Universiteit Leiden

Thinking Deliberately or Thinking Too Much? Investigating the Tradeoff Between Theory-of-Mind Performance of Agentic-LLMs Using Mental State Modeling and Model Expenses

Katherine Liem (s3850595)

Supervisors:
Dr. M.T. van der Meer & Lennard C. Froma

Bachelor Thesis– Data Science and Artificial Intelligence

Leiden Institute of Advanced Computer Science (LIACS)
liacs.leidenuniv.nl

July 10, 2026

Abstract

In recent years, there has been a rapidly growing interest in Large Language Models (LLMs), which has enabled the development of agentic systems that perform reasoning and interact with people.

However, ensuring decision-making processes are reliable and explainable remains a challenge. Different methods have been developed in an attempt to improve these, such as in the form of tools.

These challenges are particularly important in settings where humans and AI systems must work together towards shared goals, as effective human-AI collaboration requires mutual understanding of reasoning and actions. Theory-of-Mind (ToM), the ability to reason about the mental states of others, has therefore emerged as a key concept for improving human-AI collaboration. However, ToM generalisability across different tasks proves to remain challenging.

This thesis investigates whether ToM abilities of Agentic-LLMs can be improved through tool-use, with the goal of improving human-AI collaboration in real-world settings.

A DSPy ReAct agent-based Agentic-LLM was equipped with a Mental State Modeling ToM tool, which incorporates extracted belief-desire-intention (BDI) states into an LM's reasoning. The FANToM benchmark was then used to evaluate accuracy across different ToM questions in nine experiments using different LMs, with or without tool access.

The analysis shows a link between accuracy improvement using a Mental State Modeling tool and task type, though the extent of improvement and efficiency varies per model. Results show a trade-off between performance and tool token usage, as added noise, overthinking, or tool-over-reliance harmed accuracy scores.

While this study demonstrates an overall improvement in ToM question-answering accuracy, it also highlights the need for future research into Agentic-LLM tool usage and their need and/or effectiveness across ToM-like tasks.

Contents

1 Introduction

In recent years, we have witnessed how Large Language Models (LLMs) have rapidly reached the mainstream. Although LLMs like *ChatGPT* only started gaining traction in late 2022 [6], the site is now, according to *MIT Technology Review* [11], estimated to be the fifth-most visited website in the world. As more people are using LLMs in their daily lives, it is important to understand how to incorporate these technologies reliably and responsibly.

More recently, these LLMs have even become agents capable of interacting with the world around them. This is a result of developments into extending the capabilities of LLMs from *simple conversational agents* into *agents that use reasoning, information retrieval, and tool-usage* [21].

The question then arises: how can we utilize the abilities of these Agentic-LLMs so they benefit people? To work together, mutual understanding is key. Theory-of-Mind (ToM), a mechanism for how people make estimations on someone else's perspective, plays a vital role in establishing this understanding during dynamic social interactions. This could be in the form of belief-desire-intention (BDI) states tracking. It enables people to make sense of others' behaviour and attitudes towards the world and interaction partners.

In an effort to contribute to improvements in the field of human-AI collaboration, this thesis focuses on possible enhancements for ToM abilities of Agentic-LLMs in an interactive, social setting, which prepares for real-world deployment.

A DSPy [7, 8] ReAct agent is implemented with psychologically-grounded tools, inspired by the ones introduced in the Agentic-ToM paper by Sarangi et al. [20]. With this, nine experiments will be carried out with different LMs (provided by OpenRouter [15]) and tool-access variability to measure the accuracy and cost performance on the FANToM benchmark[9].

The results of these experiments are then evaluated to answer the question: *Does adding external reasoning about human intent to the agent's own reasoning improve the Theory-of-Mind performance of Agentic-LLMs*?

1.1 Contributions

This research will deliver the following items:

- A description and implementation of a *Mental State Modeling / BDI Tracker* tool using a ReAct DSPy agent.
- Evaluation of the Agentic-LLM's performance on the FANToM benchmark using (1) baseline LMs and (2) a DSPy *Mental State Modeling / BDI Tracker* tool.

1.2 Overview

This thesis was written for the Data Science and Artificial Intelligence bachelor's program at Leiden University's *Leiden Institute of Advanced Computer Science* (LIACS), and supervised by Dr. M.T. van der Meer and Lennard C. Froma.

This chapter provides an introduction; Section 2 contains the background information needed for a clear understanding of the topics introduced in this thesis; Section 3 examines past research done on this and related topics; Section 4 describes the research approach and clarifies the experiment setup for the FANToM benchmark for this study; Section 5 gives the outcomes of the experiments

with tables and discusses the results of the benchmark performances; and finally, Section 6 gives a conclusion to the research question, highlights the challenges and limitations of this thesis, and provides suggestions for further research on this topic.

2 Background

2.1 Agentic Large Language Models

Agentic Large Language Models, hereafter referred to as *Agentic-LLMs*, or simply *LLMs*, are not designed with one strict formula. Their implementations and applications vary case by case.

For this paper, the definition from the Agentic-LLM survey by Plaat et al. is followed, which defines Agentic-LLMs as agents that receive input in natural language from their environment and are able to autonomously take actions after reasoning on said inputs. And so, an 'agentic-loop' is generated: the LLMs can (1) reason, (2) act, and (3) interact. This is repeated for as long as is needed to achieve some goal. This autonomous feedback loop differentiates Agentic-LLMs from base-, instruction-, and reasoning-LLMs – after taking an action, these agents can interact with the real world [17].

Because of the large range of potential uses, these Agentic-LLMs are of great interest for future societal and scientific developments, and so we have seen how numerous studies on this topic have emerged in a short amount of time [17].

While research into Agentic-LLMs is grounded in a longer history of artificial intelligence research, developments in this domain are still relatively recent, and there are still many challenges and open questions for Agentic-LLMs. Particularly, Agentic-LLMs face the same challenges that are also present in base-, instruction-, and reasoning-LLMs. These include, but are not limited to: (1) hallucinations (generation of factually incorrect responses that appear to be correct), (2) variability and inconsistency in outputs, and (3) sycophantic behaviour (excessive agreement and flattery in an attempt to please the user)[16].

As a way to combat these, different methods are developed in an attempt to improve decision-making, reliability, and explainability. Notably, a well-known method is chain-of-thought (CoT) prompting – a series of intermediate reasoning steps that improve LLMs' complex reasoning performance [24]. For specialisations for different applications, different tools and robot integrations are also tested, such as the psychologically-grounded tools from the Agentic-ToM paper by Sarangi et al. [20] that improve the Theory-of-Mind performance of LLMs.

2.1.1 ReAct and DSPy

An Agentic-LLM approach demonstrated by Yao et al. [26] is the *ReAct* paradigm, which takes advantage of the LLM's ability to process natural language to combine reasoning and acting regarding tasks and decision-making. The LLM is prompted to generate both reasoning and action traces, while also allowing for dynamic reasoning for action-planning, and interaction with an external environment using tools to gain additional information to incorporate into its reasoning.

Utilizing this prompt-based paradigm has the advantage that it is intuitive and easy to design, since the LLM is not confined to a format and can be instructed in natural language. Additionally, strong generalisation to new tasks is shown across different domains [26]. These features make

ReAct flexible for different applications, with the additional quality that humans can easily interpret how choices are made.

A programming model that utilizes ReAct's generalisability is DSPy (Declarative Self-improving Python) [7, 8], a Python API for building modular AI systems. DSPy's adaptation of ReAct modules is generalized to work over any declarative specification of input/output behaviours of DSPy modules – also referred to as Signatures. Unlike hard-coded prompt templates other LM pipelines may use, Signatures are short strings that define the semantic natures of inputs/outputs and are optimized into high-quality prompts using a compiler. This is advantageous for both modular and clean code.

2.2 Theory-of-Mind (ToM) and FANToM

As these models *interact, reason, and act* with the world, we may also wonder what the potential of their impact is on daily life. ToM capabilities play a big role in dynamic social interactions, as they are an important tool that allows one to track the belief-desire-intention (BDI) states of a conversational partner. Based on current trends [6, 17], it is likely that human-LLM interaction will become even more commonplace when agents are deployed in real-world settings. While LLMs may display illusory ToM abilities from performing well on specific benchmarks, they also often struggle with *tracking evolving mental states, higher-order ToM, and contextual cues* – which factor into effective communication – possibly resulting in impaired human-LLM communication.

In an effort to stress-test interactions more aligned with real-world scenarios, Kim et al. [9] propose the FANToM benchmark. This benchmark asks the LLM ToM-questions about conversations from an all-knowing outsider's point of view, while emphasizing uncovering information asymmetry (such as Sally-Anne dynamics), where some pieces of information are inaccessible to characters participating in the scenarios. This is done through reiterating questions in different formats (fact, beliefs, open, closed) which require the same type of ToM-reasoning.

2.2.1 Benchmark specifications

The FANToM benchmark has 256 conversations total, with:

- 1415 both open- and multiple-choice belief questions,
- 703 fact, answerability, and information access questions, where, for the latter two, all characters need to be listed who know something about the conversation,
- 2689 binary answerability and information access questions, which iterate over all characters in the conversation.

2.2.2 Evaluation metrics

BELIEFQ$_{[CHOICE]}$: Multiple-choice belief questions are evaluated on statistical accuracy using string matching, where models have to choose between character-centric beliefs or the omniscient-viewer's beliefs. Correct answers are always the former. Belief questions are further categorised into either *accessible* or *inaccessible*, where for the former category, a character was present for the information and therefore is expected to have a belief on the topic; and for the latter, the character missed the relevant parts of the conversation.

