Agentic-ToM: Cognition-Inspired Agentic Processing For Enhancing Theory of Mind Reasoning in Large Language Models

Sneheel Sarangi
NYU Abu Dhabi
sneheelsarangi@nyu.edu

Chetan Talele
Independent
chetan.labs@gmail.com

Hanan Salam
NYU Abu Dhabi
hanan.salam@nyu.edu

Abstract

The capacity to attribute mental states like beliefs, desires, and intentions to oneself and others, known as Theory of Mind (ToM), is fundamental to human social intelligence. As Large Language Models (LLMs) are increasingly integrated into complex interactive systems, developing their ToM capabilities is crucial. Such capabilities enable LLMs to understand and predict human behavior, leading to more intuitive and productive interactions. However, current models often struggle with sophisticated reasoning about others' perspectives. In this work, we propose *Agentic-ToM*, showing that guiding LLMs by embedding psychologically-grounded functions for capabilities such as 'perspective taking' and mental state tracking markedly improves their proficiency in ToM tasks. We evaluate the approach on three diverse ToM datasets and show that this method significantly outperforms baselines across all tasks without requiring task-specific modifications. Our code is publicly available.

1 Introduction

Theory of Mind (ToM), the capacity to attribute mental states such as beliefs, desires, and intentions to oneself and others, is fundamental to human social intelligence (Premack and Woodruff, 1978; Baron-Cohen, 1995). As Large Language Models (LLMs) are increasingly integrated into complex human-interactive systems (Lalwani et al., 2025; Elgarf et al., 2024; Lalwani and Salam, 2025; Lalwani et al., 2024), robust ToM capabilities are crucial for enabling them to interpret human intentions, predict behavior, and engage in productive interactions (Bubeck et al., 2023; Kosinski, 2023).

Despite advancements, LLMs' proficiency in sophisticated ToM reasoning remains contested (Ullman, 2023; Sap et al., 2022). While some models show near-human performance on specific benchmarks like ToMi (Le et al., 2019; Sclar et al., 2023), significant shortcomings persist, especially in tasks requiring deeper inference or generalization to complex scenarios (Kim et al., 2023; Wu et al., 2023). LLMs often struggle with tracking evolving mental states, higher-order ToM, and contextual cues (Ullman, 2023). This limitation is particularly salient as we move towards LLM-based agents, which will require ToM to operate effectively in human environments, unlike simple LLM chat interfaces. (Park et al., 2023; Wang et al., 2023). Addressing this gap is essential for aligning LLM reasoning with human social understanding.

Here, cognitive psychology offers valuable insights. Human ToM often involves deliberate, structured reasoning–referred to as "System 2" thinking (Kahneman, 2011), particularly in complex social situations, rather than solely relying on intuitive processes ("System 1" thinking) (Kahneman, 2011). This includes active "perspective taking": simulating the other's viewpoint and explicit mental state tracking (Flavell, 1992; Saxe et al., 2004). This human capacity for structured, conscious reasoning about mental states motivates our work: if humans benefit from deliberate ToM processing, LLMs might similarly benefit from guided, structured reasoning.

In this paper, we propose *Agentic-ToM*, a method for enhancing LLMs' ToM capabilities by framing them as agents and guiding their reasoning with psychology-inspired prompt functions, which we term *Cognitive Tools*. Drawing on concepts such as deliberate perspective taking and systematic mental-state tracking, these tools can be invoked by the LLM to explicitly analyze different agents' perspectives and maintain coherent representations of their knowledge and beliefs, thereby imposing a structured framework on their reasoning process. We evaluate our approach on three diverse ToM datasets spanning seven tasks and show that it consistently outperforms baseline methods. Additionally, our method demonstrates strong generalizability, yielding performance gains without

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 25645–25661
November 4-9, 2025 ©2025 Association for Computational Linguistics

requiring task-specific modifications to the core functions.

2 Related Work

Machine Theory of Mind. The development of computational systems exhibiting Theory of Mind (ToM) – the capacity to attribute and reason about mental states has been a persistent objective in artificial intelligence research. Contemporary LLMs have exhibited substantial improvements, with performance metrics on established ToM benchmarks like ToMi (Kosinski, 2023) and BigToM (Gandhi et al., 2023) approaching or exceeding human accuracy. Notwithstanding these advancements, the robustness of LLM-based ToM remains a subject of scrutiny (Ullman, 2023; Shapira et al., 2023). These concerns, alongside the saturation of existing benchmarks, have necessitated advancements in ToM evaluation methodologies. These include evaluations of higher-order ToM reasoning (e.g., iterated mental state attributions) (Wu et al., 2023), performance in naturalistic dialogue contexts (Kim et al., 2023), and the creation of comprehensive datasets for evaluating a wider spectrum of ToM-related abilities. Recent benchmarks, such as OpenToM (Xu et al., 2024), aim for more holistic assessments, for example, by evaluating LLMs' capabilities to understand mental states such as emotion or through diverse psychometrically-inspired tasks like faux-pas detection (Ma et al., 2023). This trend reflects a consensus that robust machine ToM requires proficiency beyond classical false-belief tasks, extending to a broader range of social-cognitive competencies.

LLM-based Agents. The emergence of LLM-based agents signifies a notable progression towards more sophisticated AI reasoning and autonomy (Xi et al., 2023; Wang et al., 2024). Such systems typically integrate LLMs into architectures that enable complex functionality such as external tool use (Yao et al., 2023; Mialon et al., 2023). Additionally, these also make multi-step reasoning based on possible feedback, which is integral to emulating "agency" (Park et al., 2023; Shinn et al., 2023). The structured reasoning capabilities inherent in agentic designs provide an interesting parallel to human cognition and ToM reasoning.

Augmenting ToM in LLMs. Recent research has proposed several distinct methodologies for enhancing the ToM capabilities of LLMs, primarily by introducing structured reasoning frameworks. SymbolicToM (Sclar et al., 2023) employs LLMs to generate a symbolic graph representation of characters' belief states before addressing ToM queries. SimToM (Wilf et al., 2024) inspired by Simulation Theory (Shanton and Goldman, 2010), implements a two-stage process involving explicit perspective-taking by the LLM. Similarly, Decompose-ToM (**?**) demonstrates that decomposing a complex ToM problem into a series of simpler, ToM-relevant subtasks can yield performance gains. However, these methods rely on external algorithmic control or predefined procedural frameworks to structure the LLM's inference process. While effective for specific ToM problem classes, this dependency on external scaffolding contrasts with the intention to foster autonomous ToM directly within the LLM's generative capabilities. Our work seeks to integrate similar structured reasoning principles through autonomously invokable, psychologically-inspired functions within an agentic LLM.

