

Language Models Might Not Understand You: Evaluating Theory of Mind via Story Prompting

Nathaniel Getachew & Abulhair Saparov

{ngetach, asaparov}@purdue.edu

Abstract

We introduce StorySim, a programmable framework for synthetically generating stories to evaluate the *theory of mind* (ToM) and *world modeling* (WM) capabilities of large language models (LLMs). Unlike prior benchmarks that may suffer from contamination in pretraining data, or rely on an LLM for generation, StorySim produces novel, compositional story prompts anchored by a highly controllable STORYBOARD, enabling precise manipulation of character perspectives and events. Using StorySim, we evaluate LLMs across three kinds of ToM tasks: false belief, goal-directed, and preference attribution tasks. We then use our framework to design first- and second-order ToM tasks alongside WM tasks that control for the ability to track and model mental states. Our experiments across a suite of LLMs show that most models achieve higher accuracy on WM tasks than on ToM tasks, and that some models tend to reason more accurately when the subject of reasoning is a person rather than an inanimate object. Additionally, models struggle with goal-directed reasoning the most, and performance does not always scale with model size. All code for generating data and evaluations is freely available.¹

1 Introduction

Large language models (LLMs), especially models meant for reasoning, have shown increasing improvement on benchmarks that were once infamously challenging. Contemporary models have made progress in reasoning through math problems, coding, and pure logical deduction (El-Kishky, 2024; Suma & Dauncey, 2025; Xia et al., 2025). As reasoning capabilities of LLMs continue to improve, we focus on LLMs’ ability to reason about mental states and the perspectives of other intelligent agents, a concept known as *theory of mind* (ToM). This has become increasingly important, as an LLM without a sound ToM might not be able to understand the perspective of a user with harmful intent, or consider the effect its decisions could have on other people, who have their own goals and wishes. Reasoning about the perspective of other agents requires understanding how those agents’ views are affected by changes in the environment. The ability to learn and adhere to these environmental rules is called world modeling (WM) (Bower & Morrow, 1990; Ha & Schmidhuber, 2018; Vafa et al., 2024). In principle, agents with strong WM should exhibit ToM, since other minds exist within the same world.

ToM is often assessed in children by evaluating their ability to understand that others hold mental representations of the world that differ from their own. The prototypical ToM test is known as the *Sally-Anne* test (Baron-Cohen et al., 1985), where children are given a story akin to the following:

Classic Sally-Anne story

Sally and Anne walk into a bedroom. Sally puts a toy on top of the bed and leaves. Anne takes the toy and hides it under the bed. When Sally comes back into the room, where is the first place she will look for the toy?

¹Code for and data generated by StorySim is available here: github.com/ngetachew/StorySim

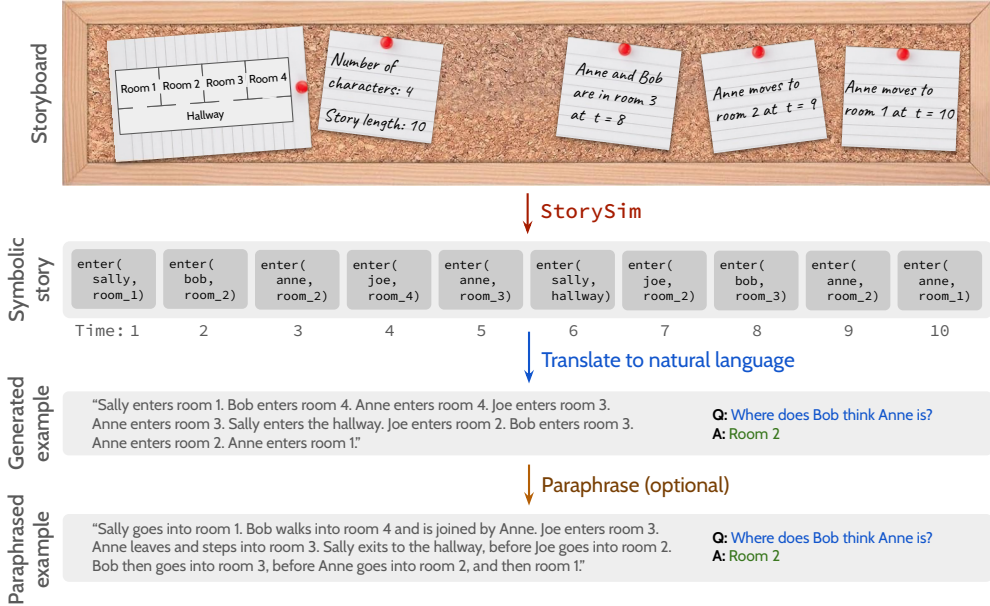


Figure 1: An example STORYBOARD is shown at the top, consisting of a graph defining the setting, story length, and timestamped key events. StorySim generates a story consistent with the graph and events, where characters move between adjacent locations (starting in the hallway). Characters can observe others’ movements when located in the same room. The resulting events are converted into a natural language QA example and may be further paraphrased into more naturalistic language. This type of story is used in our first-order mislead distance experiments.

It may seem obvious to adults that the first place Sally will look is on top of the bed. However, it has been shown that children under the age of 4.5 think that Sally will look under the bed (Memisevic et al., 2018). This is hypothesized to be due to the fact that they have not yet developed ToM, so they cannot understand the fact that Sally and Anne have a different understanding of the toy’s location.

LLMs have been evaluated on problems of this nature, and initially showed success (Kosinski, 2023). However, small perturbations of the Sally-Anne test often confound LLMs, indicating that this test has contaminated pretraining data, and LLMs may have overfit to this specific problem, and therefore, they may not necessarily have a sound ToM mechanism (Ullman, 2023).

In order to measure ToM ability in a way that is robust against data contamination, we developed a flexible, synthetic benchmark. We propose StorySim, a framework that allows us to generate stories to evaluate ToM and WM abilities of LLMs. The core events of the story are described by a high level specification called the STORYBOARD. Aside from the STORYBOARD, the rest of the story is randomly-generated. The generated story is represented symbolically and is guaranteed to be cohesive and consistent. An example of the framework in action is shown in Figure 1. The STORYBOARD provides a description of the setting in which the story will take place, including which locations are accessible from each other, as well as required actions characters need to make, the length of the story, the number of characters, etc. StorySim then generates a story that is coherent with the STORYBOARD, filling in all other details randomly in a way that is consistent with the provided constraints. Due to the exact specificity of the events that occur in StorySim via the STORYBOARD, the randomness of the events in between, and the length of the stories we generate, the chance that any synthetically-generated story in our experiments has appeared in the training data of any LLMs is astronomically small.

Benchmark	Targets LLM ToM Abilities	Guarantees Story Consistency	Tests for OOD Generalization	Diverse Story Flow	Generation Not Dependent on LLM	Tests for Higher-order ToM	Can Generate Arbitrarily Many Examples
FanTom (Kim et al., 2023)	✓	●	×	×	×	●	●
ToMBench (Chen et al., 2024b)	✓	×	×	×	×	×	×
BigToM (Gandhi et al., 2023)	✓	✓	×	×	×	×	●
OpenToM (Xu et al., 2024)	✓	×	✓	●	●	×	×
MindGames (Moore et al., 2025)	✓	×	✓	×	✓	●	×
ExploreToM (Sclar et al., 2024)	✓	✓	×	✓	×	×	✓
HiToM (Wu et al., 2023)	✓	✓	×	×	✓	×	×
ToMi (Le et al., 2019)	✓	✓	×	×	✓	×	×
StorySim (ours)	✓	✓	✓	✓	✓	✓	✓

Table 1: Comparison of existing ToM benchmarks to ours. StorySim uniquely possesses all of the characteristics described above. Check marks (✓) indicate benchmarks that implement the described feature, X’s indicate benchmarks that do not, and black dots (●) indicate benchmarks that could feasibly implement the feature, but only with substantial alteration.

StorySim allows us to systematically evaluate current LLMs, which we utilize to address the following research questions:

- RQ1** Can we measure ToM abilities of LLMs in a way that is robust to data contamination and heuristics?
- RQ2** To what extent do LLMs rely on memorization and heuristics to answer ToM questions?
- RQ3** Can ToM abilities in LLMs be evaluated independently of their general WM capabilities? Relatedly, do LLMs show stronger performance on general WM tasks or on ToM-specific reasoning?