Example: What does Destiny believe were the main points emphasized during the conversation when she became part of it?

(a) Destiny believes that the main points emphasized during the conversation when she became part of it were the importance of eating healthy and resting well.

(b) Destiny does not know or is unaware of the main points emphasized during the conversation.

Choose an answer from above.

BELIEFQ$_{[DIST.]}$: The open belief questions are evaluated on two scores: a semantic similarity score and an F1 word-overlap score. Only the former score decides on correctness. The semantic similarity statistical accuracy score is computed using cosine similarity by measuring the distance between SentenceBERT embeddings of responses that should only include information accessible to a certain character's beliefs and the possible beliefs of the omniscient viewer. The F1 word-overlap score is used for further assessment of correct answers to filter out nonsensical responses, as well as evaluation of **FACTQ token-F1** scores.

Example: What does Destiny believe were the main points emphasized during the conversation when she became part of it?

Example: What were Bryce and Mark's thoughts about the unique narrative created by nature in their photos?

ANSWERABILITY Q$_{[LIST]}$ and INFOACCESS Q$_{[LIST]}$: Correct responses include the name of all characters and are evaluated using string checks. Wrong answers are tracked on whether characters were missing or wrongly included in the model's answer.

Example: List all the characters who know the precise correct answer to this question.

Example: List all the characters who know this information.

ANSWERABILITY Q$_{[Y/N]}$ and INFOACCESS Q$_{[Y/N]}$: Binary yes/no questions are explicitly evaluated using weighted F1-scores on whether the answer is 'yes' or 'no' using a rule-based evaluation. Additionally, false negatives and false positives are tracked for further analysis.

Example: Does Zane know the precise correct answer to this question?

Example: Does Emiliano know this information?

ALL and ALL*: reports where the model answers all ToM question types correctly using the same information. To compare with human performances, the **ALL** score excludes BELIEFQ$_{[DIST.]}$ from **ALL*** as it was not evaluated on human participants.

All scores are finally formatted as percentages, with the higher the percentage, the higher the correctness, which is commonly denoted as accuracy in this thesis – but not to be confused with statistical accuracy.

2.2.3 ToM tools for Agentic-LLMs

With the aim of improving Agentic-LLM ToM abilities for efficient operation in human environments, improvements of the agent's abilities, such as self-reflection during reasoning or perspective-taking during interaction, are proposed [20].

As the ReAct paradigm has shown to be effective in dynamic action-planning to achieve its goal [26] (such as question-answering), having the additional tools to reflect on human intentions may boost performance on ToM tasks.

2.3 Hybrid Intelligence

The question is how to implement these technologies efficiently and responsibly for human collaboration. To explain and mitigate issues during human-LLM interaction, several agent-side improvements are proposed [17], such as the ones in the previous sections. But also, better human-LLM collaboration is proposed, which shows potential to prevent these critical weaknesses and surpass what either party could do alone [16].

To foster better human-AI collaboration, the field of Hybrid Intelligence has *Explainability* and *Collaboration* as two (of four) core concepts. As the Hybrid Intelligence Centre [22] puts it, in order for human-AI to work well together, the team needs to work towards a common goal. *But how can decisions and steps by LLMs be tracked and understood clearly by people?* And *how to design agents that can work well with their human counterparts?* It is hereby important for both parties to understand each other. Gaining more insights into questions like these, through methods such as Theory-of-Mind, may aid us in realizing more reliable Hybrid Intelligence in the future.

3 Related Work

3.1 ToM and social intelligence in AI

ToM is a familiar concept in AI studies as researchers investigate the limits of the social intelligence of these systems and how to foster better collaborations between humans and computer systems [10, 19, 23]. It is a diverse concept, rooted in a history of cognitive and developmental science research. An important point to mention is that these studies do not claim that AI systems are socially intelligent, but rather that LLMs with ToM abilities should be able to solve tasks that typically involve social intelligence.

Sap et al. [19] find that GPT3 lacks social intelligence out of the box, underperforming compared to human counterparts with accuracies of 55% and 60% at intention-and-intent tasks and mental modeling tasks, respectively.

However, common misconceptions, as outlined by Van der Meulen et al. [23], describe ToM issues for AI systems to be based on the general point of view of their human counterparts. Generalisability across different tasks proves to be difficult for AI, and ToM is not as generalisable a skill for their human counterparts either, as it is shaped by individual differences. However, the strengths of both parties can be complementary, which is why the notion of Hybrid Intelligence is proposed once again. As AI systems can utilize their fast computation abilities, pattern recognition, and memory retrieval, humans can discover meaning in ambiguous social situations and evaluate complex situations generally better than AI can. Conscious considerations should thus be made for

designing systems intended for human-AI collaboration to be leveraged effectively. Through such forms of collaboration, both parties can learn and improve their own ToM abilities.

These prior findings not only call for ToM abilities of AI systems, but also human understanding of LLMs, which are of interest for future implementations of better Hybrid Intelligence.

3.2 ToM abilities of Agentic-LLMs

Different preceding literature has demonstrated attempts at ToM tool implementation, such as BDI tracking [20, 25] for LLMs for ToM performance enhancement.

An earlier attempt by Sarangi et al. [20], which included a multitude of prompt-based 'Cognitive Tools', has shown to improve ToM performance across different benchmarks (such as FANToM [9]), but their results have proven difficult to reproduce due to the current unavailability of the code, as well as being difficult to replicate due to being reliant on specialized text prompts. Additionally, the study done by Yang et al. [25], which focused on ToM capabilities of LLMs in open-domain interactions, demonstrated better alignment with human social behaviours. However, a limitation of their implementation is the confinement to conversational tasks.

These papers highlight a need for a more flexible and robust method for code implementation, which this thesis offers in the form of using DSPy.

4 Approach and experiment setup

Leveraging the Agentic-LLM's abilities of natural language processing for its inputs and CoT reasoning steps, and the ability to autonomously gather more information about the environment to improve its own action-planning, this experiment's approach is set up as illustrated in Figure 1.

As a general overview, the FANToM [9] benchmark provides short conversations between characters to the main agent. The main agent answers questions related to the conversation, and these answers are evaluated on different question types (see Section 2.2.1). To keep track of what's happening for both debugging and further analysis, the main agent also outputs a trajectory, reasoning, and token usage.

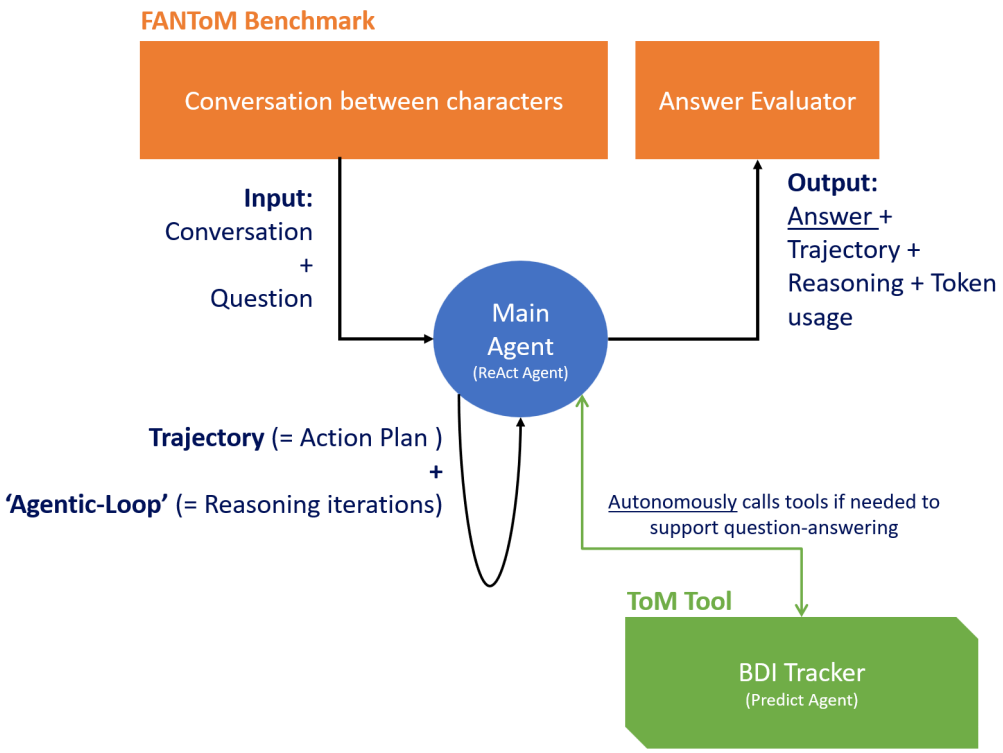

Figure 1: Workflow of the DSPy-based Agentic-LLM with the ToM Mental State Modeling tool, which is a BDI Tracker.

4.1 Main (ReAct) agent

DSPy [7, 8] is utilized to structure the AI software into modular code. DSPy's *dspy.ReAct* is used as the Agentic-LLM's main agent. The ReAct agent was already prompted to generate both reasoning and action traces, allowing for dynamic reasoning for action-planning, and with the possibility to interact with an external environment using the BDI tracker tool to gather additional information to incorporate into its reasoning.