3 Agentic-ToM: Our Approach

Our approach leverages the structured output/function-calling mechanism of LLMs (Chen et al., 2024). We structure our functions as simple but specialized text prompts, which we refer to as *Cognitive Tools*. These are inspired by fundamental concepts in psychology (detailed in Section 3). The LLM autonomously decides which of these cognitive tools to invoke iteratively, including determining any necessary input arguments. Invoking a cognitive tool returns a specific prompt that directs the LLM's subsequent reasoning process.

Cognitive Tool Characteristics. Each tool within our framework comprises two components:

- **Description**: A concise explanation provided to the LLM, outlining the tool's function and its expected input arguments. For more complex tools, this description may include illustrative examples of correct and incorrect usage.
- **Prompt**: The text-based instructions returned to the LLM upon the tool's invocation. These instructions can be dynamically modified based on the input arguments provided during the call.

Tool names and their corresponding descriptions are supplied to the LLM via the system prompt. This information is also made available to the model during each API call that enables tool use (when a structured response parseable into a tool name and arguments is anticipated), consistent with

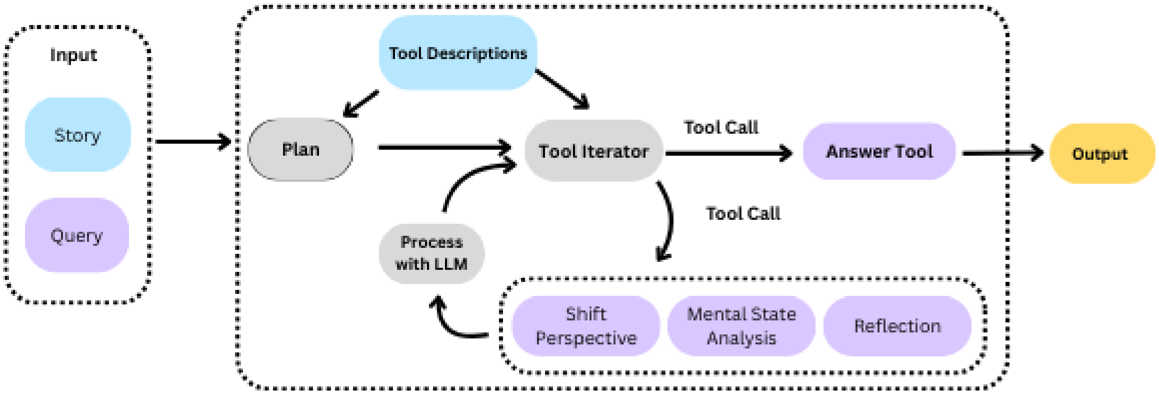

Figure 1: Workflow of Agentic-ToM.

formats such as the OpenAI API (OpenAI, 2023).

Cognitive Tools. To guide the LLM's reasoning in a manner that approximates human ToM processes, we handcraft three default tools: (1) Perspective-Taking, (2) Mental State Modeling, and (3) Reflection and Synthesis[1].

- **Perspective-Taking:** Inspired by Simulation Theory and the developmental psychology concept of "pretend-play"(Kavanaugh, 2006; Qu et al., 2015), similar to previous work by (Wilf et al., 2024; **?**), this module prompts the LLM to construct a "perspective story." That is, a given narrative is filtered to only include information a specified character would know or perceive. To compute nested perspectives (e.g., what character A thinks character B perceives), the LLM can makes sequential calls to this tool. For instance, to determine A's understanding of B's perspective, the LLM first generates A's perspective story (argument: [A]), and then, using that output, generates B's perspective as understood by A (argument: [A,B]).
- **Mental State Modeling:** Drawing from the classic belief-desire-intention (BDI) framework in cognitive science (Premack and Woodruff, 1978; Fodor, 1992), this module enables the LLM to explicitly model and track the evolution of a specific mental state (e.g., belief, desire, intention) for a designated character. The LLM invokes this tool by specifying the character and the mental state. The tool then prompts the LLM to analyze narrative events and detail how the character's mental state changes in response.
- **Reflection and Synthesis:** This module serves as a reasoning checkpoint. It prompts the LLM to synthesize information from prior operations, evaluate the plausibility of choices it is presented, and prune unlikely options. Such reflection on mental representations is key to robust ToM (Fonagy et al., 2002). The module then guides the LLM in strategically determining if, and which, subsequent tool calls are necessary to refine its understanding, mirroring deliberative processes in reasoning LLM agents (Shinn et al., 2023).

Operational Framework. The overall method for employing these tools can be delineated into two primary stages: (1) Planning, and (2) Iterative Tool Invocation (demonstrated in Figure 1).

1. ***Planning.*** Initially, the LLM is presented with a scenario, a query, and a "plan prompt." Within this stage, the LLM formulates a high-level strategy that outlines its intended approach to answering the question, including a proposed sequence of tool invocations. The LLM is prompted to consider the anticipated information state after each tool call to most effectively determine the optimal sequence of tools.
2. ***Iterative Tool Invocation.*** Following the planning phase, the model is permitted to invoke tools as deemed necessary within a loop. This iterative process continues until the LLM either calls a designated "answer" tool or reaches a predefined maximum number of allowed tool calls. Each step within this iterative invocation involves two sub-stages: first, the model invokes a tool based on its established plan and the provided tool descriptions, subsequently receiving the specialized tool prompt designed to focus its attention on a particular ToM-relevant reasoning process. Second, the LLM responds to this prompt, thereby executing the targeted reasoning. We discuss further efficiency improvements in Appendix A.

[1] The reader is referred to Appendix 6 for a detailed description of the prompts used.

