ToM-LM: Delegating Theory of Mind Reasoning to External Symbolic Executors in Large Language Models

Weizhi Tang[0009−0006−7444−292X] and Vaishak Belle[0000−0001−5573−8465]

University of Edinburgh

Abstract. Theory of Mind (ToM) refers to the ability of individuals to attribute mental states to others. While Large Language Models (LLMs) have shown some promise with ToM ability, they still struggle with complex ToM reasoning. Our approach leverages an external symbolic executor, specifically the SMCDEL model checker, and fine-tuning to improve the ToM reasoning ability of LLMs. In our approach, an LLM is first fine-tuned through pairs of natural language and symbolic formulation representation of ToM problems and is then instructed to generate the symbolic formulation with a one-shot in-context example. The generated symbolic formulation is then executed by the SMCDEL model checker to perform transparent and verifiable ToM reasoning and give the final result. We demonstrate that our approach, ToM-LM, shows a significant improvement over all the constructed baselines. Our study proposes a novel view about externalizing a particular component of ToM reasoning, mainly reasoning about beliefs, and suggests generalizing it to other aspects of ToM reasoning.

Keywords: Large Language Models · Theory of Mind · Reasoning · Neuro-symbolic AI

1 Introduction

Theory of Mind (ToM) refers to an individual's ability to conceptualize and attribute mental states to others, therefore comprehending and representing how others perceive, interpret, and represent the surrounding world [16,2,17]. This ability is essential for individuals to understand and reason about others' beliefs, intentions, emotions, obligations, and so on [16,5,17]. For Large Language Models (LLMs), in most scenarios, they are required to interact with users or other agents and to understand their needs to give effective and appropriate responses. Therefore, discovering and enhancing their ToM reasoning ability is necessary and critical. Furthermore, it is worth noting that reasoning about knowledge has been a concern in symbolic philosophy and logic for many decades [4] and certainly in computer science to model security and game theory [1,6,7]. To a large extent, it is also the case that most of the machine learning models today have not really supported epistemic modeling and learning. Therefore, the idea

that capabilities classically dominated in symbolic computing could be used in the context of LLMs is interesting and presents intriguing possibilities.

Previous studies have demonstrated the emergence of ToM reasoning ability in LLMs [10,13,8,11]. Nevertheless, most of these studies adopt the ToM evaluation problems originally designed for assessing human ToM reasoning ability, which could be widely exposed to the internet, leading to possible data leakage to the training processes of LLMs, and because of the simplicity of these ToM problems, in order to study whether LLMs can handle more complex ToM reasoning ability, new benchmarks are needed [12,20]. Recent work has developed a new benchmark called MindGames [20] aiming to give a more complex ToM benchmark specifically designed for evaluating the ToM reasoning ability of LLMs. This benchmark also indicates that despite increases in model size, LLMs still struggle with complex ToM reasoning tasks [20].

To reiterate, although ToM is widely recognized and discussed as an inherent ability in LLMs, we propose a new perspective, externalizing and delegating ToM reasoning process to external symbolic executors. Inspired by Logic-LM, which enhances the logical reasoning abilities of LLMs by leveraging external executors that provide explainable and faithful reasoning processes [15], we adopt a similar approach to improve the ToM reasoning ability of LLMs. Specifically, we utilize an external symbolic executor, the SMCDEL model checker [22] designed for handling dynamic epistemic logic problems, to tackle complex ToM reasoning tasks delegated by LLMs, thereby aiming to improve the ToM reasoning ability of LLMs while making the ToM reasoning process transparent and verifiable.

Within this approach, an LLM is first given a ToM problem, and then it is asked to convert the problem represented in the natural language to the symbolic formulation. Subsequently, the generated symbolic formulation is then executed by the SMCDEL model checker to perform the ToM reasoning in a transparent and verifiable way and give out the final answer. However, our investigations reveal that this strategy alone does not adequately enhance the ToM reasoning ability of LLMs. Therefore, we employ fine-tuning[1] to further optimize this approach, aiming to substantially improve the ToM reasoning ability of LLMs. The overview of ToM-LM is shown in Figure 1.

