important information and present it in a clear manner.
We expected the YAML format to perform better than the JSON format as the number of tokens required by the GPT tokenizers is fewer compared to the JSON format. Usually, when facing many spaces, those are compacted into a single token. This allows, in principle, for the same representation of information using a minor quantity of tokens.
But in practice, we observe that JSON has the best performance, as noted in 4.

Relation with DWM. We can view this prompting method as a special regularized case of DWM presented before 3.2.1. By setting the *splitting-context* variable to 1 and imposing a specific format for the descriptions, this method is the same as the first prompting method.

Dataset	Inspired by Theory/Test	Test Size	Construction	Example
SocialIQa based on [66]	Reasoning about motivations, what happens next and emotional reaction	400 random sample out of 37,588	Crowdsourcing	In the school play, Robin played a hero in the struggle to the death with the angry villain. How would others feel afterwards? (a) sorry for the villain **(b) hopeful that Robin will succeed** (c) like Robin should lose
ToMi [29]	Unexpected transfer task, first and second order false belief; [28]	1000 random samples	Synthetic	Jackson entered the hall. Chloe entered the hall. The boots is in the bathtub. Jackson exited the hall. Jackson entered the dining_room. Chloe moved the boots to the pantry. (Memory) Where was the boots at the beginning? *(bathtub)* (Reality) Where is the boots really? *(pantry)* (First order) Where will Chloe look for the boots? *(pantry)* (Second order) 7 Where does Chloe think that Jackson searches for the boots? *(bathtub)*
Adv-Csfb based on [20]	Unexpected content or transfer task, integrate commonsense reasoning, first-order false belief; [28]	183 questions 40 stories	Experts	On the shelf, there is a bottle. It is full of beer and the label on this bottle says "beer". Mark walks into the room looking for beer and notices the bottle. He has never seen it before. He reads the label. (a) He opens the bottle and looks inside, He can clearly see that it is full of *(beer)* (b) He believes that it is full of *(beer)* (c) He calls his friend to tell them that he has just found a bottle full of *(beer)*
MindGames based on [10]	Generated stories describing epistemic mental states; Mostly inference reasoning with dynamic epistemic logic. [**?**, 27]	3730 samples	Synthetic	There are two persons. Everyone is visible to others. It is publicly announced that someone's forehead is muddy. It is publicly announced that Alice does not know whether or not everyone's forehead is muddy. It is publicly announced that Alice's forehead is muddy. Alice can now know whether Christine's forehead is muddy. *Entailment=1*
FanToM based on [9]	Generated stories in conversation; Accessibility, first and second order false beliefs; [28]	870 conversations 2 belief questions each	GPT generation and Crowdsourced validation	... **Sabrina**: So, what was the most challenging workout experience you ever had? **Anna**: Definitely when I decided to try out CrossFit. I'm not going to lie, it kicked my butt! **Sabrina**: Wow, that sounds intense. What kind of exercises did you do? **Anna**: We did a lot of different things like high intensity interval training and Olympic lifting with barbells and dumbbells. **Sabrina**: That definitely takes dedication! How did you stay motivated during it? **Anna**: It was tough but I kept reminding myself why I wanted to get fit in the first place and that helped me stay focused on my goals. **Gina**: Hey, I'm back! What were you guys talking about? Sabrina: We were just discussing our most challenging workout experiences. What do you think about when it comes to making a good workout playlist? **Anna**: Music is really important when it comes to getting in the zone while working out. For me, I like upbeat and energetic songs that get me going. **Gina**: Yeah, something with a high tempo can really help push you during those tough workouts! I also like adding in some of my favorite classic songs that give me extra motivation to keep going. ... (Belief) What does Gina believe are the ways in which Anna motivated herself when faced with difficult workouts?

Table 1: Theory of Mind (ToM) datasets used in this work, adapted from [37]

4 Experiments

4.1 Experimental setup

4.1.1 Tasks and datasets

In this section, we will briefly present every dataset used in this study, describe how they were created, the main design principles and how we used them in our benchmarks.

Then, we summarily present the main model families used in this study and the prompting parameters used in the experiments.

ToMi, introduced in [29], is one of the earliest datasets in Neural Theory of Mind inspired by false-belief question and answering (QA) [28], created by adapting other previous QA datasets [72] and alleviating biases in datasets like ToM-bAbi [73]. Its test set comprises 999 machine-generated stories with 6 questions each. In our tests, we randomly sample 1000 story-question pairs and evaluate the LLMs on these test-set representatives.

SocialIQa, introduced in [66], is a multi-choice QA dataset build upon commonsense knowledge graphs [74, 75]. It comprises 2,224 test questions, from which we randomly sample 1000 test representatives in our prompting methods. Each sample is a statement with a question with a given stateless context about wants, reactions, motivations, needs and effects of actions based on *inferential reasoning*. We use this dataset to probe how our method works on stateless tasks. For further information on state, see 5.4.

Adv-Csfb is an expert-written dataset consisting of adversarial examples similar to [35]. It has been developed in [37] to test the ability of LLMs to reason over adversarial cases. It consists of 110 total test cases examining different ToM abilities. We use the whole dataset in a multi-choice fashion for our experiments.

FANToM introduces conversation-based evaluation of ToM abilities [9]. FANToM comprises casual dialogues among multiple characters, centered on topics like pets, risk-taking, or personal growth.
Of the total 870 conversations we test only short-context interactions involving the belief subset of the questions. We sample 1000 randomly from this set and query in an open-ended fashion evaluating with the same system as the original paper.

Mindgames [10] involves composing logical terms to solve puzzles resembling the classic muddy children problem [76, 77]. It approaches the generation of belief tasks classic in ToM analysis [27] from a dynamic epistemic logic perspective. This results in over 3.730 stories followed by a logical statement. We prompted this in a close-ended manner, asking for entailment or not entailment with 1000 randomly sampled test cases.

4.1.2 Language models

We use different language models to test the validity of our methods.

OpenAI Models. We employ the Gpt 3.5 model [15], a language model developed by OpenAI and released to the public in November 2022. Leveraging the transformer architecture [16], it produces coherent and contextually relevant responses to a wide array of prompts, spanning from casual conversation to specialized domains.

Mixstral Models. We test with the Mixstral 8x7B model [78] provided by Huggingface. Mixstral is a mixture of expert model that, as the name suggests, leverages 8 different Mixstral 7B models working in parallel. This model has a router as one of the first layers that chooses which couple of *experts* to choose in a given inference.

Llama 3 Models. Llama 3 is one of the most capable open-weight models [79] released in recent times. We employ the inference offered by the groq platform.

