

TRACKTHEMIND: PROGRAM-GUIDED ADVERSARIAL DATA GENERATION FOR THEORY OF MIND REASONING

Anonymous authors

Paper under double-blind review

ABSTRACT

Do large language models (LLMs) have theory of mind? A plethora of papers and benchmarks have been introduced to evaluate if current models have been able to develop this key ability of social intelligence. However, all rely on limited datasets with simple patterns that can potentially lead to problematic blind spots in evaluation and an overestimation of model capabilities. We introduce **TRACKTHEMIND**, the first framework to allow large-scale generation of diverse and challenging theory of mind data for robust training and evaluation. Our approach leverages an A* search over a custom domain-specific language to produce complex story structures and novel, diverse, yet plausible scenarios to stress test the limits of LLMs. Our evaluation reveals that state-of-the-art LLMs, such as Llama-3.1-70B and GPT-4o, show accuracies as low as 5% on **TRACKTHEMIND**-generated data, highlighting the need for more robust theory of mind evaluation. As our generations are a conceptual superset of prior work, fine-tuning on our data yields a 26-point accuracy improvement on the classic ToMi benchmark (Le et al., 2019). **TRACKTHEMIND** also enables uncovering underlying skills and factors missing for models to show theory of mind, such as unreliable state tracking or data imbalances, which may contribute to models’ poor performance on benchmarks.¹

1 INTRODUCTION

Reasoning about other people’s intentions, goals, thoughts, and beliefs is a foundation of social intelligence. Known as *Theory of Mind* (ToM) (Premack & Woodruff, 1978), this capability is crucial for effective human interaction. There has been a plethora of recent research that develops theory of mind benchmarks and test LLM capabilities, usually inspired in standard tests for research in children such as the Sally-Anne test (Wimmer & Perner, 1983). However, these tests are not well-suited for extensively evaluating models, as they focus on specific scenarios and lack the variability and complexity required to remain challenging after online pre-training. As a result, many existing computational benchmarks may not be effective in robustly evaluating models’ theory of mind abilities.

We introduce **TRACKTHEMIND**, an A*-powered algorithm for generating reliable, diverse, and challenging theory of mind data that can be effectively employed for testing or fine-tuning LLMs. Our approach leverages a domain-specific language to generate synthetic story structures and their character’s mental states. We then use LLMs to create plausible stories based on these plots, allowing for precise control over the narrative and tracking each character’s mental state with high confidence. We employ A* search (Hart et al., 1968) to efficiently navigate the vast space of possible narratives and pinpoint those that are most likely to fool state-of-the-art LLMs. This in turn allows to create a robust, rich dataset that effectively tests the limits of current models (Fig. 1). By generating story structures separately from lexical realizations, we can distinguish the model’s core understanding of the social reasoning from vocabulary cues that might give away stylistic hints.

Our contributions are three-fold: we algorithmically address blind spots in theory of mind evaluation, we provide a recipe to create complex training data that helps imbue models with better theory of mind reasoning skills, and we provide insights into why theory of mind skills are still elusive for LLMs.

¹We will release all code and generated data upon acceptance.

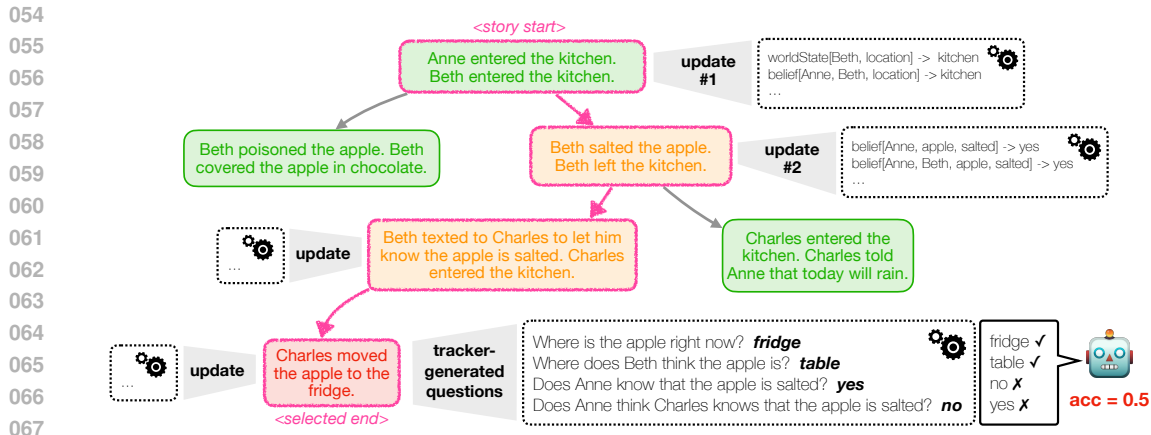


Figure 1: TRACKTHEMIND finds challenging stories for a given language model by searching through the space of stories supported by its domain-specific language for mental state tracking (⚙️), sampling k supported actions at a time (shown as a node, $k = 2$ in the example). Difficulty evaluation (simplified in the figure as *easy*, *medium*, *hard*) of each partial story is done through automatically generated questions with reliable ground-truth answers thanks to our tracking procedure.

First, our work helps to address the conflicting results from a large number of prior research on evaluating theory of mind (e.g. Sap et al., 2022; Shapira et al., 2023a; Kim et al., 2023; Zhou et al., 2023; Gandhi et al., 2024; Strachan et al., 2024), including reports that due to oversimplified datasets, prior theory of mind estimates may be overly optimistic (Ullman, 2023). Our algorithmic approach also helps address the issue of developing benchmarks that may be close to saturation at the time of release, given the increasingly harder task of anticipating LLM failures (e.g., Kim et al., 2024). TRACKTHEMIND’s adversarial nature allows for generating stories to stress test any LLM, diminishing the risk of data leakage onto training data, and thus being more robust than manually-crafted benchmarks. Our experiments show that **TRACKTHEMIND-generated data is extremely challenging, with GPT-4o and Llama-3.1 70B accuracies as low as 5%**. TRACKTHEMIND supports significantly more scenarios than previously possible, with the unique addition of knowledge gain asymmetry during an interaction, among other improvements.

Second, our method allows creating complex and diverse theory of mind data that can be leveraged for model fine-tuning. Given theory of mind’s implicit nature, it is challenging to find data that explicitly articulates the required reasoning, and existing benchmarks are not suitable to use as training data: they are often limited in scale (Xu et al., 2024), portray specific scripted scenarios (Wu et al., 2023; Le et al., 2019), and are prone to leakage risks that would make them fully unsuitable for future use. Fine-tuning with this data has been shown to overfit to specific story structures instead of learning the underlying reasoning required (Sclar et al., 2023), leading to works focused on creating inference-time algorithms to improve the model’s capabilities through prompting or more sophisticated strategies (Sclar et al., 2023; Zhou et al., 2023; Wilf et al., 2023; Jung et al., 2024). While inference-time methods have proven useful for improving performance in theory of mind benchmarks, the benefits of these methods cannot be readily transferred to downstream applications that may also require customized inference-time algorithms for their specific use cases. Fine-tuning Llama-3.1 70B on TRACKTHEMIND’s data achieves a substantial **+26 accuracy point improvement on the classic ToMi benchmark** (Le et al., 2019), **showing good generalization to even more complex TRACKTHEMIND stories than those seen during training, while still retaining general reasoning capabilities**.

Finally, TRACKTHEMIND enables providing new insights into why basic theory of mind reasoning is still challenging for LLMs. We show that LLMs struggle with basic state tracking, a fundamental skill underlying theory of mind reasoning: tracking mental states necessarily requires being able to track states. Our experiments also reveal that in order to improve on theory of mind during fine-tuning, it is crucial to use data that requires theory of mind as opposed to simply requiring state tracking. However, found data (either in-the-wild, or randomly generated) is unlikely to have this necessary property, which may be a key contributor to lagging model performance. Overall, TRACKTHEMIND offers a valuable tool for advancing the theory of mind research, enabling the development of more effective LLMs that can better handle complex social interactions.

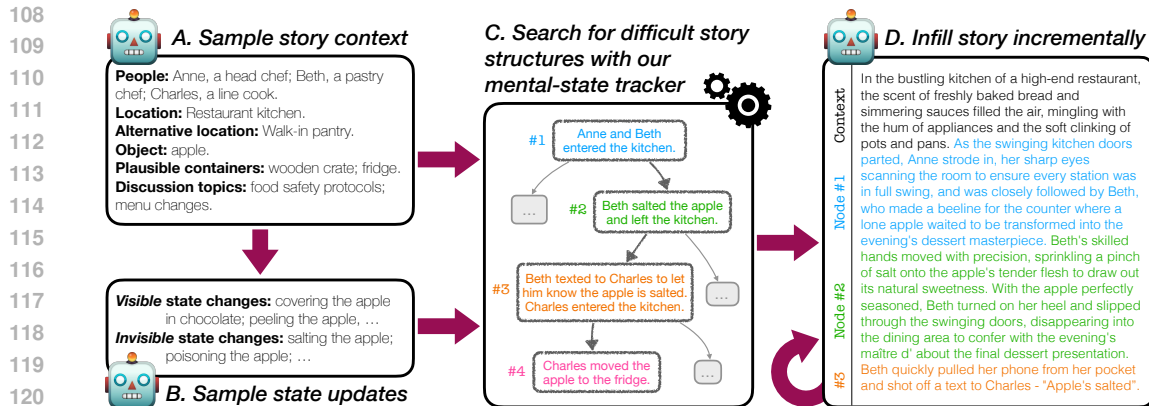


Figure 2: Overview of TRACKTHEMIND’s story generation procedure. We first sample a plausible story context using an LLM (shown in A and B). Topics discussed, location changes of objects and people, and object state updates, may all be required to track in order to pass our theory of mind tests. We then search for difficult story structures (i.e., the raw story points) by sampling and analyzing different orders in which these actions may be performed using A* search (shown in C, and Fig. 1). This ensures that the resulting stories will all be challenging tests for models, and may be used for further improvement. Finally, these story structures (nodes #1-4) are iteratively infilled, one story action at a time, using a language model, yielding a natural-sounding story. Infilled stories are used as training data; benchmarking is done with story structures since they have the highest reliability.

2 ADVERSARIALLY CONSTRUCTED STORIES WITH TRACKTHEMIND

Building on the standard approach in theory of mind of assessing mental state understanding through question answering (Wimmer & Perner, 1983; Kinderman et al., 1998; Baron-Cohen et al., 1999), TRACKTHEMIND creates stories where different characters may have different beliefs about the current world state and about other people’s beliefs, paired with questions to probe model understanding (see Fig. 1’s highlighted story, along with associated questions probing understanding that e.g. “Anne does not know that Charles knows that the apple has been salted”).

TRACKTHEMIND’s story generation process is divided into three main steps: plausible story context sampling (Section 2.1), adversarial story structure generation (Section 2.2), and optionally story infilling (Section 2.3) – an example is outlined in Figure 2. We automatically generate questions to probe understanding of said stories as part of the adversarial story structure generation process (Section 2.2.2); this process finds challenging story structures, i.e., story structures that would yield low accuracy with our generated questions. Because questions are generated automatically and directly from the tracked mental and world states, ground truth answers have a high degree of reliability: we do not use language models at all in the question-answer generation procedure.

2.1 PLAUSIBLE STORY CONTEXT SAMPLING

We use an LLM zero-shot to generate a consistent and plausible story context, comprising essential elements such as character names, roles, locations, relevant objects, object containers, and discussion topics (see Fig. 2A for a full example). This single-step process ensures a coherent and believable setup for our theory of mind stories. Previous approaches (such as ToMi (Le et al., 2019)) sample objects (e.g., an apple) and object containers independently (e.g. a bottle), often resulting in commonsense violations. Unlike these approaches, our method generates a coherent context by sampling these elements jointly in a single LLM call: autoregressive LLMs will naturally suggest contextually plausible elements based on the ones they already generated, and especially so when explicitly requesting it in the prompt. Additionally, we sample possible object state updates (Figure 2B), which are then refined through using an LLM as a judge to filter out implausible and low quality generations. The role of these state updates will be discussed in further detail in Section 2.2.1. The exact prompts used for sampling story contexts are shown in App. C.

2.2 ADVERSARIALLY GENERATING CHALLENGING YET PLAUSIBLE STORY SCRIPTS

2.2.1 THEORY OF MIND-SPECIFIC LANGUAGE DEFINITION

TRACKTHEMIND’s theory of mind-specific language consists of a diverse set of actions \mathcal{A} , each transforming the world state and the beliefs of the people involved (the story state $s \in \mathcal{S}$). A story is thus defined as a sequence of actions (a_1, \dots, a_n) , where each action $a_i \in \mathcal{A}$ is a function $a_i : \mathcal{S} \rightarrow \mathcal{S}$. Each action also has preconditions to be able to apply it, i.e., restrictions to its domain. For example, a precondition for “Charles entering the kitchen” is to not be in it already. Applying an action also automatically updates our world state tracking and belief tracking: for example, “Charles is now in the kitchen”; “Anne knows that Charles is in the kitchen since they were also in the kitchen”; “Charles knows that Anne is in the kitchen since he can see her”; and so forth. All these updates and conditions are specifically programmed and tested; see App. A.1 for the full programs.

TRACKTHEMIND enables the generation of diverse stories by significantly expanding the range of supported actions. These actions include physical changes to the world state such as entering and leaving a room (denoted a_{enter} , a_{leave}), moving an object to a container (or in general, updating its state; denoted $a_{\text{moveObjContainer}}$, $a_{\text{updateObjState}}$ respectively), relocating an object to a different room ($a_{\text{moveObjRoom}}$). Additionally, TRACKTHEMIND supports various forms of communication, including: private conversations between two characters, or public broadcasts to all characters in a room; casual discussions about a topic (denoted *chit-chat*), or notifications about changes in the world state (denoted *info*); these actions are referred to as $a_{\text{info-private}}$, $a_{\text{info-public}}$, $a_{\text{chitChat-private}}$, and $a_{\text{chitChat-public}}$. These actions can occur at any point in the story, allowing for a rich and dynamic narrative (see formal definition in App. A.1) and expanding prior work (Wu et al., 2023).

Each new action requires carefully writing the implied belief and world state updates, which precludes scaling the number of actions supported. However, we alleviate this by noting that from a theory of mind perspective, many actions are equivalent. For example, “peeling an apple” or “covering an apple in chocolate” have the same implications with respect to belief updates (a *visible property* of the apple is being updated, and the witnesses would be the same). Similarly, poisoning an apple has the same implications as moving an apple from a drawer to a fridge (an *invisible property* is updated, witnesses would be the same, and non-witnesses would not assume there has been an update). The instantiations of these equivalent state updates from a belief perspective are done with an LLM during the story context sampling (see Figure 2.B).

Asymmetric belief updates In prior work, all belief updates were *symmetric*: if A and B witnessed an action, then A knows that B witnessed the action and vice versa. Our framework introduces the ability to model asymmetric scenarios. Specifically, we enable the addition of secret witnesses to an action such as someone observing through a security camera, or removal of witnesses without others’ knowledge, as in the case of someone becoming distracted by their phone. This added nuance allows for more realistic and complex social scenarios. Asymmetries a_{peek} and $a_{\text{distracted}}$ are modifier functions, e.g., as a modifier to “Beth salted the apple” ($a_{\text{updateObjState}}(\cdot)$) there may be a secret person peeking ($a_{\text{peek}}(a_{\text{updateObjState}}(\cdot))$): “While this was happening, Diane witnessed it in secret.”

2.2.2 GENERATING QUESTIONS AND ASSESSING RESULTING STORY DIFFICULTY

We assess a model’s understanding of a generated story $s=(a_1, \dots, a_n)$ by probing it with automatically generated question-answer pairs. TRACKTHEMIND-generated answers are more reliable than purely-LLM generated ones, since they are directly produced from the states’ trajectory with our tracker. Questions may be testing first-order beliefs, second-order beliefs, or regular state tracking: *First-order* refers to asking about someone’s mental state (e.g., “Does Anne know the apple is salted?”); *Second-order* refers to one extra level of recursion in mental state tracking (e.g., “Does Anne think that Charles know the apple is salted?”); *State tracking* may probe about the current state (*ground truth*) or prior ones (*memory*).

We expand the complexity of memory questions with respect to prior work by asking about any intermediate state (e.g. “Where was the object before X happened?”) instead of solely about the initial one (“Where was the object at the beginning?”). Our generated questions are simple to evaluate: they are either binary (yes/no), or are answered by stating an object, container, or room. Specific question formulations differ based on the property, e.g., location (“Where does Charles think that

Anne will search for the apple?”) or knowledge (“Does Charles know that the apple is salted?”). See App. A.2 for the full list of supported questions.

A question is considered *interesting* if the answer would change depending on the person being asked about. For example “Does Anne think that Charles knows that the apple is salted?” is interesting because the answer would differ if asked about someone else, such as “Does Beth think that Charles knows the apple is salted?”. TRACKTHEMIND’s tracker very easily allows for automatically detecting interestingness.

2.2.3 A* SEARCH

Given a context \mathcal{C} and a set of actions \mathcal{A} , our main goal is to find challenging story structures. To increase TRACKTHEMIND’s usage flexibility, we support the option of searching for stories s that fulfill desired user conditions $\mathbf{isDesired}(s) \in \{0, 1\}$, such as the number of people involved, or the number of actions belonging to a subset $\mathcal{A}' \subseteq \mathcal{A}$ of *important actions*.

We search over the space of plausible story structures of up to m actions. We define this space as a directed graph, where each node is a sequence of valid actions $s = (a_1, \dots, a_i)$, and there is an edge between s and s' if and only if s is prefix of s' , and s' contains k more actions than s . $k \geq 1$ is the *grouping factor* for actions, defining the granularity with which we will sample and evaluate nodes. For simplicity, Figure 1 depicts only the new $k = 2$ actions that each node introduces.

To find challenging stories that simultaneously fulfill the user constraints we use A* search (Hart et al., 1968). By definition, A* selects the path that minimizes $f(s) = g(s) + h(s)$, where $g(s)$ is the cost of the path from the start to node s , and $h(s)$ is a heuristic that estimates the cost of the cheapest path from s to a *goal node* (one of the nodes where it would be acceptable to finish the search). In our context, goal nodes are those such that $\mathbf{isDesired}(s') = 1$. We choose A* as our search algorithm precisely because it enables to search this space prioritizing desired user conditions through $h(s)$, as we will detail below.

A story is said to be challenging for a model if it incorrectly answers our generated questions, i.e., it shows low accuracy. Thus, we define $g(s)$ as our target model’s accuracy among *all* questions for s . We define the heuristic function $h(s)$ as a proxy estimation of the likelihood of generating a full story $s + s'$ that fulfills user constraints $\mathbf{isDesired}(s) = 1$, where s' is the continuation of story s :

$$h(s) = \alpha \left(1 - \frac{1}{P} \sum_{i=1}^P \mathbb{1}(\mathbf{isDesired}(s + s'_i) = 1) \right)$$

Here, all s'_i are randomly sampled continuations of s and $0 \leq \alpha \leq 1$ is a scaling factor. A* requires to evaluate all neighbors of a node s . Since this would be infeasible given the vast space to explore, and that each $f(\cdot)$ evaluation requires several LLM calls (one per question), we restrict the evaluation to a pre-defined constant number of neighbors, prioritized by the closeness of this node to fulfilling the conditions described by $\mathbf{isDesired}(\cdot)$. This pre-defined constant may depend on $f(s)$ to prioritize more promising partial stories (i.e., with lower $f(s)$ values).

2.3 STORY INFILLING

Story infilling is the process of transforming a full story structure $s = (a_1, a_2, \dots, a_n)$ with a story context \mathcal{C} into a natural-sounding narration (see Fig. 2D). We infill stories iteratively with an LLM by transforming each action a into a more natural sounding one, according to some stylistic desiderata d , and conditioned on the previously infilled context z (denoted $\text{infill}(a, z, d)$). Supported stylistic desiderata d are length requests (e.g., “use up to two sentences”) or style requests (e.g., “make this into a conversation”); we optionally also include sampled character goals g and an initial narration context c based on the story s , also generated with an LLM (e.g., Anne’s goal may be to oversee that all dishes are rapidly delivered to customers; see initial context example in Fig. 2). Concretely, the full story infilling SI is as follows:

$$SI(i) = \text{infill}(a_i, SI(i-1), d_i, g) \text{ where } SI(0) = c$$

Infilling is done iteratively to ensure that the order of the actions stays the same, since this is important for keeping the mental state tracking valid. To further increase reliability, we use an LLM as a judge after each infilling step to confirm that each mental state tracked after executing the story step a_i still holds even after infilling. This discards infillings that introduced ambiguity or hallucinations.

Table 1: Accuracy results of TRACKTHEMIND’s story structures on 18 action sets \mathcal{A} , each aggregating 90 total stories from 9 different settings (number of people, actions, and rooms). Each set is either based on actions supported by well-known theory of mind tests or includes our novel expansions, and is analyzed excluding or including asymmetry (\times , \checkmark). Each setting requires at least one action in the story to be from one of the squared actions to encourage non-overlapping story structure characteristics between action sets shown. Data was generated using each model as its own evaluator (i.e., as $g(\cdot)$), and results shown include all first-order questions—the most basic theory of mind level, not requiring recursion. Lowest accuracy for each model is bolded.

TRACKTHEMIND action set $\{a_{\text{enter}}, a_{\text{leave}}, \dots\}$	Llama-3.1 70B Inst.		GPT-4o		Mixtral 8x7B Inst.	
include asymmetry modifiers? ($a_{\text{peek}}, a_{\text{distracted}}$)	\times	\checkmark	\times	\checkmark	\times	\checkmark
$\dots, \boxed{a_{\text{moveObjContainer}}}$.19	.08	.40	.34	.32	.29
$\dots, \boxed{a_{\text{updateObjState}}}$.26	.37	.17	.31	.03	.08
$\dots, \boxed{a_{\text{moveObjContainer}}}, \boxed{a_{\text{updateObjState}}}$.20	.18	.35	.40	.13	.21
$\dots, a_{\text{moveObjContainer}}, \boxed{a_{\text{moveObjRoom}}}$.11	.18	.05	.32	.05	.04
$\dots, a_{\text{moveObjContainer}}, \boxed{a_{\text{info-*}}}$.05	.08	.36	.40	.31	.39
$\dots, a_{\text{moveObjContainer}}, a_{\text{moveObjRoom}}, \boxed{a_{\text{info-*}}}$.13	.26	.24	.39	.09	.34
$\dots, a_{\text{moveObjContainer}}, a_{\text{moveObjRoom}}, a_{\text{chitChat-*}}, \boxed{a_{\text{info-*}}}$.72	.71	.71	.68	.52	.51
$\dots, \boxed{a_{\text{chitChat-private}}}$.78	.61	.76	.58	.51	.57
$\dots, \boxed{a_{\text{chitChat-public}}}$.54	.57	.46	.52	.33	.37

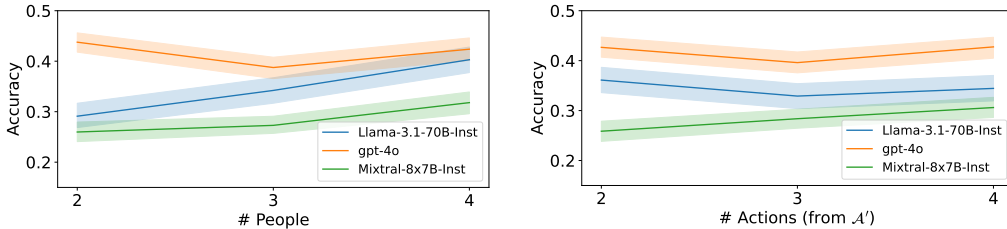


Figure 3: Accuracy on TRACKTHEMIND’s story structures when increasing the number of actions or people involved. Accuracy is computed across all story structure settings. A story with greater number of people suggests lower difficulty, possibly because there is a fixed number of actions, thus fewer actions per person. Difficulty of TRACKTHEMIND-generated stories stay similar when increasing the number of actions.

3 TRACKTHEMIND AS AN EVALUATION BENCHMARK

We begin by showcasing how TRACKTHEMIND story structures can be used as a challenging benchmark, highlighting its unique features and advantages.

Experimental setup We use TRACKTHEMIND to generate 10 story structures for each of 9 action sets (each with and without asymmetry) and each set of user conditions. Each story generation is allowed to evaluate 50 nodes. User conditions— $\text{isDesired}(\cdot)$ —require exactly $p \in \{2, 3, 4\}$ people involved, with $a \in \{2, 3, 4\}$ actions belonging to the set of important actions \mathcal{A}' , spanning across either $r = 1$ or $r = 2$ rooms, and with $m \leq 15$ actions in total—leading to a total of 162 settings. In all experiments, \mathcal{A}' are the actions that add new basic world knowledge: $\mathcal{A}' = \{a_{\text{moveObjContainer}}, a_{\text{updateObjState}}, a_{\text{moveObjRoom}}, a_{\text{chitChat-*}}\}$. We then infill every story. We use Llama-3.1-70B-Instruct (Dubey et al., 2024), GPT-4o (OpenAI, 2024), and Mixtral-8x7B-Instruct (Jiang et al., 2024) to generate story structures. A^* is run with $\alpha = 0.1$, $P = 50$, and $k = 3$ (i.e. grouping three actions per node). See generation examples in App. D.

TRACKTHEMIND finds challenging story structures for frontier models As shown in Table 1, our TRACKTHEMIND consistently identifies story structures that are highly challenging for models

Table 2: Accuracy results on TRACKTHEMIND-generated data built to minimize accuracy for each particular model. A random sample of 10000 (story structure, question) pairs is shown. Data remains challenging even if it was built with a different model, and even including questions we did not optimize for: story structures were selected adversarially towards first-order belief questions only ($g(\cdot)$), accuracies shown include all belief questions.

Model used in TRACKTHEMIND generation ($g(\cdot)$)	Model used for evaluation		
	Llama-3.1 70B Inst.	GPT-4o	Mixtral 7x8B Inst.
Llama-3.1 70B Inst.	0.56	0.63	0.46
GPT-4o	0.61	0.60	0.44
Mixtral 7x8B Inst.	0.64	0.67	0.41

across various action sets, with average performances in TRACKTHEMIND-generated datasets as low as 0.05 for GPT-4o (i.e., 5%). Performance tends to decrease when increasing the number of people involved, possibly because with a fixed number of state-changing actions, there will be fewer actions per person which may be difficult to track. There is no clear trend when increasing actions, as all settings remain similarly challenging. See Figure 3.

A* is a better strategy than over-generation and filtering Over-generation and filtering has become a standard procedure for synthetic data generation (e.g. West et al., 2022; Wang et al., 2023). We measure the effectiveness of A* by comparing the A*-generated data to the data resulting from over-generating stories with our domain-specific language—using the same **isDesired**(\cdot) criteria and budget as used in the A* search—and retaining only the most difficult stories. In a set of 81 randomly-selected settings (50% of the original 162 settings, due to the experiment’s high cost), we generate 50 stories with each method using Llama-3.1-70B-Instruct and a budget of 2500 accuracy evaluations each. A* yielded a more challenging dataset (by 2 accuracy points), with shorter stories on average (0.6 fewer actions). This length difference is possibly due to the pressures A* induces towards shorter stories through the heuristic $h(s)$. Moreover, A* successfully generated the requested 50 stories with non-perfect accuracy for 95% of settings, whereas the baseline only in 62% of cases—even with the generous budget and total action limit ($m = 15$). See Table 6 for the full distribution of results.

Story structures found adversarially for a model remain challenging for other models We evaluate the difficulty of a TRACKTHEMIND-generated dataset with each model, and find that although there is an increased difficulty towards data generated adversarially with the same model, it remains challenging for all others. Notably, the generated datasets remain challenging even when adding question types not included in the $g(\cdot)$ optimization (second-order belief questions). See Table 2.

Humans agree with TRACKTHEMIND-generated story structures labels We conducted a human evaluation to verify the quality of the story structures’ automatically-generated labels and the story infillings. For labels, we annotated 283 questions across 35 randomly-sampled story structures from all settings generated for Table 1, and found 98% agreement with our expected answers—likely due to the clear and concise nature of our stories and that the ground truth labels were generated by our domain-specific language. The annotation load was equally divided between two annotators.

Infilled stories remain challenging Infilled stories with Llama-3.1 70B yielded an average accuracy of 0.55. Although the average accuracy increased by 0.21 through the infilling process, the samples remained challenging thanks to the highly challenging underlying stories². One key factor for this accuracy difference comes from models sometimes making the mental states more explicit through the infilling process: results shown correspond to a single attempt at infilling each story (73% of the samples ended successfully in a single attempt, judged by an LLM). Although stories remain challenging, since infilling with an LLM may introduce hallucinations (especially when not constraining on infilling length or style) we only use them as training data. See App. B.2 for detailed results for all action sets.

²Infilling can be also added to the A* search; we deemed it unnecessary given that this simpler method still yields a highly challenging benchmark and it is less costly.

Table 3: Performance on major false-belief benchmarks; accuracy (in %) unless otherwise stated. Parenthesis reflect differences between out-of-the-box model and fine-tuned version using TRACKTHEMIND-generated data. **Bold** reflects higher overall performance.

	ToMi	Hi-ToM	BigToM	OpenToM (F1)	FANToM
Llama-3.1 8B Instruct	68%	30%	75%	.39	0.3%
TRACKTHEMIND-8B	94% (+26)	52% (+22)	79% (+4)	.42 (+.03)	0.5% (+0.02)

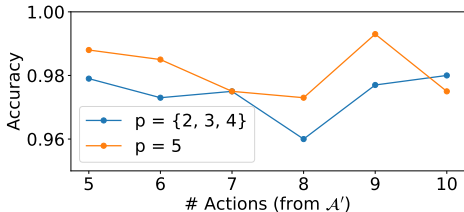


Figure 4: TRACKTHEMIND-8B accuracy when evaluating on TRACKTHEMIND-generated data with more people p and/or more actions a than seen during training ($p < 5$, $a < 5$). Performance remains high when adding several actions and/or one more person.

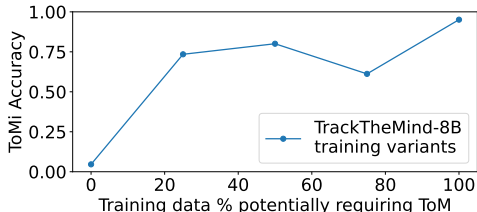


Figure 5: ToMi accuracy when training with TRACKTHEMIND-generated data with different proportions of interesting questions (i.e., questions potentially requiring theory of mind to answer). Here, all variants are fine-tuned with 85000 story structure samples for 1 epoch.

4 TRACKTHEMIND IS EFFECTIVE AS TRAINING DATA GENERATOR

Experimental setup We fine-tune Llama-3.1 8B Instruct using a dataset of 114000 (story, question, answer) triples, focusing solely on the completion tasks, and dub the resulting model TRACKTHEMIND-8B. The dataset comprises both raw story structures and infilled stories, incorporating story structures from each of the 9 action sets listed in Table 1 (excluding asymmetry, and with a balanced number of questions per story type), and various user constraints—the same as in Section 3. We do full fine-tuning with the following hyperparameters: a learning rate of 10^{-6} , 100 warm-up steps, effective batch size of 40 samples, where we fine-tune solely on completions.

Fine-tuning with TRACKTHEMIND generalizes well to TRACKTHEMIND-generated data with more people and more actions than used in training Since TRACKTHEMIND-8B is trained with TRACKTHEMIND-generated data involving $p = \{2, 3, 4\}$ people with $m = \{2, 3, 4\}$ actions from the set of important actions \mathcal{A}' , we evaluate generalization within the TRACKTHEMIND domain by evaluating on TRACKTHEMIND-generated data involving 5 people, and up to 11 actions. This data is generated with Llama-3.1, the same model as original training data.

Fine-tuning with TRACKTHEMIND improves or maintains performance on theory of mind benchmarks without hurting general reasoning capabilities We evaluate our fine-tuned TRACKTHEMIND-8B model on five prominent theory of mind benchmarks: ToMi (Le et al., 2019), Hi-ToM (Wu et al., 2023), BigToM (Gandhi et al., 2024), OpenToM (Xu et al., 2024), and FANToM (Kim et al., 2023). Results show significant improvements in performance on ToMi and HiToM, with accuracy gains of +20 points on both benchmarks (see Table 3). The model maintains or shows small gains on the remaining three similar benchmarks, indicating that fine-tuning on TRACKTHEMIND data enhances or preserves performance across a range of theory of mind tasks.

We also evaluate out-of-domain reasoning skills using the two datasets: Multi3Woz (Hu et al., 2023), a commonly-used dataset for dialogue state tracking, and MMLU (Hendrycks et al., 2021), which tests both world knowledge and problem-solving abilities. Dialogue state tracking capabilities are preserved: both the base model and TRACKTHEMIND-8B achieve 94%. Broader reasoning capabilities are also generally preserved, with a small 2% performance difference (base model achieves 69%; TRACKTHEMIND-8B, 67%). Given the out-of-domain nature, we expect that intermixing data with samples more similar to MMLU’s domains will substantially alleviate this slight regression.

Data mixture affects downstream performance We fine-tune five models, with 0%, 25%, 50%, 75%, or 100% of the stories requiring theory of mind to answer at least one question about the story. Figure 5 shows that training with as much stories that require theory of mind is crucial

for achieving high downstream performance (using ToMi as a proxy dataset), even if some of the individual questions used for training do not require theory of mind.

5 ON UNDERLYING SKILLS NEEDED FOR THEORY OF MIND

TRACKTHEMIND enables uncovering and quantifying underlying causes for models’ poor theory of mind reasoning in models out-of-the-box. We specifically focus on the lack of robust state tracking skills, and the need for targeted training data in order to improve theory of mind capabilities.

LLMs lack robust state tracking skills TRACKTHEMIND’s objective is to find story structures where models fail to answer questions; some of these questions simply require state tracking, specifically the ones where every person would give the same answer (i.e., their mental state is the same in this regard; e.g., in Fig. 1, all $X \in \{\text{Anne, Beth, Charles}\}$ would answer the same to “Where does X think Anne is right now?”). By definition (see § 2.2.2), these are the *uninteresting* questions. TRACKTHEMIND-generated questions are approximately evenly split between interesting and uninteresting, and uninteresting ones are even more challenging on average: the accuracy of interesting and uninteresting questions is 50% and 36% respectively for Llama-3.1 70B, 56% and 41% for GPT-4o, and 46% and 30% for Mixtral. See Table 6 in App. B.3 for full breakdown for all settings.

State tracking questions are a subset of theory of mind questions, and arguably an easier case since the required logic for answering questions is simpler. Therefore, improving models’ performance on state tracking may be a crucial prerequisite for achieving theory of mind reasoning in LLMs. As we have demonstrated, TRACKTHEMIND can be easily adapted to stress test pure state tracking, simply by retaining only the uninteresting questions.

Training data biases against theory of mind and its implications Figure 5 shows that to successfully improve performance on the ToMi benchmark, TRACKTHEMIND fine-tuning data needs to be biased towards *interesting* questions. However, a significant portion of models’ training data is likely biased against requiring the tracking of divergent mental states (e.g., news articles).

As a conceptual proof that this phenomena occurs even within our custom domain-specific language unless we explicitly bias towards theory of mind, we demonstrate that randomly-sampled story structures tend not to require theory of mind. Using TRACKTHEMIND’s domain-specific language, we randomly generate 1000 story structures with ToMi primitives ($\{a_{\text{enter}}, a_{\text{leave}}, a_{\text{moveObjContainer}}\}$) for stories involving $\{2, 3, 4\}$ people and $\{2, 3, 4\}$ object movements. We consider a story to not require theory of mind if all first-order and second-order theory of mind questions are *un-interesting*, as defined in § 2.2.2 (i.e., all share the same mental state). This stringent criterion evaluates *all* questions simultaneously. Nevertheless, our results show that 78% or more of the randomly-sampled stories meet this condition across all settings, with up to 87% of stories fulfilling the condition for the smallest setting (2 people, 2 object movements). When considering each question individually, 91%-95% are uninteresting questions. See App. B.4 for more details.

6 RELATED WORK

Theory of mind benchmarking for language models Theory of mind benchmarks in language models can be categorized into human-generated and model-generated datasets. While human-generated datasets (Shapira et al., 2023b; Kim et al., 2024; Chen et al., 2024) test reasoning about goals, emotions of others, and future actions, they are often limited in size and scope. Machine-generated datasets, such as foundational ToMi (Le et al., 2019) and its successor Hi-ToM (Wu et al., 2023) focus primarily on mental state tracking, but have significant limitations: ToMi only supports a restricted set of actions ($\{a_{\text{enter}}, a_{\text{leave}}, a_{\text{moveObjContainer}}\}$), while Hi-ToM adds a_{info} but only as the last action in a story, and both datasets have extremely restricted interactions to orders. In contrast, our method, TRACKTHEMIND, significantly expands the scope of machine-generated datasets by supporting a larger number of actions, diverse wording, and plausible contexts. Unlike recent approaches that rely on LLMs for generation (Kim et al., 2023; Xu et al., 2024; Gandhi et al., 2024), TRACKTHEMIND ensures reliability and multi-interaction storytelling, making it a more comprehensive and robust benchmark for theory of mind in LLMs.

Theory of mind beyond language modeling Theory of mind has been explored in various areas, including human computer interaction (Wang et al., 2021), explainable AI (Akula et al., 2022), and

486 multi-agent reinforcement learning (Rabinowitz et al., 2018; Sclar et al., 2022; Zhu et al., 2021).
 487 Recent benchmarks have evaluated theory of mind in multi-modal settings (Jin et al., 2024) and
 488 multi-agent collaboration (Bara et al., 2021; Shi et al., 2024), but these focus on goal-driven in-
 489 teractions. Psychologists distinguish between affective (emotions, desires) and cognitive (beliefs,
 490 knowledge) theory of mind (Shamay-Tsoory et al., 2010), with cognitive theory of mind developing
 491 later in children (Wellman, 2014). Our work targets cognitive theory of mind, which is well-suited
 492 for generating situations with a domain-specific language and provides unambiguous answers across
 493 cultures. By focusing on cognitive theory of mind, our approach complements existing research and
 494 provides a comprehensive benchmark for this crucial aspect of human reasoning in language models.

495 **Synthetic data generation** Synthetic data has become promising approach for acquiring high-
 496 quality data in various domains, including multihop question-answering (Lupidi et al., 2024), and
 497 language model evaluation (Wang et al., 2024). The process involves data augmentation/generation
 498 and curation, with techniques such as permutation-based augmentation (Yu et al., 2024; Li et al.,
 499 2024a) and iterative prompting (Yang et al., 2022). However, model hallucination (Guarnera et al.,
 500 2020; Van Breugel et al., 2023; Wood et al., 2021; Zhang et al., 2023) requires careful filtration and
 501 curation to ensure data quality. While prior works have used external feedback (Zelikman et al.,
 502 2022; Luo et al., 2024), our approach leverages an external LLM-as-judge to evaluate the plausi-
 503 bility and challenge of generated stories, both before and after infilling. Recently, AutoBench (Li
 504 et al., 2024b) has also been proposed to automatically search for datasets that meet a salience, nov-
 505 elty, and difficulty desiderata, highlighting the importance of careful benchmark creation. Unlike
 506 AutoBench, which over-generates under the assumption that text-based conditioning minimizes
 507 hallucinations, our approach lifts this assumption and actively searches the space of possible narra-
 508 tives. This enables to create high-quality synthetic data regardless of the likelihood of a story being
 509 generated zero-shot, and generating even more challenging stories than with over-generation.

510 7 CONCLUSIONS

511
 512 Theory of mind (ToM) is essential for social intelligence, and developing agents with theory of
 513 mind is a requisite for efficient interaction and collaboration with humans. Thus, it is important to
 514 build a path forward for imbuing agents with this type of reasoning, as well as methods for robustly
 515 assessing the of models’ theory of mind reasoning capabilities.

516 We present TRACKTHEMIND, an A*-powered algorithm for generating reliable, diverse and chal-
 517 lenging theory of mind data; specifically, creating synthetic stories that require theory of mind to
 518 understand them, along with questions to probe understanding. TRACKTHEMIND’s adversarial
 519 nature enables the stress testing of future models and making our evaluation more robust to data
 520 leakage. We show that TRACKTHEMIND generates challenging theory of mind evaluation sets for
 521 many frontier models, with accuracies as low as 5% for Llama-3.1 70B Instruct and GPT-4o. More-
 522 over, we show that TRACKTHEMIND can be used as a method for generating training data, leading
 523 to improvements of up to 26 accuracy points in well-known theory of mind benchmarks. Synthetic
 524 data is crucial for this domain, given that data that articulates theory of mind reasoning is difficult
 525 to find in the wild: children have access to a wide range of naturalistic social settings that incentivize
 526 the development of theory of mind but there is no such parallel pressure for LLMs.

527 Finally, we provide insights as to why basic theory of mind is still elusive to LLMs, including poor
 528 state tracking skills and demonstrating the need for training data that purposefully requires theory
 529 of mind, which is likely not present in the wild nor in randomly-generated data.

530 531 LIMITATIONS

532 TRACKTHEMIND offers a valuable tool for theory of mind research, and is a first step towards
 533 developing LLMs that can handle social interactions effectively. Although its data encompasses
 534 diverse and challenging settings—more than previously available—, and is grounded in established
 535 psychological tests, TRACKTHEMIND necessarily simplifies the complexity of real-world states
 536 and narratives by constraining it to the supported types of actions and interactions. Our frame-
 537 work requires manual coding of new actions, which can be time-consuming process but comes with
 538 the benefit of a significant reliability improvement. Furthermore, our stories are not necessarily
 539 goal-oriented narratives, highlighting an important avenue for future work: creating datasets where
 actions stem directly from character goals to further enhance diversity and plausibility.

REFERENCES

- 540
541
542 Arjun R. Akula, Keze Wang, Changsong Liu, Sari Saba-Sadiya, Hongjing Lu, Sinisa Todor-
543 ović, Joyce Chai, and Song-Chun Zhu. Cx-tom: Counterfactual explanations with theory-
544 of-mind for enhancing human trust in image recognition models. *iScience*, 25(1):103581,
545 2022. ISSN 2589-0042. doi: <https://doi.org/10.1016/j.isci.2021.103581>. URL <https://www.sciencedirect.com/science/article/pii/S2589004221015510>.
546
- 547 Cristian-Paul Bara, CH-Wang Sky, and Joyce Chai. Mindcraft: Theory of mind modeling for situated
548 dialogue in collaborative tasks. In *Proceedings of the 2021 Conference on Empirical Methods in*
549 *Natural Language Processing*, pp. 1112–1125, 2021.
- 550 Simon Baron-Cohen, Michelle O’Riordan, Rosie Jones, Valerie Stone, and Kate Plaisted. A new
551 test of social sensitivity: Detection of faux pas in normal children and children with asperger
552 syndrome. *Journal of Autism and Developmental Disorders*, 29(5):407–418, 1999.
- 553 Zhuang Chen, Jincenzi Wu, Jinfeng Zhou, Bosi Wen, Guanqun Bi, Gongyao Jiang, Yaru Cao,
554 Mengting Hu, Yunghwei Lai, Zexuan Xiong, et al. Tombench: Benchmarking theory of mind
555 in large language models. *arXiv preprint arXiv:2402.15052*, 2024.
- 556
557 Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha
558 Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models.
559 *arXiv preprint arXiv:2407.21783*, 2024.
- 560 Kanishk Gandhi, Jan-Philipp Fränken, Tobias Gerstenberg, and Noah Goodman. Understanding
561 social reasoning in language models with language models. *Advances in Neural Information*
562 *Processing Systems*, 36, 2024.
- 563
564 Luca Guarnera, Oliver Giudice, and Sebastiano Battiato. Deepfake detection by analyzing con-
565 volutional traces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern*
566 *recognition workshops*, pp. 666–667, 2020.
- 567 Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination
568 of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107,
569 1968.
- 570 Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob
571 Steinhardt. Measuring massive multitask language understanding. In *International Conference*
572 *on Learning Representations*, 2021.
- 573
574 Songbo Hu, Han Zhou, Mete Hergul, Milan Gritta, Guchun Zhang, Ignacio Iacobacci, Ivan Vulić,
575 and Anna Korhonen. Multi 3 woz: A multilingual, multi-domain, multi-parallel dataset for
576 training and evaluating culturally adapted task-oriented dialog systems. *Transactions of the*
577 *Association for Computational Linguistics*, 11:1396–1415, 2023.
- 578 Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bam-
579 ford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al.
580 Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.
- 581
582 Chuanyang Jin, Yutong Wu, Jing Cao, Jiannan Xiang, Yen-Ling Kuo, Zhiting Hu, Tomer Ullman,
583 Antonio Torralba, Joshua Tenenbaum, and Tianmin Shu. MMTOM-QA: Multimodal theory of
584 mind question answering. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings*
585 *of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long*
586 *Papers)*, pp. 16077–16102, Bangkok, Thailand, August 2024. Association for Computational Lin-
587 guistics. URL <https://aclanthology.org/2024.acl-long.851>.
- 588 Chani Jung, Dongkwan Kim, Jiho Jin, Jiseon Kim, Yeon Seonwoo, Yejin Choi, Alice Oh, and
589 Hyunwoo Kim. Perceptions to beliefs: Exploring precursory inferences for theory of mind in
590 large language models. *arXiv preprint arXiv:2407.06004*, 2024.
- 591 Hyunwoo Kim, Melanie Sclar, Xuhui Zhou, Ronan Bras, Gunhee Kim, Yejin Choi, and Maarten
592 Sap. Fantom: A benchmark for stress-testing machine theory of mind in interactions. In
593 *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp.
14397–14413, 2023.

- 594 Seungone Kim, Juyoung Suk, Ji Yong Cho, Shayne Longpre, Chaeun Kim, Dongkeun Yoon, Gui-
595 jin Son, Yejin Cho, Sheikh Shafayat, Jinheon Baek, et al. The biggen bench: A principled
596 benchmark for fine-grained evaluation of language models with language models. arXiv preprint
597 arXiv:2406.05761, 2024.
- 598 Peter Kinderman, Robin Dunbar, and Richard P Bentall. Theory-of-mind deficits and causal attri-
599 butions. British journal of Psychology, 89(2):191–204, 1998.
- 600 Matthew Le, Y-Lan Boureau, and Maximilian Nickel. Revisiting the evaluation of theory of mind
601 through question answering. In Proceedings of the 2019 Conference on Empirical Methods in
602 Natural Language Processing and the 9th International Joint Conference on Natural Language
603 Processing (EMNLP-IJCNLP), pp. 5872–5877, 2019.
- 604 Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. Gsm-plus: A comprehensive
605 benchmark for evaluating the robustness of llms as mathematical problem solvers. arXiv preprint
606 arXiv:2402.19255, 2024a.
- 607 Xiang Lisa Li, Evan Zheran Liu, Percy Liang, and Tatsunori Hashimoto. Autobench: Creating
608 salient, novel, difficult datasets for language models, 2024b. URL <https://arxiv.org/abs/2407.08351>.
- 609 Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing
610 Ma, Qingwei Lin, and Daxin Jiang. Wizardcoder: Empowering code large language models with
611 evol-instruct. 2024. URL <https://openreview.net/forum?id=UnUwSIgK5W>.
- 612 Alisia Lupidi, Carlos Gemmell, Nicola Cancedda, Jane Dwivedi-Yu, Jason Weston, Jakob Foer-
613 ster, Roberta Raileanu, and Maria Lomeli. Source2synth: Synthetic data generation and curation
614 grounded in real data sources. arXiv preprint arXiv:2409.08239, 2024.
- 615 OpenAI, 2024. URL <https://openai.com/index/hello-gpt-4o>.
- 616 David Premack and Guy Woodruff. Does the chimpanzee have a theory of mind? Behavioral and
617 brain sciences, 1(4):515–526, 1978.
- 618 Neil Rabinowitz, Frank Perbet, Francis Song, Chiyuan Zhang, SM Ali Eslami, and Matthew
619 Botvinick. Machine theory of mind. In International conference on machine learning, pp. 4218–
620 4227. PMLR, 2018.
- 621 Maarten Sap, Ronan LeBras, Daniel Fried, and Yejin Choi. Neural theory-of-mind? on the lim-
622 its of social intelligence in large lms. In Proceedings of the Association for Computational
623 Linguistics: EMNLP 2022, pp. 3762–3780, Abu Dhabi, United Arab Emirates, December 2022.
624 Association for Computational Linguistics. URL <https://preview.aclanthology.org/emnlp-22-ingestion/2022.emnlp-main.248>.
- 625 Melanie Sclar, Graham Neubig, and Yonatan Bisk. Symmetric machine theory of mind. In Ka-
626 malika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato
627 (eds.), Proceedings of the 39th International Conference on Machine Learning, volume 162 of
628 Proceedings of Machine Learning Research, pp. 19450–19466. PMLR, 17–23 Jul 2022. URL
629 <https://proceedings.mlr.press/v162/sclar22a.html>.
- 630 Melanie Sclar, Sachin Kumar, Peter West, Alane Suhr, Yejin Choi, and Yulia Tsvetkov. Minding
631 language models’ (lack of) theory of mind: A plug-and-play multi-character belief tracker. In
632 Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), Proceedings of the 61st Annual
633 Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 13960–
634 13980, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/
635 v1/2023.acl-long.780. URL <https://aclanthology.org/2023.acl-long.780>.
- 636 Simone G Shamay-Tsoory, Hagai Harari, Judith Aharon-Peretz, and Yechiel Levkovitz. The role of
637 the orbitofrontal cortex in affective theory of mind deficits in criminal offenders with psychopathic
638 tendencies. Cortex, 46(5):668–677, 2010.
- 639 Natalie Shapira, Mosh Levy, Seyed Hossein Alavi, Xuhui Zhou, Yejin Choi, Yoav Goldberg,
640 Maarten Sap, and Vered Shwartz. Clever hans or neural theory of mind? stress testing social
641 reasoning in large language models, 2023a.

- 648 Natalie Shapira, Guy Zwirn, and Yoav Goldberg. How well do large language models perform on
649 faux pas tests? In *Findings of the Association for Computational Linguistics: ACL 2023*, pp.
650 10438–10451, 2023b.
- 651
652 Haojun Shi, Suyu Ye, Xinyu Fang, Chuanyang Jin, Layla Isik, Yen-Ling Kuo, and Tianmin Shu.
653 Muma-tom: Multi-modal multi-agent theory of mind. *arXiv preprint arXiv:2408.12574*, 2024.
- 654 James WA Strachan, Dalila Albergo, Giulia Borghini, Oriana Pansardi, Eugenio Scaliti, Saurabh
655 Gupta, Krati Saxena, Alessandro Rufo, Stefano Panzeri, Guido Manzi, et al. Testing theory of
656 mind in large language models and humans. *Nature Human Behaviour*, pp. 1–11, 2024.
- 657
658 Tomer Ullman. Large language models fail on trivial alterations to theory-of-mind tasks. *arXiv*
659 *preprint arXiv:2302.08399*, 2023.
- 660 Boris Van Breugel, Zhaozhi Qian, and Mihaela Van Der Schaar. Synthetic data, real errors: how
661 (not) to publish and use synthetic data. In *International Conference on Machine Learning*, pp.
662 34793–34808. PMLR, 2023.
- 663
664 Qiaosi Wang, Koustuv Saha, Eric Gregori, David Joyner, and Ashok Goel. Towards mutual theory
665 of mind in human-ai interaction: How language reflects what students perceive about a virtual
666 teaching assistant. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing*
667 *Systems*, pp. 1–14, 2021.
- 668 Tianlu Wang, Iliia Kulikov, Olga Golovneva, Ping Yu, Weizhe Yuan, Jane Dwivedi-Yu,
669 Richard Yuanzhe Pang, Maryam Fazel-Zarandi, Jason Weston, and Xian Li. Self-taught eval-
670 uators. *arXiv preprint arXiv:2408.02666*, 2024.
- 671
672 Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and
673 Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions.
674 In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual*
675 *Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13484–
676 13508, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/
677 v1/2023.acl-long.754. URL <https://aclanthology.org/2023.acl-long.754>.
- 678 Henry M Wellman. *Making minds: How theory of mind develops*. Oxford University Press, 2014.
- 679
680 Peter West, Chandra Bhagavatula, Jack Hessel, Jena Hwang, Liwei Jiang, Ronan Le Bras, Ximing
681 Lu, Sean Welleck, and Yejin Choi. Symbolic knowledge distillation: from general language mod-
682 els to commonsense models. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir
683 Meza Ruiz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the*
684 *Association for Computational Linguistics: Human Language Technologies*, pp. 4602–4625,
685 Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/
686 2022.naacl-main.341. URL <https://aclanthology.org/2022.naacl-main.341>.
- 687
688 Alex Wilf, Sihyun Shawn Lee, Paul Pu Liang, and Louis-Philippe Morency. Think twice:
689 Perspective-taking improves large language models’ theory-of-mind capabilities. *arXiv preprint*
690 *arXiv:2311.10227*, 2023.
- 691
692 Heinz Wimmer and Josef Perner. Beliefs about beliefs: Representation and constraining function of
693 wrong beliefs in young children’s understanding of deception. *Cognition*, 13(1):103–128, 1983.
- 694
695 Erroll Wood, Tadas Baltrušaitis, Charlie Hewitt, Sebastian Dziadzio, Thomas J Cashman, and Jamie
696 Shotton. Fake it till you make it: face analysis in the wild using synthetic data alone. In
697 *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3681–3691, 2021.
- 698
699 Yufan Wu, Yinghui He, Yilin Jia, Rada Mihalcea, Yulong Chen, and Naihao Deng. Hi-ToM: A
700 benchmark for evaluating higher-order theory of mind reasoning in large language models. pp.
701 10691–10706, December 2023. doi: 10.18653/v1/2023.findings-emnlp.717. URL <https://aclanthology.org/2023.findings-emnlp.717>.
- 702
703 Hainiu Xu, Runcong Zhao, Lixing Zhu, Jinhua Du, and Yulan He. Opentom: A comprehensive
704 benchmark for evaluating theory-of-mind reasoning capabilities of large language models. *arXiv*
705 *preprint arXiv:2402.06044*, 2024.

702 Kevin Yang, Yuandong Tian, Nanyun Peng, and Dan Klein. Re3: Generating longer stories with
703 recursive reprompting and revision. pp. 4393–4479, December 2022. doi: 10.18653/v1/2022.
704 emnlp-main.296. URL <https://aclanthology.org/2022.emnlp-main.296>.
705

706 Longhui Yu, Weisen Jiang, Han Shi, Jincheng YU, Zhengying Liu, Yu Zhang, James Kwok, Zhen-
707 guo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical ques-
708 tions for large language models. 2024. URL <https://openreview.net/forum?id=N8N0hgNDrt>.
709

710 Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with
711 reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.
712

713 Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemaoyang Liu, Tingchen Fu, Xinting Huang, Enbo Zhao,
714 Yu Zhang, Yulong Chen, et al. Siren’s song in the ai ocean: a survey on hallucination in large
715 language models. *arXiv preprint arXiv:2309.01219*, 2023.

716 Pei Zhou, Aman Madaan, Srividya Pranavi Potharaju, Aditya Gupta, Kevin R McKee, Ari Holtzman,
717 Jay Pujara, Xiang Ren, Swaroop Mishra, Aida Nematzadeh, et al. How far are large language
718 models from agents with theory-of-mind? *arXiv preprint arXiv:2310.03051*, 2023.

719 Hao Zhu, Graham Neubig, and Yonatan Bisk. Few-shot language coordination by modeling theory
720 of mind. In *International Conference on Machine Learning*, pp. 12901–12911. PMLR, 2021.
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755

A APPENDIX

A.1 ACTIONS' FORMAL DEFINITION (CONT. FROM 2.2.1)

All actions are functions that transform a state into another state, updating the world state and the beliefs of everyone involved up to two levels of recursion. All actions have preconditions, e.g. to enter a room you need to not be in it already.

A state $\in \mathcal{S}$ is comprised of a world state ws (the things currently true physically about the world described), the first-order beliefs b_1 , and the second-order beliefs b_2 . First-order beliefs describe what each person believes to be the current world state, e.g. Anne believes that the apple is salted. Second-order beliefs describe what each person estimates that each other person believes to be the current world state, e.g. Anne believes that Beth thinks that the apple is salted.

Let's describe the definition of leaving in a room through an example: "Beth left the kitchen.", and build the definition of the action function $a_{\text{leave, Beth, kitchen}} : \mathcal{S} \rightarrow \mathcal{S}$. As described above, the state is comprised of a world state, first-order beliefs, and second-order beliefs, i.e.,

$$a_{\text{leave, Beth, kitchen}}(ws, b_1, b_2) := (ws', b'_1, b'_2)$$

Let's first describe the world state update ws' . The world state remains the same for every entity (object, container, person, etc.), except for the person leaving the room—Beth. Thus,

$$ws(q, \text{room}) = ws'(q, \text{room}) \quad \forall q \neq \text{Beth} \quad \text{and} \quad ws'(q, \text{room}) = \neg \text{kitchen}$$

Let's then describe the first-order belief updates b'_1 . Here, we assume that everyone in the same room as Beth (the kitchen) will know that Beth has left. We denote this group of people as $\text{witnesses}(\text{kitchen})$:

$$\text{witnesses}(\text{kitchen}) := \{p \mid ws(p, \text{room}) = \text{kitchen}\}$$

Everyone not in the kitchen will assume that Beth is still there unless communicated otherwise, since they have no reason to believe she has left. Thus,

$$b_1(p, \text{Beth}, \text{room}) = b'_1(p, \text{Beth}, \text{room}) = \text{kitchen} \quad \forall p \notin \text{witnesses}(\text{kitchen})$$

$$b_1(p, \text{Beth}, \text{room}) = \neg \text{kitchen} \quad \forall p \in \text{witnesses}(\text{kitchen})$$

We now describe the second-order belief updates b'_2 . Here, we assume that everyone in the kitchen (including Beth) assumes that everyone else in the kitchen knows Beth left (and only them). If someone was not in the kitchen, they will assume nothing has happened. Formally,

$$b_2(p, q, \text{Beth}, \text{room}) = b'_2(p, q, \text{Beth}, \text{room}) = \text{kitchen} \quad \forall p \notin \text{witnesses}(\text{kitchen}), \quad \forall q$$

$$b_2(p, q, \text{Beth}, \text{room}) = \neg \text{kitchen} \quad \forall p \in \text{witnesses}(\text{kitchen}) \quad \forall q \in \text{witnesses}(\text{kitchen})$$

$$b_2(p, q, \text{Beth}, \text{room}) = \text{kitchen} \quad \forall p \in \text{witnesses}(\text{kitchen}) \quad \forall q \notin \text{witnesses}(\text{kitchen})$$

Finally, the function can only be applied if Beth is in the kitchen, i.e. it has the precondition $ws(\text{Beth}, \text{room}) = \text{kitchen}$.

All other functions definitions can be found verbatim in the code to be released.

A.2 ALL SUPPORTED QUESTIONS (CONT. FROM SECTION 2.2.2)

Table 4: List of all supported TRACKTHEMIND questions per property discussed and level of theory of mind, transcribed verbatim.

Property asked about	ToM Order	Question (requesting <i>Short Answer</i> in prompt)	Expected Answers
room location	–	In which room was the <object> at the beginning?	room name
room location	–	In which room is the <object> now?	room name
room location	–	In which room was the <object> before <action>?	room name
room location	1st	In which room will <person> search for the <object>?	room name
room location	2nd	In which room does <person1> think that <person2> will search for the <object>?	room name
container location	–	In which container was the <object> at the beginning?	container name
container location	–	In which container is the <object> now?	container name
container location	–	In which container was the <object> before <action>?	container name
container location	1st	In which container will <person> search for the <object>?	container name
container location	2nd	In which container does <person1> think that <person2> will search for the <object>?	container name
abstract knowledge	topic 1st	Does <person1> know about <topicDiscussed>?	<i>yes or no</i>
abstract knowledge	topic 2nd	What does <person1> think about <person2>’s belief on <topicDiscussed>? (knows about it / does not know about it)	<i>knows about it or does not know about it</i>
knowledge about state update	1st	Does <person> believe that the <object> <newState>? Answer yes or no.	<i>yes or no</i>
knowledge about state update	2nd	Does <person1> believe that <person2> believes that the <object> <newState>? Answer yes or no.	<i>yes or no</i>

B ADDITIONAL EXPERIMENTS

B.1 A*-GENERATED STORIES ARE MORE CHALLENGING THAN OVERGENERATING AND FILTERING (CONT. FROM § 3)

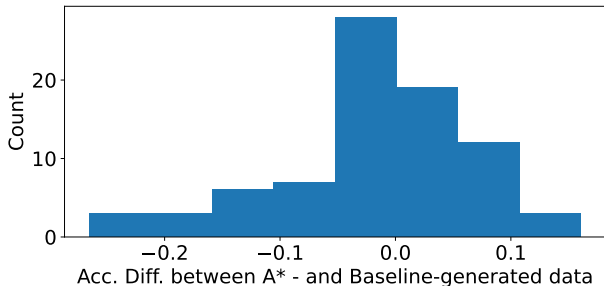


Figure 6: Histogram depicting accuracy differences between A*-generated datasets for Llama-3.1-70B-Instruct and a dataset created by over-generating and filtering with the same budget (i.e., baseline). Results show that A* is better at finding story structures that make a challenging benchmark by showing low accuracy (negative values mean A* is better at finding challenging story structures).

B.2 INFILLED STORY STRUCTURES REMAIN CHALLENGING (CONT. FROM § 3)

Table 5: Changes in accuracy when infilling TRACKTHEMIND-generated story structures to output natural-sounding stories. We only include comparison between the 73% of stories where the LLM as a judge (Llama-3.1-70B Instruct) determined that all infilled actions were high quality.

Action Set: $\{a_{\text{enter}}, a_{\text{leave}}, \dots\}$	Include asymmetry	Acc. Story Structure	Acc. Infilled	Acc. Diff.
$\dots, a_{\text{moveObjContainer}}\}$, (denoted \mathcal{A}_1)	\times \checkmark	0.21 0.09	0.48 0.38	0.27 0.29
$\dots, a_{\text{updateObjState}}\}$, (denoted \mathcal{A}_2)	\times \checkmark	0.21 0.36	0.39 0.45	0.18 0.09
$\dots, a_{\text{moveObjContainer}}, a_{\text{updateObjState}}\}$ (\mathcal{A}_3)	\times \checkmark	0.19 0.14	0.42 0.51	0.23 0.37
$\dots, a_{\text{moveObjContainer}}, a_{\text{moveObjRoom}}\}$ (\mathcal{A}_4)	\times \checkmark	0.10 0.23	0.47 0.50	0.37 0.28
$\dots, a_{\text{moveObjContainer}}, a_{\text{info-*}}\}$ (\mathcal{A}_5)	\times \checkmark	0.03 0.08	0.44 0.40	0.40 0.32
$\dots, a_{\text{moveObjContainer}}, a_{\text{moveObjRoom}}, a_{\text{info-*}}\}$ (\mathcal{A}_6)	\times \checkmark	0.16 0.29	0.57 0.66	0.41 0.37
$a_{\text{moveObjContainer}}, a_{\text{moveObjRoom}}, a_{\text{chitChat-*}}, a_{\text{info-*}}\}$ (\mathcal{A}_7)	\times \checkmark	0.73 0.74	0.77 0.79	0.04 0.06
$\dots, a_{\text{chitChat-private}}\}$ (\mathcal{A}_8)	\times \checkmark	0.77 0.61	0.79 0.68	0.02 0.07
$\dots, a_{\text{chitChat-public}}\}$ (\mathcal{A}_9)	\times \checkmark	0.56 0.57	0.65 0.63	0.10 0.06

Table 5 shows a breakdown across all settings. 73% of the stories were infilled in a single attempt. Next step options are sampled simultaneously with repetition penalty for added wording diversity. Without this constraint, 81% of the story structures are infilled within a single attempt.

918 B.3 MODELS FAIL BOTH AT THEORY OF MIND AND PURE STATE TRACKING (CONT. FROM
 919 § 5)
 920
 921

922 Table 6: Accuracy breakdown of the experiment shown in Table 1, discriminating if each question is
 923 *interesting* or not. A question is interesting if the answer would change depending on the entity asked
 924 about, thus potentially requiring theory of mind. Results show that part of a model’s difficulty with
 925 TRACKTHEMIND’s generated data can be attributed to poor state tracking (i.e., the uninteresting
 926 questions, noted \neg Int.).

Action Set	Includes Symmetry?	Llama			GPT4o			Mixtral		
		Acc. Int.	Acc. \neg Int.	% Int.	Acc. Int.	Acc. \neg Int.	% Int.	Acc. Int.	Acc. \neg Int.	% Int.
\mathcal{A}_1	\times	0.45	0.20	48%	0.47	0.45	48%	0.42	0.33	49%
	\checkmark	0.37	0.14	50%	0.51	0.35	50%	0.42	0.36	51%
\mathcal{A}_2	\times	0.58	0.25	49%	0.42	0.16	49%	0.14	0.02	49%
	\checkmark	0.52	0.34	50%	0.59	0.31	50%	0.42	0.08	42%
\mathcal{A}_3	\times	0.52	0.18	50%	0.52	0.31	49%	0.33	0.05	49%
	\checkmark	0.45	0.18	50%	0.62	0.37	48%	0.44	0.20	48%
\mathcal{A}_4	\times	0.53	0.13	27%	0.50	0.05	28%	0.60	0.05	24%
	\checkmark	0.36	0.21	46%	0.55	0.31	35%	0.47	0.04	29%
\mathcal{A}_5	\times	0.31	0.07	50%	0.50	0.38	46%	0.44	0.36	50%
	\checkmark	0.26	0.15	52%	0.52	0.42	51%	0.47	0.45	54%
\mathcal{A}_6	\times	0.44	0.17	50%	0.57	0.27	47%	0.48	0.10	49%
	\checkmark	0.39	0.28	45%	0.57	0.37	15%	0.52	0.27	33%
\mathcal{A}_7	\times	0.67	0.80	48%	0.70	0.80	48%	0.55	0.66	45%
	\checkmark	0.71	0.76	51%	0.69	0.80	49%	0.55	0.73	44%
\mathcal{A}_8	\times	0.81	0.78	56%	0.78	0.74	51%	0.52	0.55	51%
	\checkmark	0.57	0.57	73%	0.59	0.48	67%	0.55	0.55	70%
\mathcal{A}_9	\times	0.48	0.62	40%	0.48	0.41	40%	0.47	0.25	40%
	\checkmark	0.54	0.60	50%	0.58	0.46	52%	0.53	0.28	51%
Total	—	0.50	0.36	50%	0.56	0.41	48%	0.46	0.30	47%

953
 954 B.4 HOW LIKELY IS A RANDOMLY-SAMPLED STORY TO REQUIRE THEORY OF MIND? (CONT.
 955 FROM §5)
 956

957 Table 7: Probability that a randomly-sampled story would require theory of mind for answering at
 958 least one question. Actions considered are $\{a_{\text{enter}}, a_{\text{leave}}, a_{\text{moveObjContainer}}\}$, all settings of $\{2, 3, 4\}$
 959 people and $\{2, 3, 4\}$ $a_{\text{moveObjContainer}}$ movements, with 10 maximum actions, are shown.
 960

Number of people	Number of movements		
	2	3	4
2	0.131	0.208	0.235
3	0.195	0.234	0.288
4	0.210	0.259	0.315

972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025

Table 8: Probability that a randomly-sampled (story, question) pair would potentially require theory of mind, meaning that the answer to the question varies depending on the entities considered. Actions considered are $\{a_{\text{enter}}, a_{\text{leave}}, a_{\text{moveObjContainer}}\}$, all settings of $\{2, 3, 4\}$ people and $\{2, 3, 4\}$ $a_{\text{moveObjContainer}}$ movements, with 10 maximum actions, are shown.

Number of people	Number of movements		
	2	3	4
2	0.090	0.123	0.124
3	0.120	0.109	0.121
4	0.111	0.101	0.112

Table 9: Probability that a randomly-sampled (story, question, answer) triple would require an answer that is different from the true world state (i.e., it is a *false-belief question*). Actions considered are $\{a_{\text{enter}}, a_{\text{leave}}, a_{\text{moveObjContainer}}\}$, all settings of $\{2, 3, 4\}$ people and $\{2, 3, 4\}$ $a_{\text{moveObjContainer}}$ movements, with 10 maximum actions, are shown.

Number of people	Number of movements		
	2	3	4
2	0.059	0.084	0.086
3	0.065	0.065	0.072
4	0.056	0.049	0.058

1026 C PROMPTS USED FOR GENERATING AND VALIDATING TRACKTHEMIND'S
1027 DATA
1028

1029 C.1 GENERATING STORY CONTEXTS (CONT. FROM §2.1)
1030
1031

1032 Suggest a short context where {num_people} people are together in a
1033 room. It should be at most two sentences long, and they should be able
1034 to observe each other. Later in the story, characters are going to move
1035 around and store objects, so your context should be plausible under
1036 those constraints. Do not explicitly include that they can all see each
1037 other, it should be clear from context. The room could be in a house,
1038 work environment, etc.

1039 Here's an example for three people. Follow the same format.
1040

1041 LIST CHARACTERS' NAMES:
1042 1. Emily, a meticulous office manager.
1043 2. Jason, a tech-savvy intern.
1044 3. Karen, a diligent accountant.

1045 GIVE SHORT STORY CONTEXT:
1046 Emily, Jason, and Karen gathered around the central table in the sleek
1047 office's conference room, discussing the upcoming audit. As they
1048 strategized, the shelves and storage compartments lining the walls
1049 around them held the tools and documents they would soon need to
1050 organize and pack away.

1051 ROOM IN WHICH THIS STORY BEGINS:
1052
1053 NAME ONE REASONABLE ALTERNATIVE ROOM THEY COULD MOVE TO:
1054
1055 NAME ONE OBJECT TO BE MOVED BY A PERSON DURING THE STORY:
1056
1057 LIST {num_containers} REASONABLE OPAQUE CONTAINERS THAT COULD CONTAIN
1058 THIS OBJECT:
1059
1060 LIST {num_topics} DISTINCT AND REASONABLE TOPICS THEY COULD BE CHATTING
1061 ABOUT:
1062
1063 To get inspired, make this context happen in {sampled_location}.
1064 Suggested names are {sampled_names}, but feel free to come up with your
1065 own names if it would suit the story better. Be direct with your
1066 answers: do not include parentheses or clarifications beyond the
1067 responses requested. Do not refer to plural objects or give options if
1068 a singular thing is requested. The object could be anything--a pen, an
1069 apple, a spoon, a pair of scissors, etc.--, be creative!

1068 Figure 7: Prompts used for generating a story context, after infilling the variables (number of people,
1069 containers, topics, names, and location). Names and location are sampled independently to increase
1070 diversity, prompts shown in Fig. 8.
1071
1072
1073
1074
1075
1076
1077
1078
1079

```
1080
1081 List 100 names. Do not include any other text.
1082
1083 Suggest 100 different general contexts in which a story may happen. The
1084 context should be able to have several people in the same location
1085 easily listening and observing each other.
1086
1087 1. a school
1088 2. a hospital
1089 3. a vet shop
1090 4. a family living room
1091
1092 Follow the format and make the descriptions as short as possible. Do
1093 not include any text before the list.
```

Figure 8: Prompts used for generating a list of possible characters’ names and locations for the story.