2 Related Work

Theory of Mind: Existing research benchmarking ToM in language models focuses on prompting with stories followed by question-answering (Grant et al., 2017; Sap et al., 2019), including the TOMi dataset (Le et al., 2019). However, these are outdated, and have very likely leaked into the pretraining data of contemporary LLMs. In fact, Ullman (2023) demonstrates that trivial alterations of the TOMi dataset causes model performance to drop. In order to address data leakage, newer benchmarks have been created (Wu et al., 2023; Chen et al., 2024b). Specifically, we position our work as similar to Sclar et al. (2024), where the authors present EXPLORETOM, a framework that allows for graph-based synthetic story generation and question answering. As in our work, this setup ensures that generated examples are relatively unseen by LLMs. However, while EXPLORETOM uses an LLM to turn fixed contexts into stories, our framework generates sentences directly from a STORYBOARD. This gives precise control over when events occur, enabling highly targeted tests (e.g., we can vary the exact location of state-change events in order to probe LLM behavior). In Appendix D, we demonstrate that LLM-generated stories often fail to meet such constraints, while StorySim reliably produces stories of any specified length, set of characters, and events. Comparisons of our framework and others are further highlighted in Table 1. Huang et al. (2024) introduce a complexity measure for ToM tasks and a prompting method that makes implicit character state attributes explicit, offering a formal representation of ToM events similar to ours. However, we test whether LLMs can reason over implicit character states without such scaffolding. Our framework can also add extraneous state changes from irrelevant characters, but as Figure 7 demonstrates, these have little effect on ToM performance, contrasting with findings found in previous work.

World Modeling: Though there is much work evaluating aspects of WM abilities of LLMs (Kociský et al., 2017; Qiu et al., 2025; QianyiHu et al., 2025), there are none, to the best of

our knowledge, that examine the relationship between WM and ToM ability. We consider ToM to be an extension of world modeling since ToM requires the explicit modeling of mental states and how they may change over time, and mental states are dependent on the goals, limitations, and perspectives of agents. In the stories we generate, the character’s perspectives only consist of what they observe around them, so to reason over this requires understanding the world in which they live and the limitations of each character’s perception and knowledge. Wang et al. (2023) train a narrator and a role player model to interact within role-playing game environments. This setup implicitly requires both models to construct and maintain an understanding of the environment (WM), while the narrator must also infer the role player’s perspective to guide narrative choices (ToM). However, the task itself does not directly evaluate either ToM or world modeling capabilities in isolation.

Synthetic Datasets: Though existing synthetic datasets meant for benchmarking LLMs exist (Patel et al., 2024; Shvets, 2025; Mishra & Prabakeran, 2024), there are none that evaluate ToM and WM abilities as we do. Synthetic datasets have been shown to be an effective method for training LLMs to perform well on logic tasks (Xie et al., 2025), clinical text classification (Tang et al., 2023), deductive reasoning (Saparov et al., 2023), and more. The diversity of synthetic data is an important factor in the evaluation of performance on both in- and out-of-distribution data (Chen et al., 2024a; Havrilla et al., 2024). For this reason, we developed StorySim to be highly programmable and able to generate a wide range of stories.

3 Methods

Below we discuss the details of StorySim’s generation process, the breadth of stories it can create, and how we leverage it to answer our research questions.

3.1 StorySim

Broadly, the StorySim framework generates each story in two steps: (1) we construct a STORYBOARD capturing high-level story specifications, and (2) StorySim uses it to produce a story by generating event sequences that satisfy these constraints while filling in the remaining details coherently.

A STORYBOARD is defined as $D = (C, \phi, G, E, n)$, where C is a set of characters, G is a directed graph describing the locations in the story, with edges denoting which locations are immediately reachable from other locations, E is a set of programmed events, and n is the number of events. The StorySim framework generates a sequence of n events of the form $\phi(c, l)$, where ϕ is the action, $c \in C$ is the character performing the action, and $l \in V(G)$ is the location of the event. Events follow the structure of G , so characters move only between adjacent vertices. Only events in E are deterministic, specifying actions at particular time steps; all others are randomly generated but remain consistent with E . The exact parameters of D in our experiments are given in Appendix A. An example is shown in Figure 1. These events are translated into natural language via templates mapping each $\phi(c, l)$ to a sentence. Optionally, the stories are further paraphrased into more naturalistic language by GPT-4o-mini (Hurst et al., 2024).

Though a simple framework, StorySim can be used to create a wide variety of stories. Depending on the desired story, the framework requires varying degrees of manual input. However StorySim can be used to simulate any story where each action can be modeled as a relation (or set of relations) that act over a subject and a node in the graph. The subject does not necessarily need to be a human, and the nodes need not be locations. Additionally, the stories can feature manually-created actions as well, which can add another layer of complexity to the events described by StorySim. Appendix F illustrates the range of stories that can be generated by the framework for ToM evaluation, and contains examples of stories we generate in our main experiments, as well as additional stories to further highlight the framework’s breadth.

3.2 Experimental Setup

StorySim can generate stories with either simpler sentences (i.e., template-generated) or more linguistically naturalistic sentences (i.e., paraphrased). In most of our experiments, we utilize the simpler language to minimize confounding effects from linguistic complexity, ensuring errors reflect limitations in the targeted ability rather than misinterpretation—we aim to measure the model’s ToM capability independently of the model’s ability to parse sentences. In order to explore the effect of linguistic complexity on LLM ToM performance, we run experiments using LLM-paraphrased versions of the stories; these results are reported in Figure 5.

In order to test ToM abilities, we generate two types of ToM problems:

First-order ToM problem: Ask about one character’s knowledge of another character’s location. E.g.: “**Where does Bob think Alice is?**”

Second-order ToM problem: Ask about one character’s knowledge of a second character’s knowledge of a *third* character’s location. E.g.: “**Where does Bob think Charlie thinks Alice is?**”

To generate stories that allow us to ask questions like the ones above, we create two types of STORYBOARDS which specify events for characters S_1 , S_2 , and T , using the Hallways and Doors environment shown in Figure 1. In the above examples, S_1 is Bob, S_2 is Charlie, and T is Alice. We use StorySim to generate stories for first-order ToM by specifying a high-level pattern within the STORYBOARD: S_1 and T first meet at a shared location, after which T moves elsewhere. Then, without S_1 ’s knowledge, T moves to a third location and remains there for the rest of the story. Similarly, for second-order ToM, we define a pattern where S_1 , S_2 , and T initially meet, S_2 and T later meet again at a different location, and T subsequently moves to another location alone. Given these STORYBOARDS, we can ask:

ToM Prompt: *Where does S_1 think S_2 thinks T is?*

WM-Human Prompt: *When S_1 and S_2 were in the same room as T , where did T go?*

WM-Inanimate Prompt: *When S_1 and S_2 were in the same room as T , where was T moved to?*

The first question type requires the LLM to understand the perspective of S_1 and their perceived location of S_2 and T , thereby directly testing its ToM capabilities. The second question type is equivalent to the first, but instead of asking the model to consider the perspective of S_1 , we prompt it to algorithmically search through the story to find an event where S_1 ’s perception of the locations of S_2 and T changed. Thus, the second question is a test of *world modeling* capability, since we do not ask about the mental state of any character, and answering the question requires understanding the events in the story. Finally, the third question type is equivalent to the second, but instead of describing human subjects, the story describes inanimate objects, where each object is moved by some third-party outside the story. This question is meant to test whether the LLM’s world modeling ability is sensitive to whether the entities in the world are animate or inanimate. These experiments enable us to directly test RQ3. We must note that the phrasing of the **WM-Inanimate Prompt** implies that the objects are moved to different locations by some unnamed force. Our reliance on this unnamed force is unavoidable since objects cannot spontaneously move on their own in the real world, no matter how the story is phrased. The results of evaluating a suite of LLMs on these three prompts are shown in Figures 2.

Additionally, given the STORYBOARD for a first-order or second-order ToM problem, we can vary the number of time steps between the event where S_1 last perceives T ’s location and the event where S_1 ’s beliefs are no longer true. A reasonable hypothesis about the ToM abilities of LLMs is that since T ’s last movement is a more recent event in the story, LLMs may utilize a heuristic where they predict T ’s last location, despite the fact that S_1 did not see this movement. We refer to this number of time steps between these two events as the *mislead distance*. We create STORYBOARDS for both first- and second-order ToM problems, varying the mislead distance and using the first-order **ToM Prompt**. We evaluate multiple LLMs on these examples across multiple mislead distances, and conduct an error analysis to see if patterns emerge when LLMs fail to do proper ToM reasoning. The results of this are discussed in Section 4.4. Raw accuracies for additional models are presented in Figure 8,

and an analysis of these results and as well as an investigation into whether or not models suffer from recency bias are in Appendix B.

Task	GPT-4	R1
Sally-Anne	100%	100%
Random Story	75%	87%

Table 2: Accuracy on Sally-Anne vs other ToM tasks. **(top row)** Models are prompted with the exact Sally-Anne setup shown in Section 1, in a story with 5 events. **(bottom row)** Models are given a randomly-generated first-order ToM story of the same length.

and WM capabilities. We prompt the models using the first-order **ToM Prompt** and **WM-Human Prompt** and report the results in Figure 7.

To further demonstrate the breadth of StorySim, we additionally evaluate LLMs on preference attribution and goal-directed ToM tasks, and display their performance in Figure 4. This allows us to contextualize the difficulty of the two mislead distance experiments with other tasks, and gives us insight into which models have the best all-around ToM capabilities. We utilize the same synthetic generation approach to generate examples for these additional tasks as we do in the mislead distance experiments, and therefore are also less likely to be affected by LLM data contamination. Further details about the construction of the goal-directed and preference attribution ToM stories are discussed in Appendix F.