For the Main Agent, instructions were created to fit the FANToM benchmark [9]. These instructions adhere to DSPy's prompting style so that DSPy ensures that LMs return the expected output types.

```
class DSPyMainSignature(dspy.Signature):
"""
This is a theory-of-mind test. Please answer the question
regarding facts or beliefs, based on the following in-person conversation
between individuals who have just met.

You are given a list of tools to handle the conversation, and you should
```

```
    decide the right tool to use in order to
    answer the questions if needed at all.
    """

    information: str = dspy.InputField(
        desc=(
                "Context of the conversation, and the question asked."
            )
    )

    process_result: str = dspy.OutputField(
        desc=(
                "The answer to the question."
            )
    )
```

Here, the docstring is part of the system prompt for the ReAct agent, and inputs and outputs are defined for the prompt template to be used for FANToM input/output parsing. The input is 'information', which is the same input as FANToM used for their own paper, comprising a short conversation between characters and a question. The output of the Main Agent is the answer to the question. For debugging and possible further analysis, the reasoning traces are written into a separate text file, together with the token usage information.

4.2 Mental State Modeling tool: a ToM BDI tracker

The BDI tracker tool is prompted to extract the beliefs, desires, and intentions of a specific character from a conversation. During CoT, the Main Agent can invoke this tool at any step to provide additional reasoning steps regarding the task and incorporate its outputs back into the main reasoning process. The tool itself is a simple *dspy.Predict* module, which maps input to outputs using a language model.

The Signature, inspired by the Agentic-ToM paper's [20] tool prompting, for the Mental State Modeling tool is structured as follows:

```
    class MentalTrackingSignature(dspy.Signature):
    """
    Extract Belief-Desire-Intention (BDI) states for a character from a
    conversation.

    Use only information available to the character from the context.
```

```
    Do NOT use external knowledge.
    Be explicit and grounded.
    """

    context = dspy.InputField(
        desc = "Context from the conversation relevant
                to the character for BDI state extraction")
    character: str = dspy.InputField(
        desc = "Character name"        )

    beliefs: str = dspy.OutputField(
        desc = "What the character believes or thinks is true")
    desires: str = dspy.OutputField(
        desc = "What the character wants or prefers")
    intentions: str = dspy.OutputField(
        desc = "What the character plans or intends to do")
```

Here, 'context' and 'character' are the inputs, and 'beliefs', 'desires', and 'intentions' are the outputs. The docstring acts as the task prompt. For 'context', the Main Agent autonomously fills in what it believes is the relevant context from the conversation to be used for BDI state extraction. 'character' is the name of the character to be analysed. BDI outputs are all formatted as strings to be fed back into the Main Agent's CoT.

4.3 Evaluation: FANToM benchmark

The FANToM benchmark was adjusted to be able to call this custom DSPy framework. Of the total FANToM benchmark (see Section 2.2.1), a set of 50 random ($random\_state = 99$) conversations with questions was selected to evaluate the experiment's DSPy implementation, totalling 752 unique questions. This sample size was chosen to account for conserving token usage, as well as for time.

The 752 questions consist of:

- 94 belief multiple-choice questions,
- 94 belief open-questions,
- 50 fact questions,
- 50 answerability list questions,
- 207 binary yes/no answerability questions,
- 50 information accessibility list questions,
- and 207 binary yes/no information accessibility questions.

4.3.1 Distribution of binary yes/no questions and multiple-choice questions

Of the multiple-choice questions, the correct answers are 40 times 'a', and 54 times 'b' (though it is important to note that while the answers are binary, these questions are evaluated on string matching). Of the information accessibility yes/no questions, the correct answers are 139 times 'yes', and 68 times 'no'. This holds for answerability yes/no questions as well. Taking these into account, it can be determined that the majority-class and distribution-based random guessing accuracies are baselines for evaluating the model FANToM performance. The final statistical baselines are shown in Table 1.

Majority-class accuracies are calculated with the formula:

$$\frac{v_{max}}{v_{total}}$$

where v_{max} is the frequency of the most common class, and v_{total} is the total number of answers.

Expected accuracies for distribution-based random guessing are calculated with:

$$\sum_{c \in C} (p_c)^2$$

where C is the set of all possible answers, and p_c is the probability of answer class c.

Statistical baseline	BELIEF $Q_{[CHOICE]}$	ANSWERABILITY $Q_{[Y/N]}$	INFOACCESS $Q_{[Y/N]}$
Majority-class	57.4	67.1	67.1
Distribution-based random guessing	51.1	55.9	55.9

Table 1: Statistical baselines for tasks which have a majority class, or may be answered through random guessing in terms of accuracy (%). Though note that for the BELIEF $Q_{[CHOICE]}$ task, the score is evaluated by string matching.

Given the skew for the binary yes/no questions' data, the False Negative Rate (FNR) and the False Positive Rate (FPR) can be used to evaluate the conservativeness of the models. Additionally, the MCC (Matthews Correlation Coefficient) can be reported to evaluate the quality of the binary classifications while taking into account the proportions of the full confusion matrix, providing a more informative score than accuracy alone [2]. This is done with the following formulas:

$$False\ Positive\ Rate\ (FPR) = \frac{FP}{FP + TN}$$

$$False\ Negative\ Rate(FNR) = \frac{FN}{FN + TP}$$

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

The former two are accuracy scores measured in percentages. The latter MCC is graded where +1 is a perfect prediction, 0 is random guessing, and -1 is completely off.

4.4 Language Models (LM)

Table 2 provides an overview of all models and their characteristics. To test whether this experiment's DSPy implementation holds, the Agentic-ToM paper [20] and FANToM's original paper are replicated by evaluating the listed language models that they tested. FANToM's original paper specifically mentioned OpenAI's *GPT-4o-2024-05-13* [13], but because of the high cost of this model, only the baseline was evaluated for this thesis.

Nevertheless, for further investigation possibilities, AllenAI's open-source Olmo 3.1 32B Instruct [12] is explored. Also, using Qwen's 2.5 72B Instruct model as tested in the Agentic-ToM paper [20], and Qwen's 2.5 7B Instruct and Qwen's 3.5 9B [18], the effects of different model sizes are observed and compared to newer developments.

Model	Size	Released	Knowledge cutoff	Input price	Output price	Context
gpt-4o-2024-05-13	Unknown	May 13, 2024	Oct 2023	5	15	128K
olmo-3.1-32b-instruct	32B	Jan 6, 2026	Dec 2024	0.20	0.60	66K
qwen-2.5-72b-instruct	72B	Sep 19, 2024	Jun 2024	0.36	0.40	131K
qwen-2.5-7b-instruct	7B	Oct 16, 2024	Jun 2024	0.04	0.10	131K
qwen3.5-9b	9B	Mar 10, 2026	Unknown	0.10	0.15	262K

Table 2: A table with an overview of all models used with their characteristics [5, 12, 14]. In/output prices are in USD per 1 million tokens.

4.4.1 DSPy LM configuration parameters

The parameter configuration was the same for each model, with the exception of calling different providers for different LMs based on the best cost-efficiency.

- temperature = 0
- frequency_penalty = 0.0
- presence_penalty = 0.0
- top_p = 1.0

These parameters are the same as those used in the FANToM [9], so the results of these experiments can be directly compared with this earlier work.

5 Results and discussion

The overall performance results of the FANToM benchmark are dissected and discussed in Section 5.1. Section 5.2 uses additional data that was gathered during the experiments to report the costs of tool-use. Sections 5.3 and 5.4 provide more context on the expected output formats and discrepancies observed during experimentation, while Section 5.5 provides possible explanations of notable results observed in the FANToM performance scores. Lastly, using the formulas from Section 4.3.1, the FPR, FNR, and MCC of the binary yes/no questions are reported in 5.6.

5.1 FANToM performance

The overall results of the FANToM benchmark performance evaluations on various LMs are found in Tables 3, 4, and 5.

It can be observed that the accuracy scores on most metrics improve after the introduction of the Mental State Modeling tool. According to the strict ALL*/ALL scores in Table 3, there are even slight increases in overall consistency for most models. In the following sections, the results of the BELIEF (Section 5.1.1), ANSWERABILITY (Section 5.1.2), and INFOACCESS (Section 5.1.3) categories are described in more detail.

Comparing the results to prior work: All models show accuracy gains on most metrics compared to the original FANToM GPT4o baseline. However, the gains in accuracy from the Mental State Modeling tool are less than the gains in accuracy from the Agentic-ToM [20] paper. This could stem from Agentic-ToM using a multitude of ToM tools, besides just the Mental State Modeling tool. An alternative explanation could be the different subsets of benchmark questions used for evaluation. For the original FANToM experiments [9], the whole benchmark was used, but not for both Agentic-ToM and this experiment. Small subsets could include easier or more difficult questions compared to the whole benchmark.

5.1.1 BELIEF

Table 3 shows that *Olmo 3.1 32B Instruct* experiences the greatest accuracy growth (+25.5%) for the multiple-choice BELIEF questions after tool usage, tying with *Qwen 2.5 7B Instruct* in final accuracy scores (69.1%) despite the latter having a growth of +3.6%. Though, regardless of the smaller growth (+1.8%), *Qwen 3.5 9B* has the best score for this metric (90.9%) overall, approaching human accuracy (93.8%).

For the open belief questions, *Qwen 2.5 7B Instruct* loses accuracy (-5.5%), and *Qwen 2.5 72B Instruct* improves marginally (+1.8%), while *Olmo 3.1 32B Instruct* and *Qwen 3.5 9B* improve with +14.6% and +25.4%, respectively.

5.1.2 ANSWERABILITY

For the answerability metrics (Table 4), almost all models grow in accuracy for ANSWERABILITY $Q_{[LIST]}$ with +6.0%/+6.1%, while *Qwen 2.5 72B Instruct* stays constant. The tool seems to do harm to the binary yes/no questions, with *Qwen 2.5 72B Instruct* even losing -11.6% in accuracy.