Our study demonstrates that ToM-LM improves the ToM reasoning ability of LLMs significantly in terms of accuracy and Area under the ROC Curve (AUC), compared to all the constructed baselines. The contributions of this study are outlined as follows:

1. We introduce a new perspective to enhance ToM reasoning ability of LLMs, externalizing and delegating ToM reasoning process to external executors to make it transparent and explainable.
2. Our study also demonstrates that this approach makes substantial enhancements over the constructed baselines.

[1] We fine-tune the chosen LLM, *gpt-3.5-turbo*, through OpenAI Fine-tuning Platform with pairs of natural language and symbolic formulation. More details are discussed later in this paper.

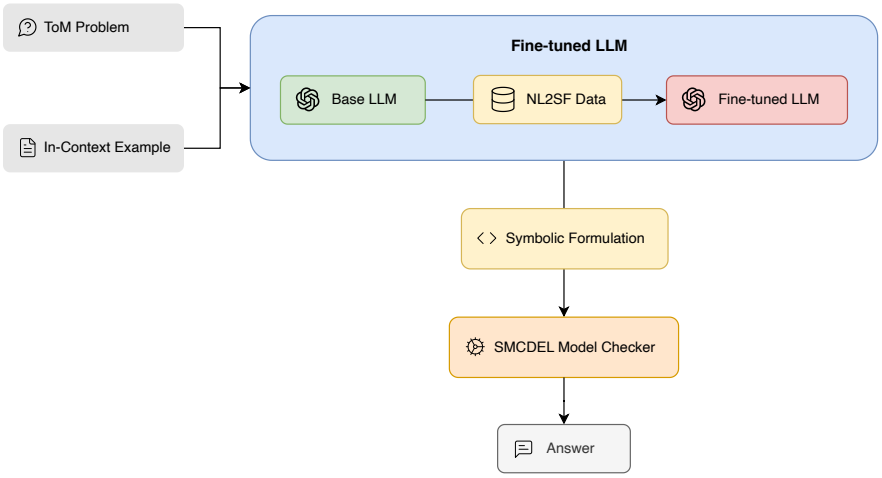

Fig. 1: Overview of ToM-LM, which consists of four principle stages: (1) *Model Fine-tuning* in which a given base model undergoes the fine-tuning process, (2) *Problem Prompting* in which a ToM problem and a one-shot example are constructed and given to the fine-tuned model, (3) *Problem Formalization* in which the model generates the corresponding symbolic formulation, and (4) *External Deterministic ToM Reasoning* in which the generated symbolic formulation is executed by SMCDEL model checker to give out the final result.

2 Related Work

In addition to the discussion in the previous section, we reiterate the following related works.

2.1 ToM In LLMs

Previous works have discussed more around whether ToM ability has emerged in LLMs [10,8,18,21,3]. Some works also try in various ways to improve the ToM reasoning ability of LLMs. As in [13], the various in-context learning methods are utilized such as *Let's think step by step* [9] and the chain-of-thought [23] to enhance the ToM reasoning ability of LLMs. Although this approach brings up an easy and flexible way to improve LLMs' ToM reasoning ability, we still hold the view that dynamic and complex ToM problems can not be simply represented as a chain-of-thought reasoning process and a more appropriate and robust way to handle them is needed. Thus, we propose an approach to use symbolic formulation to represent a general way to capture the ToM reasoning process and use external symbolic executors to perform the reasoning.

2.2 Delegating to External Executors

Our approach is significantly inspired by [19] which demonstrates delegating complex functionalities to external tools and by [15,14] which leverage symbolic formulation generation of LLMs and external symbolic executors to enhance their logical reasoning abilities. We follow a similar methodology and algorithmic strategy and hypothesize that it should be also suitable for the enhancement of complex ToM reasoning.

3 Methodology

3.1 ToM-LM

The overview of ToM-LM is shown in Figure 1. It consists of four principle stages, each of which is delineated in detail as follows.