Finally, we repeat the mislead distance experiments using the first-order **ToM Prompt** under two additional story environments, detailed in Appendix F.1, and compare these results with versions of each story paraphrased by an LLM (GPT-4o-mini). These environments are described at length in Appendix F. To encourage linguistic diversity, we set the temperature to 1.3. This experiment allows us to examine the effects of the templated structure of StorySim stories, as well as the impact of varying relations and location graphs within the STORYBOARD.

All four of the experiments above present LLMs with stories that are constructed to be dissimilar to stories seen in pre-training data and have answers that can be mapped to certain heuristics, allowing us to answer RQ1. Our error analysis in Section 4.4 allows us to gain an understanding of the heuristics used when LLMs struggle to answer StorySim’s questions, which enables us to test RQ2.

4 Results

To gauge the difficulty of our benchmark, we asked 12 graduate and undergraduate students, with no prior knowledge of the task, to answer ToM questions about StorySim stories. Each student was shown 50 examples from the Hallways and Doors environment shown in Figure 1 across various mislead distances, achieving an average accuracy of 96%.

4.1 Sally-Anne Is No Longer a Good Test for ToM

To highlight the inadequacy of benchmarks that rely on the Sally-Anne problem to measure ToM, we used StorySim to mirror the Sally-Anne experiment using GPT-4 (Achiam et al., 2023) and R1 (Suma & Dauncey, 2025), using a STORYBOARD that specifies a first-order ToM problem with 5 events matching the Sally-Anne story setup exactly. As shown in Table 2, both of these models had perfect scores on this task, far from the accuracy when prompting the models using randomly-generated stories of the same length. Additionally, in order to get a sense of whether or not our generated stories appear in pre-training data, we performed an n-gram overlap test with the C4 dataset, a web crawl corpus often included in pre-training data of LLMs (Raffel et al., 2019). We checked for exact matching 5-grams in

We also explore whether and to what extent models have overfit to simpler ToM questions by significantly increasing the number of subjects in the stories. We hypothesize that models may have seen question-answering prompts based on stories during pretraining, but it is unlikely they have seen examples where the stories contain large numbers of characters. Being able to do so would imply that the models can generalize more robustly to stories containing many characters, and would therefore provide evidence of more robust/generalizable ToM

the first-order mislead distance experiments and their paraphrased equivalents, and found no matching sequences.

4.2 Evaluating ToM vs WM Capabilities

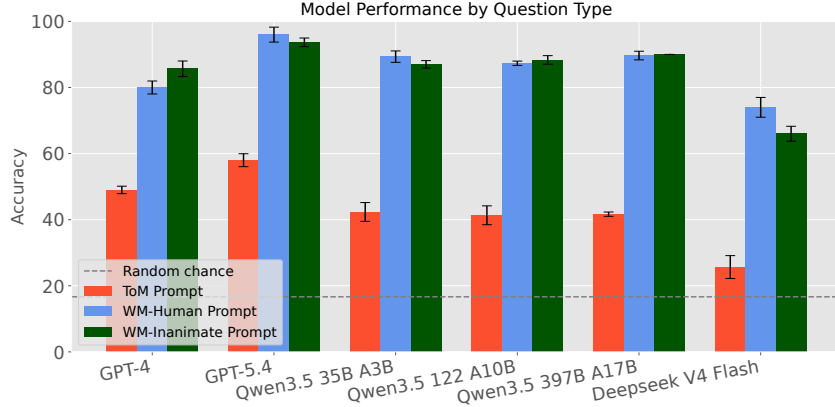


Figure 2: Zero-shot performance per model using the **ToM Prompt**, **WM-Human Prompt**, and **WM-Inanimate Prompt**. For each of these prompts, the models are given a story containing 100 events, and we give 100 such stories to each model. Black bars represent the 95% confidence interval.

We tested a suite of LLMs using the prompts discussed in Section 3. Starting with a STORYBOARD describing a second-order ToM event, we generate 100 stories consisting of 100 events with which we query each model.

Figure 2 shows the results of the experiment discussed in Section 3 using GPT-4, GPT-5.4 (OpenAI, 2025), Deepseek V4 Flash, and the Qwen 3.5 family of models (Team, 2026). Across all models, ToM performance is far worse than WM, demonstrating that it is indeed a much more difficult problem. Interestingly, GPT-4 and GPT-5.4 only have a 9% difference in performance in the ToM setting despite the two year difference in release dates. The Qwen family of models exhibit nearly identical performance across all 3 settings. In these two families, the models demonstrate a negligible difference in performance between the WM Human and WM Inanimate settings, indicating that modeling the movements of human subjects vs inanimate objects is not significantly different in terms of task complexity. However, V4 differs from the others in having the worst performance across the board, and its WM-Inanimate accuracy being much higher than WM-Animate accuracy as compared to all other models.

To understand how ToM performance relates to WM ability, we introduce the *ToM Advantage*: the number of questions answered correctly on ToM but incorrectly on WM, minus the number answered correctly on WM but incorrectly on ToM. A positive value would indicate that a model sometimes succeeds at ToM despite failing the underlying WM task, while a negative value confirms that WM is easier than ToM. As shown in Figure 3, all models have a negative ToM Advantage, confirming that WM is an upper bound on ToM performance. Differences within the GPT and Qwen 3.5 families are small, though GPT-4 has the least negative advantage of the group. Interestingly, V4’s ToM Advan-

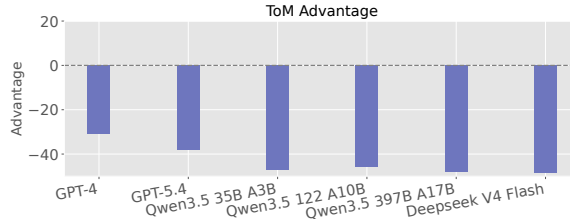


Figure 3: Performance by the Llama 3.1 Instruct family of models, of varying sizes, using 3-shot examples given the **ToM Prompt**, **WM-Human Prompt**, and **WM-Inanimate Prompt**. Similar to Figure 2, these stories have 100 events each.

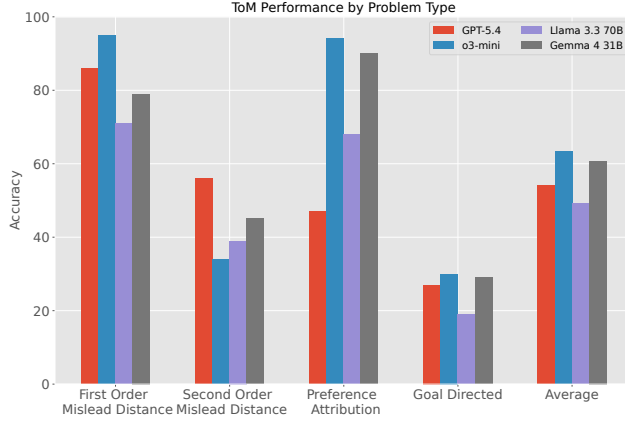


Figure 4: Using StorySim, we evaluate LLMs on the first- and second- order mislead distance task, as well as common other ToM tasks generated by our framework. In the mislead distance experiments, the distance is fixed to 30.

tage is close to the Qwen 3.5 models, despite performing far worse on the ToM task. The Qwen models are notable as the only ones that never answer ToM questions correctly while failing the equivalent WM question. Full confusion matrices for each model are provided in Appendix C.

4.3 LLM Performance on StorySim ToM Tasks

Figure 4 shows the results of evaluating models on multiple ToM tasks generated by StorySim. Broadly, the first-order mislead distance experiments are the easiest ToM task in our suite of evaluations. The most difficult task seems to be the goal-directed reasoning task, where every model sees its lowest score. Averaging across tasks, o3-mini has the best performance across all tasks, with it being the best performing model in three out of the four tasks. In the second-order mislead setting, however, it has the worst performance across all models. Additionally, the results break the parameter count ordering, with Gemma 4 31B outperforming larger models GPT-5.4 and Llama 3.3 70B.

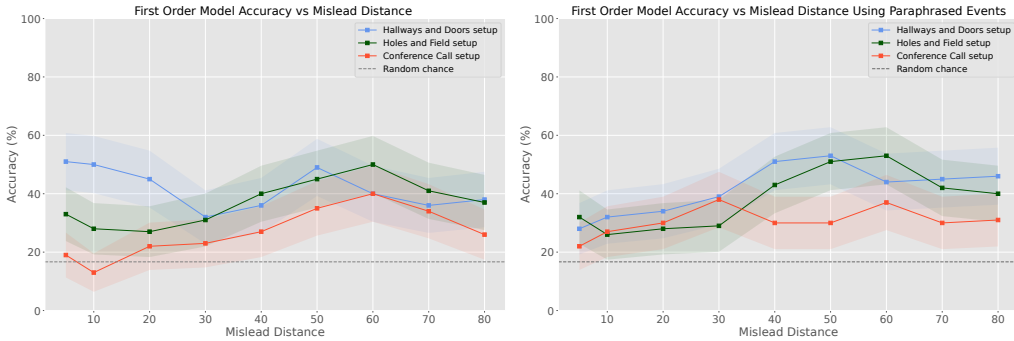


Figure 5: **(Left)** Model accuracy vs mislead distance using GPT-4 using three different STORYBOARDS featuring characters entering and exiting rooms from a hallway, characters in a field jumping in and out of holes in the ground, and characters joining and leaving conference calls. Otherwise, the stories are generated in the same way. **(Right)** The same experiment as on the left but using GPT-4o-mini to paraphrase the generated events.

The mislead distance experiments are constructed with specific STORYBOARDS as discussed in Section 3. In order to explore the effect of this specific construction on performance, we evaluate LLMs on STORYBOARDS defined with different G and ϕ . Figure 5 (left) shows that

across different environments, performance is similar across most mislead distances. Where larger differences do appear, the Hallways and Doors setup performs best, and therefore we use this setup across all other experiments. As evident in Figure 5 (right), paraphrasing the events only causes a meaningful decrease in performance for one story setup at small mislead distances. Otherwise, we do not observe any meaningful difference in performance, suggesting that the simpler language of non-paraphrased examples helps models to solve the task as compared to the more naturalistic language of paraphrased examples.

4.4 Do LLMs Rely on Heuristics for ToM?



Figure 6: Number of mistaken examples (out of 100), with each example categorized by error type, for the first-order (**top**) and second-order (**bottom**) ToM mislead experiments on (**left**) GPT-4, (**middle**) R1, and (**right**) o3-mini.

In this section, we conduct an error analysis to see if LLMs follow specific shortcuts when they fail on our ToM task. In the first-order case, we chose to focus on R1 and GPT-4 for our error analysis, since the former has the strongest evidence of a performance correlation with mislead distance, and the latter performs the worst overall in the group (see Figure 8). In the second-order case, o3-mini performs the worst despite being the best in the first-order task, so we analyze that as well. The most common error for GPT-4 is prompt refusal, where the model responds with “not enough information”, while for R1 it is insufficient context length, where the model runs out of context tokens when generating a long chain-of-thought (see Appendix G for an example). We attempted to address this by experimenting with different prompts to guide the models, but as demonstrated in Appendix E, there are no significant differences between the prompt used in the final experiments and other similarly-performing prompts and strategies.

These errors both result in the model not producing an answer, however they have different implications with respect to ToM capability. GPT-4’s prompt refusal indicates that the second-order case elicits a level of complexity that is enough to convince the model that the necessary belief state information is not available. However, as shown in Appendix G, R1 employs the general strategy of reasoning through the events of the story and the evolution of the character’s beliefs. However, it cannot reach the final answer, and instead gets lost trying to re-create the story.

Aside from this, we also found that the most common error is to repeat the location of the first place where characters are in the same location. For example, in Figure 1, this corresponds to predicting room_4, where Bob and Anne first cross paths en route to room_3, as specified by the STORYBOARD. We call this the *First Common Location Heuristic*. For both the first- and second-order ToM tasks, we quantify the relative proportions of each error type in Figure 6.

In the first-order experiments, R1’s improved performance at larger mislead distances is largely driven by fewer insufficient context length errors. GPT-4 exhibits the First Common Location Error at a roughly constant rate across mislead distances, with its accuracy varying mainly due to differing rates of prompt refusal. Moreover, GPT-4 predicts locations other than the initial crossing point much more frequently than R1, resulting in worse overall performance.

The second-order experiments do not point to a specific heuristic used by R1; instead, the task appears difficult enough to cause frequent complete failures. GPT-4 shows a similar rate of First Common Location errors as in the first-order task. o3-mini performs the worst, and Figure 6 indicates that it commits the First Common Location error far more often than the other models we analyze, despite the fact that o3-mini is specifically trained to perform reasoning tasks and is a relatively new model.

4.5 Is ToM More Difficult with More Characters?

In an effort to determine whether or not LLMs rely on having only seen ToM questions about stories with only a *few* characters, we experimented with increasing the number of characters in the stories. By keeping the length of the story at 100 events, of which only a handful are specified by the storyboard, increasing the number of characters yields stories where an increasing proportion of the events features a novel character.

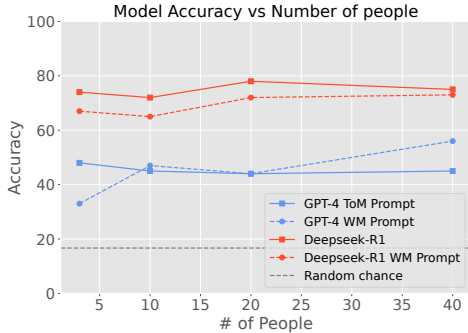


Figure 7: Model accuracy vs the number of characters in the story using ToM (solid) and WM (dashed) prompts.

In the first-order **ToM Prompt**, the number of characters has negligible effect on ToM performance for either R1 or GPT-4. However, when prompted with the WM equivalent of the first-order ToM task, GPT-4 struggles with fewer characters but improves markedly as the number of characters increases. This suggests that the presence of more distinct human entities in the prompt elicits more accurate world modeling behavior in GPT-4, enabling it to respond more accurately to our prompts.

5 Conclusion and Future Work

We present StorySim, a framework that synthetically generates stories from a STORYBOARD, a high-level outline containing a number of key events. This framework creates stories that enable us to evaluate the ToM and WM abilities of LLMs in a way that is robust against data contamination, investigate whether world modeling ability is a good predictor of ToM ability, and identify heuristics that LLMs may use in place of ToM reasoning.

Our results show that most LLMs exhibit generally stronger world modeling ability than ToM. Our error analysis reveals that LLMs often memorize incorrect character locations within the stories. Using our framework, we evaluate LLMs on a suite of ToM tasks and find that our goal-directed reasoning task is the most difficult, and that model size does not always predict performance across these tasks.

We hope this work inspires future efforts to benchmark ToM and social reasoning abilities in LLMs in a more controlled and targeted manner. Though we focus on each character’s perceptions of other’s locations, StorySim’s programmability allows for more complex stories than those presented here. As LLMs are increasingly used in socially-sensitive domains, a rigorous benchmark of their ToM and social reasoning capabilities, as well as the perturbations that affect them, is essential.

6 Reproducibility Statement

All generated stories, model outputs, and answer extraction code are made available on github.com/ngetachew/StorySim. We use the OpenAI Platform API to run experiments using GPT models, and either TogetherAI or OpenRouter for all other models. Unless otherwise stated, all models use temperature 1.0, FP8 precision. Experiments shown in Figure 2 and 3 used models accessed May 31st, 2026 at the latest, except for GPT-5.4, which was accessed March 21st, 2026. All other experiments were run with model versions dated to October 5, 2026. Model outputs are available in a compressed file in the Github repository.

References

- OpenAI Josh Achiam, Steven Adler, S. Agarwal, L. Ahmad, Ilge Akkaya, Florencia Leoni Aleman, D. Almeida, Janko Altschmidt, S. Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv*, 2023.
- Simon Baron-Cohen, Alan M Leslie, and Uta Frith. Does the autistic child have a theory of mind? *Cognition*, 21(1):37–46, 1985. doi: 10.1016/0010-0277(85)90022-8.
- Gordon H Bower and Daniel G Morrow. Mental models in narrative comprehension. *Science*, 247(4938):44–48, 1990. doi: 10.1126/SCIENCE.2403694.
- Hao Chen, Abdul Waheed, Xiang Li, Yidong Wang, Jindong Wang, Bhiksha Raj, and Marah I Abidin. On the diversity of synthetic data and its impact on training large language models. *arXiv.org*, 2024a. doi: 10.48550/arXiv.2410.15226.
- Zhuang Chen, Jincenzi Wu, Jinfeng Zhou, Bosi Wen, Guanqun Bi, Gongyao Jiang, Yaru Cao, Mengting Hu, Yungwei Lai, Zexuan Xiong, et al. Tombench: Benchmarking theory of mind in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15959–15983. Association for Computational Linguistics, 2024b. doi: 10.18653/v1/2024.acl-long.847.
- Ahmed El-Kishky. Openai o1 system card. *arXiv.org*, 2024. doi: 10.48550/arXiv.2412.16720.
- Kanishk Gandhi, Jan-Philipp Franken, Tobias Gerstenberg, and Noah D. Goodman. Understanding social reasoning in language models with language models. *Advances in Neural Information Processing Systems 36*, 36:13518–13529, 2023. doi: 10.52202/075280-0595.
- Erin Grant, Aida Nematzadeh, and Thomas L Griffiths. How can memory-augmented neural networks pass a false-belief task? In *Annual Meeting of the Cognitive Science Society*, volume 39, 2017.
- David Ha and Jürgen Schmidhuber. World models. *arXiv.org*, 2018. doi: 10.5281/zenodo.1207631.
- Alex Havrilla, Andrew Dai, Laura O’Mahony, Koen Oostermeijer, Vera Zisler, Alon Albalak, F. Milo, S. Raparthy, K. Gandhi, Baber Abbasi, et al. Surveying the effects of quality, diversity, and complexity in synthetic data from large language models. *arXiv.org*, 2024. doi: 10.48550/arXiv.2412.02980.
- X. A. Huang, Emanuele La Malfa, Samuele G. Marro, Andrea Asperti, Anthony G. Cohn, and Michael Wooldridge. A notion of complexity for theory of mind via discrete world models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 2964–2983. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.167.
- OpenAI Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv*, 2024.

- Hyunwoo Kim, Melanie Sclar, Xuhui Zhou, Ronan 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*, pp. 14397–14413. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.890.
- Tomas Kociský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2017. doi: 10.1162/tacl.a.00023. URL <https://direct.mit.edu/tacl/article/doi/10.1162/tacl.a.00023/43442>.
- Michal Kosinski. Theory of mind may have spontaneously emerged in large language models. *arXiv.org*, 4(169):2, 2023. doi: 10.48550/arXiv.2302.02083.
- Matthew Le, Y-Lan Boureau, and Maximilian Nickel. Revisiting the evaluation of theory of mind through question answering. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5872–5877, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1598. URL <https://aclanthology.org/D19-1598/>.
- Haris Memisevic, Inga Bisevic, and Arnela Pasalic. Exploring the link between a theory of mind and executive functions in preschool children. *Acta Neuropsychologica*, 16(1):17–26, 2018. doi: 10.5604/01.3001.0011.6503.
- Meta AI. Llama 4 scout, 2025a. URL <https://huggingface.co/meta-llama/llama-4-scout>. Model.
- Meta AI. Llama 4 maverick (17b, 128 experts), 2025b. URL <https://huggingface.co/meta-llama/llama-4-maverick-17b-128e-instruct>. Model.
- Anand Kunal Mishra and S Prabakeran. Kidney disease prediction using gen-ai approach. In *Challenges in Information, Communication and Computing Technology*, pp. 857–862. CRC Press, 2024. doi: 10.1201/9781003559092-148.
- Jared Moore, Ned Cooper, Rasmus Overmark, Beba Cibralic, Nick Haber, and Cameron R Jones. Do large language models have a planning theory of mind? evidence from mindgames: a multi-step persuasion task. *Annual Meeting of the Cognitive Science Society*, 2025. doi: 10.48550/arXiv.2507.16196.
- OpenAI. Gpt-5.4, 2025. URL <https://deploymentsafety.openai.com/gpt-5-4-thinking/gpt-5-4-thinking.pdf>. Model.
- Ajay Patel, Colin Raffel, and Chris Callison-Burch. Datadreamer: A tool for synthetic data generation and reproducible llm workflows. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3781–3799. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.208.
- QianyiHu QianyiHu, Xinhui Tu, Guo Cong, and Shun Zhang. Time-aware react agent for temporal knowledge graph question answering. In *North American Chapter of the Association for Computational Linguistics*, pp. 6013–6024. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-naacl.334. URL <https://aclanthology.org/2025.findings-naacl.334/>.
- Shi Qiu, Shaoyang Guo, Zhuo-Yang Song, Yunbo Sun, Zeyu Cai, Jiashen Wei, Tianyu Luo, Yixuan Yin, Haoxu Zhang, Yi Hu, et al. Phybench: Holistic evaluation of physical perception and reasoning in large language models. *Advances in Neural Information Processing Systems* 38, pp. 1345–1385, 2025. doi: 10.52202/085713-0044.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2019.

- Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 4462–4472. Association for Computational Linguistics, 2019. doi: 10.18653/v1/d19-1454.
- Abulhair Saparov, Richard Yuanzhe Pang, Vishakh Padmakumar, Nitish Joshi, Mehran Kazemi, Najoung Kim, and He He. Testing the general deductive reasoning capacity of large language models using ood examples. *Advances in Neural Information Processing Systems* 36, 36:3083–3105, 2023. doi: 10.52202/075280-0136.
- Melanie Sclar, Jane Yu, Maryam Fazel-Zarandi, Yulia Tsvetkov, Yonatan Bisk, Yejin Choi, and Asli Celikyilmaz. Explore theory of mind: program-guided adversarial data generation for theory of mind reasoning. *arXiv.org*, 2024. doi: 10.48550/arXiv.2412.12175.
- Alexander Shvets. Emo pillars: Knowledge distillation to support fine-grained context-aware and context-less emotion classification. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 174–191. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.10.
- Adam Suma and Sam Dauncey. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv.org*, 2025. doi: 10.48550/arXiv.2501.12948.
- Ruixiang Tang, Xiaotian Han, Xiaoqian Jiang, and Xia Hu. Does synthetic data generation of llms help clinical text mining? *arXiv.org*, 2023. doi: 10.48550/arXiv.2303.04360.
- Qwen Team. Qwen3.5: Accelerating productivity with native multimodal agents, February 2026. URL <https://qwen.ai/blog?id=qwen3.5>.
- Nguyen Thanh Tung. Deepseek-r1-distill-qwen-14b-abliterated-v2-gguf, 2026. URL <https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-14B>. Dense distilled 14B model from DeepSeek-R1 reasoning LLM.
- Tomer Ullman. Large language models fail on trivial alterations to theory-of-mind tasks. *arXiv.org*, 2023. doi: 10.48550/arXiv.2302.08399.
- Keyon Vafa, Justin Y Chen, Ashesh Rambachan, Jon Kleinberg, and Sendhil Mullainathan. Evaluating the world model implicit in a generative model. *Advances in Neural Information Processing Systems* 37, 37:26941–26975, 2024. doi: 10.52202/079017-0846.
- Xintao Wang, Yunze Xiao, Jen-Tse Huang, Siyu Yuan, Rui Xu, Haoran Guo, Quan Tu, Yaying Fei, Ziang Leng, Wei Wang, et al. Incharacter: Evaluating personality fidelity in role-playing agents through psychological interviews. In *Annual Meeting of the Association for Computational Linguistics*, pp. 1840–1873. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2024.acl-long.102.
- Yufan Wu, Yinghui He, Yilin Jia, Rada Mihalcea, Yulong Chen, and Naihao Deng. 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*, pp. 10691–10706. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.717.
- Xi Xia, Bowen Shen, Cici, Dawei Zhu, Di Zhang, Gang Wang, Hailin Zhang, Huaqiu Liu, Jiebao Xiao, Jinhao Dong, et al. Mimo: Unlocking the reasoning potential of language model—from pretraining to posttraining. *arXiv.org*, 2025. doi: 10.48550/arXiv.2505.07608.
- Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning. *arXiv.org*, 2025. doi: 10.48550/arXiv.2502.14768.
- Hainiu Xu, Runcong Zhao, Lixing Zhu, Jinhua Du, and Yulan He. 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)*, pp. 8593–8623. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.466.

A Further Experiment Details

Component	Specification
Characters (C)	8 randomly sampled names
Actions (ϕ)	{enters}
Location graph (G)	6 vertices (locations) that are fully connected
Horizon (n)	100
Event sequence (E)	1. <code>cross_paths($S_1, T, \text{location}_1, t = 10$)</code> 2. <code>move($T, \text{second_location}, t = 11$)</code> 3. <code>exclusive_random($S_1, T, t = 12 \rightarrow 12 + \text{mislead_distance}$)</code> 4. <code>move($T, \text{third_location}, t = 12 + \text{mislead_distance}$)</code> 5. <code>exclusive_random($S_1, T, t \rightarrow n$)</code>

Table 3: Storyboard configuration for mislead distance experiments.

Table 3 displays the exact parameters of the mislead distance experiments. The “exclusive_random” event makes StorySim randomly-generate sequences that don’t affect the actors specified. Note that for the second-order experiments, the only difference is the introduction of S_2 , and an additional cross paths event is added to create the second layer of ToM. In every trial, S_1 , S_2 , and T are randomly picked from C . Further, *location_1*, *second_location*, and *third_location* are all randomly generated from G .

The storyboard used for the experiments described in the ToM vs WM capabilities evaluation is the same as the second-order mislead distance experiment, but the mislead distance is fixed at 30.

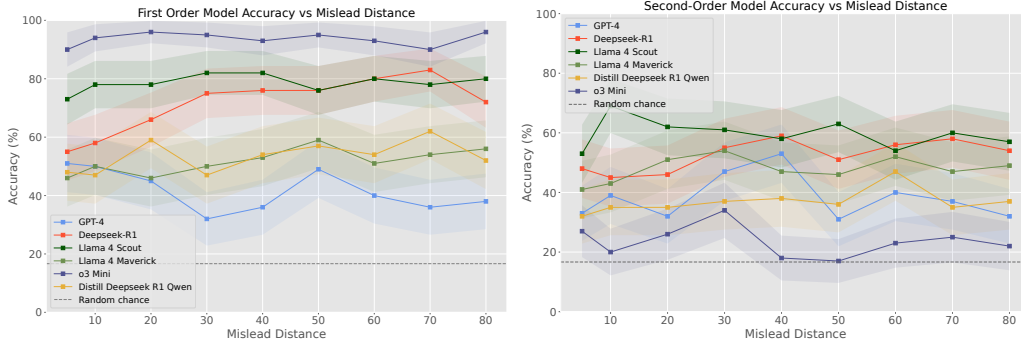


Figure 8: **(left)** Model accuracy on the first-order ToM task as a function of the mislead distance (i.e., the number of steps after S_1 sees T move to another location and before T moves to a different location altogether). **(right)** Model accuracy on the second-order ToM task as a function of mislead distance. Shading indicates a 95% confidence band.

In the subsequent experiments, we aim to address whether or not LLMs are exploiting simple heuristics to answer our questions rather than relying on an actual ToM mechanism. We use StorySim to generate stories while controlling for numerous confounding variables, including but not limited to the story length, number of characters, locations, events, and the order of events. As a result, the resulting observed differences in LLM behavior can be attributed to specific changes in the experimental variables.

B How Sensitive Are LLMs to Story Construction in ToM?

A reasonable hypothesis is that LLMs would predict T ’s last location without considering S_1 ’s mental state, especially if the final location change occurred later in the story. This would be a form of *recency bias*. To test this, we evaluated GPT-4, R1, R1-Distilled Qwen, Llama 4 Scout, Llama 4 Maverick, and o3-mini on first- and second-order ToM problems while increasing the mislead distance (i.e., the number of steps after S_1 sees T move to another location before T moves to a different location altogether) (Meta AI, 2025a;b; Tung,

2026). We chose this set of models because they can be organized into pairs based on model family with differing numbers of parameters. Though we see no significant evidence of recency bias, we present values across mislead distances to demonstrate StorySim’s ability to generate targeted stories, and the LLM’s variance across slight changes in construction.

In the first-order setting, we see a wide spread of model performances, with R1, GPT-4, Qwen, and Maverick performing nearly identically at smaller mislead distances, then varying more at larger mislead distances. Llama 4 Scout outperforms Maverick despite its smaller size. But in all other pairs of models from the same family, the larger model outperforms the smaller one, providing further evidence of a correlation between model size and ToM ability.

Figure 8 (right) shows the results of the second-order ToM version of the experiment. Again, Llama 4 Scout outperforms Maverick, despite the difference in model size, but R1 outperforms its distilled Qwen counterpart. The performance gaps between pairs of models of the same family shrinks in the second-order setting, pointing to the difficulty of this ToM task. The most clear differences in performance appear at the largest mislead distances, with the Llama models and R1 finishing above the rest. GPT-4 is the only model that noticeably improves performance relative to other models. Interestingly, the worst performing model in this setting is o3-mini, which is the best performing model in the first-order setting.

We do not see any correlation between performance and mislead distance, indicating that there is no recency bias at play in the second-order setting. Since StorySim stories are symbolic and relatively simple, we are able to automatically analyze the model’s errors in the next section.

C World Modeling vs ToM Confusion Matrices

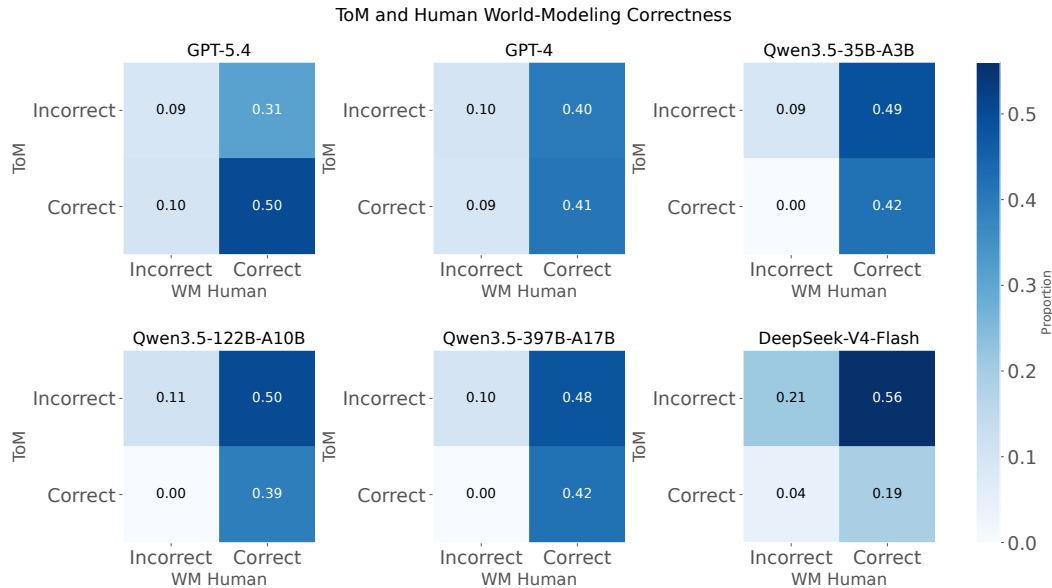


Figure 9: Confusion Matrices for the ToM vs WM Experiment shown in Figure 2

Figure 9 shows the confusion matrices for the ToM vs WM experiments discussed in Section 4.2. Across all models, very few answer story questions correctly in the ToM setting and incorrectly in the WM setting, showing how much more difficult ToM is. That being said, GPT-5.4 has the highest amount of this occurring, getting 10 of the 19 examples it answers incorrectly in WM setting correct in the WM setting.

D LLM Storyboard Adherence

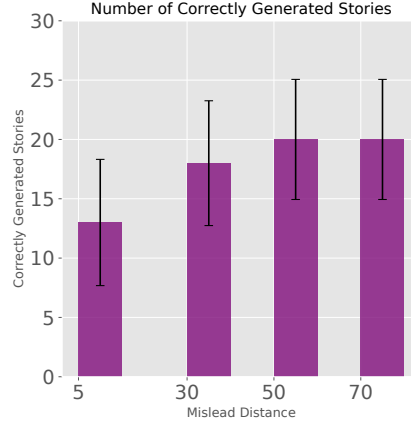


Figure 10: Number of correctly generated stories out of 30 trials using GPT-4. The LLM was prompted to generate stories according to constraints described by a Storyboard that specifies the first-order ToM problem across various mislead distances.

A core feature of StorySim is its ability to generate stories that adhere to the exact specifications of the Storyboard. In Figure 10, we display the performance of GPT-4 generating stories.

E Prompt Experiments

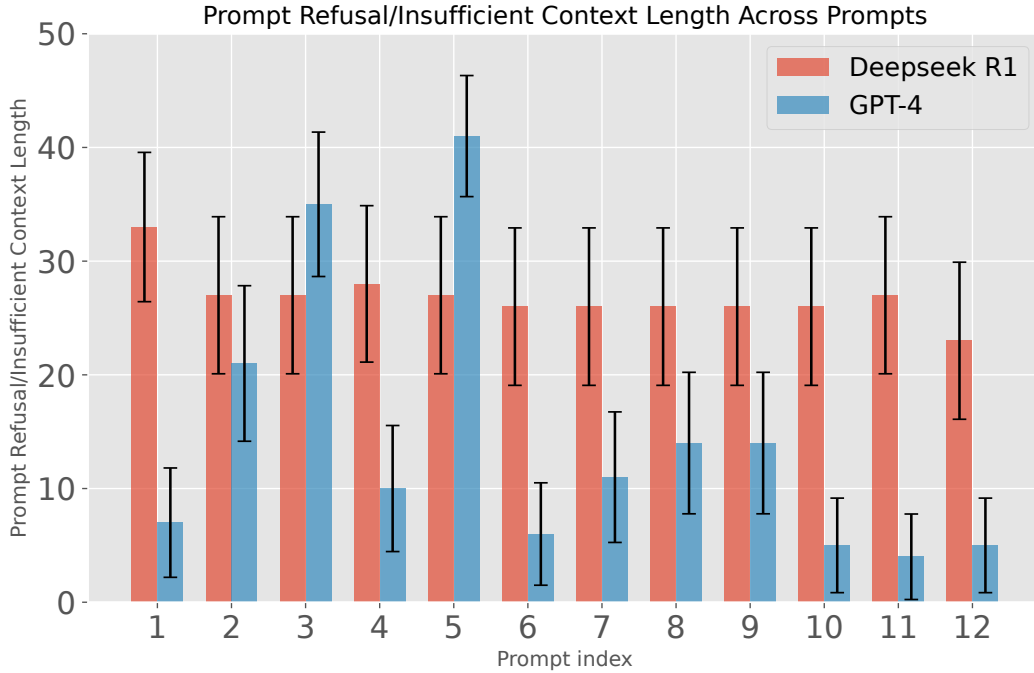


Figure 11: Number of prompts refused by GPT-4 and R1 over 50 trials, with a fixed Storyboard. Indices on the horizontal axis refer to prompts shown in Table E.1. Black bars indicate a 95% confidence interval.

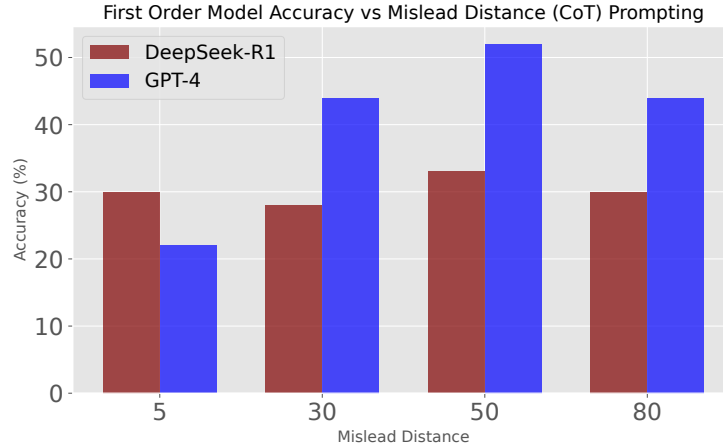


Figure 12: GPT-4 and R1 performance on the first-order ToM task using chain of thought prompting.

In this section, we present supplementary results from experiments with different prompt templates and strategies for avoiding prompt refusal, as initially referenced in Section 4.4. We find that there is no significant difference across most of the prompt variants we tested; when large performance differences do occur, they typically reflect that our chosen template (index 1) performs substantially better than the alternatives. Additionally, Figure 12 evaluates chain-of-thought prompting with GPT-4 and R1 and shows no significant improvement in performance. We further explore the effect of in-context examples across model sizes in the Qwen3.5 family in Figure 14. We see that in the 3-shot setting, the Qwen3.5 family performs similar to zero-shot, with the only notable difference being the smallest model, Qwen3.5 45B A3B, gaining 6% compared to the results in Figure 2. Finally, Figure 13 reports results for GPT-4 in the first-order ToM setting under varying sampling temperatures. Changing the temperature has either a negligible or negative effect on performance. Overall, these findings suggest that the prompting and sampling configurations used in our main analysis elicit the strongest performance from the evaluated LLMs.

E.1 Table of Prompts

#	Prompt Text
1	Read the following story and answer the question at the end. Note that all characters start in the hallway. Characters in the same location can see where each other go when someone leaves. If characters are in different locations, they cannot see each other. There is enough information to answer every question.
2	Please read the following story and answer the concluding question. Every character starts in the hallway. Characters who occupy the same location can observe each other's departures, while those in separate locations cannot. All questions are answerable, so be sure to provide an answer.
3	INSTRUCTIONS: Read the story and answer the question that follows. All characters begin in the hallway. When two characters share a space, they know where the other goes upon leaving; characters in different spaces lack this information. Every question has a sufficient basis in the story, and you must provide an answer. Story:
4	Review the story and give the answer to the final question. Characters start out in the hallway. Characters in the same room always observe each other's movements, while characters in different rooms remain unaware. The necessary information is guaranteed to be available; provide the answer succinctly.
5	Read the following narrative and answer the question that appears at the end. All characters begin in the hallway. Shared locations allow characters to see one another move; separated characters have no visibility. Rely solely on the information in the story and give a direct answer.
6	Carefully read the story and respond to the final question. Every character starts in the hallway. Characters sharing a location know where others go when they leave; characters in different locations do not. The question can be answered from the text alone—state your conclusion briefly.
7	Instructions Read the story and answer the question that follows. Rules: • All characters begin in the hallway. • When characters share a location, they observe each other's movements; those in different places cannot. • When characters share a location, they observe each other's movements; those in different places cannot. • The answer is fully determined by the story. Provide a concise, final answer.
8	Instructions You will read a story involving characters moving between locations and then answer comprehension questions. Rules: • All characters start in the hallway. • Characters who share a location can track each other's departures; characters who are apart cannot • The question is completely answerable from the story. • Provide a single, final answer.
9	Instructions: Read the story about characters moving between different locations, then answer the question that follows. Rules: • All characters begin in the hallway. • Characters in the same place can observe each other's departures; characters in different places cannot. • The question is fully answerable from the provided text. • Respond with one clear final answer. Story:

- 10 You are an assistant tasked with reading a story about characters moving between locations and then answering questions based on it. Follow these rules:
- All characters start in the hallway.
 - Characters who share a location can see where others go; characters who are apart lack this visibility.
 - The question is fully answerable from the narrative.
 - Provide one final answer.
- If it ever seems like information is missing, re-check the story—it always contains enough clues. Avoid repeating yourself. Story:
-
- 11 You are a helpful reading assistant that will read the provided story about characters moving between locations, then answer comprehension questions at the end. Here are some rules that you must remember:
- All of the characters start in the hallway.
 - Characters who share a location know where the other goes when leaving; characters in separate locations have no visibility.
 - The question is 100 percent answerable using the given information.
 - Output a single final answer.
- If you ever think there isn't enough information in the story, check again, because the questions are designed to be answerable by the prompt. Try not to repeat yourself. Story:
-
- 12 INSTRUCTIONS: Read the following story and answer the question at the end. Note that all characters start in the hallway. If two characters are in the same location, then they know where each other are. If one of them leaves the location, then the other will know where they went. There is enough information to answer every question. You must provide an answer to every question. Story:
-

Table 4: Full prompt texts used in the experiments.

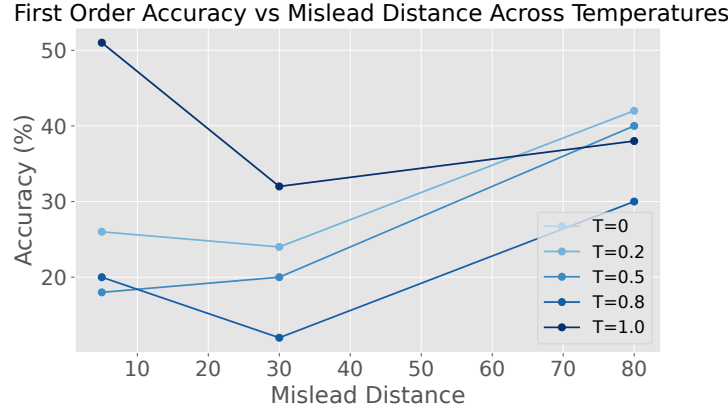


Figure 13: First-order ToM performance of GPT-4 using various temperatures.

F Narrative Variations

In this section, we explore different types of stories generated by StorySim some of which are discussed in the main text.

F.1 Alternative Storyboard Environments

Figure 15 below shows examples of the stories generated with alternative STORYBOARD in the experiments discussed at the end of Section 3. Figure 5 compares LLM performance on the text shown in Figure 15 to an LLM paraphrased version of it.

F.2 Examples of More Complex Stories

While the generated stories used in our experiments are repetitive in structure, StorySim can generate more complex stories given the correct STORYBOARD. We provide Figures 16 through 19 as a supplementary example of this. If one wishes to make more detailed versions of the stories generated by our framework, it’s possible to use an LLM to follow the generated events and generate a polished story. The story shown in Figure 17 is the result of asking GPT-4o to do such a thing. Figure 18 demonstrates two possible stories that can be generated to ask counterfactual and goal-directedness questions that require ToM reasoning. In our main study, we generate the types of stories shown in Figure 18 (left) and Figure 19, leaving other types of ToM to future work.

To motivate how StorySim could generate further types of stories, recalling the notation from the Methods section, imagine a story generated where $\phi := \text{whisper}$ and G was an adjacency graph specifying a social network of characters who can only whisper to other specified characters. Further, imagine creating multiple StorySim objects and creating a story by sampling randomly generated

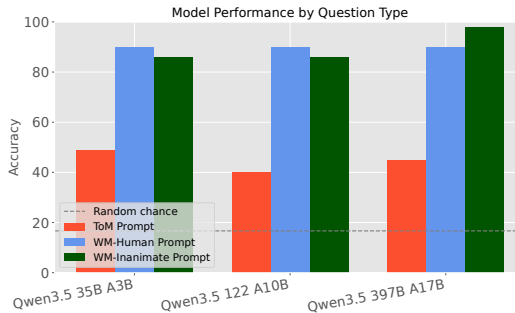


Figure 14: The Qwen3.5 family of models performance on the ToM vs WM task given 3 in-context examples each.

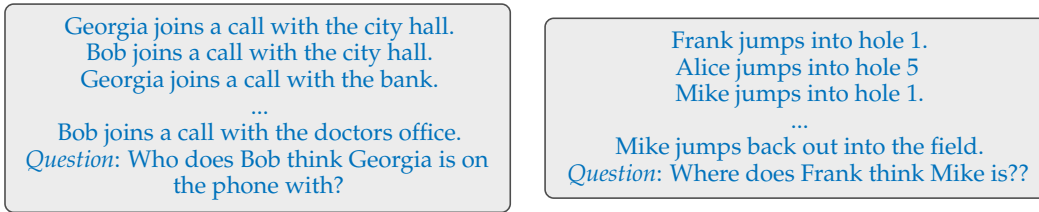


Figure 15: Examples of two different STORYBOARD environments used in experiments in the main text. Right is the Holes and Field environment, on the left is the Conference Call environment.

events from both of them. This could be used to create stories that target characteristics with multiple dimensions. We invite readers to use StorySim to create custom stories that may fit their needs.