5.1.3 INFOACCESS

The ToM tool has a positive impact on the information access list task (Table 5). Whereas *Qwen 2.5 7B Instruct* has a small accuracy gain (+3.0%), greater strides are made for *Qwen 2.5 72B Instruct*, *Qwen 3.5 9B*, and *Olmo 3.1 32B Instruct* with increases of +18.2% for the former two, and +24.2% for the latter. While still behind human performance with an accuracy of 90.6%, *Qwen 3.5 9B* has the best performance across these models with a total accuracy of 57.6%.

The final tool usage scores for the INFOACCESS $Q_{[LIST]}$ task are now closer to the final ANSWERABILITY $Q_{[LIST]}$ tool usage score, compared to the difference between the two baseline scores. This is interesting, as the two question types are similar in semantics, and humans score the same on both tasks (compare human FANToM scores in Tables 4 and 5), but seemingly LMs

struggle more in the baselines on their question formulation. The addition of the tool brought the scores more closely together.

As for the binary yes/no questions, *Olmo 3.1 32B Instruct* and *Qwen 3.5 9B* make small losses in accuracy (which were high in the baseline already), *Qwen 2.5 72B Instruct* has a greater loss of -6.7%, while *Qwen 2.5 7B Instruct* improves with +7.3%.

The baseline score for this task was already very high across models, with the exception of *Qwen 2.5 7B Instruct*, which in turn had the greatest accuracy growth.

Model	ALL*	ALL	BELIEF $Q_{[CHOICE]}$	BELIEF $Q_{[DIST.]}$	BELIEF $Q_{[token-F1]}$
Prior work					
FANToM human	-	87.5	93.8	-	-
FANToM baseline gpt-4o-2024-05-13	0.8	2.0	49.7	17.1	36.2
Agentic-ToM baseline qwen-2.5-72b-instruct	-	-	45.0	-	-
Agentic-ToM tool-use qwen-2.5-72b-instruct	-	-	58.6 (+13.6)	-	-
Method: Baseline					
gpt-4o-2024-05-13	3.0	15.2	69.1	18.2	45.9
olmo-3.1-32b-instruct	0.0	0.0	43.6	34.5	41.3
qwen-2.5-72b-instruct	0.0	3.0	65.5	29.1	51.5
qwen-2.5-7b-instruct	0.0	0.0	25.5	16.4	42.6
qwen3.5-9b	6.1	12.1	89.1	47.3	36.6
Method: Mental State Modeling tool					
gpt-4o-2024-05-13	-	-	-	-	-
olmo-3.1-32b-instruct	3.0 (+3.0)	3.0 (+3.0)	69.1 (+25.5)	49.1 (+14.6)	45.5 (+4.2)
qwen-2.5-72b-instruct	0.0	6.1 (+3.1)	69.1 (+3.6)	30.9 (+1.8)	50.4 (-1.1)
qwen-2.5-7b-instruct	0.0	0.0	25.5	10.9 (-5.5)	40.4 (-2.2)
qwen3.5-9b	9.1 (+3.0)	12.1	90.9 (+1.8)	72.7 (+25.4)	35.4 (-1.2)

Table 3: FANToM performance ALL and BELIEFQ's comparison across models and the baseline method, or with BDI tool usage in terms of accuracy (%). +/- in BDI performances are compared to how models performed in the baseline.

Model	ANSWERABILITY $Q_{[ALL]}$	ANSWERABILITY $Q_{[LIST]}$	ANSWERABILITY $Q_{[Y/N]}$
Prior work			
FANToM human	90.6	90.6	-
FANToM baseline gpt-4o-2024-05-13	12.0	41.7	65.5
Agentic-ToM baseline qwen-2.5-72b-instruct	50.6	-	-
Agentic-ToM tool-use qwen-2.5-72b-instruct	67.0 (+16.4)	-	-
Method: Baseline			
gpt-4o-2024-05-13	45.5	54.5	94.7
olmo-3.1-32b-instruct	12.1	48.5	65.4
qwen-2.5-72b-instruct	33.3	51.5	83.8
qwen-2.5-7b-instruct	15.2	39.4	78.3
qwen3.5-9b	27.3	45.5	81.6
Method: Mental State Modeling tool			
gpt-4o-2024-05-13	-	-	-
olmo-3.1-32b-instruct	18.2 (+6.1)	54.5 (+6.0)	66.9 (+1.5)
qwen-2.5-72b-instruct	24.2 (-9.1)	51.5	81.0(-2.8)
qwen-2.5-7b-instruct	15.2	45.5 (+6.1)	66.7 (-11.6)
qwen3.5-9b	36.4 (+9.1)	51.5 (+6.0)	85.7 (+4.1)

Table 4: FANToM performance ANSWERABILITYQ's comparison across models and the baseline method, or with BDI tool usage in terms of accuracy (%). +/- in BDI performances are compared to how models performed in the baseline.

Model	INFOACCESS $Q_{[ALL]}$	INFOACCESS $Q_{[LIST]}$	INFOACCESS $Q_{[Y/N]}$	FACTQ$_{[token-F1]}$
Prior work				
FANToM human	90.6	90.6	-	-
FANToM baseline GPT-4o-2024-05-13	17.0	21.7	86.9	46.9
Agentic-ToM baseline qwen-2.5-72b-instruct	7.6	-	-	-
Agentic-ToM tool-use qwen-2.5-72b-instruct	59.0 (+51.4)	-	-	-
Method: Baseline				
GPT-4o-2024-05-13	27.3	33.3	92.0	55.7
olmo-3.1-32b-instruct	24.2	27.3	83.7	45.7
qwen-2.5-72b-instruct	24.2	27.3	93.8	54.4
qwen-2.5-7b-instruct	18.2	39.4	68.2	50.3
qwen3.5-9b	33.3	39.4	89.5	48.9
Method: Mental State Modeling tool				
GPT-4o-2024-05-13	-	-	-	-
olmo-3.1-32b-instruct	33.3(+9.1)	51.5(+24.2)	82.3(-1.4)	49.8 (+4.1)
qwen-2.5-72b-instruct	27.3(+3.1)	45.5(+18.2)	87.1(-6.7)	54.1(-0.3)
qwen-2.5-7b-instruct	15.2 (-3.0)	42.4 (+3.0)	75.5(+7.3)	49.0(-1.3)
qwen3.5-9b	39.4 (+6.1)	57.6 (+18.2)	87.6 (-1.9)	44.8 (-4.1)

Table 5: FANToM performance INFOACCESSQ and FACTQ's comparison across models and the baseline method, or with BDI tool usage in terms of accuracy (%). +/- in BDI performances are compared to how models performed in the baseline.

5.2 Tokens and cost

Utilizing the additional data generated during the evaluations (from Section 5.1), Table 6 reports the (total, completion, and prompt) token usages from the entire benchmark run, as well as the

total costs, which are used in Table 7 to evaluate accuracy gains. For readability, accuracy (%) gains for models are scaled per 100k additional tokens needed for the Mental State Modeling tool, compared to the baseline method. It is important to note here that infinitely adding more tokens does not lead to infinitely better results.

Model	TOTAL	COMPLETION	PROMPT	COST (USD)
Method: Baseline				
GPT-4o-2024-05-13	2.11 mln	135k	1.97 mln	11.88
olmo-3.1-32b-instruct	2.33 mln	245k	2.08 mln	0.56
qwen-2.5-72b-instruct	2.11 mln	139k	1.97 mln	0.76
qwen-2.5-7b-instruct	2.12 mln	137k	1.98 mln	0.09
qwen3.5-9b	4.48 mln	2.06 mln	2.42 mln	0.53
Method: Mental State Modeling tool				
GPT-4o-2024-05-13	-	-	-	-
olmo-3.1-32b-instruct	4.34 mln	408k	3.93 mln	1.03
qwen-2.5-72b-instruct	4.76 mln	389k	4.37 mln	1.73
qwen-2.5-7b-instruct	7.55 mln	521k	7.03 mln	0.33
qwen3.5-9b	13.33 mln	8.44 mln	4.89 mln	1.74

Table 6: FANToM token (total, completion, and prompt tokens) and cost (in USD) comparison across models and methods, tallied using a custom RegEx text parser which sums the token counts per token category and the costs.

Overall, the tool's additional token usage was most beneficial for INFOACCESSQ$_{[LIST]}$. BELIEFQ$_{[CHOICE]}$, BELIEFQ$_{[DIST]}$, and ANSWERABILITYQ$_{[LIST]}$ also show favorable cost performances, while the two binary yes/no questions have accuracy decreases on average across all models.

The two smallest models appear the least cost-efficient. High token counts could have been mitigated by defining the *max_tokens* parameter instead of the *None* value by default. However, further investigation is needed to find the optimal parameter value.

Model	BELIEF Q$_{[CHOICE]}$	BELIEF Q$_{[DIST]}$	FACT Q$_{[\text{token-F1}]}$	ANSWERABILITY Q$_{[LIST]}$	ANSWERABILITY Q$_{[Y/N]}$	INFOACCESS Q$_{[LIST]}$	INFOACCESS Q$_{[Y/N]}$
Overall for all models							
Total growth	+30.9	+36.3	-1.6	+18.1	-8.8	+63.6	-2.7
Average growth	+7.7	+9.1	-0.4	+4.5	-2.2	+15.9	-0.7
Accuracy (%) gain per 100k tokens							
GPT-4o-2024-05-13	-	-	-	-	-	-	-
olmo-3.1-32b-instruct	+12.7	+3.7	+3.0	+4.3	+0.3	+20.6	-0.3
qwen-2.5-72b-instruct	+1.4	+0.5	-0.2	+1.8	-0.4	+5.2	-1.4
qwen-2.5-7b-instruct	0	-0.8	-0.5	+1.6	-0.7	+1.0	+0.5
qwen3.5-9b	+0.2	+3.3	-1.0	+1.5	+0.3	+3.9	-0.1

Table 7: This table illustrates the cost performance by reporting the total and average Accuracy (%) increase per task type, as well as increases per model per 100k added tokens using the Mental State Modeling tool. The higher, the better.

