

Mind the Perspective: Let’s Reason Recursively for Theory of Mind

Chao Lei¹, Guang Hu¹, Meng Yang², Yanbei Jiang¹, Nir Lipovetzky¹

¹ School of Computing and Information Systems, The University of Melbourne, Australia

² SensiLab, Monash University, Australia

{clei1, yanbeij}@student.unimelb.edu.au

Meng.Yang@monash.edu

{guang.hu, nir.lipovetzky}@unimelb.edu.au

Abstract

Theory of Mind (ToM) reasoning requires inferring agents’ beliefs from partial and asymmetric observations, which remains an open challenge for LLMs. Existing prompting-based approaches improve ToM reasoning through observable-event filtering or temporal belief chains, without explicitly modeling nested beliefs. We introduce RECTOM, an inference-time framework for ToM reasoning that models nested beliefs via recursive perspective construction. RECTOM constructs each character perspective from the preceding character perspective along the character chain specified by the question, reducing higher-order belief questions to actual-world questions within the final constructed perspective. We further provide a KD45 analysis showing that RECTOM’s perspective construction induces a well-formed belief modality beyond simple event filtering. Experiments on ToM benchmarks, including Hi-ToM, Big-ToM, and FanToM, across multiple LLM backbones show that RECTOM consistently outperforms recent advanced approaches, achieving state-of-the-art performance. Notably, RECTOM reaches 100% accuracy on Hi-ToM with GPT-5.4 and Qwen3.5, a benchmark requiring higher-order ToM reasoning.

1 Introduction

Theory of Mind (ToM), the ability to reason about others’ beliefs, knowledge, and perspectives, is a central component of social intelligence (Premack and Woodruff, 1978; Wimmer and Perner, 1983; Baron-Cohen et al., 1985). For Large Language Models (LLMs) in interactive settings, ToM is central to handling asymmetric information, multi-agent coordination, and belief-dependent decision making (Rabinowitz et al., 2018; Sap et al., 2022; Gandhi et al., 2023). However, recent studies show that even strong LLMs remain unreliable on ToM tasks that require reconstructing agent-specific beliefs from partial observations, rather

than predicting the final world state (Sap et al., 2022; Gandhi et al., 2023; Wu et al., 2023; Kim et al., 2023). For instance, in a Sally-Anne-style false-belief paradigm, if Alice leaves after placing an object in a box and Bob later moves it to a drawer, Alice would believe the object remains in the box, whereas predicting the drawer reflects an omniscient-state bias (Wimmer and Perner, 1983; Baron-Cohen et al., 1985). Such cases illustrate the epistemic nature of ToM reasoning, where successful reasoning must identify agent-specific observability, preserve beliefs across unobserved intervals, and revise beliefs only under observed relevant evidence. Higher-order questions, such as the second-order question “Where does Alice think Bob will search?”, further challenge ToM reasoning since they require nested belief construction, in which Bob’s belief must be represented within Alice’s perspective rather than inferred as Bob’s actual belief (Perner and Wimmer, 1985; Wu et al., 2023).

Recent prompting-based methods address parts of this challenge through structured intermediate reasoning. SIMTOM uses a two-stage prompting procedure: it first filters the story to the events observable to each character in question, and then prompts the model to answer the ToM question using only the filtered context (Wilf et al., 2024). TIMETOM (Hou et al., 2024) introduces a temporal space by assigning time points to story sentences or dialogue utterances, and constructs a Temporal Belief State Chain (TBSC) for each character. It further separates TBSC into self-world beliefs, which record a character’s belief, and social-world beliefs, which record beliefs about other characters’ actions that may create belief gaps. The former is used for first-order ToM questions, whereas the latter supports higher-order ToM reasoning. For higher-order questions, its Time-Aware Belief Solver identifies each character’s accessible time points, intersects them as belief-communication periods, and reduces higher-order questions to first-order ques-

tions within these periods. SIMTOM and TIMETOM show that ToM reasoning benefits from intermediate representations that specify observable events for each character and the temporal evolution of character beliefs. However, neither SIMTOM nor TIMETOM explicitly constructs nested beliefs for reasoning over one character’s belief within another character’s perspective in higher-order ToM questions. See Appendix A for detailed related work.

To address this limitation, we introduce RECTOM, an inference-time framework that formulates ToM reasoning as recursive symbolic perspective construction for nested belief modeling. RECTOM explicitly models belief states under partial observability, enabling beliefs to persist across unobserved intervals, update under observed evidence, and nest across character perspectives. For each ToM task, each narrative statement or dialogue utterance is abstracted into a fact-based symbolic event and classified as either persistent or transient. Persistent events introduce or revise ontic facts, such as object locations, character locations, and character presence, whereas transient events, such as communications, claims, and questions, allow belief updates under partial observability. RECTOM constructs a global state-event sequence by accumulating ontic facts over the event sequence to build fact-based states and pairing each state with its corresponding persistent or transient event.

From the global state-event sequence, RECTOM constructs perspectives recursively. The global perspective preserves the complete state-event sequence and serves as the initial source perspective. For each character specified in the belief question, RECTOM constructs the character’s perspective in order by completing the character’s partial observation over the current source perspective. Observable states and events are retained locally, unobservable events are removed, and unobservable states are completed by inheriting the preceding belief state and revising it with the paired observable events. The newly constructed perspective then becomes the source perspective for the next character. In this way, nested beliefs are evaluated relative to preceding perspectives rather than the omniscient narrative, reducing higher-order belief questions to actual-world questions within the final constructed perspective, whose final state determines the answer under a closed-world assumption (Reiter, 1981).

When evaluated on well-established ToM bench-

marks, including Hi-ToM (Wu et al., 2023), Big-ToM (Gandhi et al., 2023), and FanToM (Kim et al., 2023), RECTOM consistently outperforms SIMTOM and TIMETOM across multiple LLMs, demonstrating state-of-the-art performance. Its advantage is most evident on Hi-ToM, which includes up to fourth-order questions, where RECTOM achieves 100% accuracy on most evaluated LLMs. These results indicate that recursive perspective construction with explicit fact-based state representations yields robust gains in higher-order and information-asymmetric ToM reasoning. We outline our contributions as follows:

- We introduce RECTOM, an inference-time framework that formulates ToM reasoning as recursive perspective construction, modeling nested beliefs with respect to preceding character perspectives.
- We provide a formal analysis showing that RECTOM’s perspective construction induces a well-formed belief modality satisfying KD45.
- We conduct extensive experiments on Hi-ToM, Big-ToM, and FanToM across diverse LLM backbones, showing that RECTOM consistently outperforms current state-of-the-art approaches.

2 Problem Formulation

We define a ToM instance as

$$\mathcal{I} = (E, q, y), \quad E = (e_1, \dots, e_T),$$

where E is an ordered event sequence, each event $e_t \in E$ corresponds to a narrative statement or dialogue utterance, q is the question, and y is the ground-truth answer.

Let $\mathcal{A}(E)$ denote the characters appearing in E . For a question q , we define its character chain as

$$C(q) = (a_1, \dots, a_K), \quad a_i \in \mathcal{A}(E),$$

where $C(q)$ lists the characters in q from the outermost belief holder a_1 to the innermost belief holder a_K , and K denotes the belief-nesting order. Figure 1 illustrates a Hi-ToM instance, where each e_t is a narrative statement. The zero-order (actual-world) question has $K = 0$ and $C(q) = \emptyset$; the first-order belief question has $K = 1$ and $C(q) = (\text{Elizabeth})$; and the third-order belief question, a higher-order case with $K > 1$, has $K = 3$ with $C(q) = (\text{Elizabeth}, \text{Isabella}, \text{Jacob})$.

3 RECTOM Overview

RECTOM consists of three main steps: 1) global state-event sequence construction; 2) recursive perspective generation for the characters in $C(q)$; and

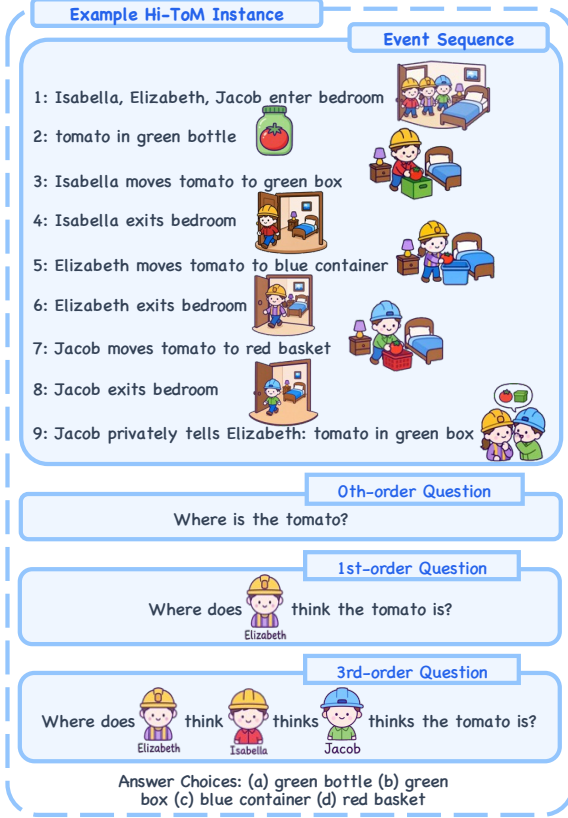


Figure 1: An example Hi-ToM instance with zero-order, first-order, and third-order belief questions over a narrative event sequence.

3) answer inference from the constructed perspective. The overall procedure of RECTOM is illustrated in Figure 2.

3.1 Global State-Event Sequence Construction

3.1.1 Fact-Based Event Abstraction.

Given the event sequence E , RECTOM first abstracts each event $e_t \in E$ into a fact-based symbolic representation and classifies it as either a persistent event π_t or a transient event τ_t . A persistent event π_t introduces or revises ontic facts about world conditions, such as object locations, character locations, and character presence, which remain valid until explicitly revised by later persistent events. In contrast, a transient event τ_t records an event occurrence, such as a communication, claim, or question, that updates characters' beliefs by modifying the facts when they are unobservable. We note that RECTOM generates the fact-based representation using the structured abstraction prompt and determines π_t and τ_t according to the task description. For concision, we use e_t to denote the abstracted fact-based event in the following sec-

tions, unless explicitly noted.

3.1.2 Global State-Event Sequence

RECTOM constructs the symbolic state s_t for each event e_t by updating s_{t-1} with added facts Δ_t^+ and removed facts Δ_t^- , initialized with $s_0 = \emptyset$:

$$s_t = (s_{t-1} \setminus \Delta_t^-) \cup \Delta_t^+, t = 1, \dots, T. \quad (1)$$

When $e_t = \pi_t$, Δ_t^+ and Δ_t^- are specified by the persistent event; when $e_t = \tau_t$, $\Delta_t^+ = \Delta_t^- = \emptyset$, yielding $s_t = s_{t-1}$. Thus, s_t accumulates the ontic facts that hold after processing e_t , providing a complete state representation for belief modeling. RECTOM preserves each abstract event e_t at its original position, resulting in the global state-event sequence:

$$G = \{(s_t, e_t)\}_{t=1}^T,$$

where each pair (s_t, e_t) aligns the symbolic state with its corresponding event, instantiated as either π_t or τ_t . Step 1 in Figure 2 illustrates the construction of G for the Hi-ToM example in Figure 1.

3.2 Perspective Construction

3.2.1 Global Perspective

RECTOM treats the global state-event sequence G as the global perspective, denoted by $P^{(*|G)}$. Here, $*$ indicates that the perspective P is not associated with a specific character, and G serves as its source. We note that G contains complete state-event information independent of any character's beliefs or observations. This allows RECTOM to answer zero-order questions about real-world conditions, which are encoded in the final state of $P^{(*|G)}$, without character-specific belief inference.

3.2.2 Character Perspective

A character perspective encodes the character's belief across the states, combining what the character has observed with what the character continues to believe for unobserved states. Given a source perspective $P^{(\cdot)} = \{(s_t, e_t)\}_{t=1}^T$, RECTOM constructs the perspective of character a_i , denoted by $P^{(a_i|P^{(\cdot)})}$, in two steps.

Character Partial Observation. To model what character a_i has observed, RECTOM derives a partial observation sequence $O^{(a_i|P^{(\cdot)})} = \{(\bar{s}_t, \bar{e}_t)\}_{t=1}^T$ over $P^{(\cdot)}$ for a_i , where $\bar{s}_t = s_t$ if s_t is observable to a_i and $\bar{s}_t = \emptyset$ otherwise, and $\bar{e}_t = e_t$ if e_t is observable to a_i and $\bar{e}_t = \emptyset$ otherwise. Thus, observable states and events in $P^{(\cdot)}$ are retained, unobservable events are discarded, and unobservable

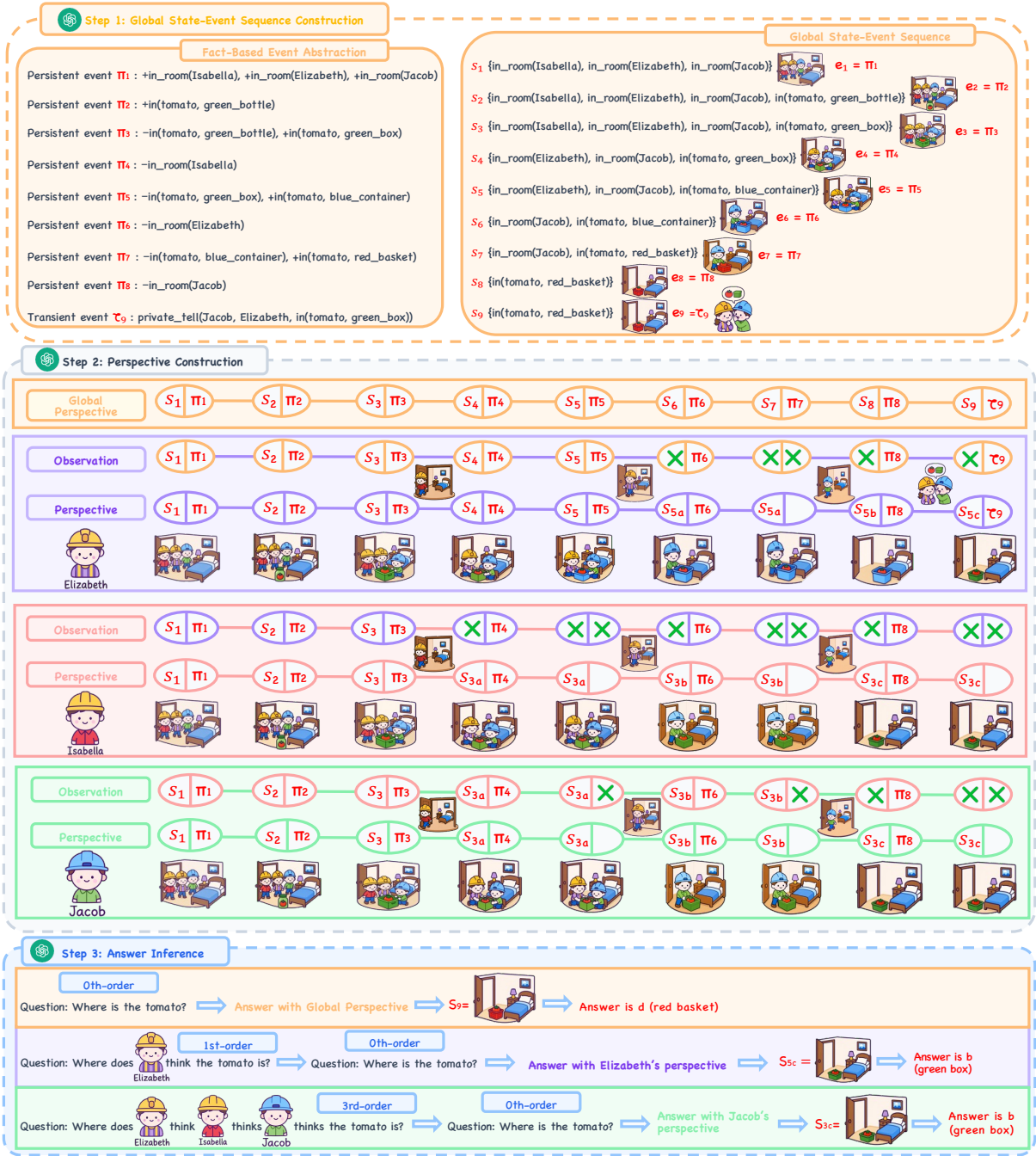


Figure 2: Illustration of the full RECTOM procedure for solving the Hi-ToM instance in Figure 1. Narrative events are abstracted into fact-based persistent events π_i and transient events τ_i , and accumulated into a global state-event sequence $\{(s_t, e_t)\}_{t=1}^T$, with $T = 9$ in this example (Step 1). Character perspectives are recursively constructed along the character chain (Elizabeth, Isabella, Jacob) (Step 2). Crosses mark unobservable states or events, while changes in the suffixes of state labels, such as from s_{5a} to s_{5b} , denote belief-state updates induced by observable events during partial-observation completion. Each constructed perspective serves as the source for constructing the perspective of the next character. The zero-order, first-order, and third-order questions are answered from the final states of the global, Elizabeth's, and Jacob's perspectives, respectively, after reducing belief questions to zero-order questions within their corresponding perspectives (Step 3).

states are masked for later completion. RECTOM evaluates each pair (s_t, e_t) in $P^{(\cdot)}$ according to the task-specified observability rules. For example, in Hi-ToM, a state s_t is observable to a_i when it con-

tains presence facts for a_i , such as $\text{in_room}(a_i)$; a persistent event $e_t = \pi_t$ is observable if its paired state s_t is observable, while room-entry and room-exit events, such as $+\text{in_room}(\text{character})$

or `-in_room(character)`, are observable to all characters; and a transient event $e_t = \tau_t$ is observable when a_i is a participant in the event, such as being the speaker or listener in `private_tell(speaker, listener, fact)`.

Partial Observation Completion. To model what character a_i continues to believe for unobserved states, RECTOM completes the partial observation sequence of a_i . In detail, RECTOM maps the partial observation sequence $O^{(a_i|P^{(\cdot)})} = \{(\bar{s}_t, \bar{e}_t)\}_{t=1}^T$ into the perspective of a_i :

$$P^{(a_i|P^{(\cdot)})} = \{(\hat{s}_t, \bar{e}_t)\}_{t=1}^T,$$

where the event component $\bar{e}_t$ is preserved and the state component $\bar{s}_t$ is completed as $\hat{s}_t$:

$$\hat{s}_t = \begin{cases} \bar{s}_t, & \bar{s}_t \neq \emptyset, \\ f_{\text{com}}(\hat{s}_{t-1}, \bar{e}_t), & \bar{s}_t = \emptyset. \end{cases} \quad (2)$$

Here, the completion function f_{com} is applied when state is unobservable, i.e., $\bar{s}_t = \emptyset$. It inherits the preceding completed state as the basis of the character’s belief to preserve belief continuity, and revises the inherited state with $\bar{e}_t$ when it is observable, i.e., $\bar{e}_t \neq \emptyset$, to account for belief updates. This follows the belief-persistence assumption that a character’s belief remains unchanged unless revised by observable evidence, consistent with prior work (Hu et al., 2023; Goldman, 1979). In practice, RECTOM updates $\hat{s}_{t-1}$ with $\bar{e}_t$ using Eq. (1) when $\bar{e}_t = \tau_t$, and revises $\hat{s}_{t-1}$ according to the task-specified transient-event rules when $\bar{e}_t = \tau_t$.

Recursive Perspective Generation For a belief question of order $K > 0$, the global perspective $P^{(*|G)}$ serves as the source perspective for recursively constructing character perspectives along the character chain $C(q)$. Each constructed perspective then serves as the source for the next character, thereby modeling nested beliefs relative to the preceding perspective, as shown in Step 2 of Figure 2.

3.3 Answer Inference

Recursive perspective construction reduces first-order and higher-order questions to zero-order questions within the final constructed perspective. Under the closed-world assumption, RECTOM answers the reduced question using the final state of this perspective, which encodes the innermost character’s belief conditioned on the preceding perspectives in $C(q)$, as illustrated in Step 3 of Figure 2.

Algorithm 1: Pseudocode of RECTOM

```

Input   : Event sequence  $E = (e_1, \dots, e_T)$ ; ToM
           : question  $q$ ; LLM  $\mathcal{M}$ 
Output : Predicted answer  $\hat{y}$ 
/* Global Perspective Construction */
// Fact-based event abstraction and
// event-type classification
1  $\mathcal{F} = \{(\Delta_t^+, \Delta_t^-, e_t), e_t \in \{\pi_t, \tau_t\}\}_{t=1}^T \leftarrow \mathcal{M}(E)$ ;
// Construct the global perspective
2  $P^{(*|G)} \leftarrow \text{BuildGlobalPerspective}(\mathcal{F})$ ;
/* Recursive Perspective Construction */
// Extract the character chain
3  $C(q) = (a_1, \dots, a_K) \leftarrow \mathcal{M}(q)$ ;
//  $C(q) = \emptyset$  indicates a zero-order question
4 if  $C(q) = \emptyset$  then
    // Answer from the global perspective
    5  $\hat{y} \leftarrow \mathcal{M}(q, \text{FinalState}(P^{(*|G)}))$ ;
    6 return  $\hat{y}$ ;
    // Initialize source perspective
    7  $P^{(\cdot)} \leftarrow P^{(*|G)}$ ;
    8 foreach  $a_i \in C(q)$  do
        // Build partial observation sequence
        9  $O^{(a_i|P^{(\cdot)})} \leftarrow \mathcal{M}(a_i, P^{(\cdot)})$ ;
        // Complete partial observation
        10  $P^{(a_i|P^{(\cdot)})} \leftarrow \text{Complete}(O^{(a_i|P^{(\cdot)})}, \mathcal{M})$ ;
        // Update source perspective
        11  $P^{(\cdot)} \leftarrow P^{(a_i|P^{(\cdot)})}$ ;
    /* Answer Inference */
    // Reduce to a zero-order question
    12  $q^{(0)} \leftarrow \mathcal{M}(q, C(q))$ ;
    // Answer from the final perspective
    13  $\hat{y} \leftarrow \mathcal{M}(q^{(0)}, \text{FinalState}(P^{(\cdot)}))$ ;
    14 return  $\hat{y}$ ;

```

3.4 Pseudocode of RECTOM

Algorithm 1 summarizes the full procedure of RECTOM. The LLM $\mathcal{M}$ is used for natural-language interpretation and semantic reasoning, including event abstraction and event-type classification (Line 1), character-chain extraction (Line 3), observability judgment (Line 9), transient-event belief revision within `Complete(·)` (Line 10), question reduction (Line 12), and answer inference (Lines 5 and 13). In contrast, global state accumulation from persistent events during global-perspective construction (Line 2) and persistent-event updates inside `Complete(·)` (Line 10) follow the deterministic update rule in Eq. (1). RECTOM combines the semantic reasoning capacity of LLMs with deterministic updates to improve the reliability of belief persistence, belief revision, and recursive perspective construction in ToM reasoning.

4 KD45 Analysis of RECTOM

We use KD45, a standard modal logic for belief, to show that RECTOM’s perspective construction

induces a well-formed belief modality beyond simple event filtering (Malcolm, 1952; Fagin et al., 1995). In this analysis, K requires a constructed perspective to support standard logical inference, D requires internal consistency, and 4 and 5 concern the preservation of beliefs and non-beliefs under repeated perspective construction for the same character. For concision, we provide a proof sketch here and show the full proof in Appendix C.

Let φ and ψ denote belief-query formulas, which may be symbolic facts or nested belief statements. For a character a_i , let $B_{a_i}\varphi$ denote that a_i believes φ . RECTOM interprets $B_{a_i}\varphi$ by constructing a_i 's perspective, $P(a_i|P^{(\cdot)})$, from a source perspective $P^{(\cdot)}$, and evaluating φ inside $P(a_i|P^{(\cdot)})$:

$$P^{(\cdot)} \models B_{a_i}\varphi \quad \text{iff} \quad P(a_i|P^{(\cdot)}) \models \varphi. \quad (3)$$

Thus, φ is evaluated in the constructed character perspective rather than in the source perspective.

4.1 Self-Perspective Idempotence

The key property for proving KD45 is that the perspective is stable under repeated construction for the same character:

$$P(a_i|P(a_i|P^{(\cdot)})) = P(a_i|P^{(\cdot)}). \quad (4)$$

RECTOM satisfies this property because reconstructing the partial observation sequence of a_i from $P(a_i|P^{(\cdot)})$ yields the same partial observation sequence, so the subsequent completion remains the same and produces the same perspective.

4.2 KD45 Satisfaction

For RECTOM, K holds because the constructed perspective supports standard logical inference (if $\varphi \rightarrow \psi$ and φ hold then ψ holds). According to Eq. (3), $B_{a_i}(\varphi \rightarrow \psi)$ and $B_{a_i}\varphi$ holding in $P^{(\cdot)}$ implies that $\varphi \rightarrow \psi$ and φ hold in $P(a_i|P^{(\cdot)})$; hence ψ holds in $P(a_i|P^{(\cdot)})$, therefore $B_{a_i}\psi$ holds in $P^{(\cdot)}$. D holds because each constructed perspective maintains the consistent symbolic state under the closed-world assumption, so a fact and its negation cannot both hold. 4 and 5 follow from Eq. (4): reconstructing the same character's perspective returns the same perspective, so both beliefs and non-beliefs are preserved under repeated construction. Therefore, the belief modality induced by RECTOM's perspective construction satisfies KD45.

5 Experiments

We first evaluate RECTOM on Hi-ToM, a benchmark designed to assess higher-order ToM reasoning over narrative event sequences (Figure 1). We select 400 tasks from Hi-ToM, covering question orders from zero to four, with 80 instances per order. Each task requires selecting the correct answer from 15 candidate choices. We compare RECTOM against current state-of-the-art approaches: Chain-of-Thought prompting (CoT), SIMTOM, and TIMETOM. To examine robustness across model families and scales, we evaluate all methods with multiple LLM backbones: proprietary models GPT-5.4 (OpenAI, 2026) and Gemini-3-Flash (DeepMind, 2025), the open-source dense model Qwen3.5-27B (Qwen, 2026), and the open-source Mixture-of-Experts model Gemma-4-26B-A4B (DeepMind, 2026). See Appendix D for parameter settings.

5.1 Results on Hi-ToM

Table 1 reports Hi-ToM accuracy for zero-order to fourth-order questions, together with overall performance. RECTOM achieves the highest overall accuracy across all LLM backbones, reaching 100.00% with GPT-5.4 and Qwen3.5, 98.50% with Gemini-3, and 99.50% with Gemma-4, demonstrating state-of-the-art performance. Compared with the strongest baseline for each backbone, RECTOM yields absolute overall accuracy gains of 9.75%, 6.25%, 9.25%, and 15.75%, respectively, with the largest improvement over SIMTOM using Gemma-4. The order-wise results show that all methods perform strongly on zero-order questions, whereas the baselines generally degrade as the order increases, reflecting the difficulty of inferring nested character beliefs under asymmetric information. In contrast, RECTOM maintains near-perfect accuracy across question orders, with the lowest order-wise accuracy still reaching 97.50% on Gemini-3 and Gemma-4. The few remaining errors arise from LLM semantic interpretation failures, such as incorrect observability identification. These results indicate that recursive perspective construction provides a robust mechanism for higher-order ToM reasoning and demonstrate the model-independent nature of RECTOM.

5.2 Generalization across ToM Scenarios

We further evaluate RECTOM on benchmarks, BigToM and FanToM, under the same experimen-

Model	Method	Hi-ToM					
		0th-Order	1st-Order	2nd-Order	3rd-Order	4th-Order	Overall
GPT-5.4	CoT	<u>100.00</u>	88.75	60.00	68.75	71.25	77.75
	SIMToM	<u>100.00</u>	<u>92.50</u>	88.75	<u>87.50</u>	80.00	89.75
	TIMEToM	<u>100.00</u>	<u>92.50</u>	<u>90.00</u>	<u>87.50</u>	<u>81.25</u>	<u>90.25</u>
	RECToM (ours)	100.00	100.00	100.00	100.00	100.00	100.00
Gemini-3	CoT	<u>100.00</u>	<u>100.00</u>	92.50	85.00	83.75	92.25
	SIMToM	<u>100.00</u>	91.25	90.00	<u>88.75</u>	86.25	91.25
	TIMEToM	<u>100.00</u>	78.75	86.25	86.25	<u>87.50</u>	87.75
	RECToM (ours)	100.00	100.00	97.50	97.50	97.50	98.50
Qwen3.5	CoT	<u>100.00</u>	<u>95.00</u>	88.75	83.75	85.00	90.50
	SIMToM	<u>100.00</u>	87.50	<u>90.00</u>	<u>90.00</u>	<u>86.25</u>	<u>90.75</u>
	TIMEToM	<u>100.00</u>	70.00	87.50	86.25	82.50	85.25
	RECToM (ours)	100.00	100.00	100.00	100.00	100.00	100.00
Gemma-4	CoT	<u>100.00</u>	<u>93.75</u>	65.00	62.50	58.75	76.00
	SIMToM	98.75	91.25	<u>81.25</u>	<u>73.75</u>	<u>73.75</u>	83.75
	TIMEToM	97.50	70.00	60.00	63.75	60.00	70.25
	RECToM (ours)	100.00	100.00	100.00	100.00	97.50	99.50

Table 1: Accuracy (%) on Hi-ToM across question orders and overall performance. Each order contains 80 instances, with 400 tasks in total. The best result for each backbone and metric is in bold. The second-best result is underlined.

Model	Method	Big-ToM	FanToM		
		Overall	1st-Order	2nd-Order	Overall
GPT-5.4	CoT	<u>99.00</u>	88.95	86.71	87.96
	SIMToM	<u>99.00</u>	91.16	<u>95.80</u>	<u>93.21</u>
	TIMEToM	97.75	<u>92.27</u>	92.31	92.28
	RECToM (ours)	99.50	96.13	99.30	97.53
Gemini-3	CoT	<u>98.75</u>	77.35	<u>90.21</u>	83.02
	SIMToM	98.00	<u>82.32</u>	89.51	<u>85.49</u>
	TIMEToM	92.50	74.59	91.61	82.10
	RECToM (ours)	99.00	91.16	91.61	91.36
Qwen3.5	CoT	<u>96.25</u>	78.45	<u>85.31</u>	81.48
	SIMToM	94.75	<u>89.50</u>	81.82	<u>86.11</u>
	TIMEToM	91.00	86.74	<u>85.31</u>	<u>86.11</u>
	RECToM (ours)	98.50	92.82	86.01	89.81
Gemma-4	CoT	<u>93.25</u>	63.54	75.52	68.83
	SIMToM	88.00	71.27	<u>80.42</u>	75.31
	TIMEToM	80.00	<u>76.24</u>	79.72	<u>77.78</u>
	RECToM (ours)	99.00	87.29	84.62	86.11

Table 2: Accuracy (%) on Big-ToM and FanToM. Big-ToM contains 400 instances, while FanToM contains 181 first-order and 143 second-order instances, with 324 instances in total. The best result for each backbone and metric is in bold. The second-best result is underlined.

tal settings as Table 1 to examine its generality across different ToM scenarios. Similar to Hi-ToM, Big-ToM follows the Sally–Anne paradigm, while presents stories in more natural language and extends beyond object-location changes. Following prior work (Wilf et al., 2024; Hou et al., 2024), we evaluate 400 forward-belief questions from Big-ToM, consisting of 200 false-belief and 200 true-belief first-order questions. FanToM evaluates ToM reasoning in interactive dialogue scenarios, where characters enter and leave ongoing conversations, creating asymmetric information access and dis-

tinct mental states (Quesque and Rossetti, 2020). We evaluate 324 FanToM belief questions, including 181 first-order and 143 second-order questions. Detailed benchmark descriptions for Big-ToM and FanToM are provided in Appendix E.

The results are reported in Table 2. RECToM achieves the highest overall accuracy for every backbone on both datasets, reaching up to 99.50% on Big-ToM and 97.53% on FanToM, with the largest absolute gain of 8.33% over the second-best method on FanToM with Gemma-4. It also obtains the best or tied-best order-specific accuracy on FanToM. These results show that RECToM generalizes from narrative event sequences to interactive dialogue scenarios. For RECToM, errors in FanToM mainly reflect incorrect semantic grounding by LLMs. For example, partial exposure, where a character enters the conversation late and hears only the “tail end” of a conversation, is incorrectly treated as access to the full preceding dialogue. Moreover, when later utterances are semantically related to earlier ones, LLMs may incorrectly assume that a character who joined later also knows information mentioned before.

5.3 Cost Analysis

Table 3 compares the average token usage per problem (Tok) and token efficiency (Eff) relative to CoT across the evaluated approaches using GPT-5.4 and Gemini-3 as backbones. RECToM shows higher token usage. However, when evaluated by token efficiency, measured as the overall accuracy gain

Model	Method	Hi-ToM		Big-ToM		FanToM	
		Tok	Eff	Tok	Eff	Tok	Eff
GPT-5.4	CoT	0.6K	–	0.3K	–	0.9K	–
	SIMToM	2.3K	7.3	0.5K	0.0	4.0K	1.7
	TIMEToM	3.0K	5.4	1.1K	-1.5	5.4K	1.0
	RECToM (ours)	6.6K	3.8	3.4K	0.2	8.7K	1.2
Gemini-3	CoT	1.1K	–	0.4K	–	1.1K	–
	SIMToM	3.7K	-0.4	1.2K	-1.0	5.7K	0.5
	TIMEToM	7.5K	-0.7	3.0K	-2.4	9.4K	-0.1
	RECToM (ours)	9.4K	0.7	4.8K	0.1	10.2K	0.9

Table 3: Cost analysis in the proprietary backbones GPT-5.4 and Gemini-3. Tok denotes the average token usage per question. K is 10^3 . Eff denotes token efficiency relative to CoT, computed as $\Delta\text{Acc}/\Delta\text{Tok}$, where ΔAcc and ΔTok are the overall accuracy gain and additional token usage over CoT, respectively. Higher values indicate better token efficiency. Negative values indicate that a method consumes more tokens than CoT while achieving lower overall accuracy.

obtained from each additional 1K tokens relative to CoT, RECToM achieves the highest efficiency across all Gemini-3 settings and on Big-ToM with GPT-5.4. It is lower than SIMToM on FanToM with GPT-5.4, while efficiency remains comparable (1.7 vs. 1.2). RECToM is less token-efficient on Hi-ToM with GPT-5.4, where higher-order questions require deeper perspective construction. However, this additional computation supports near-perfect accuracy across all LLM backbones on this challenging benchmark, as reported in Table 1.

5.4 Ablation Study

To examine the contribution of deterministic state updates and symbolic state representation, we compare RECToM with two ablated variants. The *w/o-det* variant retains symbolic states, while replacing deterministic state updates with LLM-based execution under the same update rule in Eq. (1). The *w/o-state* variant removes state representation from each perspective while preserving recursive perspective construction along $C(q)$. Instead of constructing perspectives through observation-based completion over state-event pairs, it maintains a cumulative observable event history at each step, $\bar{E}_t = \{\bar{e}_1, \dots, \bar{e}_t\}$, yielding the perspective $\tilde{P} = \{\bar{E}_t\}_{t=1}^T$.

Table 4 reports the ablation results. The *w/o-det* variant leads to slight performance degradation, suggesting that deterministic updates reduce variability in belief-state maintenance. Removing symbolic state representation, the *w/o-state* variant,

Backbone	Method	Hi-ToM		Big-ToM		FanToM	
		Acc	Δ	Acc	Δ	Acc	Δ
GPT-5.4	RECToM	100.00	–	99.50	–	97.53	–
	<i>w/o-det</i>	100.00	0.00	99.50	0.00	97.53	0.00
	<i>w/o-state</i>	93.00	7.00	99.25	0.25	94.44	3.09
Gemini-3	RECToM	98.50	–	99.00	–	91.36	–
	<i>w/o-det</i>	98.00	0.50	99.00	0.00	90.43	0.93
	<i>w/o-state</i>	94.25	4.25	98.75	0.25	87.65	3.70
Qwen3.5	RECToM	100.00	–	98.50	–	89.81	–
	<i>w/o-det</i>	100.00	0.00	98.50	0.00	89.50	0.31
	<i>w/o-state</i>	93.50	6.50	97.25	1.25	87.35	2.47
Gemma-4	RECToM	99.50	–	99.00	–	86.11	–
	<i>w/o-det</i>	98.00	1.50	98.75	0.25	85.19	0.93
	<i>w/o-state</i>	88.25	11.25	95.25	3.75	80.86	5.25

Table 4: Ablation study of RECToM. *w/o-det* replaces deterministic state updates with LLM-based state updating. *w/o-state* removes symbolic state construction and maintains cumulative observable event sequences in each perspective. Acc denotes overall accuracy (%). Δ denotes the accuracy decrease relative to RECToM.

consistently reduces performance across all benchmarks and backbones, with the largest degradation of 11.25% recorded on Hi-ToM using Gemma-4. These results demonstrate that fact-based state representations provide a more reliable basis for belief reasoning than cumulative observable event histories. The *w/o-state* variant requires the LLM to implicitly infer the observability from accumulated event semantics, as well as belief persistence and revision across observed and unobserved events, which can lead to incorrect observability judgments, inconsistent beliefs, and erroneous answer inference. In contrast, RECToM explicitly constructs fact-based states after each event, encoding belief-relevant conditions through ontic fact updates. The fact-based state representation supports accurate observability evaluation via explicit character presence and location facts and enables answer inference directly from ontic facts in the constructed perspective. These advantages are evident in higher-order reasoning on Hi-ToM and FanToM, where errors can propagate through recursively constructed perspectives.

6 Conclusion

We introduced RECToM, an inference-time framework for ToM reasoning. RECToM models nested beliefs by recursively constructing character perspectives, where each constructed perspective becomes the source for constructing the next character’s perspective. This reduces higher-order belief questions to actual-world questions within the per-

spective of the innermost character specified by the question. We further provided a KD45 analysis showing that RECTOM’s perspective construction induces a well-formed belief structure beyond simple event filtering. Experiments on Hi-ToM, Big-ToM, and FanToM benchmarks demonstrate that RECTOM consistently outperforms recent advanced approaches, across multiple LLM backbones, achieving state-of-the-art performance with the strongest gains on higher-order ToM questions.

7 Limitations

RECTOM is designed for controlled text-based ToM benchmarks where event-transition rules and observability assumptions are specified by the task description. While this setting covers higher-order beliefs, asymmetric information, and dialogue-based belief reasoning, extending RECTOM to open-ended or multimodal environments may require additional grounding of implicit observations and event-transition rules. Moreover, RECTOM explicitly constructs character perspectives before deriving the final answer, which introduces additional inference-time computation. Future work can reduce this cost through prompt compression, caching shared perspective states, or lightweight symbolic abstraction.

References

- Alexandru Baltag, Lawrence S. Moss, and Slawomir Solecki. 1998. The logic of public announcements, common knowledge, and private suspicions. In *Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge, TARK ’98*, page 43–56, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Simon Baron-Cohen, Alan M Leslie, and Uta Frith. 1985. Does the autistic child have a “theory of mind”? *Cognition*, 21(1):37–46.
- Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. [Large language monkeys: Scaling inference compute with repeated sampling](#). *Preprint*, arXiv:2407.21787.
- Zhuang Chen, Jincenzi Wu, Jinfeng Zhou, Bosi Wen, Guanqun Bi, Gongyao Jiang, Yaru Cao, Mengting Hu, Yunghwei Lai, Zexuan Xiong, and Minlie Huang. 2024. [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)*, pages 15959–15983, Bangkok, Thailand. Association for Computational Linguistics.
- Google DeepMind. 2025. [Gemini 3 developer guide](#). Accessed: 2026-05-21.
- Google DeepMind. 2026. [Gemma 4](#). Accessed: 2026-05-21.
- Ronald Fagin, Joseph Y Halpern, Yoram Moses, and Moshe Vardi. 1995. *Reasoning about knowledge*. The MIT press.
- Kanishk Gandhi, J.-Philipp Fränken, Tobias Gerstenberg, and Noah D. Goodman. 2023. Understanding social reasoning in language models with language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NeurIPS*, Red Hook, NY, USA.
- Alvin I. Goldman. 1979. [What is justified belief?](#) In *Justification and Knowledge: New Studies in Epistemology*, pages 1–23. Springer Netherlands, Dordrecht.
- Guiyang Hou, Wenqi Zhang, Yongliang Shen, Linjuan Wu, and Weiming Lu. 2024. Timetom: Temporal space is the key to unlocking the door of large language models’ theory-of-mind. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11532–11547, Bangkok, Thailand. Association for Computational Linguistics.
- Guang Hu, Tim Miller, and Nir Lipovetzky. 2023. Planning with multi-agent belief using justified perspectives. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 33, pages 180–188.
- Chani Jung, Dongkwan Kim, Jiho Jin, Jiseon Kim, Yeon Seonwoo, Yejin Choi, Alice Oh, and Hyunwoo Kim. 2024. Perceptions to beliefs: Exploring precursory inferences for theory of mind in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19794–19809, Miami, Florida, USA. Association for Computational Linguistics.
- Hyunwoo Kim, Melanie Sclar, Xuhui Zhou, Ronan Bras, Gunhee Kim, Yejin Choi, and Maarten Sap. 2023. FANToM: A benchmark for stress-testing machine theory of mind in interactions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14397–14413, Singapore. Association for Computational Linguistics.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NeurIPS*, Red Hook, NY, USA.
- Matthew Le, Y-Lan Boureau, and Maximilian Nickel. 2019. Revisiting the evaluation of theory of mind through question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5872–5877.

- Norman Malcolm. 1952. [Knowledge and belief](#). *Mind*, 61(242):178–189.
- OpenAI. 2026. [Introducing gpt-5.4](#). Accessed: 2026-05-21.
- Josef Perner and Heinz Wimmer. 1985. “john thinks that mary thinks that. . .” attribution of second-order beliefs by 5-to 10-year-old children. *Journal of experimental child psychology*, 39(3):437–471.
- David Premack and Guy Woodruff. 1978. Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences*, 1(4):515–526.
- François Quesque and Yves Rossetti. 2020. What do theory-of-mind tasks actually measure? theory and practice. *Perspectives on psychological science*, 15(2):384–396.
- Qwen. 2026. [Qwen3.5: Towards native multimodal agents](#). Accessed: 2026-05-21.
- Neil Rabinowitz, Frank Perbet, Francis Song, Chiyuan Zhang, SM Ali Eslami, and Matthew Botvinick. 2018. Machine theory of mind. In *International conference on machine learning*, pages 4218–4227. PMLR.
- Raymond Reiter. 1981. On closed world data bases. In *Readings in artificial intelligence*, pages 119–140. Elsevier.
- Maarten Sap, Ronan Le Bras, Daniel Fried, and Yejin Choi. 2022. Neural theory-of-mind? on the limits of social intelligence in large lms. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pages 3762–3780.
- Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. 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)*, pages 4463–4473.
- Melanie Sclar, Jane Dwivedi-Yu, Maryam Fazel-Zarandi, Yulia Tsvetkov, Yonatan Bisk, Yejin Choi, and Asli Celikyilmaz. 2025. Explore theory of mind: program-guided adversarial data generation for theory of mind reasoning. In *The Thirteenth International Conference on Learning Representations*.
- Melanie Sclar, Sachin Kumar, Peter West, Alane Suhr, Yejin Choi, and Yulia Tsvetkov. 2023. Minding language models’ (lack of) theory of mind: A plug-and-play multi-character belief tracker. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13960–13980, Toronto, Canada. Association for Computational Linguistics.
- Natalie Shapira, Mosh Levy, Seyed Hossein Alavi, Xuhui Zhou, Yejin Choi, Yoav Goldberg, Maarten Sap, and Vered Shwartz. 2024. Clever hans or neural theory of mind? stress testing social reasoning in large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2257–2273.
- Damien Sileo and Antoine Lerneoud. 2023. MindGames: Targeting theory of mind in large language models with dynamic epistemic modal logic. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4570–4577, Singapore. Association for Computational Linguistics.
- Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*.
- Tomer Ullman. 2023. [Large language models fail on trivial alterations to theory-of-mind tasks](#). *Preprint*, arXiv:2302.08399.
- Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. 2008. *Dynamic Epistemic Logic*. Springer, Dordrecht, Nederland.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. [Self-consistency improves chain of thought reasoning in language models](#). In *The Eleventh International Conference on Learning Representations*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NeurIPS*, Red Hook, NY, USA.
- Alex Wilf, Sihyun Lee, Paul Pu Liang, and Louis-Philippe Morency. 2024. Think twice: Perspective-taking improves large language models’ theory-of-mind capabilities. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8292–8308, Bangkok, Thailand. Association for Computational Linguistics.
- Heinz Wimmer and Josef Perner. 1983. Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children’s understanding of deception. *Cognition*, 13(1):103–128.
- Yufan Wu, Yinghui He, Yilin Jia, Rada Mihalcea, Yulong Chen, and Naihao Deng. 2023. Hi-ToM: A benchmark for evaluating higher-order theory of mind reasoning in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10691–10706, Singapore. Association for Computational Linguistics.
- Yuheng Wu, Jianwen Xie, Denghui Zhang, and Zhaozhuo Xu. 2025. [Del-tom: Inference-time scaling for theory-of-mind reasoning via dynamic epistemic logic](#). In *Proceedings of the 2025 Conference*

on Empirical Methods in Natural Language Processing, pages 11383–11397, Suzhou, China. Association for Computational Linguistics.

Hainiu Xu, Siya Qi, Jiazheng Li, Yuxiang Zhou, Jinhua Du, Caroline Catmur, and Yulan He. 2025. EnigmaToM: Improve LLMs’ theory-of-mind reasoning capabilities with neural knowledge base of entity states. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 13598–13622, Vienna, Austria. Association for Computational Linguistics.

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

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Zhining Zhang, Chuanyang Jin, Mung Yao Jia, Shunchi Zhang, and Tianmin Shu. 2025. Autotom: Scaling model-based mental inference via automated agent modeling. In *Advances in Neural Information Processing Systems*, pages 129659–129699.

Appendix

A Related Work

Early ToM benchmarks adapted classic false-belief paradigms into text-based question answering. ToMi (Le et al., 2019), for example, evaluates whether models can distinguish reality from an agent’s belief in Sally–Anne-style narratives. SocialIQA (Sap et al., 2019) broadened the scope from belief tracking to everyday social common-sense, including intents, reactions, and social consequences. However, subsequent evaluations showed that LLMs often lack robust ToM-like reasoning. Sap et al. (2022) found that large pretrained models underperform humans on social reasoning and false-belief tasks. Ullman (2023) further showed that seemingly minor perturbations to classic ToM scenarios can cause dramatic failures. Moreover, Shapira et al. (2024) argued that apparent success on ToM-style problems may reflect shallow heuristics rather than stable mental-state reasoning. These findings motivate more controlled, diverse, and leakage-resistant evaluations of ToM in LLMs.

Recent work has therefore developed richer ToM benchmarks that move beyond isolated first-order false-belief questions. Big-ToM (Gandhi et al., 2023) uses causal templates to generate controlled scenarios involving percepts, beliefs, desires, and actions, enabling systematic tests of forward belief inference, forward action inference, and backward belief inference. FanToM (Kim et al., 2023) shifts evaluation from passive narratives to multi-party conversations in which characters enter and leave discussions, creating information asymmetries between participants. Its results show that models may answer one question format correctly while failing logically related answerability or information-access questions, revealing inconsistent performance across question types. Hi-TOM (Wu et al., 2023) focuses on higher-order recursive beliefs, extending evaluation to third-order and fourth-order ToM and incorporating public and private deceptive communication. More recent benchmarks further broaden the evaluation landscape: ToMBench (Chen et al., 2024) introduces a large-scale bilingual benchmark spanning multiple mental-state categories and abilities; OpenToM (Xu et al., 2024) evaluates longer stories with richer psychological states; and ExploreToM (Sclar et al., 2025) uses program-guided adversarial generation to produce challenging and diverse ToM scenarios. Together, these benchmarks show that robust

ToM evaluation requires tracking event access, information asymmetry, and higher-order recursive beliefs.

A parallel line of work studies how to improve LLMs’ ToM reasoning. General-purpose reasoning methods such as chain-of-thought prompting (Wei et al., 2022; Kojima et al., 2022), self-consistency (Wang et al., 2023), and tree-of-thought search (Yao et al., 2023) improve performance on many symbolic and mathematical reasoning tasks, but their effect on ToM is mixed. Several ToM benchmarks report that CoT provides limited gains or can even amplify errors when models adopt an incorrect perspective or propagate a mistaken intermediate belief (Gandhi et al., 2023; Kim et al., 2023; Wu et al., 2023). This suggests that ToM reasoning requires not simply longer rationales, but appropriate intermediate representations for tracking the evolution of agents’ beliefs. SIMTOM (Wilf et al., 2024) addresses this by decomposing ToM into perspective-taking and question answering: the model first filters the story to what the target character knows, and then answers from that character’s perspective. This two-stage decomposition substantially improves performance over standard prompting baselines, showing that explicitly constructing character perspectives is beneficial.

Subsequent methods make this intermediate structure more explicit. SYMBOLICTOM (Sclar et al., 2023) introduces a plug-and-play multi-character belief tracker, using graphical symbolic representations to track entities’ beliefs, their estimates of other entities’ beliefs, and higher-order beliefs rather than relying on implicit language-model inference alone. TIMETOM (Hou et al., 2024) further emphasizes temporal structure by constructing a temporal space and per-character Temporal Belief State Chains (TBSCs), distinguishing self-world beliefs from social-world beliefs and using shared belief-communication periods with a Time-Aware Belief Solver to support higher-order reasoning. PERCEPTOM (Jung et al., 2024) decomposes ToM into perception inference and perception-to-belief inference, showing that LLMs may identify what an agent can perceive while still failing to convert that perception into the correct belief state. ENIGMATOM (Xu et al., 2025) extends this direction with structured entity-state knowledge and spatial scene graphs for belief tracking, together with iterative perspective masking for perspective-taking in higher-order ToM reasoning. These studies collectively suggest that ToM failures often arise from in-

Method	Intermediate Representation	Belief-Tracking Mechanism	Higher-Order / Nested-Belief Mechanism
SIMToM	Character-specific filtered context	Context filtering by character knowledge	Answering from the filtered context
SYMBOLICToM	Graphical symbolic belief representations	Tracking entities’ beliefs and beliefs about others	Tracking beliefs about others’ beliefs across higher orders
TIMEToM	Temporal Belief State Chains (TBSCs)	Per-character self-world and social-world belief chains	Belief-communication periods with a Time-Aware Belief Solver
PERCEPToM	Character-specific perception information	Perception-to-belief inference	No
ENIGMAToM	Entity-state knowledge and spatial scene graphs	Entity-state-based belief tracking across ToM orders	Iterative perspective masking
DEL-ToM	DEL-grounded belief-update traces	Step-wise belief-update scoring with a Process Belief Model	DEL-grounded belief updates for higher-order reasoning
RECToM (ours)	Fact-based symbolic state-event sequences	Belief persistence and updates via partial-observation completion	Recursive perspective construction from the preceding perspective

Table 5: Comparison of structured Theory-of-Mind reasoning methods in terms of intermediate representations, belief-tracking mechanisms, and higher-order or nested-belief mechanisms. “No” indicates that the method does not explicitly define the corresponding mechanism.

sufficient tracking of belief evolution: models struggle to determine which information is observable to each character and whether subsequent events should update that character’s beliefs.

Formal and verifier-based approaches provide another route to structured ToM reasoning. Dynamic epistemic logic (DEL) offers a principled formalism for representing belief states, event models, and belief updates in multi-agent settings (Baltag et al., 1998; van Ditmarsch et al., 2008). MINDGAMES (Sileo and Lernould, 2023) uses epistemic logic to generate controlled reasoning problems. DEL-ToM (Wu et al., 2025) formalizes ToM as a sequence of DEL-grounded belief updates. DEL-ToM trains a Process Belief Model using labels generated by a DEL simulator to score intermediate belief updates and applies inference-time scaling to select high-scoring belief traces from multiple LLM-generated candidates. This connects ToM reasoning to broader work on process supervision and inference-time scaling, where voting, search, or verifiers are used to select among candidate reasoning traces (Wang et al., 2023; Brown et al., 2024; Snell et al., 2025). This line of work highlights the importance of intermediate belief-update supervision, yet typically relies on external simulators or verifier-based selection rather than direct inference-time perspective construction.

Table 5 compares representative structured ToM

reasoning methods in terms of their intermediate representations, belief-tracking mechanisms, and higher-order or nested-belief mechanisms.

RECToM differs from these approaches primarily in how nested beliefs are represented and constructed. SIMToM reasons from character-specific filtered contexts; SYMBOLICToM explicitly tracks entities’ beliefs and beliefs about others; TIMEToM constructs temporal belief-state chains and uses a Time-Aware Belief Solver to identify belief-communication periods for higher-order reasoning; and ENIGMAToM performs iterative perspective masking over structured entity-state information. In contrast, RECToM constructs fact-based symbolic state-event sequences and recursively derives each character perspective from the already constructed perspective of the preceding character in the query chain. Thus, an inner character’s belief is evaluated relative to the outer character’s constructed perspective rather than directly from the original omniscient narrative. Within each perspective, partial-observation completion preserves belief persistence, with observable events triggering belief updates and unobservable states inheriting the preceding belief state.

RECToM is an inference-time neuro-symbolic framework that combines LLM-based semantic interpretation with explicit symbolic belief reasoning. The LLM handles fact-based event ab-

Model	Method	Big-ToM		
		True-belief	False-belief	Overall
GPT-5.4	CoT	<u>98.50</u>	99.50	<u>99.00</u>
	SIMToM	99.50	<u>98.50</u>	<u>99.00</u>
	TIMEToM	96.00	99.50	97.75
	RECToM	99.50	99.50	99.50
Gemini-3	CoT	<u>98.50</u>	99.00	<u>98.75</u>
	SIMToM	98.00	<u>98.00</u>	98.00
	TIMEToM	92.00	93.00	92.50
	RECToM	99.00	99.00	99.00
Qwen3.5	CoT	<u>93.00</u>	99.50	<u>96.25</u>
	SIMToM	92.50	97.00	94.75
	TIMEToM	84.00	98.00	91.00
	RECToM	98.00	<u>99.00</u>	98.50
Gemma-4	CoT	88.00	<u>98.50</u>	<u>93.25</u>
	SIMToM	78.50	97.50	88.00
	TIMEToM	69.00	91.00	80.00
	RECToM	98.50	99.50	99.00

Table 6: Accuracy (%) on Big-ToM forward-belief questions. Big-ToM contains 400 forward-belief instances, with 200 true-belief and 200 false-belief instances. The best result for each backbone and metric is in bold, and the second-best result is underlined.

straction, observability judgments, and semantic reasoning for belief updates, while explicit state-transition and belief-update mechanisms maintain symbolic states and character perspectives. This design separates natural-language interpretation from explicit perspective construction and nested-belief reasoning. Unlike verifier-based approaches such as DEL-TOM, which score and select candidate belief-update traces, RECToM performs answer inference from the recursively constructed perspective itself. Its perspective construction further supports formal analysis of the induced belief modality under KD45.

B Additional Experimental Results

Table 6 reports separate true-belief and false-belief accuracies on Big-ToM forward-belief questions, complementing the overall accuracy in Table 2. RECToM performs better across both belief settings, with clear gains on true-belief questions. For Gemma-4, RECToM outperforms CoT, the second-best method, by 10.50% on true-belief accuracy. These results show that recursive perspective construction supports reliable reasoning across both belief conditions.

Table 7 reports the performance of RECToM on Hi-ToM under low, medium, and high reasoning-effort settings for GPT-5.4 and Gemini-3. With GPT-5.4, RECToM achieves 100% accuracy across all question orders and reasoning-effort set-

tings. With Gemini-3, accuracy remains at 100% on 0th-order and 1st-order questions across all settings, while higher-order accuracy decreases only slightly as reasoning effort is reduced. RECToM decomposes nested-belief reasoning by recursively constructing each character’s perspective from the source perspective. This design avoids simultaneous reasoning over multiple nested perspectives, thereby reducing the method’s sensitivity to lower reasoning-effort settings.

We further compare RECToM with two additional training-free ToM methods, SYMBOLICToM (Sclar et al., 2023) and AUTOToM (Zhang et al., 2025). SYMBOLICToM constructs explicit symbolic graphical representations for each character, tracking both the character’s own beliefs and their estimates of other characters’ beliefs, and extends these representations to higher-order beliefs by iteratively processing the story one sentence at a time. AUTOToM instead formulates ToM reasoning as automated Bayesian inverse planning: it first proposes a Bayesian ToM model for the given problem, performs inference over the target mental variables, and iteratively refines the model by adding mental variables or additional timesteps when the inferred posterior remains uncertain. As shown in Table 8, both baselines underperform CoT overall and degrade with increasing belief order, while RECToM remains near-perfect across all orders. Their performance degradation mainly arises from difficulties in handling communication events in Hi-ToM, leading to incorrect belief updates across character perspectives. We then compare RECToM with AUTOToM on Big-ToM, omitting SYMBOLICToM due to its incompatible event representations. As shown in Table 9, AUTOToM performs slightly below CoT with both GPT-5.4 and Qwen3.5, whereas RECToM achieves the highest overall accuracy.

C KD45 Proof for RECToM

We provide the full derivation of the KD45 axioms for the belief modality induced by RECToM’s recursive perspective construction. The proof has three steps. First, we define the language of belief queries. Second, we define how such queries are evaluated by RECToM. Third, we show that this evaluation satisfies the K , D , 4, and 5 axioms of KD45.

Let $\mathcal{W}$ be the set of well-formed perspectives. In this proof, we simplify the notation and write

Model	Method	Effort	0th-Order	1st-Order	2nd-Order	3rd-Order	4th-Order	Overall
GPT-5.4	RECToM	Low	100.00	100.00	100.00	100.00	100.00	100.00
		Medium (default)	100.00	100.00	100.00	100.00	100.00	100.00
		High	100.00	100.00	100.00	100.00	100.00	100.00
Gemini-3	RECToM	Low	100.00	100.00	96.25	95.00	95.00	97.25
		Medium	100.00	100.00	97.50	97.50	96.25	98.25
		High (default)	100.00	100.00	97.50	97.50	97.50	98.50

Table 7: Accuracy (%) of RECToM on Hi-ToM under different reasoning-effort settings. The default reasoning effort is medium for GPT-5.4 and high for Gemini-3.

Model	Method	Hi-ToM					
		0th-Order	1st-Order	2nd-Order	3rd-Order	4th-Order	Overall
GPT-5.4	CoT	100.00	<u>88.75</u>	60.00	<u>68.75</u>	<u>71.25</u>	<u>77.75</u>
	SYMBOLICToM	93.75	82.50	<u>67.50</u>	56.25	48.75	69.75
	AUTOToM	98.75	87.50	66.25	61.25	55.00	73.75
	RECToM (ours)	100.00	100.00	100.00	100.00	100.00	100.00
Qwen3.5	CoT	100.00	<u>95.00</u>	<u>88.75</u>	83.75	<u>85.00</u>	<u>90.50</u>
	SYMBOLICToM	98.75	88.75	75.00	66.25	61.25	78.00
	AUTOToM	98.75	88.75	86.25	<u>85.00</u>	81.25	88.00
	RECToM (ours)	100.00	100.00	100.00	100.00	100.00	100.00

Table 8: Accuracy (%) on Hi-ToM with additional training-free baselines, including SYMBOLICToM and AUTOToM. Each order contains 80 instances, with 400 tasks in total. The best result for each backbone and metric is in bold, and the second-best result is underlined.

Model	Method	Big-ToM Overall
GPT-5.4	CoT	<u>99.00</u>
	AUTOToM	98.50
	RECToM (ours)	99.50
Qwen3.5	CoT	<u>96.25</u>
	AUTOToM	95.00
	RECToM (ours)	98.50

Table 9: Accuracy (%) on Big-ToM with AUTOToM included as an additional training-free baseline. The best result for each backbone is in bold, and the second-best result is underlined.

P for an arbitrary perspective, e.g., $P \in \mathcal{W}$. We assume that $\mathcal{W}$ is closed under RECToM’s perspective construction: for each character a_i , the constructed perspective $P^{(a_i|P^{(\cdot)})}$ also belongs to $\mathcal{W}$, where $P^{(\cdot)} \in \mathcal{W}$ is the source perspective. We define $F_{a_i}(P^{(\cdot)}) = P^{(a_i|P^{(\cdot)})}$, where F_{a_i} maps any source perspective $P^{(\cdot)}$ to the constructed perspective $P^{(a_i|P^{(\cdot)})}$ of character a_i . For clarity, we omit the source-perspective indicator $(\cdot)$ in the following proof.

Given an atomic symbolic fact x and a set of characters $\mathcal{A}$, we define belief-query formulas using the grammar:

$$\varphi, \psi ::= x \mid \neg\varphi \mid (\varphi \rightarrow \psi) \mid B_{a_i}\varphi, \quad a_i \in \mathcal{A}.$$

Here, $B_{a_i}\varphi$ means that character a_i believes φ . An example of a nested belief is $B_{a_i}B_{a_j}\varphi$, which means that character a_i believes that character a_j believes φ . This grammar includes only negation, implication, and belief as primitive operators. Other Boolean connectives can be introduced as abbreviations in the usual classical way; for example, $\varphi \vee \psi$ abbreviates $\neg\varphi \rightarrow \psi$, and $\varphi \wedge \psi$ abbreviates $\neg(\varphi \rightarrow \neg\psi)$. Thus, the grammar remains compact while still allowing standard Boolean combinations of belief queries.

In the following proof, we fix an arbitrary character a_i and show that B_{a_i} satisfies the KD45 axioms. We write $P \models \varphi$ to mean that formula φ is satisfied, or evaluates to true, under perspective P . For an atomic symbolic fact x , $P \models x$ iff x holds in the final state of P ; if x is absent from the final state, it is treated as false under the closed-world assumption. The primitive Boolean connectives are interpreted classically: $P \models \neg\varphi$ iff $P \not\models \varphi$, and $P \models \varphi \rightarrow \psi$ iff either $P \not\models \varphi$ or $P \models \psi$. Connectives such as $\wedge$ and $\vee$, when used, inherit their meanings through the abbreviations defined above. The belief operator is interpreted by switching from the current perspective to the constructed perspective of the queried character:

$$P \models B_{a_i}\varphi \quad \text{iff} \quad F_{a_i}(P) \models \varphi. \quad (5)$$

Thus, evaluating $B_{a_i}\varphi$ in source perspective P means: first construct a_i 's perspective, and then evaluate φ inside that perspective. Nested belief queries are handled by repeatedly applying the same rule.

Self-perspective idempotence. The key property is that constructing the same character's perspective twice does not change the result:

$$F_{a_i}(F_{a_i}(P)) = F_{a_i}(P). \quad (6)$$

This follows from the construction rule in Eq. (2). Once $F_{a_i}(P)$ has already been constructed, the event sequence contains only events observable to a_i or $\emptyset$, and every unobservable state has already been completed by inheriting the previous belief state and revising it only with observable evidence. Therefore, reapplying the same construction does not remove any additional information or add any new information. It preserves the same observable event sequence and returns the same completed state sequence.

C.1 KD45 Axiom Satisfaction

We now verify the KD45 axioms under the belief semantics in Eq. (5). The main idea is simple: K and D follow from ordinary classical reasoning inside the constructed perspective $F_{a_i}(P)$, while 4 and 5 follow from self-perspective idempotence.

K: Distribution. The K axiom states $B_{a_i}(\varphi \rightarrow \psi) \rightarrow (B_{a_i}\varphi \rightarrow B_{a_i}\psi)$. Intuitively, this axiom says that belief is closed under implication: if a character's constructed perspective supports an implication and also supports its premise, then it should also support the conclusion. To verify this for RECTOM, suppose $P \models B_{a_i}(\varphi \rightarrow \psi)$ and $P \models B_{a_i}\varphi$. By Eq. (5), both formulas are evaluated inside the same constructed perspective $F_{a_i}(P)$, so $F_{a_i}(P) \models \varphi \rightarrow \psi$ and $F_{a_i}(P) \models \varphi$. Since implication is interpreted classically inside this perspective, $F_{a_i}(P) \models \psi$. Applying Eq. (5) again gives $P \models B_{a_i}\psi$. Therefore, K holds.

D: Consistency. The D axiom states $B_{a_i}\varphi \rightarrow \neg B_{a_i}\neg\varphi$. Intuitively, this axiom says that a character's constructed perspective should not support both a statement and its negation. To verify this for RECTOM, suppose $P \models B_{a_i}\varphi$. By Eq. (5), this means $F_{a_i}(P) \models \varphi$. Since $F_{a_i}(P)$ is a well-formed perspective and formulas are evaluated with classical negation, $F_{a_i}(P) \not\models \neg\varphi$. Hence

$P \not\models B_{a_i}\neg\varphi$, so $P \models \neg B_{a_i}\neg\varphi$. Therefore, D holds.

4: Positive introspection. The 4 axiom states $B_{a_i}\varphi \rightarrow B_{a_i}B_{a_i}\varphi$. Intuitively, this axiom says that if a character's constructed perspective supports φ , then recursively asking what the same character believes should not change that perspective. To verify this for RECTOM, suppose $P \models B_{a_i}\varphi$. Then $F_{a_i}(P) \models \varphi$. By self-perspective idempotence (Eq. (6)), applying the same construction again gives $F_{a_i}(F_{a_i}(P)) = F_{a_i}(P)$. Thus, when $B_{a_i}\varphi$ is evaluated inside $F_{a_i}(P)$, it returns to the same constructed perspective where φ already holds. Hence $F_{a_i}(P) \models B_{a_i}\varphi$, and by Eq. (5), $P \models B_{a_i}B_{a_i}\varphi$. Therefore, 4 holds.

5: Negative introspection. The 5 axiom states $\neg B_{a_i}\varphi \rightarrow B_{a_i}\neg B_{a_i}\varphi$. Intuitively, this axiom says that if φ is not supported by a character's constructed perspective, then recursively querying that same character's belief should not make φ become supported. To verify this for RECTOM, suppose $P \models \neg B_{a_i}\varphi$. Then $F_{a_i}(P) \not\models \varphi$. By self-perspective idempotence (Eq. (6)), applying the same construction again still yields the same perspective, so $F_{a_i}(F_{a_i}(P)) \not\models \varphi$. Therefore, inside $F_{a_i}(P)$, the character does not believe φ , i.e., $F_{a_i}(P) \models \neg B_{a_i}\varphi$. Applying Eq. (5) again gives $P \models B_{a_i}\neg B_{a_i}\varphi$. Therefore, 5 holds.

Since the belief operator B_{a_i} satisfies K , D , 4, and 5, RECTOM's recursive perspective construction induces a KD45 belief modality.

D Parameter Settings

RECTOM operates on all LLMs through API access. For proprietary models, GPT-5.4 and Gemini-3 Flash, we use the default parameter settings. For open-source models, we set temperature to 1.0 and top- p to 0.95 for both Qwen3.5-27B and Gemma-4-26B-A4B, with top- k set to 20 for Qwen3.5-27B and 64 for Gemma-4-26B-A4B to reduce repetitive outputs and filter rare tokens while preserving generation diversity. All experiments are conducted on a virtual machine with four NVIDIA A100 80GB GPUs.

E Benchmark Details

E.1 Big-ToM

Big-ToM (Gandhi et al., 2023), a GPT-4-generated benchmark, evaluates belief reasoning in natural-language stories based on the Sally-Anne false-

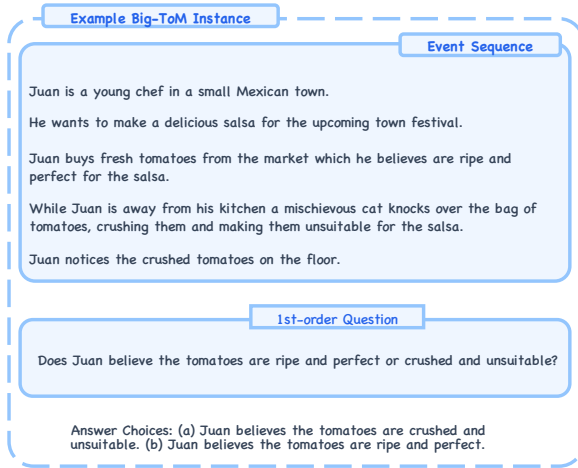


Figure 3: An example Big-ToM instance with the first-order belief question over a natural-language event sequence.

belief paradigm. We use the forward-belief subset, which asks models to infer what a character believes after the character either observes or misses a belief-relevant event. This subset focuses on first-order true-belief and false-belief questions, evaluating whether a character’s belief is consistent or inconsistent with reality. The task is formulated as binary multiple choice, with a random baseline accuracy of 50%. Figure 3 shows an example true-belief instance from the Big-ToM forward-belief subset.

E.2 FanToM

FanToM (Kim et al., 2023) evaluates ToM reasoning in multi-party dialogue settings. Its dialogues introduce asymmetric information by allowing characters to enter and leave while the conversation continues, resulting in different characters observing different parts of the conversation. We focus on FanToM belief questions, including first-order and second-order questions, which align with the scope of this work. These questions are formulated as binary multiple-choice tasks, with a random baseline accuracy of 50%. Compared with narrative benchmarks, FanToM dialogues are longer and involve more characters and subtopics, requiring models to integrate extended dialogue context and maintain character-specific beliefs. Figure 5 illustrates the FanToM dialogue structure with first-order and second-order belief questions.

F Error Analysis

The explicit state-event representation and recursive perspective construction in RECTOM provide

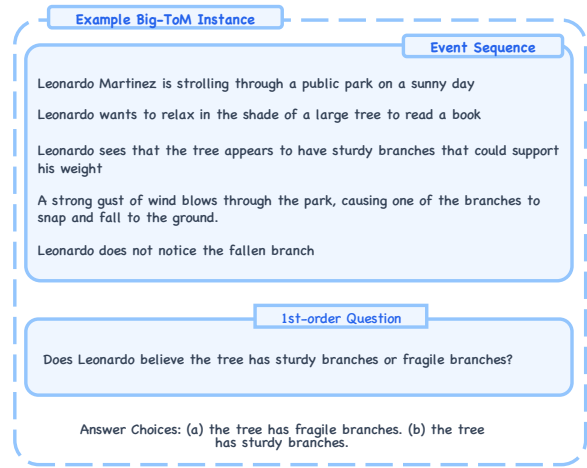


Figure 4: An example Big-ToM instance where RECTOM makes an incorrect observability judgment.

a structured basis for nested-belief reasoning. Nevertheless, semantic misinterpretations by the LLM can still lead to reasoning failures. For example, Figure 4 illustrates a Big-ToM instance in which RECTOM produces an incorrect answer due to an erroneous observability judgment by the LLM. Leonardo first observes that a tree appears to have sturdy branches. Later, a branch snaps and falls, while the narrative explicitly states that Leonardo does not notice this event. Leonardo’s perspective should therefore preserve the earlier belief that the tree has sturdy branches. However, the LLM treats the branch-falling event as observable to Leonardo, resulting in an erroneous belief update and final answer.

Figure 6 illustrates a FanToM example in which RECTOM constructs the state-event sequence and character perspective correctly, but the final inference step fails to map the natural-language question to the relevant symbolic fact. Victor leaves the conversation, Nyla and Jude continue talking during his absence, and Victor later returns. The question asks who Victor believes joined the conversation later, and the correct answer is Victor. The constructed state-event sequence accurately records Victor’s status using the facts `depart(Victor)`, `return(Victor)`, and `present(Victor)`, and Victor’s observation mask correctly excludes the dialogue during his absence and retains the dialogue when he is present. However, the LLM fails to associate the phrase “joined the conversation later” with `return(Victor)` in the constructed perspective, resulting in an incorrect answer.

These examples show that LLM interpretation errors can occur at different stages of RECTOM: incorrect observability judgments affect perspective construction, whereas semantic grounding errors can lead to incorrect final inference even when the constructed perspective is correct.

G Prompt Templates for RECTOM

Tables 10–14 present the main prompt templates used by RECTOM on Hi-ToM. They cover fact-based event abstraction, state and event observability identification, transient-event belief revision, and final answer inference. For readability, we show compact templates; the complete prompts are provided in the released code.



Figure 5: An example FanToM instance with first-order and second-order belief questions over a dialogue sequence.



Figure 6: An example FanToM instance where RECToM incorrectly grounds a natural-language question to the relevant symbolic facts.

Prompt for Fact-based Event Abstraction in RECTOM

You are extracting step-wise symbolic deltas from a ToM story.

Return valid JSON only.

Example output schema:

```
{
  "characters": ["Avery", "Charlotte"],
  "steps": [
    {
      "step_index": 1,
      "step_text": "Avery entered the living_room.",
      "event_type": "persistent",
      "added_facts": ["in_room(Avery, living_room)"],
      "removed_facts": []
    }
  ]
}
```

Important definition.

- Do not output a full state for each step. Output only the delta for each step.
- event_type is either persistent or transient. Persistent events introduce or revise state facts; transient events record communication, claims, or questions.
- added_facts are facts that become true because of the current step.
- removed_facts are facts that stop being true because of the current step.

Extraction guidelines.

- Include every story step exactly once and include all human characters appearing in the story.
- Extract concise symbolic facts for belief-relevant changes, such as character locations, object locations, communication events, and stated claims.
- Remove a fact only when the current step makes it false.
- Distinguish spoken claims from actual world facts when communication events appear.
- Represent private communication as `private_tell(speaker, listener, proposition)` and public communication as `public_claim(speaker, proposition)`.
- If the spoken content is an object-location statement, represent the proposition as `in(object, container)`.

Story steps: {story_steps}

Assumptions: {assumptions}

Table 10: Prompt template used by RECTOM for fact-based event abstraction. The LLM extracts symbolic deltas and classifies each event as persistent or transient. The complete state sequence is then computed externally using the deterministic update rule.

Prompt for State Observability Identification in RECTOM

You are deciding whether a target character is present in each aligned source state.

Return valid JSON only.

Example output schema:

```
{
  "character": "Alice",
  "observation_basis": [
    "at(Alice,kitchen)",
    "at(Alice,kitchen)",
    "not_observable"
  ]
}
```

Task.

- Return exactly one string for each aligned source state.
- The n th string corresponds to the n th aligned source state.
- Each output string should be either a fact indicating the target character's location or not_observable.

Decision rules.

- If the aligned source state contains a fact indicating that the target character is present in a location, return that fact.
- Presence may be expressed by facts such as `at(character,location)` or `in_room(character,location)`.
- If no fact indicates the target character's presence, return not_observable.
- Do not return bare location names.

Target character: {target_character}

Aligned source states: {aligned_source_states}

Table 11: Prompt template used by RECTOM for state observability identification. Given aligned source states, the LLM derives the observation basis for the target character by identifying explicit presence facts or returning not_observable.

Prompt for Event Observability Identification in RECTOM

You are deciding whether a target character can observe each aligned event fact.
Return valid JSON only.

Example output schema:

```
{
  "character": "Alice",
  "observation_basis": [
    "-in_room(Isabella, living_room)",
    "private_tell(Alice, Bob, in(ball, bedroom))",
    "public_claim(Bob, in(ball, kitchen))",
    "not_observable"
  ]
}
```

Task.

- Return exactly one string for each aligned event fact.
- The n th string corresponds to the n th aligned event fact.
- Each output string should be either the exact observable event fact or not_observable.

Decision rules.

- If the aligned event is empty, return not_observable.
- For private communication, such as private_tell(speaker, listener, proposition), only the speaker and listener can observe the event.
- For public communication, such as public_claim(speaker, proposition), all characters can observe the event.
- For room-entry or room-exit facts, such as +in_room(character, room) or -in_room(character, room), the event is observable to all characters.
- Do not rewrite or normalize the event fact; return it exactly as it appears in the aligned event facts.

Target character: {target_character}

Aligned event facts: {aligned_event_facts}

Table 12: Prompt template used by RECTOM for event observability identification. Given aligned event facts, the LLM derives the observation basis for the target character by identifying observable communication events and room-transition events, returning either the exact observable facts or not_observable.

Prompt for Transient-Event Belief Revision in RECTOM

You are applying an observable transient event to a symbolic belief state.
Return valid JSON only.

Example output schema:

```
{
  "state_before": [
    "in(lettuce,green_drawer)",
    "in_room(Avery,living_room)"
  ],
  "action": [
    "public_claim(Isabella,in(lettuce,green_bathtub))"
  ],
  "action_added_facts": [
    "in(lettuce,green_bathtub)"
  ],
  "action_removed_facts": [
    "in(lettuce,green_drawer)"
  ],
  "state_after": [
    "in(lettuce,green_bathtub)",
    "in_room(Avery,living_room)"
  ]
}
```

Task.

- Start from the given state.
- Apply the observable transient event to this state.
- Output the facts added or removed by the event.
- Output the updated state after applying the event.

Revision rules.

- The transient event may represent communication, claims, questions, or other belief-relevant actions.
- If the event contains an object-location fact, use the fact to revise the current state.
- Preserve state facts that are not affected by the event.

State: {state}

Observable transient event: {observable_transient_event}

Table 13: Prompt template used by RECTOM for transient-event belief revision. Given the current belief state and an observable transient event, the LLM outputs the event-induced added and removed facts and the resulting updated state.

Prompt for Answer Inference in RECTOM

You are answering a multiple-choice Theory-of-Mind question.

Return valid JSON only.

Example output schema:

```
{
  "reasoning_summary": "short summary",
  "predicted_answer": "C"
}
```

Task.

- Use the provided state facts to answer the question.
- Select exactly one option from the candidate choices.
- The selected answer should correspond to the facts in the given state.

Output constraints.

- predicted_answer must contain only the option letter, such as A, B, or C.
- Do not output the choice text in predicted_answer.
- Do not include additional characters or punctuation in predicted_answer.

State: {state}

Question: {question}

Choices: {choices}

Table 14: Prompt template used by RECTOM for answer selection. Given the final state from the constructed perspective, the LLM selects one candidate option and returns the predicted answer as an option letter.