4.1.3 Prompting parameters

Most of the language models used in this work follow the Language Models as a Service (LMaaS) paradigm [80]. This model of service does not help with reproducibility. Because of the classic CI/CD pipeline in software development, the underlying model and weights could change at any time while retaining the same name in the interface.
Therefore, to offer the highest possible grade of reproducibility, we set the temperature to 0.0 or enable greedy decoding. In prompting methods where the creativity of the in response is exploited for better performance, e.g., Tree of Thoughts [53]. We set the temperature to 0.7, the value proposed in the paper.

4.1.4 Baselines

We compare our prompting techniques with CoT [51] and ToT [53].
In each experiment, we evaluate 1000 randomly sampled examples of the dataset and run a series of queries to the language models using similar instructions in each dataset. See 4.1.1 for a description of every single dataset.
We keep the wording of the same prompting technique across each dataset as similar as possible to better compare them and limit the influence of single wording matters.

4.1.5 Design of the prompts

We took great care not to guide excessively the model in a given task and thus invalidating the result, and to ensure fair comparison between the prompting methods.

Same pattern for every dataset. We kept the description as similar as possible across every single prompt approach in order to have a fair comparison between the prompt methods.
For this reason, we used the same <answer>FORMAT</answer> string across every prompt. We gave the same description of the dataset among every prompting method. For example, FanToM comprises of dialogues between agents, and we told the model that there was a dialogue, in mindgames we specified that the answer was always entailment or 'not_entailment.

Not guiding the model. We did not specify anything related to solving the specific dataset in every prompt method. For example, in ToMi, it is known in the original research [29] that only two are the possible answers. In our design, we did not specify this and let the model infer this by itself. The relative comparison of the prompting methods is coherent, so even if the reported accuracy could differ from other papers testing the same ability, our experiments' relative gain in accuracy should be consistent.

4.2 Analysis of the results

The main difference from our method is as follow: while CoT and ToT attempt to describe steps of reasoning steps useful to reach to the conclusion, DWM **splits** the input and attempts to describe different states of the environment with information relevant to the question.
This might seem like a small change, but in practice, it makes a noticeable difference in zero-shot prompting examples.

4.2.1 Is problem splitting helpful?

From the above results we can assert that in some cases providing a description of the state of the world of a part of the problems provides boosts in the model's performance as much as 10%.
In the case of the ToMi dataset, we claim that the drop in performance of our model is caused by the contamination of the dataset during training phase. More on this problem in the section 4.3.
In the case of stateless datasets like socialIQa, where there is no dialogue or change in the environment, nor sub-states where it could be useful to split, our prompting technique falls back on a special type of Chain-of-Thought prompting, as it just gives a description (a reasoning path) of the final state.

From an empirical point of view, most of the observations made during the description of a sub-state are correct. The main ability required is to rephrase

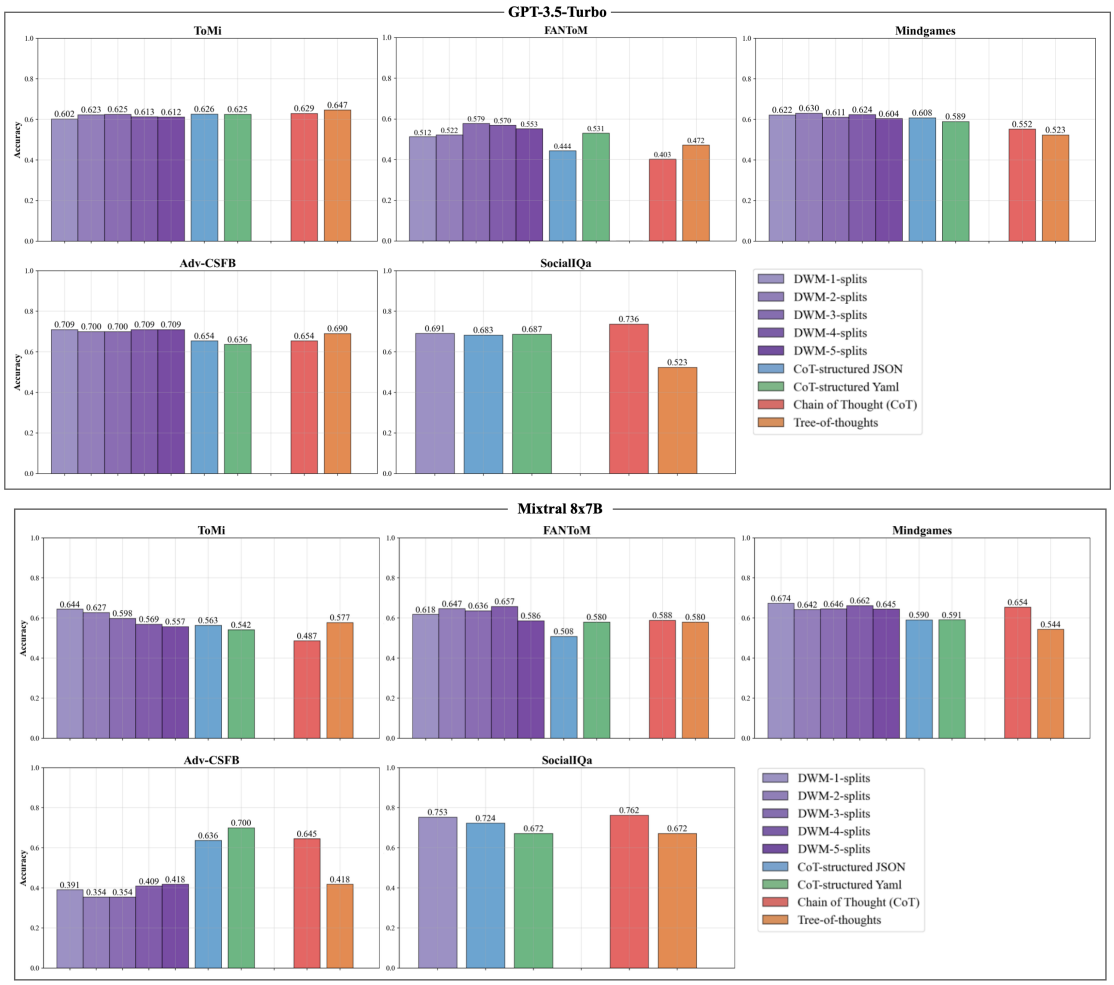

Figure 7: Benchmarks of GPT-3.5-Turbo (top) and Mixtral 8x7B (bottom) models on different ToM tasks for DWM (one to five splits), CoT, ToT and structured prompts (JSON and YAML). Results from [21].