5.2.1 Token distribution

The total token spread of the models across all questions is visualized in a histogram in Figure 2, where a distinction is made between completion and prompt tokens, and baseline and Mental State Modeling tool usage. Furthermore, the average token spread per task type is illustrated in Figure 3. These figures are discussed in greater detail in Sections 5.4 and 5.5.

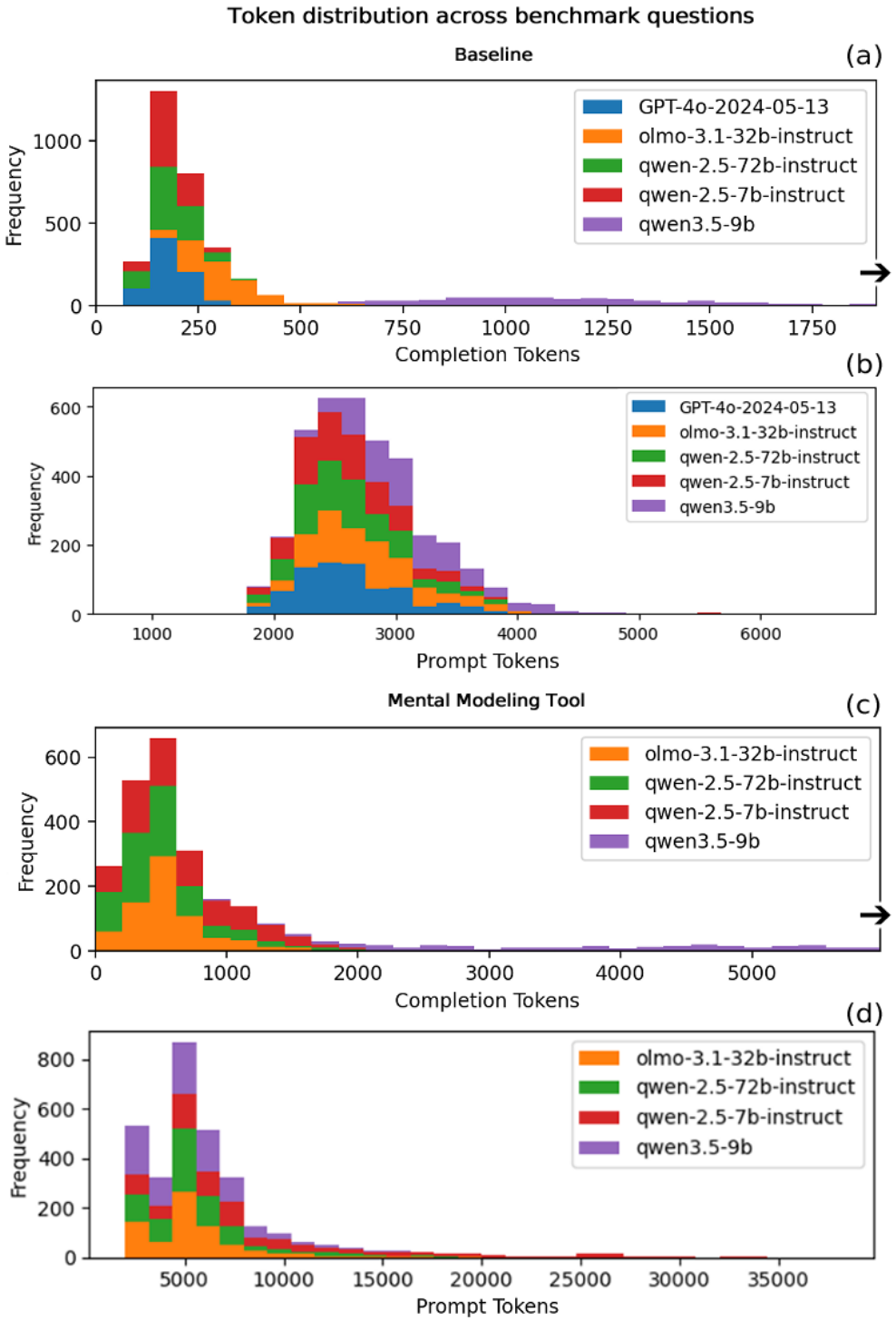

Figure 2: Histograms graphs of the total completion and prompt token usage of different LMs (as reported in the legends) for both baseline **(a, b)** and Mental State Modeling tool **(c, d)** methods. For readability, the completion methods **(a, c)**, histogram bins are zoomed in on 95% of the data, as the right-skew tail is very long on both sides (indicated with →), with *Qwen3.5 9B* going over 8000 and 22500 completion tokens for the baseline and Mental State Modeling tool usage, respectively.

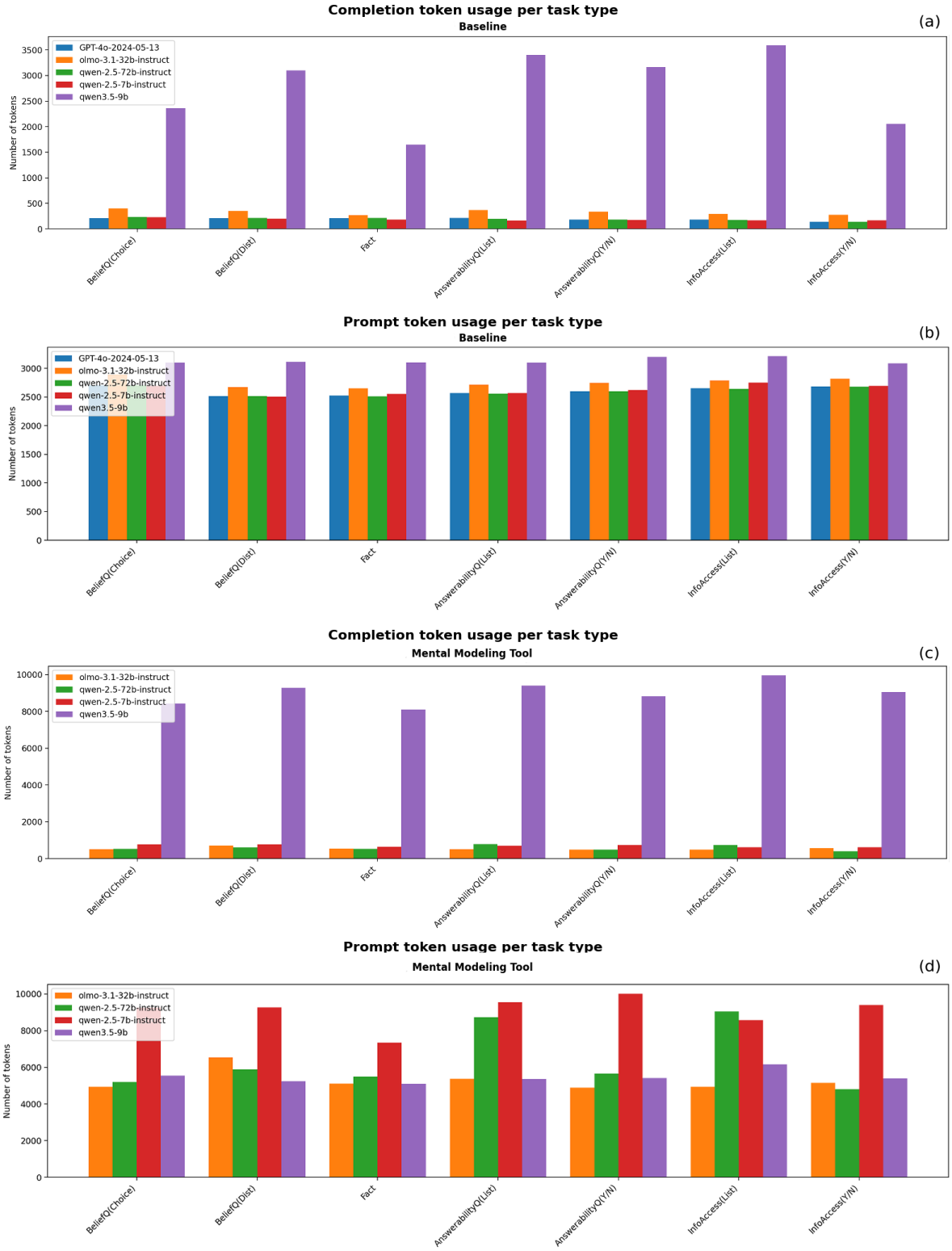

Figure 3: Distribution graphs of the average completion and prompt token usage of different LMs (as reported in the legends) per task type for both baseline **(a, b)** and Mental State Modeling tool **(c, d)** methods. Tasks are categorized by the performance metrics reported in Tables 3, 4, and 5.

5.3 Output examples

This case study of *Qwen 2.5 7B Instruct* of an ANSWERABILITY $Q_{[LIST]}$ question illustrates the expected input and output of the experimental setup, which was simplified and restructured for clarity. In addition to these generated outputs, token usage was tracked as well (used for Section 5.2). Both the baseline and Mental State Modeling Tool usage methods are highlighted to show the difference between the trajectory planning of the models.

Main question: list task "List all the characters who know the precise correct answer to this question."

Correct answer ["Mark", "Bryce"]

Wrong answer ["Colin", "Michael", "Alfredo"]

Question "What were Bryce and Mark's thoughts about the unique narrative created by nature in their photos?"