Model Fine-tuning During this stage, we fine-tune the LLM, specifically the *gpt-3.5-turbo*, using our newly constructed *Symbolic Formulation Generation Prompting Fine-tuning Dataset*[2] which consists of pairs of ToM problems represented in both natural language and symbolic formulation. The fine-tuning was being processed on the OpenAI Fine-tuning Platform with 3 epochs. This process aims to improve the performance of the generation of symbolic formulations.

Problem Prompting Subsequently, a ToM problem and a one-shot in-context example are constructed as a prompt and given to the fine-tuned LLM to instruct it to give an appropriate symbolic formulation. Figure 2 shows the template used for prompting in which the texts within the braces serve as placeholders to be replaced with actual example or problem contents. The placeholders beginning with "example" indicate they are for the one-shot in-context example while the placeholders starting with "problem" mean they are for the ToM problem.

Problem Formalization The fine-tuned LLM is then expected to generate the symbolic formulation to appropriately represent the problem described in the natural language. An example symbolic formulation is illustrated in Figure 3. The generated symbolic formulations should conform to this structure and are anticipated to be executable.

External Deterministic ToM Reasoning Finally, the generated symbolic formulation is then executed by the external symbolic executor, specifically the SMCDEL model checker, to perform the ToM reasoning in a transparent and verifiable way and give out the final result.

[2] We explain *Symbolic Formulation Generation Prompting Fine-tuning Dataset* later in Section 3.2.

```
You should transform the natural language representation to the correct SMCDEL symbolic formulation. You should only output
the transformed symbolic formulation as shown in the example.
------
=== Example Begin ===
Natural Language Representation:
Context:
{example_context}
Hypothesis:
{example_hypothesis}

Symbolic Formulation:
{example_sf}
=== Example End ===
------
Natural Language Representation:
Context:
{problem_context}
Hypothesis:
{problem_hypothesis}

Symbolic Formulation:
```

Fig. 2: Symbolic Formulation Generation Prompting Template

3.2 Dataset

We select MindGames [20] as our evaluation and fine-tuning dataset. It is a dataset designed specifically for evaluating the ToM reasoning ability of the LLMs. Each item in the dataset includes a premise which describes the ToM problem situation and context, a hypothesis which describes a situation needing LLMs to judge its validity, a Boolean label which is the ground truth of the validity of the hypothesis, and a corresponding symbolic formulation to the premise and hypothesis.

For example, as illustrated in Figure 3, the premise describes a scenario involving four agents, each drawing a face unrevealed card, and someone may reveal their card to others. The hypothesis to be validated based on this given scenario is "Cara can now know whether Conrad picked a red card". The corresponding symbolic formulation representation of the premise and hypothesis is also given. To briefly explain[3], "VARS 1,2,3,4" defines four atomic propositions which represent the status of each agent, such as that "2" represents "Conrad picked a red card". "LAW Top" indicates that all possible Boolean states of these statuses should be considered. "OBS" simply denotes which agent observes which status given in "VARS", such as that "Agentc:1" means Cara observes the status that "Vasiliki picked a red card". The "VALID?" line queries an agent's belief about a certain status, in which "!" means a public announcement to all agents and "|"

[3] Refer to https://github.com/jrclogic/SMCDEL for more syntax details.

is a logical *OR*, such as that "!(1|2|3|4)" means "It is publicly announced that someone picked a red card". It represents and encapsulates the main part of the premise and hypothesis.