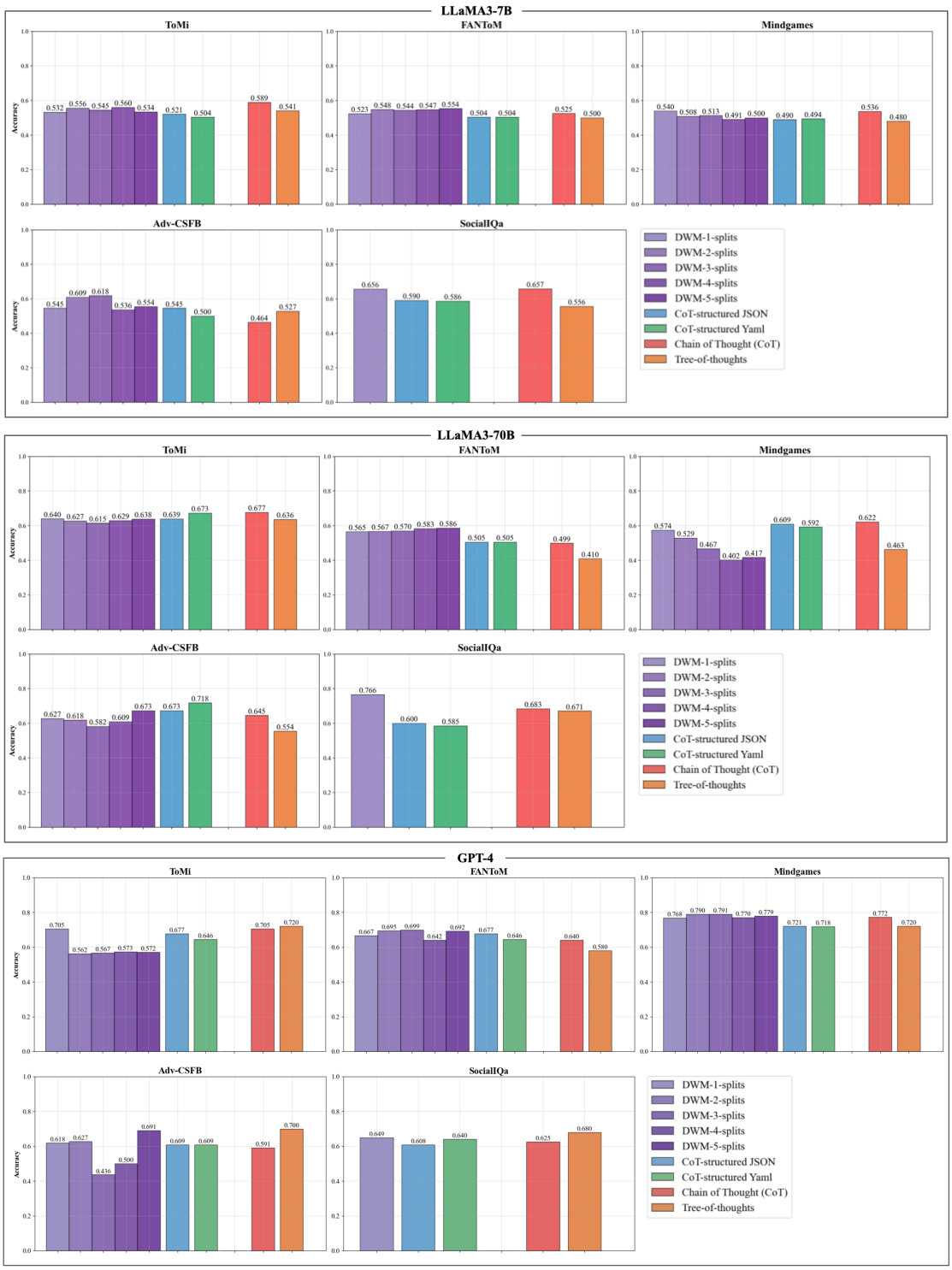

Figure 8: Benchmarks of LLaMA3-7B (top), LLaMA3-70B (middle) and GPT-4 (bottom) models on different ToM tasks for DWM (one to five splits), CoT, ToT and structured prompts (JSON and Yaml). For GPT-4 and ToT, we tested 50 samples (instead of 1000). Results from [21].

known information into a different, almost standardized format chosen by the model.

4.3 The contamination problem

Data contamination and memorization are a major problem in benchmarking modern LLMs [80]. These models have been trained on internet-wide data. The origin of the datasets is not tracked clearly and openly. Therefore, leaks of test splits of datasets used to benchmark the models are possible to have been incorporated in the training data of the language models.
Using benchmarks during the evaluation of methods that depend on contaminated LLMs could severely overestimate the performance and, as a consequence, produce biased and deceptive results [81].
Although some methods were proposed to diminish this problem [82], they mostly rely on the honour code: dataset creators shall not upload the test set of their benchmarks, and model builders shouldn't include benchmarks into their models' training data.

In our setting, we focus on the effects of data contamination in explaining some of the results we observe in the ToMi [29] dataset. We conduct an experiment following the method from [83]and conclude that ToMi has been contaminated in the training data of GPT-3.5. See 4.3 for further details.

We hypothesize that our self-prompting method is worse than the CoT [51] as the produced description of the state breaks the learned pattern during the pre-training phase.

In order to verify this claim, we run an ablation test, described in the next section 4.3.

On the contamination of ToMi. We prompt the gpt−3−5−instruct model with a prefix of the database, splitted by line. As the model should be trained to complete the text, if we observe a complete match between a long enough section of the original and generated text, it can be safely assumed that there has been a leak in the training data. The inverse is false.
We observe that gpt−3.5−instruct completes the input text verbatim for the ToMi dataset [29].
We split the dataset into 761 prompts with 70 lines from the dataset, and check if the produced tokens match exactly the continuation of the prompts. We report that 484 promps, that is 63% of the dataset, does. We report this as hard evidence of the contamination.

Example prompt.

1 Hannah entered the patio.

2 Noah entered the patio.
... (Omitted for clarity)
1 Carter entered the cellar.
2 Jacob entered the workshop.

Example output from the model.

3 Mia is in the cellar.
4 The gloves is in the envelope.
5 Mia moved the gloves to the container.
6 Jacob hates the pumpkin

Correct continuation from the dataset.

3 Mia is in the cellar.
4 The gloves is in the envelope.
5 Mia moved the gloves to the container.
6 Jacob hates the pumpkin

Ablation tests for memorization. We claim that the small percentages in different performances are caused by memorizing the ToMi dataset during training. To empirically verify this claim, we set up two experiments in the following way:

We insert random strings, like "ignore me" or just straight up "AAAAA" after some sentences in ToMi dataset. And run CoT prompting. We observe a loss of about "4%" when running GPT3.5 over these new samples.

We run a second experiment using ToT. With this prompting method, after the thoughts are chosen, the final question is posed. We ran two cases, in one the question was just after the input sample, as if it was in the ToMi dataset, in the other case, we moved the question to the end. In this way, we claim the dataset pattern is broken by the observations that are chosen during the breath-first-search in the tree of thought algorithm.
We observe a drop in performance of 2.7% with this simple change, given other things equal, which is significant compared to the best performance of the other prompting methods.