	HiToM				FANTOM			OpenToM		
Model	1st	2nd	3rd	4th	AnsL	IAL	FB	Loc-FB	Mh-A	Mh-F
Method: Baseline										
Qwen-32B	71.7	47.5	32.5	37.5	46.0	32.0	39.0	59.0	58.4	60.8
Qwen-72B	65.0	50.0	30.0	25.0	50.6	7.6	45.0	64.0	57.4	59.8
GPT-4o-mini	55.1	40.8	29.2	26.7	48.7	25.3	23.7	67.5	62.8	64.7
GPT-4o	60.0	40.8	22.5	21.7	60.4	53.2	50.0	81.0	58.8	64.7
Method: CoT										
Qwen-32B	78.3	46.7	45.8	42.5	50.0	42.0	45.0	70.5	59.3	75.5
Qwen-72B	76.7	64.2	48.3	38.3	63.0	41.7	44.4	70.0	61.8	76.5
GPT-4o-mini	60.0	51.7	39.9	25.8	56.0	24.0	20.0	75.0	63.2	71.3
GPT-4o	74.2	53.3	45.0	42.5	59.6	50.0	47.7	81.0	61.6	75.5
Method: Agent Prompt										
Qwen-32B	77.5	50.8	32.5	35.8	55.3	38.0	33.3	67.5	77.9	87.3
Qwen-72B	80.8	70.8	50.8	37.5	54.0	48.7	38.0	68.0	75.5	81.3
GPT-4o-mini	70.0	61.7	42.5	34.2	65.3	52.7	43.3	63.5	74.4	84.3
GPT-4o	76.7	58.3	54.2	52.5	73.3	76.0	75.7	65.0	85.7	87.3
Method: Agentic-ToM										
Qwen-32B	71.4 (-6.9)	57.5 (+6.7)	53.3 (+7.5)	44.2 (+1.7)	52.0 (-3.3)	46.0 (+4.0)	50.6 (+5.6)	72.5 (+2.0)	77.2 (-0.7)	86.9 (-0.4)
Qwen-72B	76.3 (-4.5)	65.8 (-5.0)	57.5 (+6.7)	53.8 (+15.5)	67.0 (+4.0)	59.0 (+10.3)	58.6 (+13.6)	73.5 (+3.5)	71.8 (-3.7)	82.0 (+0.7)
GPT-4o-mini	63.2 (-6.8)	56.7 (-5.0)	50.8 (+8.3)	37.6 (+3.4)	58.6 (-6.7)	39.3 (-13.4)	46.7 (+3.4)	73.5 (-1.5)	70.4 (-4.0)	79.4 (-4.9)
GPT-4o	79.5 (+2.8)	74.3 (+16.0)	70.8 (+16.6)	68.3 (+15.8)	71.3 (-2.0)	77.3 (+1.3)	82.0 (+6.3)	82.0 (+1.0)	84.7 (-1.0)	87.3 (+0.0)
o4-mini	85.8	77.5	65.0	62.5	69.3	59.3	55.3	77.0	71.5	80.4

Table 1: Performance comparison across models and methods on the HiToM, FANToM, and OpenToM datasets in terms of accuracy (%), with gains over the best performing method among Baseline, CoT, and the agent prompt

4 Experiments and Results

4.1 Datasets and Tasks

We experiment by sampling examples from 3 datasets, covering 7 tasks. We provide detailed descriptions of the datasets, alongside our sampling procedure and example questions in Appendix C.

- From **HiToM (Wu et al., 2023)**, a dataset built using templatized generation, we sampled 480 story-question pairs for its false-belief task, which spans up to fourth-order ToM reasoning. Results for each order are presented independently.
- The **FANToM (Kim et al., 2023)** dataset contributed examples for three tasks: a false-belief task (300 questions), an Info-Accessibility task “Who in the story has access to this information?”(150 questions), and an Answerability task “Who can answer this question?” (150 questions).
- From **OpenToM (Xu et al., 2024)**, which contains LLM-generated long narratives based on false-belief scenarios, we sampled 100 questions for each of: first and second-order fine-grained location, first and second-order multihop-accessibility “How has accessibility of an object changed?”, and first-order multihop-fullness tasks “How has fullness of a container changed?”.

4.2 Evaluation

We experiment with four base LLMs, and evaluate each of them via:
(1) **Zero-shot prompting (Baseline)**: Model directly returns the answer;
(2) **Zero-shot chain of thought prompting**: Model "thinks step-by-step" before outputting the answer;
(3) **Ablation: Agent prompting**: Simplified version of our Agentic-ToM approach involving compressing the workflow in a single prompt;
(4) **Our approach (Agentic-ToM)**: Use the designed agentic framework to reason through the problem;
Each method is applied consistently to compare performance across models and datasets. We evaluate the open-source models Qwen-2.5-32B-Instruct and 72B-Instruct (Qwen, 2024a,b), alongside the closed source models GPT-4o, and GPT-4o-mini (OpenAI, 2024a,b). We share the model version details in E. We also present results for the state-of-the-art reasoning model o4-mini (OpenAI, 2025) to compare our approach against the current post-training methods.

4.3 Results and Discussion

The application of the Agentic-ToM methodology yields discernible performance enhancements across the evaluated tasks when compared to the stronger of the Baseline or Chain-of-Thought (CoT) methods. Additionally, using just the agent prompt also shows significant performance improvements beyond the baselines.

Performance across Tasks. Comparing Agentic-ToM to the CoT and Baseline methods, the most substantial improvements are observed for the FANTOM-FB (Overall) task, which registered an average increase of 18.55 percentage points. Following this, notable gains were also recorded for FANTOM-IAL (+14.85 percentage points), OpenToM-Mh-A (+14.55 percentage points), and higher-order HiToM tasks (4th order: +13.48 percentage points; 3rd order: +13.35 percentage points). Conversely, the HiToM 1st-order task exhibited the most modest improvement, with an average gain of 0.3 percentage points. As such, the most significant gains came through the tasks that required higher levels of reasoning. The simpler Agent prompt ablation also performs significantly better than the baselines across tasks

Performance across Model Size. Analysis of performance gains relative to model scale indicates that more capable models tend to derive greater benefit from the Agentic-ToM approach. Specifically, Qwen-72B achieved an average improvement of 7.98 percentage points over its best alternative baseline method, surpassing the 5.52 percentage point gain observed for Qwen-32B. A more pronounced trend is evident with the GPT series, where GPT-4o demonstrated an average uplift of 18.08 percentage points, substantially exceeding the 8.34 percentage point improvement seen with GPT-4o-mini.

Impact of the Agentic Approach The Agent Prompt method, a simpler ablation, provides a significant boost on its own, demonstrating the inherent value of a basic agentic framing, and providing a simple yet strong baseline for future ToM datasets. For example, on the OpenToM-Mh-A task, the Agent Prompt method was often the top performer, with GPT-4o reaching a score of 85.7, slightly ahead of the 84.7 achieved with the more complex Agentic-ToM.

However, the full Agentic-ToM methodology consistently delivers the most substantial gains, particularly on the most challenging tasks that require deep, iterative reasoning. On all FanToM False-Belief tasks and the higher-order HiToM tasks (3rd and 4th order), the Agentic-ToM method consistently outperforms the simpler Agent Prompt across all models. For instance, with GPT-4o, the Agentic-ToM method outperforms the Agent prompt by >15% on the 2nd, 3rd, and 4th order HiToM tasks. Thus, while a simple agent prompt is effective, the more structured and comprehensive reasoning process of the Agentic-ToM framework is crucial for more complex reasoning.

Comparison to Post-Training approaches While o4-mini shows strong performance on lower-order HiToM questions, Agentic-ToM used with GPT-4o outperforms it across all other tasks. In general, LLM post-training remains a viable approach to instill ToM in LLMs. Recent work has shown that LLMs' ToM capability can be drastically improved by Reinforcement Learning based post-training (Lu et al., 2025). However, the generalizability of this approach is contested (Sarangi and Salam, 2025). The Agentic-ToM approach can likely help guide reward design and provide a baseline reasoning structure to motivate further work in the domain.

5 Conclusion

This paper introduces Agentic-ToM, a novel and generalizable method that enhances LLMs' ToM by integrating psychologically inspired Cognitive Tools. These handcrafted functions enable LLMs to autonomously engage in structured reasoning, such as perspective-taking and mental state tracking, during inference. Our comprehensive evaluation across multiple diverse ToM tasks demonstrates that Agentic-ToM consistently and significantly outperforms existing baselines, with more capable LLMs showing the most pronounced improvements, all without requiring task-specific modifications. We also show that an ablation of our methodology involving a single prompt encompassing our planning and tool prompts also significantly outperforms baselines, but fails at tasks requiring more complex reasoning. Our work represents a step towards LLMs that are autonomously capable of ToM, and exhibit higher social intelligence by extension.

6 Limitations

Our method demands considerably more computational resources compared to baseline approaches, primarily because the agents maintain a working memory that allows them to process and repeatedly re-process information as they reason. This added memory footprint and iterative reasoning cycle introduce a meaningful trade-off between cognitive depth and efficiency. Moreover, although ToM encompasses a wide and diverse range of tasks, the current landscape of evaluation benchmarks is still evolving. As such, it remains an open question how well our method generalizes to more challenging forms of ToM, particularly those that involve deeper reasoning chains or are embedded in dynamic, real-world contexts.

Acknowledgements

This work is supported in part by the NYUAD Center for Artificial Intelligence and Robotics, funded by Tamkeen under the NYUAD Research Institute Award CG010 and in part by the NYUAD Center for Interdisciplinary Data Science & AI (CIDSAI), funded by Tamkeen under the NYUAD Research Institute Award CG016

References

Simon Baron-Cohen. 1995. *Mindblindness: An Essay on Autism and Theory of Mind*. The MIT Press.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *Preprint*, arXiv:2303.12712.

Yi-Chang Chen, Po-Chun Hsu, Chan-Jan Hsu, and Da shan Shiu. 2024. Enhancing function-calling capabilities in llms: Strategies for prompt formats, data integration, and multilingual translation. *Preprint*, arXiv:2412.01130.

Maha Elgarf, Hanan Salam, and Christopher Peters. 2024. Fostering children's creativity through llm-driven storytelling with a social robot. *Frontiers in Robotics and AI*, 11:1457429.

John H. Flavell. 1992. Perspectives on perspective taking. In *Piaget's Theory*, 1st edition, page 33. Psychology Press.

Jerry A. Fodor. 1992. A theory of the child's theory of mind. *Cognition*, 44(3):283–296.

Peter Fonagy, György Gergely, and Elliot L. Jurist, editors. 2002. *Affect Regulation, Mentalization and the Development of the Self*, 1st edition. Routledge.

Kanishk Gandhi, Jan-Philipp Fränken, Tobias Gerstenberg, and Noah D. Goodman. 2023. Understanding social reasoning in language models with language models. *Preprint*, arXiv:2306.15448.

Daniel Kahneman. 2011. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, New York.

R D Kavanaugh. 2006. Pretend play and theory of mind. In L S Balter C, editor, *Child psychology: A handbook of contemporary issues*, pages 153–166. Psychology Press.

Hyunwoo Kim, Melanie Sclar, Xuhui Zhou, Ronan Bras, Gunhee Kim, Yejin Choi, and Maarten Sap. 2023. FANToM: A benchmark for stress-testing machine theory of mind in interactions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14397–14413, Singapore. Association for Computational Linguistics.

Michal Kosinski. 2023. Theory of mind may have spontaneously emerged in large language models. *Preprint*, arXiv:2302.02083.

Himanshi Lalwani, Maha Elgarf, and Hanan Salam. 2024. Productivity coachbot: a social robot coach for university students with adhd. In *A3DE, ACM/IEEE International Conference on Human-Robot Interaction*. IEEE.

Himanshi Lalwani and Hanan Salam. 2025. Supporting productivity skill development in college students through social robot coaching: A proof-of-concept. In *Ro-MAN*. IEEE.

Himanshi Lalwani, Mira Saleh, and Hanan Salam. 2025. A study companion for productivity: exploring the role of a social robot for college students with adhd. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 1438–1442. IEEE.

Matthew Le, Y-Lan Boureau, and Maximilian Nickel. 2019. 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)*, pages 5872–5877, Hong Kong, China. Association for Computational Linguistics.

Yi-Long Lu, Chunhui Zhang, Jiajun Song, Lifeng Fan, and Wei Wang. 2025. Do theory of mind benchmarks need explicit human-like reasoning in language models? *Preprint*, arXiv:2504.01698.

Ziqiao Ma, Jacob Sansom, Run Peng, and Joyce Chai. 2023. Towards a holistic landscape of situated theory of mind in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1011–1031, Singapore. Association for Computational Linguistics.

Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. 2023. Augmented language models: a survey. *Preprint*, arXiv:2302.07842.

OpenAI. 2023. *Function calling*. OpenAI Platform. OpenAI Platform Documentation.

OpenAI. 2024a. Gpt-4o. `https://openai.com/index/hello-gpt-4o/`. OpenAI's flagship multimodal model capable of processing and generating text, images, and audio in real-time.

OpenAI. 2024b. Gpt-4o-mini. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/. A smaller, faster, and more efficient version of GPT-4o, designed to deliver quick responses for a wide range of everyday tasks.

OpenAI. 2025. Openai o3 and o4-mini system card. Technical report, OpenAI.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. *Preprint*, arXiv:2304.03442.

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

Li Qu, Pinxiu Shen, Yu Yan Chee, and Luxi Chen. 2015. Teachers' theory-of-mind coaching and children's executive function predict the training effect of sociodramatic play on children's theory of mind. *Social Development*, 24(4):716–733.

Qwen. 2024a. Qwen2.5-32b-instruct. https://huggingface.co/Qwen/Qwen2.5-32B-Instruct. An instruction-tuned 32 billion parameter language model from the Qwen2.5 series, designed for enhanced instruction-following capabilities.

Qwen. 2024b. Qwen2.5-70b-instruct. https://deepinfra.com/Qwen/Qwen2.5-72B-Instruct. A 72 billion parameter instruction-tuned model from the Qwen2.5 series, offering improvements in knowledge, coding, mathematics, and instruction following.

Maarten Sap, Ronan Le Bras, Daniel Fried, and Yejin Choi. 2022. Neural theory-of-mind? on the limits of social intelligence in large LMs. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3762–3780, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Sneheel Sarangi and Hanan Salam. 2025. Small llms do not learn a generalizable theory of mind via reinforcement learning. *Preprint*, arXiv:2507.15788.

Rebecca Saxe, Susan Carey, and N Kanwisher. 2004. Understanding other minds: Linking developmental psychology and functional neuroimaging. *Annual review of psychology*, 55:87–124.

Melanie Sclar, Sachin Kumar, Peter West, Alane Suhr, Yejin Choi, and Yulia Tsvetkov. 2023. Minding language models' (lack of) theory of mind: A plug-and-play multi-character belief tracker. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13960–13980, Toronto, Canada. Association for Computational Linguistics.