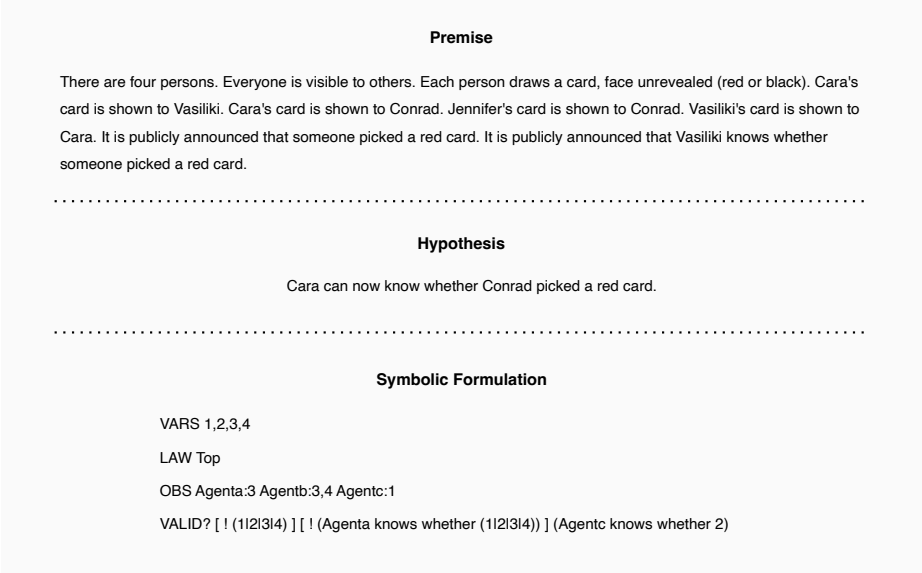

Fig. 3: An example of premise, hypothesis, and symbolic formulation in MindGames.

Each problem in MindGames can be categorized as either a true-belief or false-belief problem. LLMs are expected to handle them correctly to show their ToM reasoning ability.

We then randomly sample 200 items from the test set of MindGames dataset, as the evaluation dataset, to ensure an even distribution of labels. It means the ground truth labels in this sampled evaluation dataset are meticulously balanced to achieve an exact distribution of 50% *True* label and 50% *False* label.

Additionally, we randomly sample 150 items from the train set of MindGames dataset to form as the fine-tuning dataset. While maintaining the same data, the dataset is further preprocessed in two separate ways to suit different fine-tuning purposes: (1) *Direct Prompting Fine-tuning Dataset*, wherein each problem is given to the *Direct Prompting Template* as shown in Figure 4 to generate the appropriate prompt, and the corresponding target ground truth Boolean label is stringified as "TRUE" or "FALSE" and it is prepared specifically for the *Direct Prompting With Fine-tuning* baseline; (2) *Symbolic Formulation Generation Prompting Fine-tuning Dataset*, in which each problem is given to *Symbolic Formulation Generation Prompting Template* as shown in Figure 2 to generate the

appropriate prompt, and the corresponding target is not the Boolean label but the ground truth symbolic formulation, which is prepared for ToM-LM.

3.3 Baselines

We choose *gpt-3.5-turbo* as the base model and form three baselines to compare with our approach. The baselines are then differentiated as follows: (1) *Direct Prompting* or *Symbolic Formulation Generation Prompting*, and (2) whether or not Fine-tuning is applied. Our approach, ToM-LM, can be recognized as *Symbolic Formulation Generation Prompting with Fine-tuning*.

DP It stands for *Direct Prompting Without Fine-tuning*. In this setting, we directly ask the base model to answer the problem in "TRUE", "FALSE", or "I DON'T KNOW", with a one-shot in-context example provided. Answering in "TRUE" means it considers the hypothesis is valid in the given situation, "FALSE" means it thinks the hypothesis is invalid, and "I DON'T KNOW" means it has no idea about the validity of the hypothesis. The prompting template is illustrated in Figure 4.

```
You will be given a problem with a context and hypothesis. You should judge whether the hypothesis is "TRUE", "FALSE", or "I
DON'T KNOW" in the given context. "TRUE" means the hypothesis is true in the context, "FALSE" means the hypothesis is
false in the context, and "I DON'T KNOW" means you cannot determine the truth value of the hypothesis in the context. You
should provide your answer in the form of "TRUE", "FALSE", or "I DON'T KNOW".
------
Example:

Context:
{example_context}
Hypothesis:
{example_hypothesis}
Answer:
{example_answer}
------
Context:
{context}
Hypothesis:
{hypothesis}
Answer:
```

Fig. 4: Direct Prompting Template

SFGP It stands for *Symbolic Formulation Generation Prompting Without Fine-tuning*. In this setting, the workflow is exactly as same as ToM-LM but the model is not fine-tuned.