This is the example of the given template:

Given this dialogue and possible observations, answer the question with one word and provide the answer between <answer></answer> tags.
{problem}
{question}
{observations}

For example, reply with <answer>vase</answer>.

Example of the first case, using the pattern {problem}{question} directly recreates the pattern stored in the dataset.

Given this dialogue and possible observations, answer the question with one word and provide the answer between <answer></answer> tags.
{problem}
{observations}
{question}

For example, reply with <answer>vase</answer>.

A single change, moving the question down, has a decrease in performance comparable to our prompting methods.

4.4 Analysis of the Complexity

In this section, we propose the analysis present in my work in [21], as the experiments for the paper share some of the same insights on this topic.

We used the complexity framework introduced in Section 3.1 to characterise the statefulness and statelessness of the five ToM benchmarks used for the experimental evaluation.

We randomly sampled 50 problems from each dataset, identified the objects, and manually labelled stateful and stateless *state events*. We release the split samples alongside a web application that facilitates manual labelling, available here in the code repository associated to this work.

As illustrated in Figure 10 (left), the statefulness of each problem, i.e., that of the object a model must track to answer correctly, **strongly correlates** with the best-performing DWM split.

The statelessness complexity, reported in Figure 10 (middle), i.e., that of objects that a model does not need to track, grows larger for problems such as FANToM, only partially influencing the models' performance. We hypothesize that the most potent models developed some competency in discerning the relevant part of a prompt from the confounding ones, suggesting an ability of discerning between stateful and stateless.

We finally report, in Figure 10 (right), the complexity of each problem computed as per Eq. 3.1.2, with τ set in a range between 0.05 and 0.2 (i.e., the relative weight of stateless compared to stateful events). Results suggest that FANToM is the most difficult ToM task for humans and LLMs (see Figure 7), followed by ToMi (the second most difficult for LLMs as well) and Adv-CSFB (which seems easier than the others); in contrast, Mindgames and SocialIQa tend to be easier.

Finally, in Figure 9}, we compare the accuracy of GPT-3.5-Turbo, GPT-4, Mixtral 8x7B and LLaMA3-70B when prompted with CoT (i.e., without split) on the five

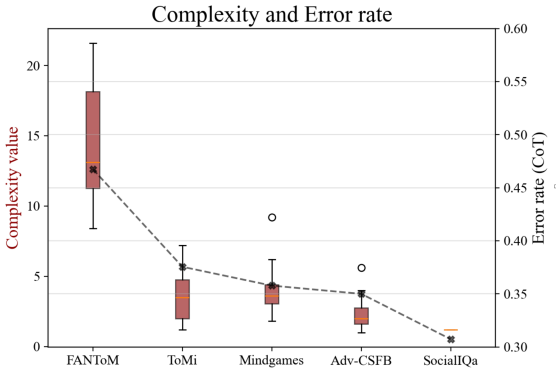

Figure 9: Each boxplot summarizes the complexity analysis of the five ToM benchmarks in ascending order. We report the average **error rate** (i.e., 1-accuracy) of GPT-3.5-Turbo, GPT-4, Mixtral 8x7B and LLaMA3-70B on the task when prompted with CoT. Results from [21]

ToM benchmarks with the complexity of the task as per Def. 3.1.2}. We observe a **strong correlation** between the error-rate and the complexity of a task, i.e., our framework correctly identifies the tasks that are harder both for humans and current state-of-the-art LLMs.

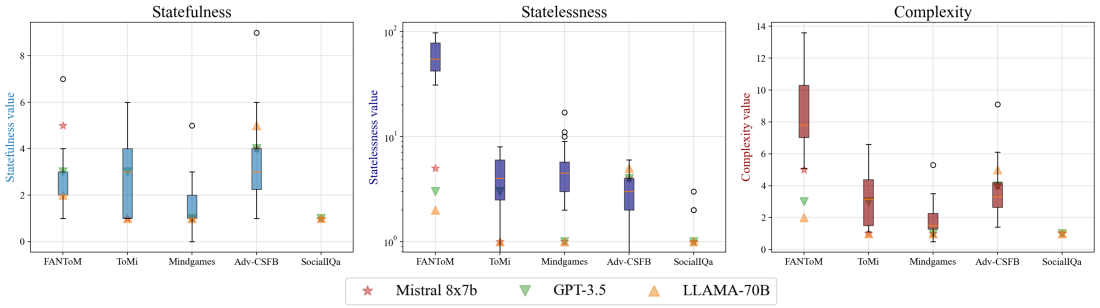

Figure 10: Each boxplot summarizes the statefulness (left), statelessness (middle, y-axis in log-scale) and complexity analysis (right) of the five ToM benchmarks. We report mean, standard deviation and outliers alongside the best DWM method (by the number of prompt splits) and observe a strong correlation between the number of splits and the statefulness. Results from [21]

5 Discussion

5.1 Are prompting methods a good approach to test ToM?

Human's theory of mind is, roughly speaking, assessed by the correctness of their answer after a given question, specifically designed to test some mental property. Currently, models are tested in the same way. Even though the underlying principles that enable ToM abilities to emerge could be different, if the model behaves similarly to humans in test regarding ToM, we argue they have similar degrees of ToM ability.

In [37], Shapira states that tests specifically designed to test humans are not automatically transferable to test models. This observation is generally true only if the test was designed to detect latent abilities different between humans and machines.

On the other hand, if the model manifestly behaves in the same way compared to humans, such that they fail when a human fails, and are correct when a human is correct, the model is functionally equivalent to the role of the human in that context. There is no clear distinction between the two. Even if the inner mechanism of the two entities are different, we have complete interchangeability between the two.

We argue for the concept of **functional similarity**: a comparison grounded to human capability, reminiscent of the Turing-Test [84]. We say that two objects are **functionally similar** if given the same context, task or problem, they will behave in the same manner.

In this framework, prompting methods are effective approaches to test ToM abilities inside language models as the test method assesses for the same *linguistic behaviour*

in humans and machines. Nevertheless, the unbounded nature of possible human manifest behaviours complicates an operational application of **functional similarity**. Yet, most behaviours can be categorized within similar buckets of interest. A possible direction of research would be to explicitly present what are the sets of most general interactions in humans and develop datasets for each category. This dataset should be operational, meaning they should test for effective competence using ToM-related skills rather than latent variables correlated to these skills. To the best of my knowledge, such general categorization has not been attempted and is probably quite expensive. On the other hand, it would provide a fair comparison between humans and machines.