C.2 PROMPTS USED FOR STORY INFILLING

```
1095
1096
1097
1098 You are an expert writer that uses simple language, avoiding sounding
1099 unnatural or cliché. You are clear, creative, and helpful.
1100
```

Figure 9: System prompt used for story infilling

```
1104
1105 Given the following story and knowing the description of the characters
1106 involved, write the start of a story. Don't actually describe any
1107 actions in the story, just the setting in which the story will happen.
1108 Only include the characters that are mentioned in the story.
1109
1110 STORY:
1111 {story_script}
1112
1113 CHARACTERS:
1114 {characters_description}
1115
1116 TWO-SENTENCE STORY BEGINNING THAT DOES NOT INCLUDE OR SUGGEST ANY
1117 INFORMATION OF WHAT WILL HAPPEN IN THE STORY. DO NOT MENTION PEOPLE:
```

Figure 10: System prompt used for sampling narration (the start of the story, before infilling).

```
1118
1119 Given the following story and knowing the description of the characters
1120 involved, suggest a reasonable goal for each character. Only include
1121 the characters that were mentioned in the story.
1122
1123 STORY:
1124 {story_script}
1125
1126 CHARACTERS:
1127 {characters_description}
1128
1129 CHARACTERS GOALS: <insert here>
1130
1131 Follow the format and do not include any other text. Only include the
1132 characters mentioned in the story, and do not even mention the others
1133 in your list.
```

Figure 11: System prompt used for sampling character goals.

1134
 1135 Continue the story {story_length}, clearly conveying the action or
 1136 information below without altering it. Do not contradict any prior
 1137 information. Avoid repeating the information verbatim, instead
 1138 naturally (and possibly implicitly, but still unambiguously) conveying
 1139 the meaning. Do not add characters or actions that were not explicitly
 1140 described. Do not replace characters even if this would improve flow.
 1141 Combining actions into a single sentence is OK as long as you do not
 1142 alter the original information. {infilling_text_type}

1143
 1144 Make it a short, yet an interesting story to read. Make the text
 1145 exciting to read as well as each character's speech, so try to avoid
 1146 e.g. starting all the sentences the same way. The story needs to follow
 1147 common sense, e.g. do not magically change an object's location without
 1148 mentioning it. Do not include any notes, comments, parentheses, or any
 1149 other form of extra text that would not belong in a story. Feel free to
 1150 hint or describe characters' goals and motivations for performing the
 1151 actions if it would make the story flow better.

1152
 1153 As a warning, take into account that when someone tells someone
 1154 privately they might not be in the same location, e.g. they might be
 1155 sending a text message or making a phone call; they might also be in
 1156 the same location, in that case they could also communicate through a
 1157 gesture, a whisper, etc. Do not assume a person is in the same room if
 1158 it has not been made explicit before. Also, if someone was spying, or
 1159 if they were distracted and did not listen/saw something happen, do not
 1160 forget to include it!

1161
 1162 Give {num_tries_completions} responses, ensuring to give
 1163 {num_tries_completions} different phrasings of continuing the story
 1164 conveying the action. Use very different wordings and sentence
 1165 structures!

1166
 1167 WHO ARE THE CHARACTERS: {people_with_personas}

1168
 1169 WHAT ARE THEIR GOALS: {optional_characters_goals}

1170
 1171 NEW ACTION OR INFORMATION TO INCLUDE: {new_information}

1172
 1173 CURRENT SHORT STORY: {story_context}

1174
 1175 Follow the format and do not include any other text. Do not include any
 1176 text before the list. Do not enumerate. Continue the story
 1177 {story_length}. Avoid repeating the information verbatim, instead
 1178 naturally (and possibly implicitly, but still unambiguously) conveying
 1179 the meaning.

1180
 1181 STORY CONTINUATION: <fill>

1182
 1183 STORY CONTINUATION: <fill>

1184
 1185
 1186
 1187

1177 Figure 12: Prompt used for iterative story infilling including characters' goals, and allowing for
 1178 simultaneous sampling of several possible infillings, which when associated with repetition penalty,
 1179 yields more diverse infillings. Infilling length is uniformly chosen between 'with a single sentence'
 1180 and 'with up to two sentences', and infilling text type is uniformly chosen between 'Make the new
 1181 text be declarative, without including conversations.' and 'Make the new text conversational, using
 1182 direct quotes to convey the words spoken by a character.'

1188 D TRACKTHEMIND EXAMPLES (CONT. FROM §3)

1189

1190 D.1 STORY STRUCTURE EXAMPLES

1191

1192

1193

1194

1195

1196

1197

1198

- Addison entered the monastery dining hall.
- Addison filled the large ceramic vase with fresh sunflowers.
- Addison left the monastery dining hall.
- Charlotte entered the monastery dining hall.
- Charlotte painted the large ceramic vase with intricate designs in gold.
- Charlotte glued a few loose diamonds around the neck of the large ceramic vase. While this was happening, Addison witnessed this action in secret (and only this action).

1199

1200

1201

1202

1203

1204

1205

1206

1207

- Amelia entered the staff room.
- Amelia moved the large first aid kit to the plastic storage bin, which is also located in the staff room. While this was happening, Alexis witnessed this action in secret (and only this action).
- Amelia entered the equipment storage room.
- Amelia left the equipment storage room.
- Amelia entered the staff room.
- Amelia moved the large first aid kit to the equipment storage room, leaving the plastic storage bin in its original location.
- Amelia moved the large first aid kit to the metal cabinet, which is also located in the equipment storage room.

1208

1209

1210

1211

1212

1213

1214

1215

- Alexander entered the city hall planning department.
- Alexander moved the large map of the city to the cardboard tube, which is also located in the city hall planning department.
- Alexander told privately to Leslie that the large map of the city is in the cardboard tube.
- Victoria entered the city hall planning department. Leslie told privately to Peyton that the large map of the city is in the cardboard tube.
- Victoria moved the large map of the city to the plastic storage bin, which is also located in the city hall planning department.

1216

1217

1218

1219

1220

1221

1222

1223

1224

1225

1226

1227

1228

1229

1230

1231

1232

1233

1234

1235

1236

1237

1238

1239

1240

1241

Figure 13: Examples of the following story structures: $a_{\text{updateObjState}}$ (\mathcal{A}_2) with asymmetry; $\{a_{\text{moveObjContainer}}, a_{\text{moveObjRoom}}\}$ (\mathcal{A}_4) with asymmetry; and $\{a_{\text{moveObjContainer}}, a_{\text{info}}\}$ (\mathcal{A}_5) without asymmetry.

1242 D.2 STORY INFILLING EXAMPLES
1243
12441245 Table 10: Example of adversarially sampled story structure and iterative infilling with Llama-3.1
1246 70B Instruct, including a story context (first row). This example showcases how the infilling process
1247 is able to group sentences when they depict the same event but for more than one person, allowing
1248 for an even more natural-sounding story.

1249 Story Structure (list of actions)	1249 Infilling for each action
1250 <initial context>	1251 The operations center, a bustling hub of activity during the 1252 park’s peak festival season, was dimly lit except for a few 1253 overhead fluorescent lights that cast a sterile glow over the 1254 rows of workstations and storage units. The air was thick 1255 with the smell of stale coffee and fresh printer paper, a 1256 familiar scent to anyone who spent time behind the scenes of the busy amusement park.
1257 Brooklyn entered the operations 1258 center. 1259 James entered the operations center.	As she entered the operations center, Brooklyn’s presence was swiftly followed by James, whose rugged demeanor stood in contrast to the sterile environment of the room.
1260 James moved the large festival 1261 schedule poster to the cardboard 1262 box, which is also located in the op- 1263 erations center.	James neatly relocated the oversized festival schedule poster, carefully placing it within the confines of a card- board box, a receptacle that had been securely positioned in a corner of the operations center.
1264 James left the operations center. 1265 Brooklyn left the operations center.	Brooklyn stepped out into the bright sunlight, a stark contrast to the dimly lit operations center she had just left, while James followed suit, the warm rays illuminating his weathered features. The festival preparations were now unfolding across the park, with both of them poised to tackle their respective tasks.
1270 Tyler entered the operations center. 1271 Brooklyn entered the operations 1272 center.	As Tyler walked into the operations center, the soft hum of idle machinery and the faint scent of printer paper enveloped him, a familiar yet calming atmosphere before the chaos of the festival. The room’s sterile lighting seemed to intensify, reflecting off the polished surfaces as Brooklyn slipped in, her footsteps quiet on the worn floor.
1276 Brooklyn moved the large festival 1277 schedule poster to the plastic storage 1278 bin, which is also located in the op- 1279 erations center.	Brooklyn’s hands moved swiftly as she carefully slid the oversized festival schedule poster into the large plastic storage bin in the operations center.

1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295

1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349

Table 11: Example of adversarially sampled story structure and simple iterative infilling with Llama-3.1 70B Instruct, along with example questions and model answers.

Story Structure (list of actions)	Infilling for each action		
David entered the study room. Sarah entered the study room.	David walked into the study room, scanning the organized chaos of files on the desk, just as Sarah stepped in, clutching her tablet and a determined look.		
Sarah moved the prototype model to the metal filing cabinet, which is also located in the study room.	Sarah moved the prototype model to the metal filing cabinet, her fingers brushing against the cool surface before turning to David with a confident smile.		
David left the study room.	After a brief exchange with Sarah, David grabbed his briefcase and left the study room, mindful of the upcoming meeting downtown.		
Mark entered the study room.	Mark entered the study room, his eyes quickly assessing the scattered documents and half-finished sketches on the table, before settling into his favorite chair by the window to brainstorm new ideas.		
Mark moved the prototype model to the wooden chest, which is also located in the study room.	Mark, contemplating his latest concept, carefully picked up the prototype model from the metal filing cabinet and transferred it to the wooden chest, a better place for such an inspiration-spurring creation.		
Selected Questions	TRACKTHEMIND-generated answer using our domain-specific language	GPT-4o's responses	Correct?
In which container will David search for the prototype model?	metal filing cabinet	david will search for the prototype model in the metal filing cabinet.	Yes
In which container will Sarah search for the prototype model?	wooden chest	sarah will search for the prototype model in the metal filing cabinet.	No
In which container will Mark search for the prototype model?	wooden chest	mark will search for the prototype model in the wooden chest.	Yes
In which container will David search for the prototype model?	metal filing cabinet	the metal filing cabinet.	Yes
In which container will Sarah search for the prototype model?	wooden chest	sarah will search for the prototype model in the metal filing cabinet.	No
In which container will Mark search for the prototype model?	wooden chest	the wooden chest.	Yes