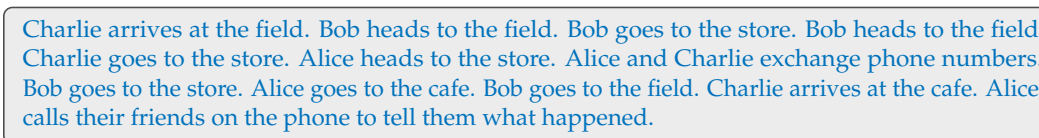


Figure 16: A story about characters meeting and exchanging phone numbers using the actions *arrives*, *heads*, *goes to*, and *heads to*. Each sentence is a generated event.

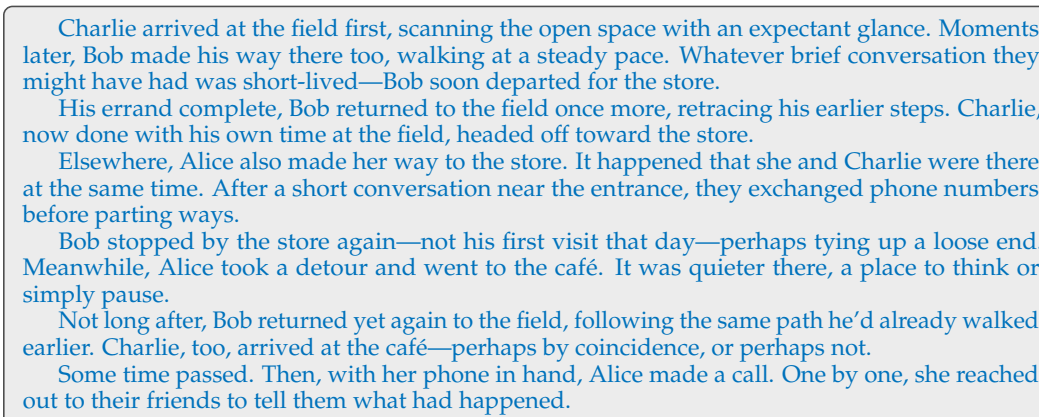


Figure 17: Using events from Figure 16 to create a more interesting story by asking an LLM (GPT-4o) to use the structure of the generated events to generate a more naturalistic story, without adding any new events that deviate from the original.

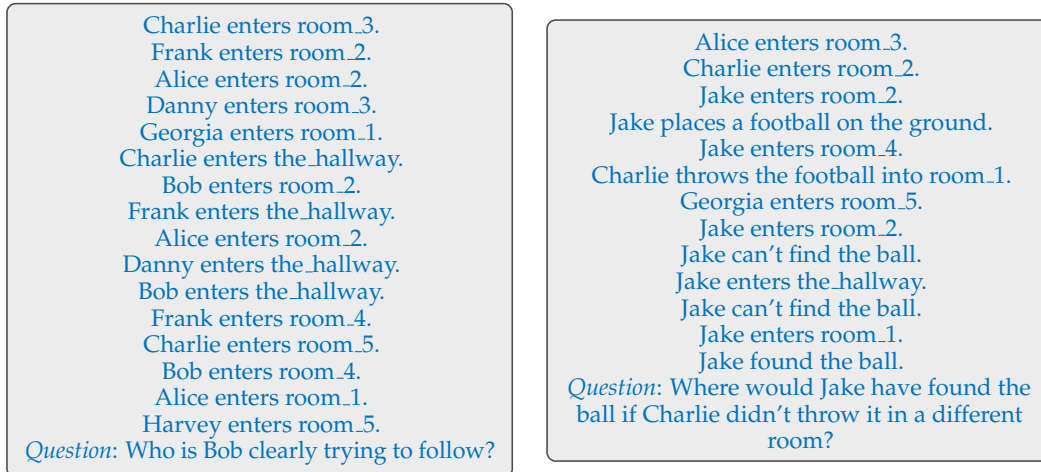


Figure 18: **(Left)** A story generated to ask a goal-directedness ToM question, as used in the experiments for Figure 4. **(Right)** A story generated to ask a counterfactual ToM example.

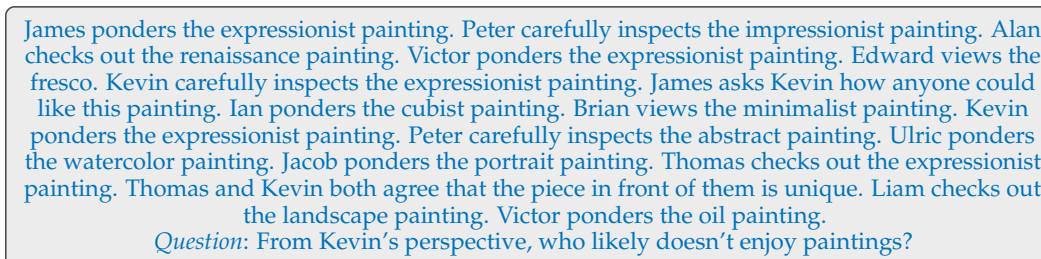


Figure 19: A story generated to ask a preference attribution ToM example. This type of setup is use in the preference attribution task shown in 4.

G Example of Insufficient Context Length Response for R1.

Below is an example of one of R1's responses where it runs out of context length. Note that all errors that it reports finding in the story are untrue. The ellipsis indicates truncation of the prompt for brevity, but where the sentence cuts off in the last step is not truncation. Rather, it is where the model ran out of context length. These erroneous responses from R1 often involve the model attempting to enumerate the story's events, trying to determine when S_1 last saw T , and then restarting its reasoning chain. When R1 does produce a concrete but incorrect answer, it is most often a First Common Location Error, suggesting a tendency to memorize the first location where S_1 and T are seen together.

<think>

Okay, let's try to figure out where Alice and Bob were last in the same room and where Alice saw Bob go when he left. The question is a bit tricky because I need to track both Alice and Bob's movements through all the rooms and see when they were last together. Then, once they part ways, I need to note where Bob went next from Alice's perspective.

First, I'll start by going through the story step by step, noting each time Alice or Bob moves. Since all characters start in the hallway, their initial positions are there. Let's list out their movements in order:

1. Edward enters room_1.
2. Alice enters room_1. So now Alice is in room_1 with Edward.
...
35. Georgia enters room_5. Georgia was in room_5, but maybe she left? Wait, step 23: Georgia was in room_5. Then step 24: Frank leaves room_5. Then step 25: Charlie enters room_2. Then step 26: Charlie to hallway. Step 27: Frank to room_4. Step 28: Danny to room_2. Step 29: Frank to room_3. So Georgia was in room_5 until step 35: Georgia enters room_5 again? Wait, maybe she was in room_1 before. Let me check. Step 15: Georgia enters room_1. Then step 23: Georgia enters room_5. So she was in room_1, then room_5. Then step 35: Georgia enters room_5 again? Wait, maybe that's a typo. Let me check the original movements again.

Looking back at the original list:

After step 15: Georgia enters room_1.

Then step 23: Georgia enters room_5. So she moves from room_1 to room_5.

...
Step 34: Frank enters room_2.

Step 35: Georgia enters room_5. Wait, but she was already in room_5 at step 23. So maybe she left room_5 at some point? Let me check. The steps after step 23:

After step 23 (Georgia enters room_5), the next movement involving Georgia is step 35: Georgia enters room_5 again. But unless she left room_5 in between, she would still be there. But according to the given movements, characters only move when specified. So if Georgia entered room_5 at step 23, she stays there until she moves again. The next move for Georgia is step 35: Georgia enters room_5 again. Wait, that's redundant. Maybe it's a mistake, or perhaps she was in another room. Let me check the steps again.

Wait, step 23: Georgia enters room_5. Then step 24: Frank leaves room_5. Then step 25: Charlie enters room_2. Step 26: Charlie to hallway. Step 27: Frank to room_4. Step 28: Danny to room_2. Step 29: Frank to room_3. Step 30: Frank to room_4. Step 31: Frank to room_3. Step 32: Frank to room_2. Step 33: Frank to room_1. Step 34: Frank to room_2.

Georgia's movements:

- Enters room_4 (step 8)
- Enters room_1 (step 15)
- Enters room_5 (step 23)
- Then step 35: Georgia enters room_5 again.

Maybe she was in room_1 again? Wait, no. After step 15, she was in room_1. Then step 23, she moves to room_5. Then step 35: she's entering room_5 again.

That doesn't make sense unless she left. But according to the given steps, she didn't leave. So perhaps this is a mistake, but I'll proceed as per the given data. Maybe she was in room_5 and then enters again, which doesn't change her location. So she's still in room_5.

Continuing:

35. Georgia enters room_5. (Still there)
...
53. Danny enters room_1. (From room_4 to room_1)
54. Georgia enters <|EOS|>