5.2 Performance gap in prompting methods

From a purely empirical point of view, we have observed that prompting methods have an enormous gap of performance in a wide number of tasks. Surprisingly, similar behaviour has been seen in humans. In the remarkable "Copy Machine" study [85], some researchers were in line for a printer. The experimenter asked to use the machine first to copy 5 pages in the following ways:

1. "Excuse me, I have 5 pages. May I use the xerox machine?"
2. "Excuse me, I have 5 pages. May I use the xerox machine, because I have to make copies?"
3. "Excuse me, I have 5 pages. May I use the xerox machine, because I'm in a rush?"

The surprising datum is that the percentage of success across 120 people were 60, 93, 94, respectively. It seems that just rephrasing the question, in the context of two strangers asking for permission, is sufficient to uncover a wide gap in "performance".

Communication is often multi-channel in humans. Often, the same information can be conveyed through voice-tone, gestures, or facial expressions. In contrast, the language models we tested in this work, operate only with the language channel. We envision two possible paths in exploring prompting methods, given these observations from human communication. (1) Designing an experiment that explores techniques similar to prompting in humans by isolating a single channel of communication. (2) Trying to extend the analysis of current prompting methods to multi-modal models. [86] provides compelling evidence that multi-modal transformer-based models are able to recognize human emotions in complex movie settings. This indicates the need for more general multi-modal benchmarks for ToM.

5.3 On the spontaneous Theory of Mind

Spontaneous thoughts are mental processes that arise without any conscious act. Classical ToM tests as the Sally-Anne [28] require a conscious effort in understanding and answering the question.
Gurney and Pynadah [87] argue that a **spontaneous** approach to ToM could have better results. This nudges toward experimenting with a computational entity able to draw from spontaneous channels as social cues and use them to predict the behaviour of the entity they are communicating with and plan accordingly [88].
This observation was largely drawn from work on human ToM. Gurney and Pynadah argued that prompting tests, which explicitly question for ToM abilities as the famous Sally-Anne test does, are fragile. Humans could be easily persuaded into believing something, even if it never happened before [89], or could be influenced just by the wording of the request [85]. For this reason, they advocate more from an operational point of view. For example, in humans, the ability to form correct true or false beliefs assumptions could be derived from eye fixation times [90]. Is it even possible to test for spontaneity in machines?

Even if, in theory, the spontaneous approach may be sound, in practice, there is no clear definition of what is spontaneous for machines. The above definition of spontaneous thought is linked to consciousness, which is currently out of scope and has no meaning when applied to machines. Chat-bots like ChatGPT have no body and can't interact without being prompted first. Furthermore, they don't have other channels of communication, such as facial expression, from which states could be inferred, as in humans. Nonetheless, if it is possible to design a task that requires a certain degree of ToM, but doesn't directly test for it, we can abductively infer that it is more probable that the model has ToM. This is closer to that spontaneous use of ToM. Therefore, a possible approach in testing ToM is to design tasks and benchmarks that directly need ToM abilities and compare them to the human baseline. If the models can generalize on those tasks with just a few examples, asserting they have human-level ToM could be possible.

5.4 Limitations

Higher order belief tracking. As specified in the limitation section of [21], our theoretical framework reduces the problem of solving a belief ToM problem to finding the correct descriptions that need to be tracked. It extends seamlessly to tasks with much higher complexity, however, we have not had the opportunity to test this theory in those settings. We noticed that most theory of mind tasks available in the community only require one to five states to be correctly answered. A possible extension would be testing the theory upon tasks with higher state complexity, e.g. k^{th}-order belief tracking tasks. However, it is unclear whether this

could be useful in real applications as most human belief tracking is limited to 5 or 6 orders [91, 92]

Where is the best location to place a split? Intuitively, a good place where to put a split is after a change of state interesting for answering the question. We expect the new description at this place of the text to be helpful in giving valuable information to answer the question. But a few technical problems arise: how do you decide if a change of state is of interest? How can we choose from many possible split points to stay within budget constraints? In this work, we just set the number of wanted splits k and created k equal-sized portions of text. A possible path is creating a heuristic or prompting a LLM to decide a better place to allocate for the split. But we leave this to future research.

On the performance over stateless problems. We described a method that shows improved performance over tasks where a description of the change of the environment provides useful notions. For some problems, however, the answer is stateless: the answer does not depend on any past interaction, it doesn't benefit from any retrieved knowledge, and does not require the state to be tracked. Examples as socialIQa seem to be easy to answer, but only if the model possesses some *parametric knowledge* about the task, which is often difficult to quantify.
Our prompting method is usually not useful on this type of tasks as the implicit knowledge is not explained just by describing the state.

Memorization analysis. Training and evaluating on the same dataset produces positively biased data on the model's performance. While running our benchmarks on ToMi, we discovered that the GPT-3.5 model had completely memorized parts of the dataset. This motivated us to extend the memorization test to the other tasks. We urge the research community to include a memorization section on every benchmark study with public datasets used in their works. This data is crucial to conduct fair and unbiased research on evaluating LLMs' abilities [82].
Future works will include an analysis of the memorisation rate of other ToM tasks alongside tests to quantify their impact on different models.

6 Conclusion

This thesis introduces a complexity framework to measure the difficulty of Theory of Mind (ToM) problems, currently published as a preprint on the archive [21]. It quantifies the difficulty by tracking necessary states (stateful) and unnecessary states (stateless), with the latter discounted in the complexity computation. The framework evidences a strong correlation between complexity and model performance.
Inspired by this work, we propose a new prompting method which attempts to extract a structure from a given task, interpretable as a local world model built by the LLM itself. Along with this method, we propose a structured version where we ask for a YAML or JSON format of the state. We test the ability of these prompting methods in enhancing ToM reasoning in common ToM and commonsense datasets and discover in the best case, we reach a 9.99% increase in performance when compared to the baselines. We provide compelling evidence on the contamination of the training data in some datasets tested in this work.

6.1 Final notes

6.1.1 Usage of Language Models as Writing tools

I declare that all the paragraphs in this work are written by a human. ChatGPT and other language models have been helpful in activities like

- Finding synonyms
- Rewording sentences
- Correct grammar errors

No entire paragraph in this work has been written by Language Models.

6.1.2 Acknowledgements

My heartfelt thanks go to my supervisor Emanuele La Malfa, whose support, both intellectually and personally, was invaluable to this project. I am grateful to my professor, Andrea Asperti, for giving me the opportunity to pursue this work.

References

[1] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis. Mastering Chess and Shogi by Self-Play

with a General Reinforcement Learning Algorithm. [Online]. Available: http://arxiv.org/abs/1712.01815

[2] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, p. d. u. family=Driessche, given=George, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," vol. 529, no. 7587, pp. 484–489. [Online]. Available: https://www.nature.com/articles/nature16961

[3] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Cai, J. P. Agapiou, M. Jaderberg, A. S. Vezhnevets, R. Leblond, T. Pohlen, V. Dalibard, D. Budden, Y. Sulsky, J. Molloy, T. L. Paine, C. Gulcehre, Z. Wang, T. Pfaff, Y. Wu, R. Ring, D. Yogatama, D. Wünsch, K. McKinney, O. Smith, T. Schaul, T. Lillicrap, K. Kavukcuoglu, D. Hassabis, C. Apps, and D. Silver, "Grandmaster level in StarCraft II using multi-agent reinforcement learning," vol. 575, no. 7782, pp. 350–354. [Online]. Available: https://www.nature.com/articles/s41586-019-1724-z

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, vol. 25. Curran Associates, Inc. [Online]. Available: https://papers.nips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. [Online]. Available: http://arxiv.org/abs/1506.02640

[6] J. Abramson, J. Adler, J. Dunger, R. Evans, T. Green, A. Pritzel, O. Ronneberger, L. Willmore, A. J. Ballard, J. Bambrick, S. W. Bodenstein, D. A. Evans, C.-C. Hung, M. O'Neill, D. Reiman, K. Tunyasuvunakool, Z. Wu, A. Žemgulytė, E. Arvaniti, C. Beattie, O. Bertolli, A. Bridgland, A. Cherepanov, M. Congreve, A. I. Cowen-Rivers, A. Cowie, M. Figurnov, F. B. Fuchs, H. Gladman, R. Jain, Y. A. Khan, C. M. R. Low, K. Perlin, A. Potapenko, P. Savy, S. Singh, A. Stecula, A. Thillaisundaram, C. Tong, S. Yakneen, E. D. Zhong, M. Zielinski, A. Žídek, V. Bapst, P. Kohli, M. Jaderberg, D. Hassabis, and J. M. Jumper, "Accurate structure prediction of biomolecular interactions with AlphaFold 3," pp. 1–3. [Online]. Available: https://www.nature.com/articles/s41586-024-07487-w

[7] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, H. Nori, H. Palangi, M. T. Ribeiro, and Y. Zhang. Sparks of Artificial General Intelligence: Early experiments with GPT-4. [Online]. Available: http://arxiv.org/abs/2303.12712

[8] D. Ha and J. Schmidhuber, "World Models." [Online]. Available: http://arxiv.org/abs/1803.10122

[9] H. Kim, M. Sclar, X. Zhou, R. L. Bras, G. Kim, Y. Choi, and M. Sap. FANToM: A Benchmark for Stress-testing Machine Theory of Mind in Interactions. [Online]. Available: http://arxiv.org/abs/2310.15421

[10] D. Sileo and A. Lernould, "MindGames: Targeting Theory of Mind in Large Language Models with Dynamic Epistemic Modal Logic." [Online]. Available: https://arxiv.org/abs/2305.03353

[11] Z. Chen, J. Wu, J. Zhou, B. Wen, G. Bi, G. Jiang, Y. Cao, M. Hu, Y. Lai, Z. Xiong, and M. Huang. ToMBench: Benchmarking Theory of Mind in Large Language Models. [Online]. Available: http://arxiv.org/abs/2402.15052

[12] H. M. Wellman, "The Development of Theory of Mind: Historical Reflections," vol. 11, no. 3, pp. 207–214. [Online]. Available: https://srcd.onlinelibrary.wiley.com/doi/10.1111/cdep.12236

[13] J. McCarthy, "Programs with Common Sense."

[14] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners." [Online]. Available: https://paperswithcode.com/paper/language-models-are-unsupervised-multitask

[15] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language Models are Few-Shot Learners. [Online]. Available: http://arxiv.org/abs/2005.14165

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention Is All You Need. [Online]. Available: http://arxiv.org/abs/1706.03762

[17] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals,

P. Liang, J. Dean, and W. Fedus. Emergent Abilities of Large Language Models. [Online]. Available: http://arxiv.org/abs/2206.07682

[18] S. Hao, Y. Gu, H. Ma, J. J. Hong, Z. Wang, D. Z. Wang, and Z. Hu. Reasoning with Language Model is Planning with World Model. [Online]. Available: http://arxiv.org/abs/2305.14992

[19] F. Lin, E. La Malfa, V. Hofmann, E. M. Yang, A. Cohn, and J. B. Pierrehumbert. Graph-enhanced Large Language Models in Asynchronous Plan Reasoning. [Online]. Available: http://arxiv.org/abs/2402.02805

[20] M. Kosinski. Theory of Mind Might Have Spontaneously Emerged in Large Language Models. [Online]. Available: http://arxiv.org/abs/2302.02083

[21] X. A. Huang, E. La Malfa, S. Marro, A. Asperti, A. Cohn, and M. Wooldridge. A Notion of Complexity for Theory of Mind via Discrete World Models. [Online]. Available: http://arxiv.org/abs/2406.11911

[22] D. C. Dennett, "Intentional Systems," vol. 68, no. 4, pp. 87–106. [Online]. Available: https://www.jstor.org/stable/2025382

[23] D. Premack and G. Woodruff, "Does the chimpanzee have a theory of mind?" vol. 1, no. 4, pp. 515–526. [Online]. Available: https://www.cambridge.org/core/journals/behavioral-and-brain-sciences/article/does-the-chimpanzeehave-a-theory-of-mind/1E96B02CD9850016B7C93BC6D2FEF1D0

[24] M. Tomasello, J. Call, and B. Hare, "Chimpanzees understand psychological states – the question is which ones and to what extent," vol. 7, no. 4, pp. 153–156. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364661303000354

[25] A. Gopnik and H. M. Wellman, "The theory theory," in *Mapping the Mind: Domain Specificity in Cognition and Culture*, L. A. Hirschfeld and S. A. Gelman, Eds. Cambridge University Press, pp. 257–293. [Online]. Available: https://www.cambridge.org/core/books/mapping-the-mind/theory-theory/5B959278347235DE3177E162C8224BF1

[26] B. J. Scholl and A. M. Leslie, "Modularity, Development and 'Theory of Mind'," vol. 14, no. 1, pp. 131–153. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1111/1468-0017.00106

[27] H. Wimmer and J. Perner, "Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children's understanding

of deception," vol. 13, no. 1, pp. 103–128. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0010027783900045

[28] S. Baron-Cohen, A. M. Leslie, and U. Frith, "Does the autistic child have a "theory of mind" ?" vol. 21, no. 1, pp. 37–46. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0010027785900228

[29] M. Le, Y.-L. Boureau, and M. Nickel, "Revisiting the Evaluation of Theory of Mind through Question Answering," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, pp. 5871–5876. [Online]. Available: https://www.aclweb.org/anthology/D19-1598

[30] J. McCarthy, M. L. Minsky, N. Rochester, and C. E. Shannon, "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955," vol. 27, no. 4, pp. 12–12. [Online]. Available: https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/1904

[31] J. McCarthy, "Ascribing Mental Qualities To Machines."

[32] B. Scassellati, "Theory of Mind for a Humanoid Robot," vol. 12, no. 1, pp. 13–24. [Online]. Available: https://doi.org/10.1023/A:1013298507114

[33] R. A. Brooks, C. Breazeal, M. Marjanović, B. Scassellati, and M. M. Williamson, "The Cog Project: Building a Humanoid Robot," in *Computation for Metaphors, Analogy, and Agents*, C. L. Nehaniv, Ed. Springer Berlin Heidelberg, vol. 1562, pp. 52–87. [Online]. Available: http://link.springer.com/10.1007/3-540-48834-0_5

[34] Y. Choi, "The Curious Case of Commonsense Intelligence," vol. 151, no. 2, pp. 139–155. [Online]. Available: https://direct.mit.edu/daed/article/151/2/139/110627/The-Curious-Case-of-Commonsense-Intelligence

[35] T. Ullman. Large Language Models Fail on Trivial Alterations to Theory-of-Mind Tasks. [Online]. Available: http://arxiv.org/abs/2302.08399

[36] M. Sap, R. LeBras, D. Fried, and Y. Choi. Neural Theory-of-Mind? On the Limits of Social Intelligence in Large LMs. [Online]. Available: http://arxiv.org/abs/2210.13312

[37] N. Shapira, M. Levy, S. H. Alavi, X. Zhou, Y. Choi, Y. Goldberg, M. Sap, and V. Shwartz. Clever Hans or Neural Theory of Mind? Stress Testing Social Reasoning in Large Language Models. [Online]. Available: http://arxiv.org/abs/2305.14763

[38] M. Sclar, S. Kumar, P. West, A. Suhr, Y. Choi, and Y. Tsvetkov. Minding Language Models' (Lack of) Theory of Mind: A Plug-and-Play Multi-Character Belief Tracker. [Online]. Available: http://arxiv.org/abs/2306.00924

[39] M. Beck, K. Pöppel, M. Spanring, A. Auer, O. Prudnikova, M. Kopp, G. Klambauer, J. Brandstetter, and S. Hochreiter. xLSTM: Extended Long Short-Term Memory. [Online]. Available: http://arxiv.org/abs/2405.04517

[40] B. Peng, E. Alcaide, Q. Anthony, A. Albalak, S. Arcadinho, S. Biderman, H. Cao, X. Cheng, M. Chung, M. Grella, K. K. GV, X. He, H. Hou, J. Lin, P. Kazienko, J. Kocon, J. Kong, B. Koptyra, H. Lau, K. S. I. Mantri, F. Mom, A. Saito, G. Song, X. Tang, B. Wang, J. S. Wind, S. Wozniak, R. Zhang, Z. Zhang, Q. Zhao, P. Zhou, Q. Zhou, J. Zhu, and R.-J. Zhu. RWKV: Reinventing RNNs for the Transformer Era. [Online]. Available: http://arxiv.org/abs/2305.13048

[41] P. W. Anderson, "More Is Different," vol. 177, no. 4047, pp. 393–396. [Online]. Available: https://www.science.org/doi/10.1126/science.177.4047.393

[42] D. R. Hofstadter, *I Am a Strange Loop*. Basic Books. [Online]. Available: https://psycnet.apa.org/record/2007-01197-000

[43] S. Kambhampati, "Can Large Language Models Reason and Plan?" p. nyas.15125. [Online]. Available: http://arxiv.org/abs/2403.04121

[44] S. Lu, I. Bigoulaeva, R. Sachdeva, H. T. Madabushi, and I. Gurevych. Are Emergent Abilities in Large Language Models just In-Context Learning? [Online]. Available: http://arxiv.org/abs/2309.01809

[45] R. Schaeffer, B. Miranda, and S. Koyejo, "Are Emergent Abilities of Large Language Models a Mirage?" [Online]. Available: https://arxiv.org/abs/2304.15004

[46] A. Petrov, P. H. S. Torr, and A. Bibi. When Do Prompting and Prefix-Tuning Work? A Theory of Capabilities and Limitations. [Online]. Available: http://arxiv.org/abs/2310.19698

[47] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. [Online]. Available: http://arxiv.org/abs/2107.13586

[48] B. Lester, R. Al-Rfou, and N. Constant. The Power of Scale for Parameter-Efficient Prompt Tuning. [Online]. Available: http://arxiv.org/abs/2104.08691

[49] K. Hambardzumyan, H. Khachatrian, and J. May, "WARP: Word-level Adversarial ReProgramming," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds. Association for Computational Linguistics, pp. 4921–4933. [Online]. Available: https://aclanthology.org/2021.acl-long.381

[50] X. L. Li and P. Liang, "Prefix-Tuning: Optimizing Continuous Prompts for Generation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds. Association for Computational Linguistics, pp. 4582–4597. [Online]. Available: https://aclanthology.org/2021.acl-long.353

[51] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. [Online]. Available: http://arxiv.org/abs/2201.11903

[52] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa. Large Language Models are Zero-Shot Reasoners. [Online]. Available: http://arxiv.org/abs/2205.11916

[53] S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, and K. Narasimhan. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. [Online]. Available: http://arxiv.org/abs/2305.10601

[54] J. Long. Large Language Model Guided Tree-of-Thought. [Online]. Available: http://arxiv.org/abs/2305.08291

[55] D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, C. Cui, O. Bousquet, Q. V. Le, and E. H. Chi, "Least-to-Most Prompting Enables Complex Reasoning in Large Language Models." [Online]. Available: https://openreview.net/forum?id=WZH7099tgfM

[56] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou. Self-Consistency Improves Chain of Thought Reasoning in Language Models. [Online]. Available: http://arxiv.org/abs/2203.11171

[57] P. Zhou, J. Pujara, X. Ren, X. Chen, H.-T. Cheng, Q. V. Le, E. H. Chi, D. Zhou, S. Mishra, and H. S. Zheng. Self-Discover: Large Language Models Self-Compose Reasoning Structures. [Online]. Available: http://arxiv.org/abs/2402.03620

[58] Applications of General Linguistics - Firth - 1957 - Transactions of the Philological Society - Wiley Online Library. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-968X.1957.tb00568.x

[59] R. Cotterell, A. Svete, C. Meister, T. Liu, and L. Du. Formal Aspects of Language Modeling. [Online]. Available: http://arxiv.org/abs/2311.04329

[60] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative Pre-Training."