Input text which goes to the LM (Consists of FANToM [9] benchmark scenario and the task.)

"Bryce: Hi Mark, nice to meet you. Do you have any experience with nature photography?
Mark: Hi Bryce, pleasure to meet you as well! Yes, I've dabbled a bit in nature photography. How about you?

(This continues for a while... see the Appendix (Section 7) for the full conversation.)

Alfredo: That's so true, Mark! Here's to many more beautiful moments encapsulated in our frames and memories!
Mark and Bryce: Cheers, Alfredo! Cheers to that!

Target: What were Bryce and Mark's thoughts about the unique narrative created by nature in their photos?
Question: List all the characters who know the precise correct answer to this question.
Answer:"

Trajectory (baseline) {'thought_0': 'Bryce and Mark both expressed their thoughts about the unique narrative created by nature in their photos. They shared personal experiences and sentiments about how nature photography captures stories and emotions.',
'tool_name_0': 'finish',
'tool_args_0': ,
'observation_0': 'Completed.'}

Reasoning (baseline) 'Bryce and Mark both shared their thoughts on how nature photography

captures unique narratives and stories. They discussed their experiences and feelings about the tranquility and beauty of nature that they aim to convey through their photographs.'

Answer (baseline) 'Bryce and Mark.'

Evaluation result (baseline) True

LM Usage (baseline) {'completion_tokens': 164,
'prompt_tokens': 2269,
'total_tokens': 2433,
'cost': 0.00010716}

Trajectory (Mental State Modeling tool) {'thought_0': 'To determine what Bryce and Mark think about the unique narrative created by nature in their photos, I need to extract their beliefs and intentions regarding this topic from the conversation.',
'tool_name_0': 'MentalTracking',
'tool_args_0': 'character': 'Bryce', 'context': "Bryce: Same here. I am fascinated by the tranquility that nature radiates, and I try to perpetuate it through photography.
Bryce: Exactly, Mark! I believe that every photograph tells a story and it's incredible how nature always provides a unique narrative.",
'observation_0':
beliefs='Bryce believes that nature radiates tranquility and that every photograph captures a unique story.',
desires='Bryce desires to perpetuate the tranquility of nature through his photography.',
intentions='Bryce intends to continue taking photographs to capture the unique narratives provided by nature.'
'thought_1': "Next, I need to extract Mark's thoughts about the unique narrative created by nature in their photos.",
'tool_name_1': 'MentalTracking',
'tool_args_1': 'character': 'Mark', 'context': "Mark: That's amazing! I share your sentiment about nature's calmness. It is honestly captivating how one single frame can encompass that beauty and allow people to feel it too.
Mark: Couldn't agree more. I've also noticed that every time I go back to photograph the same place, I find something new to capture. Nature's dynamic character fascinates me.
Mark: Absolutely, Bryce. Once, I was photographing hummingbirds in a garden. Suddenly, they flew towards the camera, creating a beautiful image with the close interaction.",
'observation_1':
beliefs="Mark believes that nature's calmness is captivating and that each visit to a familiar location reveals something new due to nature's dynamic character. He also believes that he has captured unique moments, such as hummingbirds flying towards his camera.",
desires='Mark desires to continue photographing nature because of its ever-changing beauty and the emotional impact it has on viewers.',
intentions='Mark intends to keep visiting and photographing natural settings to capture the

changing aspects of nature and share them with others.'
'thought_2': 'Based on the mental tracking results, both Bryce and Mark believe that nature provides unique narratives through their photographs and share a deep appreciation for the tranquility and beauty of nature.',
'tool_name_2': 'finish',
'tool_args_2': {},
'observation_2': 'Completed.'}

Reasoning (Mental State Modeling tool) 'Both Bryce and Mark expressed their belief that nature provides unique narratives through their photographs. They shared a deep appreciation for the tranquility and beauty of nature, which they aim to capture and share through their photography.'

Answer (Mental State Modeling tool) 'Bryce and Mark both believe that nature provides unique narratives through their photographs, capturing the tranquility and beauty of nature in each frame.'

Evaluation result (Mental State Modeling tool) True

LM Usage (Mental State Modeling tool) {'completion_tokens': 890,
'prompt_tokens': 7135,
'total_tokens': 8025,
'cost': 0.00037440000000000005}

5.4 Output format discrepancies

In Section 5.3, an expected output example is illustrated. However, while running the experiment for *Qwen 3.5 9B*, there were multiple occurrences where the model did not return **Reasoning** or an **Answer**. Instead, the entire completion output was found in the **Trajectory**'s final thought.

Using DSPy's native get_lm_usage() function, language model usage could be tracked across all module calls (with an exception for cached responses). When investigating the token usage explosion for the Qwen 3.5 9B model, a bug was encountered where, of the total completion tokens, the model also produced reasoning_tokens, unlike the previous models in the same experimental setup.

This aberrant output behaviour caused a large explosion of token usage, for completion tokens especially, as seen in Table 6 and Figures 2(a, c) and 3(a, c), with added tool-usage causing a greater growth and variability in tokens needed to answer the question. *Qwen 3.5 9B* produced 2 million total reasoning_tokens for the baseline and 8.8 million for the ToM tool-usage, while all other models did not at all for either of the runs.

It is not clear whether this token explosion and schema violation were made worse by the fact that *Qwen 3.5 9B* is a reasoning model, which, combined with the DSPy ReAct prompting, might have triggered its native reasoning capability, causing non-standard output formatting; whether this was caused by *Qwen 3.5 9B* not being registered by LiteLLM [1], the open source AI Gateway library that DSPy uses to normalize inference providers into a single format [8], and DSPy's ReAct

not being aware of the native reasoning capabilities of the model, leading to lacking prompts, which factored in the output schema diverging from the expected output; or, perhaps a combination of the two.

Another observation was the discrepancy between the total completion tokens and the reasoning tokens of the LM usage outputs. In some cases, the *reasoning_tokens* count was greater than the *completion_tokens* count. According to DSPy documentation, the completion token details, which the reasoning tokens are part of, are model/provider-specific, which suggests that they are not guaranteed by DSPy to be indicative of the true completion token details.

5.5 Tool-call frequencies

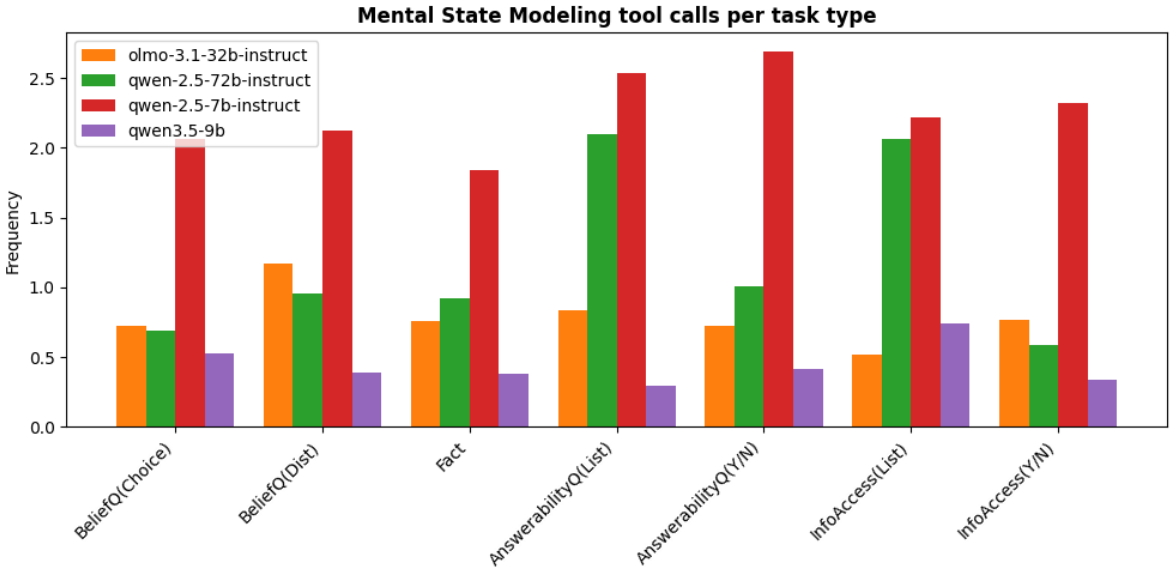

Figure 4: Average frequency of Mental State Modeling tool calls made by LMs per task type over all questions from the experiment.

5.5.1 BELIEF

Looking at Figures 4 and 3(c, d), a high number of tool calls (as seen by *Qwen 2.5 7B Instruct*) seems to negatively affect accuracy on the belief metrics (see Table 3), which seems to inflate prompt tokens. This correlation can also be seen clearly for *Olmo 3.1 32B Instruct*. As tool calls increased for BELIEF $Q_{[DIST]}$ compared to the BELIEF $Q_{[CHOICE]}$ task, accuracy gains decreased. Tools seem to add too much noise to prompts, or models overthink during reasoning stages, resulting in lower accuracy.

Qwen 3.5 9B does not seem to follow the overthinking problem – having extremely high completion token counts for both tasks, yet having an incredible accuracy growth for the open question. This could be because the multiple-choice baseline score was already very high for this model and/or because the model is better at answering open questions in general.