Karen Shanton and Alvin Goldman. 2010. Simulation theory. *WIREs Cognitive Science*, 1(4):527–538.

Natalie Shapira, Mosh Levy, Seyed Hossein Alavi, Xuhui Zhou, Yejin Choi, Yoav Goldberg, Maarten Sap, and Vered Shwartz. 2023. Clever hans or neural theory of mind? stress testing social reasoning in large language models. *Preprint*, arXiv:2305.14763.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Preprint*, arXiv:2303.11366.

Tomer Ullman. 2023. Large language models fail on trivial alterations to theory-of-mind tasks. *Preprint*, arXiv:2302.08399.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An open-ended embodied agent with large language models. *Preprint*, arXiv:2305.16291.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6).

Alex Wilf, Sihyun Lee, Paul Pu Liang, and Louis-Philippe Morency. 2024. Think twice: Perspective-taking improves large language models' theory-of-mind capabilities. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8292–8308, Bangkok, Thailand. Association for Computational Linguistics.

Yufan Wu, Yinghui He, Yilin Jia, Rada Mihalcea, Yulong Chen, and Naihao Deng. 2023. Hi-ToM: A benchmark for evaluating higher-order theory of mind reasoning in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10691–10706, Singapore. Association for Computational Linguistics.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, Zhangyue Yin, Shihan Dou, Rongxiang Weng, Wensen Cheng, Qi Zhang, Wenjuan Qin, Yongyan Zheng, Xipeng Qiu, Xuanjing Huang, and Tao Gui. 2023. The rise and potential of large language model based agents: A survey. *Preprint*, arXiv:2309.07864.

Hainiu Xu, Runcong Zhao, Lixing Zhu, Jinhua Du, and Yulan He. 2024. OpenToM: A comprehensive benchmark for evaluating theory-of-mind reasoning capabilities of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8593–8623, Bangkok, Thailand. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. *Preprint*, arXiv:2210.03629.

Note: We have used AI tools for editing some of the content here.

A Using an LLM as a submodule for efficiency

We note that our method is computationally intensive, especially due to the redundant concatenation of identical tool prompts upon multiple invocations of the same tool in different contexts. As such, to improve our system's efficiency, we use an LLM call independent of the broader agent history (such as the planning step and previous tool calls) to conduct the intended reasoning step. To facilitate this, we store the results of prior tool calls (such as previously computed perspective stories) and pass only relevant context to the LLM call. We empirically observe that this modification results in no change in overall performance.

B Analyzing Failure Cases

We observe a couple of common failure modes in our method that result in most of the errors:

a) Under-Reasoning at the Tool Solving Level: LLMs of all sizes often fail to execute the base reasoning steps as instructed by the tools. For example, when asked to simulate the story from a character's point of view. The LLM might include events that the character does not know, or exclude events that the character does know. Given that each step in our method builds upon the previous step, even minor errors can snowball and cause a wrong answer.

b) Reasoning confusion before final answer: While largely mitigated by prompt engineering, models sometimes still fail to arrive at the correct answer after undertaking all the tool-based reasoning steps correctly. This happens because the LLM might fail to pay attention to the reasoning conducted in the tool steps and attempt to provide a straightforward answer.

We believe that with further prompt engineering, these issues can be mitigated further.

C Tool Prompts and Descriptions

C.1 Perspective Shift Prompts

C.1.1 Direct shift to perspective

Perspective Shift Prompt

You'll now simulate the perspective of `{characters[0]}` by drawing from the original story. From the given story, directly output ONLY the events that the specified character, `{characters[0]}`, knows about. **Knowledge of event is determined by commonsense assumptions.** For example, an agent likely won't know about occurrences in a room after leaving it. On the other hand, they will likely know of events in which they participate or which they witness. In case it is unclear whether the character knows about an event, choose to include it. **Example:** if original story:

1. Alice and Jack enter the red_room.
2. The hat is in box.
3. Alice leaves the red_room.
4. Jack moves the hat to the basket.
5. Alice enters the blue_room

From the perspective of Alice:

1. Alice and Jack enter the red_room.
2. The hat is in box.
3. Alice leaves the red_room.
4. Alice enters the blue_room

First, provide brief reasoning about which events `{characters[0]}` doesn't know. **Answer Format :**

- Reasoning
- **[Story from `{characters[0]}`'s perspective:]**
  - 1.Full Event/Dialogue - {characters[0]}'s reason to know event 1 in <5 words))
  - 2.Full Event/Dialogue - {characters[0]}'s reason to know event 2 in <5 words))
  - ....

C.1.2 Nested shift to perspective

Perspective Shift Prompt

You'll now simulate `{result}`'s perspective of what `{characters[-1]}` believes/knows, by conducting the following procedure: **Procedure:**

- Consider the story from `{characters[-2]}`'s perspective as deduced earlier (tagged previously as **[Story from `{characters[-2]}`'s perspective:]**).
- From this story, select and output ONLY the events that `{characters[-1]}` knows about.

Knowledge of events is determined by commonsense assumptions:

- An agent likely won't know about occurrences in a room after leaving it.
- They will likely know of events in which they participate or which happen when they are in the room.
- **IMPORTANT: Do not add any events that are not present in [Story from `{characters[-2]}`'s perspective:]**.
- As a double check, make sure that all of `{characters[:-1]}` know the printed event too.
- In case it is unclear whether the character knows about an event, choose to include it.

Example:

- Original story:
  1. Alice and Jack enter the red_room.
  2. The hat is in the box.
  3. Jack leaves the red_room.
  4. Alice moves the hat to the basket.
- From the perspective of Jack:
  1. Alice and Jack enter the red_room.
  2. The hat is in the box.
  3. Jack leaves the red_room.
- Simulating Jack's perspective of Alice's perspective:
  1. Alice and Jack enter the red_room.
  2. The hat is in the box.
  3. Jack leaves the red_room.

Answer Format:

- (Brief reasoning about which events from `{result}`'s perspective `{characters[-1]}` will not know about. Specifically consider the simplified and previously computed **[Story from `{characters[-2]}`'s perspective:]**. Since this perspective is being computed recursively from `{result}`'s perspective, make sure that each event listed is also known by `{characters[:-1]}`.)

- **[From `{result}`'s perspective:] [Story from `{characters[-1]}`'s perspective:]**
  1. Full Event/Dialogue - (`{characters[-1]}`'s reason to know event 1 in <5 words, any relevant change to state (eg. exited/left))
  2. Full Event/Dialogue - (`{characters[-1]}`'s reason to know event 2 in <5 words, any relevant change to state (eg. exited/left))
  3. . . .

C.2 Instructions for Mental State Analysis

Direct Mental State Analysis

Task: For each important actor in the scenario, track their mental states (such as **beliefs, desires, and intentions**), including any changes over time. **Guidelines:**

- Note key events that clarify ambiguous mental states or cause shifts in existing states.
- Specifically highlight an event if it is key to answering the question (e.g., if it is referenced in the question).
- Examine each statement in the scenario carefully with these considerations in mind.

Answer Format:

1. **Initial State:** Describe each actor's initial mental states.
2. **Event-by-Event Analysis:** For each significant event, explain how it influences the actors' mental states.

C.3 Instructions for Reflection

Choice Reflection

Task: Re-state the query, review the steps previously taken, and combine the outputs of the modules to reason towards an answer to the query. **Guidelines:**

- Re-examine the scenario to identify potential details that might have been overlooked.
- Consider if more information is necessary.
- Consider all possible options to answer with. Eliminate any options that are not possible, and then consider the remaining options to make your decision.

Answer Format:

1. **Re-state the question precisely.**
2. **Review:** Reason to combine outputs from previous steps towards an answer.
3. **Re-Evaluate:** Consider any details from the scenario that might have been overlooked in current reasoning.
4. **Option Elimination:** Consider all possible options that you can answer with. Eliminate any options that are not possible.
5. **Remaining Options:** Evaluate remaining options. *Note: Do not make decisions based solely on unstated or future assumptions; re-evaluate evidence thoroughly.*
6. **Further Action/Most Likely Answer:** If an answer isn't clear, consider if more tool calls need to be made to clarify the answer, and if so, which ones. If not, reason to provide the most likely answer/s.

C.4 Planning

Problem Analysis and Plan generation

Here's the scenario and question: **{prompt}**
Initial Analysis:

- Here's the scenario and question: prompt
- As a first step, identify the **type of question**/what **ability/reasoning** the question will need to be answered.
- Carefully look at all **tool descriptions**, **correct/incorrect ways to use the tools**, and **example use**, to decide tool calls and input arguments. Summarize example usage/important instructions for tools as mentioned in the tool guide. Then reason clearly about how to solve the problem, following the guidelines.
- Then come up with a **multi-step plan** to solve the question using the tools provided strictly following the tool description guidelines.
- Remember, you can use a tool multiple times and chain them together sequentially (i.e., use a tool, get a response, then use the same tool again, or use another tool). After each planned step, consider what information you are likely to have af-

ter conducting the reasoning step. You always have information from all the previously conducted steps. Analyze the question carefully (if necessary, consider potential ambiguities), and make sure of the answer before providing it.

Follow the format:

- (Tool use guide with exact examples and details on correct/incorrect ways to pass parameters)
- (Initial Reasoning)
- (Step 1) : (Expected Information State after Step 1)
- (Step 2) : (Expected Information State after Step 2)
- ...

C.5 Tool Description Prompts:

shift_perspective description

Description:

- Simulates the perspective of a specified character by simplifying the story to events only known by that character. Returns a perspective story.
- For multi-character input (nested perspectives), it takes a previously computed perspective story for character n-1 and simulates events known to the n'th character.
- **Nested perspectives must be built sequentially from OUTERMOST to INNERMOST!**
- **Example:** To simulate Character 1's perspective of Character 2's perspective of Character 3's perspective (or what Char1 thinks of what Char2 thinks of what Char3 thinks):
  - **FIRST,** call with **characters = [char1]** (Char1's perspective (outermost))
  - **THEN,** call with **characters = [char1, char2]** (Char1's perspective of Char2's perspective)
  - **THEN,** call with **characters = [char1, char2, char3]** (Full nested perspective)
- **Rule:** Never call **[char1, char2, char3]** without first calling **[char1]** and **[char1, char2]**. Build perspectives in order to maintain accuracy.
- **CORRECT ORDER:** char1 → char1, char2 → char1, char2, char3
- **INCORRECT ORDER (do not do this!):** char3 → char2, char3 → char1, char2, char3
- **OPTIONAL:** If the goal is to simulate the story only up to a specific event (such as when a question refers to a particular moment), provide that event as the **simulate_till** argument. Be sure to verify that the question and its options do not involve events occurring after this point. If they do, do not call this function.

Parameters:

- **characters** (array of strings): Names of characters in the sequence of being simulated. E.g., **[char1, char2, char3]** is the simulation of char1's perspective of char2's perspective of char3's perspective.
- **simulate_till** (string or null): The event up to which the story should be simulated.

mental_state_analysis description

Description:

- **description:** "Explicitly track and analyzed the specified mental states of a given agent. Useful to conduct an agent-specific analysis. Provide name of character to be analyzed and mental state to be analyzed. Use perspective_shift instead if nested character perspectives are required or if a complete set of events known to a character is needed."
- **parameters:**
  - **type:** "object"
  - **properties:**
    - **character:**
      - **type:** "string"
      - **description:** "Name of character to track"
    - **mental_states:**
      - **type:** "string"
      - **description:** "Mental states to track and analyze. Eg. Emotion, Desire, Intention, Knowledge, Beliefs. Be specific as to what exactly to track (eg. Knowledge of spoken dialogue A or Desire to do B). Multiple mental states can be provided if necessary."

choice_reflection description

function:

- **name:** "choice_reflection"
- **description:** "Provides instructions on how to combine steps taken till this point, to reason towards a solution, to eliminate impossible or unlikely options, or to decide to get more information by calling more tools. Consider using this tool before the answer module to ensure a more accurate response."
- **parameters:**
  - **type:** "object"
  - **properties:**

D More on Datasets

D.1 Hi-ToM:

Hi-ToM (Wu et al., 2023). Hi-ToM is designed to evaluate the higher-order theory of mind abilities of language models, going up to the fourth-order theory of mind reasoning tasks. It is based on the Sally-Anne test (**?**) with characters entering, leaving, and moving items in rooms. The provided task is a multiple-choice question-answering task with 15 choices per question. The dataset consists of 600 questions spanning two categories: "Tell" and "no-Tell". The ”Tell” category adds an element of deception by asking LLMs to reason whether an agent is lying by inferring when they left the room. We present our results averaged over both categories. We exclude order 0 questions from our analysis.

D.1.1 Order 4 Hi-ToM Example:

Order 4 Hi-ToM Example:

Original Story:

1. Chloe, Nathan, Evelyn, Jacob and Lily entered the patio.
2. The strawberry is in the red_box.
3. Chloe made no movements and stayed in the patio for 1 minute.
4. Chloe exited the patio.
5. Nathan moved the strawberry to the blue_bottle.
6. Nathan exited the patio.
7. Evelyn made no movements and stayed in the patio for 1 minute.
8. Lily lost his gloves.
9. Evelyn exited the patio.
10. Jacob made no movements and stayed in the patio for 1 minute.
11. Jacob exited the patio.
12. Lily moved the strawberry to the red_box.
13. Lily exited the patio.
14. Chloe, Nathan, Evelyn, Jacob and Lily entered the waiting_room.
15. Evelyn privately told Chloe that the strawberry is in the blue_crate now.

Question: Where does Nathan think Jacob thinks Evelyn thinks Lily thinks the strawberry is?

Answer: blue_bottle

D.2 FANToM:

FANToM (Kim et al., 2023). The FANToM dataset consists of dialogue-based interactions between characters. Characters may leave or enter the conversation at any time, and questions are based on correctly navigating this information incompleteness. This provides a naturalistic setting to test our model on. We use the answerability-list and knowledge awareness-list tasks for our evaluations due to their challenging nature, which requires multi-step reasoning. Additionally, we also include the false-belief task. The dataset consists of 1,540 questions, containing first and second-order ToM questions. We show averaged results over these categories due to generally similar accuracy gains for models across evaluation methods. We use the long-length variant of the dataset for its higher toughness.

D.2.1 FANToM Example:

Belief about Shared Experiences Prompt

Short Context: Gianna: Guys, I've really enjoyed sharing our pet stories, but I need to excuse myself. I need to change clothes for a meeting later. Talk to you later! Sara: Sure thing, Gianna. Take care! Javier: Catch you later, Gianna. Sara: So Javier, have you ever tried training Bruno? Javier: Yes, I did actually. It was a challenge at times, but rewarding nevertheless. How about you? Did you try training Snowflake? Sara: Oh gosh, trying to train a cat is a whole different ball game. But I did manage to teach her a few commands and tricks.

She was quite an intelligent little furball. Gianna: Hey guys, I'm back, couldn't miss out on more pet stories. Speaking of teaching and training pets, it is amazing how that further strengthens the bond between us and our pets, right? Sara: Absolutely, Gianna! The fact that they trust us enough to learn from us is really special. Javier: I can't agree more. I believe that's one of the ways Bruno conveyed his love and trust towards me. It also gave me a sense of responsibility towards him. Gianna: Just like Chirpy. Once she began to imitate me, we connected in a way I never imagined. She would repeat words that I was studying for exams and that somehow made studying less stressful. Javier: Pets are indeed lifesavers in so many ways. Sara: They bring so much joy and laughter too into our lives. I mean, imagine a little kitten stuck in a vase! I couldn't have asked for a better stress buster during my college days. Gianna: Totally, they all are so amazing in their unique ways. It's so nice to have these memories to look back on.
Question: What does Gianna believe about who discussed their experiences training their pets, Bruno and Snowflake?
Question Type: tom:belief:inaccessible
ToM Type: first-order
Correct Answer: Gianna knows that Javier discussed training his pet, Bruno. However, Gianna will not know training a pet named Snowflake.
Wrong Answer: Gianna believes that Sara and Javier discussed their experiences training their pets, Bruno and Snowflake.
Missed Info Accessibility: inaccessible

Question Type: answerability_list
Target: Who discussed their experiences training their pets, Bruno and Snowflake?
Question: List all the characters who know the precise correct answer to this question.
Correct Answer: [Javier, Sara]
Wrong Answer: [Gianna, Alondra, Angela]
Missed Info Accessibility: inaccessible

Question Type: info_accessibility_list
Information: Who discussed their experiences training their pets, Bruno and Snowflake? Sara and Javier discussed their experiences training their pets, Bruno and Snowflake.
Question: List all the characters who know this information.
Correct Answer: [Javier, Sara]
Wrong Answer: [Gianna, Alondra, Angela]
Missed Info Accessibility: inaccessible

D.3 OpenToM:

OpenToM (Xu et al., 2024). The OpenToM dataset consists of LLM-written stories based on the Sally-Anne false belief task. The dataset provides questions for the course-grained location, fine-grained location, multihop-fullness, multihop-accessibility tasks, each for both the first and second order, and an attitude task. The dataset also consists of an extension with longer narratives. We use the longer variant of OpenToM, and sample 100 questions each for the first-order and second-order variants of the fine-grained location and multihop accessibility tasks, alongside the first-order variant for the multihop fullness task. Additionally, while the original dataset does not have an equal distribution of labels, the reason behind using F1 scores, we sample equal distributions for each label and use an accuracy score metric. We exclude the Attitude task and the second-order multihop fullness task from our analysis due to the presence of ambiguities/mislabeled samples.

D.3.1 OpenToM Example:

Attitude towards Observed Action Prompt

Plot: Diego entered the patio. Amir entered the patio. Both Diego and Amir noticed that the scarf is in the basket in the patio. Diego moved the scarf to the a donation bin and Amir witnessed the action.
Plot Info:
- **Mover:** Diego
- **EOI:** scarf
- **Original Place:** basket
- **Move To Place:** a donation bin
- **Observer:** Amir

Preferences:

- **Mover:** Diego hates scarf.
- **Observer:** Amir likes scarf.
- **Mover-Observer:**
- **Observer-Mover:** Amir mistakenly think that Diego likes scarf.

Personality: Diego is an inconsiderate person.
Sentiment Statement: Diego hates scarf.
True Sentiment: the most negative
Intention: get rid of it and make it accessible to someone who may actually want it
New Location: a donation bin
Observed: true
Narrative: Diego and Amir were both residents of the same apartment complex. They had known each other for quite some time, but they couldn't be more different in their tastes and preferences. One thing that particularly divided them was their opinion on scarves. Diego despised scarves, finding them to be unnecessary and bothersome. On the other hand, Amir adored scarves, always wearing one to complete his outfit.
One sunny afternoon, both Diego and Amir happened to stroll into the patio at the same time. As they approached the central basket, their eyes fell upon a colorful scarf lying inside. Diego's face contorted in disdain while Amir's eyes lit up with delight.
In that moment, without exchanging any words, Diego swiftly reached into the basket and snatched the scarf. Amir watched curiously as Diego took a few steps towards a nearby donation bin. With a resolute expression, Diego dropped the scarf into the bin, relieving himself of its presence.
And just like that, the scarf that once rested in the patio basket had found a new temporary home in the donation bin, waiting to be discovered by someone who would appreciate its warmth and beauty. Diego turned around to leave the patio, completely unaware that his actions had not gone unnoticed by Amir.
Question:

- **Question:** As Amir, what is your attitude towards Diego's action of moving scarf from basket to a donation bin?
- **Answer:** negative
- **Type:** attitude

Question:

- **Question:** From Amir's perspective, how does Diego think basket's fullness would change by the end of the story?
- **Answer:** less full
- **Type:** multihop-so

Question:

- **Question:** From Amir's perspective, how would scarf's accessibility change by the end of the story?
- **Answer:** less accessible
- **Type:** multihop-fo

Question:

- **Question:** From Diego's perspective, where precisely does Amir think that the scarf is located by the end of the story?
- **Answer:** a donation bin
- **Type:** location-so

E Model Version Details:

We run the open-sourced Qwen-2.5 72B and 32B Instruct models on 4 A100 80G GPUs. All our experiments took approximately 15 hours to run. Additionally, we use the GPT-4o and GPT-4o-mini checkpoints dated 26-03-2025. All our experiments are run with temperature 0, and seed 0.

F Example Log:

Raw Data Formatting

- **system_prompt:** "You are a Theory-of-Mind reasoning agent. You conduct multi-step reasoning over a given scenario and question to finally provide an answer.You have access to a series of tools using which will provide you instructions on conducting a particular step of reasoning, to finally better answer a given question based on a scenario. You can use a tool multiple times.At each step, reason if using a tool will be helpful, and decide on a tool to be used. After selecting a tool, output a valid function call. The tools are:

- **shift_perspective:** Simulates the perspective of a specified character by simplifying the story to events only known by the character. Returns a perspective story. For multi-character input (nested perspectives), takes prior computed perspective story for character n-1, and simulates events known to n'th character. Nested perspectives must be built sequentially from OUTERMOST to INNERMOST!!!
  - ∗ **Example:** To simulate Character 1's perspective of Character 2's perspective of Character 3's perspective:
  - ∗ **FIRST,** call with **characters = [char1]** (Char1's perspective (outermost))
  - ∗ **THEN,** call with **characters = [char1, char2]** (Char1's perspective of Char2's perspective)
  - ∗ **THEN,** call with **characters = [char1, char2, char3]** (Full nested perspective)
  - ∗ **Rule:** Never call **[char1, char2, char3]** without first calling **[char1]** and **[char1, char2]**. Build perspectives in order to maintain accuracy.
  - ∗ **CORRECT ORDER:** char1 → char1, char2 → char1, char2, char3
  - ∗ **INCORRECT ORDER (do not do this!!!!):** char3 → char2, char3 → char1, char2, char3
  - ∗ **OPTIONAL:** If it is only required to simulate a story till a certain point, for eg. when a question specifies a point in the story, pass this event as the simulate_till argument.
- **mental_state_analysis:** Explicitly track and analyze the specified mental states of a given agent. Useful to conduct an agent-specific analysis. Provide name of character to be analyzed and mental state to be analyzed. Use shift_perspective instead if nested character perspectives are required or if a complete set of events known to a character is needed.
- **choice_reflection:** Provides instructions on how to combine steps taken till this point, to reason towards a solution, to eliminate impossible or unlikely options, or to decide to get more information by calling more tools. Consider using this tool before the answer module to ensure a more accurate response.
- **answer_module:** If all required reasoning and tool use is finished, use this tool to get instructions on providing a final answer.

After conducting the tool reasoning step: If no further tool usage is necessary, use the answer_module tool to return the answer."

- **user_prompt_init:** "The provided scenario is: Rory and Maddox shared a mutual appreciation for the zestful and tart essence of lemons, both finding pleasure in the fruit's tangy palate. Their affinity for the citrus had become a known fact among their circle of friends, and many an afternoon was spent reveling in the pleasures of lemon-flavored delights. It was a bond that spoke silently yet vividly of shared tastes.

On an ordinary yet fateful day, the duo made their way into the lushness of the garden, a sanctuary where the mundane transformed into moments of simple joys. The garden itself was an amphitheater of greenery, with plants and flowers performing their silent, tranquil ballet in the breeze. There, amidst the verdant foliage, lay a plump lemon that seemed to capture the very essence of the color yellow – bold and bright, its skin a canvas painted with the sun's rays.

Drawn to the fruit as if by an invisible thread, Rory and Maddox approached the lemon that sat proudly within its container. It was a common specimen yet rendered extraordinary by the glint in the garden's dappled light. The moment was poised, like a picture waiting to be captured, as they leaned in to better appreciate the lemon's splendid isolation.

However, life's imperatives interrupted the scene. Maddox departed momentarily from the garden, beckoned by the call of an urgent matter that required attention beyond the confines of their shared paradise.

This turn of events left Rory standing solo before the citrus treasure.

The lemon, nestled innocuously in its place, became the subject of a silent decision. Without the hesitation that marks indecision, Rory extended a hand towards the fruit, intent on a relocation that promised future delights of culinary or quenching nature. The act was swift, the motion unaccompanied by fanfare, as Rory transported the lemon from its outdoor cradle to a new abode on the kitchen counter.

Upon Maddox's return, the garden presented a void where the lemon had once been the center of potential. The container that had housed the vibrant orb now sat unadorned, its contents spirited away without a trace. Maddox perused the space where they had previously stood together, unaware of the lemon's journey and its new resting place.

The hours of the day unspooled like threads from a weaver's loom, each moment interlacing with the next in patterns of routine and task. Rory, in possession of the secret, was the sole guardian of the lemon's fate, the fruit itself assuming its position on the kitchen counter, a silent testament to a simple, yet decisive act.

Thus, the tale concluded, the lemon's relocation to Rory's kitchen counter an event unobserved by Maddox. The air of mystery that hung in the garden where the lemon had first been discovered now clung to its absence, the knowledge of its whereabouts securely harbored by Rory alone. And so the story remained, a narrative chapter sealed with the finality of an unspoken secret.

You need to answer the question: From Rory's perspective, where precisely does Maddox think that the lemon is located by the end of the story?

As a first step, identify the type of question/ what ability/reasoning the question will need to be answered. Carefully consider tool descriptions. Provide example usage/ important instructions for tools in the tool guide. Mention all tools available. Then come up with a multi-step plan to solve the question using the tools provided - strictly following the tool description guidelines. Remember, you can use a tool multiple times and chain them together sequentially (ie. use a tool, get a response, then use the same tool again, or use another tool). After each planned step, consider what information you are likely to have after conducting the reasoning step. You always have information from all the previously conducted steps. Analyze the question carefully, (if necessary, consider potential ambiguities), and make sure of the answer before providing it. Follow the format: (Tool use guide with details on correct/incorrect ways to pass parameters and clear example usage (if provided)) (Step 1) : (Expected Information State after Step 1) (Step 2) : (Expected Information State after Step 2) ..."

- **init_plan:**

1. **Tool Use Guide**

   (a) **shift_perspective**: Simulates the perspective of a character. Build nested perspectives sequentially. Example calls:

      i. shift_perspective(characters=["char1"]

      ii. shift_perspective(characters=["char1", "char2"])

      iii. shift_perspective(characters=["char1", "char2","char3"])

      Optional parameter: `simulate_till`.

   (b) **mental_state_analysis**: Example: `mental_state_analysis(character="Rory", mental_state="belief")`

   (c) **choice_reflection**: Example: `choice_reflection()`

   (d) **answer_module**: Example: `answer_module()`

2. **Multi-Step Plan**

(Step 1) Use `shift_perspective` to simulate Rory's perspective. Expected info: simplified story from Rory's perspective, including relocating the lemon.

(Step 2) Use `shift_perspective` again for Maddox's perspective from Rory's perspective.

(Step 3) Use `mental_state_analysis` for Rory's belief about Maddox's knowledge.

(Step 4) Use `choice_reflection` to combine reasoning.

(Step 5) Use `answer_module` to finalize the answer.