DP$_{FT}$ It stands for *Direct Prompting With Fine-tuning*. In this setting, the workflow is exactly as same as the DP baseline but the model is fine-tuned based on the *Direct Prompting Fine-tuning Dataset*.

4 Results

The responses given by DP, SFGP, DP$_{FT}$, and ToM-LM can be categorized and summarized into three types in general: (1) *TRUE*, (2) *FALSE*, or (3) *I DON'T KNOW*.

For *Direct Prompting* setting, *TRUE* simply means the LLM answers in "TRUE" literally, *FALSE* means the LLM answers in "FALSE" literally, and answering in "I DON'T KNOW" literally or other contents are considered as answering in *I DON'T KNOW*.

For *Symbolic Formulation Generation Prompting* setting, *TRUE* means the LLM generates an executable symbolic formulation and the model checker executes it to get *TRUE* value towards the problem, *FALSE* means the LLM generates an executable symbolic formulation and the model checker executes it to get *FALSE* value towards the problem, and *I DON'T KNOW* in this case means the LLM fails to generate an executable symbolic formulation.

4.1 Metrics

The evaluation metrics consist of three aspects, (1) Execution Rate, (2) Accuracy, and (3) Area under the ROC Curve (AUC). Table 1 shows all the three metrics for DP, SFGP, DP$_{FT}$, and ToM-LM.

Table 1: The Metrics of DP, SFGP, DP$_{FT}$, and ToM-LM

Approach	Execution Rate(%)	Accuracy(%)	AUC
DP	99.50	58.00	0.58
SFGP	78.00	49.00	0.60
DP$_{FT}$	**100**	76.00	0.76
ToM-LM	94.50	**91.00**	**0.94**

Execution Rate This metric is used to measure how successfully the LLM can generate available or executable answers. In the case of *Direct Prompting* setting, it is *TRUE* or *FALSE* and in the case of *Symbolic Formulation Generation Prompting* setting, it is executable symbolic formulation. A higher value of execution rate means higher available or executable answers generated by the LLM. But it does not mean the generated answers are correct.

The execution rate of baselines in *Direct Prompting* setting are almost around 100%, which means the LLM can generate available answers at most of time in this setting. Instead, in the case of *Symbolic Formulation Generation Prompting* setting, the execution rate is 78% without fine-tuning but can be high up to 94.5% in ToM-LM.

Accuracy As demonstrated in Table 1, ToM-LM reaches up to 91% accuracy which significantly surpasses all the baselines. It achieves the 33% improvement over the DP baseline, the 42% increases compared to the SFGP baseline, the 15% enhancement over the DP_{FT} baseline.

AUC Additionally, AUC is utilized to provide a more holistic indicator and assessment of the performance of ToM-LM and the baselines. As shown in Table 1, while the AUC for DP and SFGP are around 0.60, ToM-LM reaches up to the 0.94 AUC and outperforms all the baselines.

4.2 Output Distributions

We have also illustrated the distributions of outputs in Figure 5 to provide an intuitive visualization of the results and performance. Given that the evaluation set is evenly distributed across the ground truth labels, the distributions of outputs can intuitively reflect performance differences. While the output distributions for all baseline models exhibit significant bias, the output distribution of ToM-LM is notably more balanced.

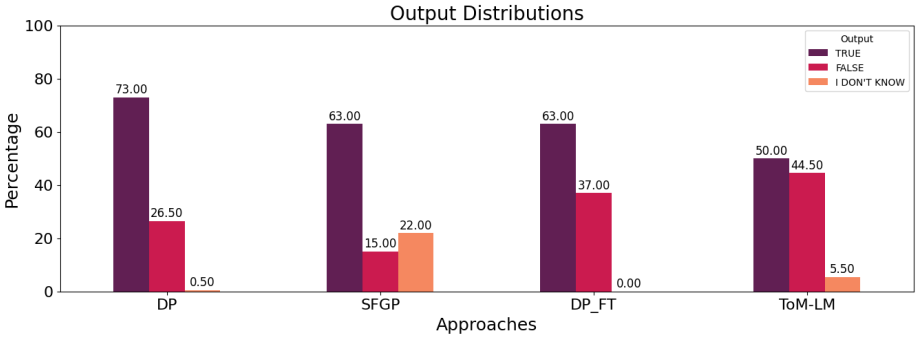

Fig. 5: Output Distributions of DP, SFGP, DP_{FT}, and ToM-LM

5 Discussion

5.1 Results Analysis

The results demonstrate that ToM-LM has successfully enhanced the ToM reasoning ability of LLMs, surpassing the other three baselines in terms of both accuracy, which reaches as high as 91%, and AUC which attains to 0.94.

From the results of the DP baseline, the accuracy is 58% and AUC is 0.58, meaning although the LLM itself may possess a modest degree of ToM reasoning ability, it still struggles with it and its answers are highly dependent on random guessing. From the SFGP baseline, although AUC is shown with a marginal improvement by 0.02 and reaches to 0.6, indicating that it modestly enhances the performance of the LLM against random guessing, the gain comes at the cost of reduced accuracy and a highly increased incidence of non-executable symbolic formulations.

As anticipated, the DP_{FT} baseline markedly surpasses both of the DP and SFGP baseline in terms of both accuracy and AUC. When comparing it to our approach, ToM-LM shows significant improvement over it with both of accuracy and AUC. In addition, within the same fine-tuning settings, we also observe that the training loss of the DP_{FT} baseline has a violent oscillation and does not converge at the end but the training loss of the ToM-LM has a very slighter oscillation and it converges at the end of fine-tuning.

Furthermore, the distributions of outputs indicate a significant bias among the three baselines, with all tending to respond *TRUE* to the given ToM problems. This pattern reveals their inability to accurately handle false-belief tasks, underscoring a deficiency in ToM reasoning ability. And as shown in Figure 5, it is noteworthy that even in the SFGP baseline, the LLM fails to generate appropriate executable symbolic formulations for false-belief tasks, while predominantly generating symbolic formulations for true-belief tasks instead. However, ToM-LM significantly improves the balance of results compared to all the three baselines.

5.2 Implications

Our study introduces a novel approach to externalize the ToM reasoning process to make it transparent and explainable and to significantly improve the ToM reasoning ability of LLMs in cases of true-belief and false-belief tasks. We hope it can potentially provide a new perspective on the feasibility of externalizing and delegating other aspects of ToM reasoning to external executors.

Furthermore, given that the essence of ToM involves understanding others in multi-agent contexts, this study is also anticipated to provide valuable insights for real-world applications, including multi-agent collaboration, autonomous driving, and human-machine interactions.

5.3 Limitations and Future Work

ToM could and is supposed to encompass a broad range of aspects, such as emotions, goals, intentions, desires, and beyond, besides the foundation of handling false-belief and true-belief. Due to the expressive limitations of the current external executor, our study focuses solely on a specific aspect of ToM, handling false-belief and true-belief. Nevertheless, the framework established here has the potential to be generalized to address various ToM-related tasks. Future research should explore additional components of ToM using external executors and may also aim to develop a more expressive and capable external executor or language to handle a diverse array of ToM tasks.

Moreover, our research primarily utilizes *gpt-3.5-turbo*. Future studies should investigate whether this approach is replicable with language models in smaller parameter sizes or explore it in more powerful models to determine if the approach can consistently and efficiently enhance ToM reasoning ability.

6 Conclusion

In this study, we propose a novel approach, ToM-LM, to externalize the ToM reasoning process of LLMs by employing the external symbolic executor, specifically the SMCDEL model checker. It aims to improve the ToM reasoning ability of LLMs while also making the ToM reasoning process transparent and explainable. Our approach demonstrates significant improvements over the three constructed baselines. We also anticipate that this research can offer new insights about improving other aspects of ToM reasoning ability of LLMs.

References

1. Ittai Abraham, Lorenzo Alvisi, and Joseph Y. Halpern. Distributed computing meets game theory: combining insights from two fields. *ACM SIGACT News*, 42(2):69–76, June 2011.
2. Simon Baron-Cohen, Alan M. Leslie, and Uta Frith. Does the autistic child have a "theory of mind"? *Cognition*, 21(1):37–46, October 1985.
3. Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.
4. Ronald Fagin, Joseph Y Halpern, Yoram Moses, and Moshe Vardi. *Reasoning about knowledge*. MIT press, 2004.
5. Chris D. Frith and Uta Frith. The neural basis of mentalizing. *Neuron*, 50(4):531–534, May 2006.
6. Joseph Y. Halpern. Substantive rationality and backward induction. *Games and Economic Behavior*, 37(2):425–435, November 2001.
7. Joseph Y. Halpern, Rafael Pass, and Vasumathi Raman. An epistemic characterization of zero knowledge. In *Proceedings of the 12th Conference on Theoretical Aspects of Rationality and Knowledge*, TARK '09, page 156–165, New York, NY, USA, July 2009. Association for Computing Machinery.

8. Mohsen Jamali, Ziv M Williams, and Jing Cai. Unveiling theory of mind in large language models: A parallel to single neurons in the human brain.
9. Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. (arXiv:2205.11916), January 2023. arXiv:2205.11916 [cs].
10. Michal Kosinski. Evaluating large language models in theory of mind tasks. 2023.
11. Huao Li, Yu Quan Chong, Simon Stepputtis, Joseph Campbell, Dana Hughes, Michael Lewis, and Katia Sycara. Theory of mind for multi-agent collaboration via large language models. (arXiv:2310.10701), October 2023. arXiv:2310.10701 [cs].
12. Ziqiao Ma, Jacob Sansom, Run Peng, and Joyce Chai. Towards a holistic landscape of situated theory of mind in large language models. (arXiv:2310.19619), October 2023. arXiv:2310.19619 [cs].
13. Shima Rahimi Moghaddam and Christopher J Honey. Boosting theory-of-mind performance in large language models via prompting.
14. Theo Olausson, Alex Gu, Ben Lipkin, Cedegao Zhang, Armando Solar-Lezama, Joshua Tenenbaum, and Roger Levy. Linc: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, page 5153–5176, Singapore, 2023. Association for Computational Linguistics.
15. Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. (arXiv:2305.12295), October 2023. arXiv:2305.12295 [cs].
16. David Premack and Guy Woodruff. Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences*, 1(4):515–526, December 1978.
17. François Quesque and Yves Rossetti. What do theory-of-mind tasks actually measure? theory and practice. *Perspectives on Psychological Science*, 15(2):384–396, March 2020.
18. Maarten Sap, Ronan LeBras, Daniel Fried, and Yejin Choi. Neural theory-of-mind? on the limits of social intelligence in large lms. (arXiv:2210.13312), April 2023. arXiv:2210.13312 [cs].
19. Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. (arXiv:2302.04761), February 2023. arXiv:2302.04761 [cs].
20. Damien Sileo and Antoine Lernould. Mindgames: Targeting theory of mind in large language models with dynamic epistemic modal logic. (arXiv:2305.03353), November 2023. arXiv:2305.03353 [cs].
21. Sean Trott, Cameron Jones, Tyler Chang, James Michaelov, and Benjamin Bergen. Do large language models know what humans know? *Cognitive Science*, 47(7):e13309, 2023.
22. Johan Van Benthem, Jan Van Eijck, Malvin Gattinger, and Kaile Su. Symbolic model checking for dynamic epistemic logic — s5 and beyond*. *Journal of Logic and Computation*, 28(2):367–402, March 2018.
23. Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. 2022.