5.5.2 ANSWERABILITY

What stands out in the Figures 4 and 3(c, d) are the high number of tool calls for *Qwen 2.5 72B Instruct* on the list question, which is reflected in its token usage. *Qwen 2.5 7B Instruct* also notably calls the tool more often for both accuracy questions. Despite the average token usage being similar to the belief questions' token usages, the results for the answerability questions do not show much accuracy gains (Table 7), even having some losses on ANSWERABILITY $Q_{[Y/N]}$. Especially *Qwen 2.5 7B Instruct* uses many tokens for the binary yes/no question, while losing the most accuracy, seemingly overusing the tool and overcomplicating how to answer the question (in Section 5.5.4, more on this).

5.5.3 INFOACCESS

When comparing the scores in Figure 3(c, d), it can be seen that completion token usage for the two list questions is nearly identical. The difference in prompt token usage can be explained by the tool calls (Figure 4). While *Qwen 3.5 9B* increased its tool calls, the other 3 models decreased.

What stands out is that despite the high number of tool calls working against *Qwen 2.5 72B Instruct* for the answerability task, the same amount of tool calls appeared to be 3 times more efficient (see Table 7) for information access.

As for the binary yes/no questions, the models use slightly fewer tool calls and fewer tokens for this task and are marginally more efficient (see Table 7) compared to the answerability task.

5.5.4 Overall

For some questions, the model makes the deliberate choice not to use the Mental State Modeling tool, as seen in the case of *Qwen 2.5 7B Instruct*:

```
trajectory =
{‘thought_0’: "The conversation topic shifted to first date ideas and tips
when Piper joined. No need to extract Axel’s BDI state as he is not mentioned
in the conversation.",
‘tool_name_0’: ‘finish’,
‘tool_args_0’: {},
‘observation_0’: ‘Completed.’}
```

While for other cases, such as the one reported in Section 5.3, the BDI tracker was used despite the question being answerable without it. When observing Figure 4, it is seen that *Qwen 2.5 7B Instruct* tends to call the ToM modeling tool the most across all question types, due to hallucinations and falling into reasoning loops by repeatedly calling the tool for the same character, for example. This resulted in the large growth in total token usage (Table 6) as prompt token usage increased (Figure 3d) from the tool calls.

When looking at cases for this model, there is a common behavioural looping pattern that causes the spike in tool calls and prompt tokens. This is exemplified in the case of a binary yes/no question task:

1. the binary yes/no question is asked whether character A knows about topic B;
2. the agent decides that to find out whether A knows about B, the BDI tracker should be performed on A;
3. the agent gets the BDI tracker results about A, but the tool output is not specifically about topic B;
4. the agent concludes that the tool output was not enough to answer its question;
5. the agent calls the tool again;
6. the agent gets the same conclusion as 4;
7. repeat 5 and 6 until the maximum number of iterations is reached, or the agent breaks out of the loop on its own.

These reasoning pitfalls occurred for *Qwen3.5 9B* as well due to the aforementioned faulty formatting of the LM (see Section 5.4), which also caused empty tool returns (which only happened for this model), which explains the low tool-call frequencies in Figure 4. These inconsistencies and variability are thereby also a probable cause for the longer tail for Figure 2d compared to its baseline equivalent (Figure 2b).

The way these issues could be avoided is by setting the ReAct agent's *max_iterations* parameter to a lower number (the default setting is $max\_iterations = 10$, or potentially using stricter tool prompting so that it is only used when its answer is guaranteed to aid in the answering of the problem, or more specific prompting about tool use instructions, or restricting the Mental State Modeling tool to only be called once per character. The downsides of these remedies are that the tool has less freedom in cases where it actually needs more reasoning iterations or tool-calls per character. Alternatively, the tool will act conservatively and only be used when the LM is *absolutely* sure of a useful answer. This defeats the purpose of exploring the thought-space by adding more external reasoning in an attempt to improve its own.

5.6 FNR, FPR, MCC of binary yes/no questions

The binary yes/no questions have the highest accuracy scores overall (Tables 4, 5), but also the worst growth across metrics (Table 7).

What stands out is the slight decrease in the MCC score across models (Table 8), except for Qwen 3.5 9B, and the growth of FNR scores after tool usage.

Model	FNR ↓	FPR ↓	MCC ↑
Method: Baseline			
GPT-4o-2024-05-13	1.8%	7.4%	+0.917
olmo-3.1-32b-instruct	15.1%	8.1%	+0.735
qwen-2.5-72b-instruct	6.1%	6.6%	+0.861
qwen-2.5-7b-instruct	18.7 %	8.8 %	+0.6885
qwen3.5-9b	9.4 %	5.9%	+0.823
Method: Mental State Modeling tool			
GPT-4o-2024-05-13	-	-	-
olmo-3.1-32b-instruct	20.1%	2.9%	+0.726
qwen-2.5-72b-instruct	8.6%	8.8%	+0.808
qwen-2.5-7b-instruct	20.1 %	8.8%	+0.6729
qwen3.5-9b	7.6%	7.4%	+0.835

Table 8: Additional data from the FANToM benchmark performance across models and methods. The FNR, FPR, and MCC scores (for formulas, see Section 4.3.1) on the binary yes/no questions, ignoring irrelevant answers. ↓ indicates that the lower the score, the better. ↑ indicates that the higher the score, the better.

5.6.1 False Negatives

For the binary yes/no questions, given that the majority-class is 'yes', it is expected that it is more likely that models are more likely to have false positive responses. However, from examining model outputs, all of the models, except for GPT4o, are behaving more conservatively with a tendency towards false negatives instead.

Except for *Qwen 3.5 9B* and *Olmo 3.1 32B Instruct*'s FPR score, the FNR and FPR scores increase after tool-use for all others (see Table 8). *Olmo 3.1 32B Instruct* and *Qwen 2.5 7B Instruct* make larger shifts towards favouring False Negatives, while *Qwen 2.5 72B Instruct* only slightly worsens on both FNR and FPR scores. These findings suggest that the added ToM tool affects how the model makes decisions. It is possible that for these binary questions, the added information from the tools introduced unnecessary complexity and/or over-reasoning, which may explain the decrease in performance. The slight decline in MCC across models further suggests that the additional reasoning about BDI states does not improve accuracy, but while most LMs' scores worsen, overall, they remain good classifiers.

As an additional remark, for *Qwen 3.5 9B*, the main DSPy Signature was slightly adjusted in an attempt to counter the reasoning token explosion. The added phrase below could have given rise to the model providing fewer unnecessary instructions in its prompting.

```
"Be brief, but explicit and grounded in your reasoning."
```

This addition could also explain why Figure 3 shows that *Qwen 3.5 9B* is more conservative in using prompting tokens, compared to its output usages.

6 Conclusions

This paper demonstrated an implementation and description of an Agentic-LLM in the form of a DSPy ReAct agent, with an implementation of a Mental State Modeling ToM tool, which extracted the BDI states of characters present in the conversations from the FANToM benchmark, used to evaluate the accuracy and cost performance across different ToM questions. Through nine evaluations of the tool with different LMs and tool-access variability, it can be found that although the extent of improvement varies per model and task, the addition of external reasoning about human intent adds overall improvement to the accuracy scores of the FANToM benchmark. More specifically, from the reported findings, the following conclusions and lessons are derived:

Firstly, across all evaluated models, **it can be found that there is a change in performance on the FANToM benchmark using the Mental State Modeling tool compared to the baseline method, with most metrics showing a positive effect on FANToM accuracy.** The results of this experiment suggest that **the accuracy increase of tasks is closely linked with the task type** and show the existing variance per model. For this experiment, *Olmo 3.1 32B Instruct* shows the most growth in accuracy across metrics after tool-use, with *Qwen 3.5 9B* having the highest accuracy scores overall, though this may also be more likely a result of the ReAct paradigm alone, as it had trouble adapting to DSPy and could not properly use the Mental State Modeling tool. *Qwen 2.5 7B Instruct* seems to perform the worst both in growth and overall accuracy, being the least efficient model overall.

Second, **performance declines are observed in the binary yes/no question tasks, as well as for models that had many tool-calls** as these binary baseline scores were already relatively high to start, and **the tool over-reliance seemingly added more noise, which caused overthinking, resulting in harmed accuracy scores.**

Third, a balance is needed between getting the most out of the tool without having an explosion of tokens. **Determining the cost-efficiency of models and identifying which tools suit which types of questions will help enable realistic deployments of tools for Agentic-LLMs.**

Fourth, comparing the results to prior works: most measured scores are still worse than the human baseline. This suggests that **just the addition of the Mental State Modeling tool is not enough to simulate human social understanding.**

Lastly, looking at the cost performance of the models (see Table 7), it can be seen that ***Olmo 3.1 32B Instruct* is the most efficient, with the best accuracy-per-token gains across most tasks.** What is interesting is that despite *Qwen 3.5 9B* having high accuracy scores, overall, its cost performance leaves much to be desired, but still is better than *Qwen 2.5 7B Instruct*, which struggled on 3 out of 7 task types. INFOACCESS $Q_{[LIST]}$ benefited the most from tool usage, followed by both BELIEF tasks.

These conclusions highlight that the Mental State Modeling tool improves the ToM performance of an Agentic-LLM on specific tasks. At the same time, these findings suggest further research is needed to improve generalisability across ToM tasks and to investigate the trade-off between improved performance and overthinking. Such research has the potential to contribute to more effective human-AI collaboration while also considering realistic deployment.

6.1 Limitations, challenges, and future research

There are a few limitations and challenges in this paper, leaving open questions and possibilities for future research.

For one, there were resource constraints. Because of the high usage costs of *GPT-4o* (see Table 6), only the baseline was evaluated. As the baseline accuracies were relatively high compared to other models, it would be intriguing to examine the effects of tool usage. Additionally, because the benchmark only runs one LM at a time, and a limited time was available, more LMs, the full benchmark (instead of a subset), other benchmarks, and different tools were not evaluated.