[61] J. Dodge, M. Sap, A. Marasović, W. Agnew, G. Ilharco, D. Groeneveld, M. Mitchell, and M. Gardner. Documenting Large Webtext Corpora: A Case Study on the Colossal Clean Crawled Corpus. [Online]. Available: http://arxiv.org/abs/2104.08758

[62] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving. Fine-Tuning Language Models from Human Preferences. [Online]. Available: http://arxiv.org/abs/1909.08593

[63] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. [Online]. Available: http://arxiv.org/abs/2305.18290

[64] N. Kim and S. Schuster. Entity Tracking in Language Models. [Online]. Available: http://arxiv.org/abs/2305.02363

[65] J. Pilault, R. Li, S. Subramanian, and C. Pal, "On Extractive and Abstractive Neural Document Summarization with Transformer Language Models," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds. Association for Computational Linguistics, pp. 9308–9319. [Online]. Available: https://aclanthology.org/2020.emnlp-main.748

[66] M. Sap, H. Rashkin, D. Chen, R. Le Bras, and Y. Choi, "Social IQa: Commonsense Reasoning about Social Interactions," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, pp. 4463–4473. [Online]. Available: https://aclanthology.org/D19-1454

[67] M. Sclar, Y. Choi, Y. Tsvetkov, and A. Suhr, "Quantifying Language Models' Sensitivity to Spurious Features in Prompt Design or: How I learned to start worrying about prompt formatting." [Online]. Available: https://openreview.net/forum?id=RIu5lyNXjT

[68] P. Pezeshkpour and E. Hruschka. Large Language Models Sensitivity to The Order of Options in Multiple-Choice Questions. [Online]. Available: http://arxiv.org/abs/2308.11483

[69] B. Fatemi, J. Halcrow, and B. Perozzi. Talk like a Graph: Encoding Graphs for Large Language Models. [Online]. Available: http://arxiv.org/abs/2310.04560

[70] T. Bray, "The JavaScript Object Notation (JSON) Data Interchange Format." [Online]. Available: https://datatracker.ietf.org/doc/rfc8259

[71] R. Polli, E. Wilde, and E. Aro, "YAML Media Type." [Online]. Available: https://datatracker.ietf.org/doc/rfc9512

[72] J. Weston, A. Bordes, S. Chopra, A. M. Rush, p. u. family=Merriënboer, given=Bart, A. Joulin, and T. Mikolov. Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks. [Online]. Available: http://arxiv.org/abs/1502.05698

[73] A. Nematzadeh, K. Burns, E. Grant, A. Gopnik, and T. L. Griffiths. Evaluating Theory of Mind in Question Answering. [Online]. Available: http://arxiv.org/abs/1808.09352

[74] M. Sap, R. LeBras, E. Allaway, C. Bhagavatula, N. Lourie, H. Rashkin, B. Roof, N. A. Smith, and Y. Choi. ATOMIC: An Atlas of Machine Commonsense for If-Then Reasoning. [Online]. Available: http://arxiv.org/abs/1811.00146

[75] A. Bosselut, H. Rashkin, M. Sap, C. Malaviya, A. Celikyilmaz, and Y. Choi. COMET: Commonsense Transformers for Automatic Knowledge Graph Construction. [Online]. Available: http://arxiv.org/abs/1906.05317

[76] p. u. family=Eijck, given=Jan, "Dynamic Epistemic Logics," pp. 175–202.

[77] Y. Shoham and K. Leyton-Brown, *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press.

[78] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, p. l. u. family=Casas, given=Diego, E. B. Hanna, F. Bressand, G. Lengyel, G. Bour, G. Lample, L. R. Lavaud, L. Saulnier, M.-A. Lachaux, P. Stock, S. Subramanian, S. Yang, S. Antoniak, T. L. Scao, T. Gervet, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed. Mixtral of Experts. arXiv.org. [Online]. Available: https://arxiv.org/abs/2401.04088v1

[79] Introducing Meta Llama 3: The most capable openly available LLM to date. Meta AI. [Online]. Available: https://ai.meta.com/blog/meta-llama-3/

[80] E. La Malfa, A. Petrov, S. Frieder, C. Weinhuber, R. Burnell, R. Nazar, A. G. Cohn, N. Shadbolt, and M. Wooldridge. Language Models as a Service: Overview of a New Paradigm and its Challenges. [Online]. Available: http://arxiv.org/abs/2309.16573

[81] O. Sainz, J. Campos, I. García-Ferrero, J. Etxaniz, p. u. family=Lacalle, given=Oier Lopez, and E. Agirre, "NLP Evaluation in trouble: On the Need to Measure LLM Data Contamination for each Benchmark," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, H. Bouamor, J. Pino, and K. Bali, Eds. Association for Computational Linguistics, pp. 10 776–10 787. [Online]. Available: https://aclanthology.org/2023.findings-emnlp.722

[82] A. Jacovi, A. Caciularu, O. Goldman, and Y. Goldberg. Stop Uploading Test Data in Plain Text: Practical Strategies for Mitigating Data Contamination by Evaluation Benchmarks. [Online]. Available: http://arxiv.org/abs/2305.10160

[83] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson, A. Oprea, and C. Raffel. Extracting Training Data from Large Language Models. [Online]. Available: http://arxiv.org/abs/2012.07805

[84] A. M. Turing, "I.—COMPUTING MACHINERY AND INTELLIGENCE," vol. LIX, no. 236, pp. 433–460. [Online]. Available: https://academic.oup.com/mind/article/LIX/236/433/986238

[85] E. J. Langer, A. Blank, and B. Chanowitz, "The mindlessness of ostensibly thoughtful action: The role of "placebic" information in interpersonal interaction," vol. 36, no. 6, pp. 635–642.

[86] D. Srivastava, A. K. Singh, and M. Tapaswi. How you feelin'? Learning Emotions and Mental States in Movie Scenes. [Online]. Available: http://arxiv.org/abs/2304.05634

[87] N. Gurney, D. V. Pynadath, and V. Ustun. Spontaneous Theory of Mind for Artificial Intelligence. [Online]. Available: http://arxiv.org/abs/2402.13272

[88] M. K. Ho, R. Saxe, and F. Cushman, "Planning with Theory of Mind," vol. 26, no. 11, pp. 959–971. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1364661322001851

[89] K. A. Braun, R. Ellis, and E. F. Loftus, "Make My Memory: How Advertising Can Change Our Memories of the Past."

[90] A. Senju, V. Southgate, S. White, and U. Frith, "Mindblind eyes: An absence of spontaneous theory of mind in Asperger syndrome," vol. 325, no. 5942, pp. 883–885.

[91] D. C. Dennett, "The intentional stance in theory and practice," in *Machiavellian Intelligence: Social Expertise and the Evolution of Intellect in Monkeys, Apes, and Humans*. Clarendon Press/Oxford University Press, pp. 180–202.

[92] J. Cargile, "A Note on "Iterated Knowings"," vol. 30, no. 5, pp. 151–155. [Online]. Available: https://www.jstor.org/stable/3328051