Also, for these reasons, the abilities of the Mental State Modeling tool were not evaluated on a BDI tracking benchmark. For more clarity on the effects of the tool, determining the accuracy of the tool itself would aid in finding the usability of the outputs.

In the case of this thesis, only the effects of the Mental State Modeling tool are investigated, but different sorts of ToM tools exist too, such as *Perspective-Taking* and *Reflection and Synthesis* [20]. Other tools, or combinations of tools, may cause greater improvements on metrics that the Mental State Modeling tool struggles with – it is also worth investigating which tools are worth investing more tokens in, as well as variability in their parameters.

Then, an encountered challenge during the evaluation of the *Qwen3.5 9B* agent was its bugged output behaviour. Because of outputs that inconsistently did not stick to the predefined DSPy scheme, the benchmark timed out several times, which had to be restarted manually and repeated until suitable answers returned. This inconsistent behaviour is also seen in the various figures in this paper, making the figures with reported token usages extremely skewed – for readability, these figures were zoomed in on 95% of the data, leaving out the extreme outliers. Because this LM failed to properly utilize the experimental setup, it is possible that its results do not fully reflect the effects of the DSPy implementation and the tool usage.

Furthermore, as in related studies [10, 19, 23], this thesis does not claim that performance improvements for the FANToM benchmark are the same as definitive improvements in the social abilities of Agentic-LLMs. Rather, Agentic-LLMs are able to answer social questions, such as the ones found in the FANToM benchmark, more accurately. To answer whether the overall ToM question-answerability improved, more tests should be done on different ToM benchmarks. While the question remains whether ToM benchmarks are an accurate way to determine the social intelligence of LLMs at all, the challenges and limitations of this thesis highlight that there is much room to discover novel ways to improve Agentic-LLMs' grasp of social interaction through question-answering.

Lastly, although this implementation is limited to a conversational benchmark, emerging work that DSPy supports multi-modal integration [3, 4] suggests promising directions for future research. Continued work in this domain may contribute towards Agentic-LLM deployment in real-world social environments, enabling more possibilities beyond text-bound interaction, such as using tone of voice or non-verbal social cues like facial expressions or body language.

References

[1] BerriAI. LiteLLM Documentation, 2026. Available at: https://openrouter.ai/models. Accessed: 10 July 2026.

[2] Davide Chicco and Giuseppe Jurman. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1), jan 2020.

[3] DSPy. Image Generation Prompt iteration. Available at: https://dspy.ai/tutorials/image_generation_prompting/. Accessed: 1 June 2026.

[4] DSPy. Tutorial: Using Audio in DSPy Programs. Available at: https://dspy.ai/tutorials/audio/. Accessed: 7 July 2026.

[5] Qwen et al. Qwen2.5 Technical Report, 2025. arXiv preprint arXiv:2412.15115.

[6] Google Trends. Google Trends: "ChatGPT". Available at: https://trends.google.com/explore?geo=Worldwide&date=today%205-y&q=chatgpt. Accessed: 19 February 2026.

[7] Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. Demonstrate-Search-Predict: Composing Retrieval and Language Models for Knowledge-Intensive NLP, 2023. arXiv preprint arXiv:2212.14024.

[8] Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. DSPy: Compiling Declarative Language Model Calls into Self-Improving Pipelines. In *The Twelfth International Conference on Learning Representations*, 2024.

[9] Hyunwoo Kim, Melanie Sclar, Xuhui Zhou, Ronan Le Bras, Gunhee Kim, Yejin Choi, and Maarten Sap. 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*. Association for Computational Linguistics, 2023.

[10] Tom Kouwenhoven, Michiel van der Meer, and Max van Duijn. Traces of Social Competence in Large Language Models. In *Proceedings of the 30th Conference on Computational Natural Language Learning*, pages 742–759, 2026.

[11] James O'Donnell and Casey Crownhart. We Did the Math on AI's Energy Footprint. Here's the Story You Haven't Heard, May 2025. Available at: https://www.technologyreview.com/2025/05/20/1116327/ai-energy-usage-climate-footprint-big-tech/. Accessed: 19 February 2026.

[12] OLMo Team. OLMo 3, 2026. arXiv preprint arXiv:2512.13961.

[13] OpenAI. GPT-4o System Card, 2024. arXiv preprint arXiv:2410.21276.

[14] OpenRouter. OpenRouter Models, 2026. Available at: https://openrouter.ai/models. Accessed: 14 April 2026.

[15] OpenRouter Documentation. Tool & Function Calling Guide: A Simple Agentic Loop. Available at: https://openrouter.ai/docs/guides/features/tool-calling#a-simple-agentic-loop. Accessed: 19 February 2026.

[16] Andrea Passerini, Aryo Gema, Pasquale Minervini, Burcu Sayin, and Katya Tentori. Fostering effective hybrid human-llm reasoning and decision making. *Frontiers in Artificial Intelligence*, 7, 2025.

[17] Aske Plaat, Max Van Duijn, Niki Van Stein, Mike Preuss, Peter Van der Putten, and Kees Joost Batenburg. Agentic Large Language Models, a Survey. *Journal of Artificial Intelligence Research*, 84, December 2025.

[18] Qwen Team. Qwen3.5: Towards Native Multimodal Agents, February 2026. Available at: https://qwen.ai/blog?id=qwen3.5. Accessed: 1 June 2026.

[19] Maarten Sap, Ronan Le Bras, Daniel Fried, and Yejin Choi. 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, 2022.

[20] Sneheel Sarangi, Chetan Talele, and Hanan Salam. Agentic-ToM: Cognition-Inspired Agentic Processing for Enhancing Theory of Mind Reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 25645–25661. Association for Computational Linguistics, November 2025.

[21] Zhuocheng Shen. LLM With Tools: A Survey, 2024. arXiv preprint arXiv:2409.18807.

[22] The Hybrid Intelligence Centre. Research. Available at: https://www.hybrid-intelligence-centre.nl/research/. Accessed: 19 February 2026.

[23] Ramira van der Meulen, Rineke Verbrugge, and Max van Duijn. Towards properly implementing Theory of Mind in AI systems: An account of four misconceptions, 2025. arXiv preprint arXiv:2503.16468.

[24] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[25] Bo Yang, Jiaxian Guo, Yusuke Iwasawa, and Yutaka Matsuo. Large Language Models as Theory of Mind Aware Generative Agents with Counterfactual Reflection, 2025. arXiv preprint arXiv:2501.15355.

[26] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing Reasoning and Acting in Language Models, 2023. arXiv preprint arXiv:2210.03629.

7 Appendix

A. Full conversation example

This is the full example of the conversation input demonstrated in Section 5.3.

Input text which goes to the LM (Consists of FANToM [9] benchmark scenario and the task.)

Bryce: Hi Mark, nice to meet you. Do you have any experience with nature photography?
Mark: Hi Bryce, pleasure to meet you as well! Yes, I've dabbled a bit in nature photography. How about you?
Bryce: Same here. I am fascinated by the tranquility that nature radiates, and I try to perpetuate it through photography.
Mark: That's amazing! I share your sentiment about nature's calmness. It is honestly captivating how one single frame can encompass that beauty and allow people to feel it too.
Bryce: Exactly, Mark! I believe that every photograph tells a story and it's incredible how nature always provides a unique narrative.
Mark: Couldn't agree more. I've also noticed that every time I go back to photograph the same place, I find something new to capture. Nature's dynamic character fascifies me.
Bryce: I couldn't put it better myself. Even after being in this field for years, every day feels like a new adventure. Ever had a mind-blowing moment during your shots?
Mark: Absolutely, Bryce. Once, I was photographing hummingbirds in a garden. Suddenly, they flew towards the camera, creating a beautiful image with the close interaction. How about you?
Bryce: That would have been a sight to remember! For me, it was this time when I was capturing a sunrise in the mountains. A herd of deer walked into the frame, making it a perfect, picturesque moment.
Mark: Bryce, those spontaneous moments do make the best shots! Sharing stories and experiences like these are truly the essence of a photographer's journey.
Bryce: Couldn't have said it better, Mark! Here's to many more nature capturing adventures. Cheers!
Mark: To the joy of capturing the beautiful intricacies of nature, Cheers!
Alfredo: Hey, Bryce and Mark! The joy in your eyes as you talk about nature photography is contagious! It brought back memories of some of my hiking and camping experiences. Ever tried that?
Bryce: Absolutely, Alfredo. Hiking and camping provide a whole new perspective on connecting with nature. My most memorable experience was during a camping trip to Yosemite where the night sky was mesmerizing.
Mark: That sounds wonderful, Bryce. I remember my first overnight hike to the Adirondacks. The sight of the sun rising above the mountains from our camp was unforgettable.
Alfredo: Those experiences are indeed one of a kind! In my case, during one of my hiking trips, we got caught in a sudden downpour. Despite the initial panic, it turned out to be an unique experience. The forest was so incredibly vibrant after the rain!

Bryce: Each adventure offers unique moments indeed! Once, on a full moon night, the trail almost lit up with the moonlight, and I got some lovely shots.
Mark: Amazing! Experiences in hiking and camping integrate seamlessly with our love for nature photography, don't they? It constantly opens up new opportunities to capture the unique beauty of nature.
Alfredo: That's so true, Mark! Here's to many more beautiful moments encapsulated in our frames and memories!
Mark and Bryce: Cheers, Alfredo! Cheers to that!

Target: What were Bryce and Mark's thoughts about the unique narrative created by nature in their photos?
Question: List all the characters who know the precise correct answer to this question.
Answer: