DEL-ToM: Inference-Time Scaling for Theory-of-Mind Reasoning via Dynamic Epistemic Logic

Yuheng Wu[1] **Jianwen Xie**[2] **Denghui Zhang**[3,†] **Zhaozhuo Xu**[3,†]
[1]Stanford University [2]Lambda, Inc. [3]Stevens Institute of Technology
yuhengwu@stanford.edu jianwen.xie@lambda.ai {dzhang42,zxu79}@stevens.edu

Abstract

Theory-of-Mind (ToM) tasks pose a unique challenge for large language models (LLMs), which often lack the capability for dynamic logical reasoning. In this work, we propose DEL-ToM, a framework that improves verifiable ToM reasoning through inference-time scaling rather than architectural changes. Our approach decomposes ToM tasks into a sequence of belief updates grounded in Dynamic Epistemic Logic (DEL), enabling structured and verifiable dynamic logical reasoning. We use data generated automatically via a DEL simulator to train a verifier, which we call the Process Belief Model (PBM), to score each belief update step. During inference, the PBM evaluates candidate belief traces from the LLM and selects the highest-scoring one. This allows LLMs to allocate extra inference-time compute to yield more transparent reasoning. Experiments across model scales and benchmarks show that DEL-ToM consistently improves performance, demonstrating that verifiable belief supervision significantly enhances LLMs' ToM capabilities without retraining. Code is available at https://github.com/joel-wu/DEL-ToM.

1 Introduction

> "To know what John knows is to know the worlds that are compatible with his belief, and to know which ones are not." — *Jaakko Hintikka* (Hintikka and B. P. Hintikka, 1989)

The ability to attribute beliefs, desires, and intentions to others, known as Theory-of-Mind (ToM) (Premack and Woodruff, 1978; C. Dennett, 1978; Apperly and Butterfill, 2009), is a fundamental component of social intelligence (Baron-Cohen, 1991). ToM enables agents to reason about what others think, want, or know, and to anticipate their subsequent behavior (Rabinowitz et al., 2018).

Recent studies suggest that large language models (LLMs) (Brown et al., 2020) exhibit ToM abilities (Strachan et al., 2024; Lin et al., 2024; Street et al., 2024; Amirizaniani et al., 2024; Sclar et al., 2025; Wu et al., 2025). However, ToM performance follows a scaling law (Kosinski, 2024), with smaller models showing limited ability on ToM tasks. This limitation poses a challenge for low-resource deployments, where edge agents are expected to robustly infer users' intentions and act in alignment with human expectations. At the same time, current evaluations compare only the final output to the ground-truth label (Chen et al., 2024), leaving it unclear whether correct answers result from genuine reasoning or from lucky guessing (Ullman, 2023). Consequently, existing ToM reasoning remains unverifiable and not applicable in practice. This paper addresses the question: *How can we enable LLMs to perform verifiable ToM reasoning, especially in low-resource settings?*

Following process reliabilism (Goldman, 1979), verifiable ToM reasoning requires a sequence of intermediate belief states that reliably support the final conclusion. We formalize this reasoning process using Dynamic Epistemic Logic (DEL) (Baltag et al., 1998; Van Benthem, 2001; Plaza, 2007; Van Ditmarsch et al., 2007; Aucher and Schwarzentruber, 2013), a logic system grounded in the traditions of formal logic and semantics (Frege, 1879; Russell and Whitehead, 1910; Wittgenstein, 1922; Tarski, 1956; Hintikka, 1962; Kripke, 1963). DEL models agents' beliefs with epistemic models, actions with event models, and belief change via product updates, allowing us to view ToM reasoning as dynamic logical reasoning.

Within this framework, transparent belief traces are generated and evaluated by a Process Belief Model (PBM). By scoring multiple candidates, the PBM enables us to select the most reliable trace. This constitutes inference-time scaling: spending more computation during inference to obtain more

[†]Corresponding authors.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 11395–11409
November 4-9, 2025

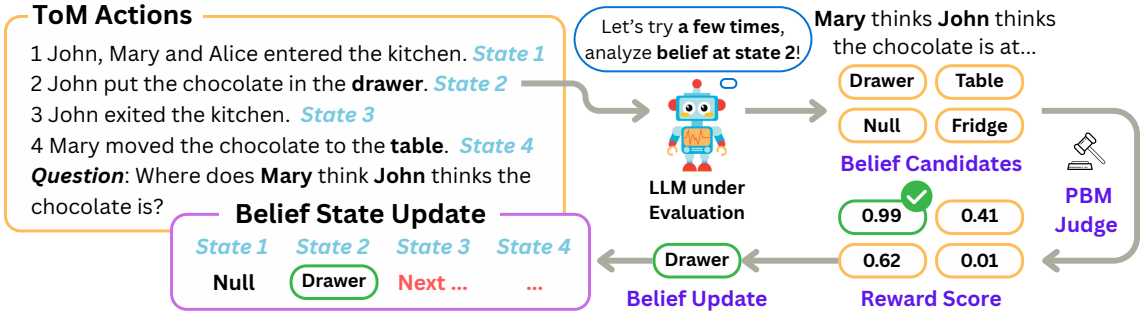


Figure 1: Overview of the DEL-ToM framework. Each belief state is inferred from the previous state and the current action. The LLM generates multiple candidate belief traces in parallel, and the PBM assigns reward scores to filter a top-scoring subset, which is then used to continue reasoning toward the next belief state.

reliable reasoning traces, which in turn allows smaller models to achieve stronger ToM performance while remaining efficient for deployment. We experiment with different trace selection and search strategies for ToM reasoning.

To train the PBM, we first generate ToM-related questions and use DEL to produce belief process labels. We then use GPT-4o-mini (Hurst et al., 2024) to answer these questions. Finally, DEL-generated gold labels are used to automatically score GPT-generated traces, producing positive and negative examples for PBM training. Unlike other process-level reward modeling datasets, which rely on human annotation or LLM assistance (Wang et al., 2024), our labels are derived from a formal DEL system, which guarantees correctness.

In conclusion, we approach ToM reasoning through the lens of formal logic. Using a PBM trained via DEL, we make each intermediate belief update explicit and employ search-based methods to select the most reliable trace. This enables inference-time scaling and yields dynamic logical reasoning grounded not only in model outputs, but in verifiable, structured belief updates. Our contributions are threefold:

- We propose a new perspective on ToM reasoning by framing it as a problem of process reliability. By modeling reasoning as a multi-step dynamic belief-update process, we can apply inference-time scaling to select more reliable belief traces.
- We formalize ToM reasoning in the framework of DEL and construct a PBM dataset with noise-free supervision derived from DEL. This enables training PBMs for stepwise reasoning evaluation.
- We evaluate our approach across different model scales and search strategies. Our method consistently improves LLM performance on standard ToM benchmarks.

2 Background and Motivation

ToM in LLMs. Researchers have designed various tasks to evaluate the ToM capabilities of LLMs. Among these, false belief tasks are the most widely used, typically in two forms:

- Unexpected Contents: A protagonist is shown an object with misleading external cues (e.g., an opaque crayon box that actually contains candles). The LLM under evaluation must identify that the actual content is candles while recognizing that the protagonist holds the mistaken belief that the box contains crayons.
- Unexpected Transfer: An object is moved without the protagonist's knowledge, and the LLM must predict where the protagonist will search for it – based on the protagonist's outdated belief.

Among the two, the unexpected transfer task is more commonly used. Figure 1 illustrates a typical instance of this task setup.

Illustrative Example. As shown in Figure 1, the story consists of four sentences, each describing an action that updates the characters' belief state. The goal of ToM reasoning is to infer the sequence of belief states, culminating in the final belief state.

In this example, after Action 1, John, Mary, and Alice are all present in the kitchen, but the chocolate has not been introduced, so no beliefs are yet established. After Action 2, John places the chocolate in the drawer, and everyone present observes this action. Hence, Mary believes that John believes the chocolate is in the drawer. Following Action 3, John exits the kitchen. Then, in Action 4, Mary moves the chocolate to the table, an action that John is unaware of. As a result, Mary thinks John still believes the chocolate is in the drawer.

From this example, we see that ToM reasoning can be understood as an action applied to a prior

belief state, causing characters to gain or lose information and thereby forming a new state. This process naturally aligns with DEL, which represents each belief state with an epistemic model, each action with an event model, and updates beliefs via the product update by combining the state with an action. Together, these elements yield a formal dynamic-logic system that derives the full belief-state trace over time.

Our Objective: Inference-Time Scaling for Verifiable ToM in LLMs. Our goal is to enable LLMs to perform ToM reasoning in an efficient and verifiable manner. To this end, we adopt an inference-time scaling strategy that allocates extra compute during inference to improve the reliability of reasoning. This approach not only enhances the reasoning capability of large models but also allows smaller models to remain deployment-efficient while achieving performance competitive with closed-source LLMs.

3 Inference-Time Scaling for ToM

In this section, we first formulate ToM reasoning as a DEL process. We then describe how the PBM is constructed and trained to evaluate belief traces, and present inference-time scaling pipelines that use the PBM to guide ranking and selection of reasoning traces.

3.1 Formulating ToM Reasoning within DEL

We formulate ToM reasoning within the framework of DEL, which is based on Kripke's possible-world semantics (Kripke, 1963). Let $\mathcal{P}$ be a countable set of atomic propositions, representing basic facts about the world, and let $\mathcal{A}$ be a finite, non-empty set of agents. The epistemic language $\mathcal{L}(\mathcal{P}, \mathcal{A})$ is defined by the Backus-Naur form (Knuth, 1964):

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid B_i\varphi,$$

where $p \in \mathcal{P}$, $i \in \mathcal{A}$, and φ ranges over well-formed formulas. The formula $B_i\varphi$ is read as "agent i believes φ." For example, "John believes the chocolate is in the drawer" can be written as $B_{\text{John}}(chocolate\_in\_drawer)$. Based on this language, we define epistemic models, event models, and the product update.

Definition 1 (Epistemic Model)**.** An *epistemic model* over agent set $\mathcal{A}$ and proposition set $\mathcal{P}$ is a triple $\mathcal{M} = (W, R, V)$, where:

- W is a set of possible worlds, where each world is a complete valuation of $\mathcal{P}$;
- $R : \mathcal{A} \to 2^{W \times W}$ assigns each agent $a \in \mathcal{A}$ an accessibility relation R_a;
- $V : \mathcal{P} \to 2^W$ maps each atomic proposition $p \in \mathcal{P}$ to the set of worlds where p is true.

A *state* is a pointed epistemic model $(\mathcal{M}, w)$ where $w \in W$ is the designated actual world.

We write wR_av to denote that world v is accessible from world w according to agent a: in world w, agent a considers v possible.

On the basis of an epistemic model $\mathcal{M} = (W, R, V)$ and a designated world $w \in W$, the satisfaction relation $\models$ for $\mathcal{L}(\mathcal{P}, \mathcal{A})$ is defined as follows:

- $\mathcal{M}, w \models p$ iff $w \in V(p)$;
- $\mathcal{M}, w \models B_a\varphi$ iff for all $v \in W$ such that wR_av, we have $\mathcal{M}, v \models \varphi$.

Definition 2 (Event Model)**.** An *event model* is a tuple $\varepsilon = (E, Q, \mathsf{pre}, \mathsf{post})$, where:

- E is a finite, non-empty set of events;
- $Q : \mathcal{A} \to 2^{E \times E}$ assigns to each agent $a \in \mathcal{A}$ an indistinguishability relation Q_a over events;
- $\mathsf{pre} : E \to \mathcal{L}(\mathcal{P}, \mathcal{A})$ assigns to each $e \in E$ a precondition specifying when e is executable;
- $\mathsf{post} : E \to \mathcal{L}(\mathcal{P}, \mathcal{A})$ assigns to each $e \in E$ a postcondition describing how the world changes.

We refer to a pointed event model (ε, e) as an *action*, where $e \in E$ is the actual event that occurs.

Definition 3 (Product Update)**.** Let $(\mathcal{M}, w)$ be a state with $\mathcal{M} = (W, R, V)$, and let (ε, e) be an action with $\varepsilon = (E, Q, \mathsf{pre}, \mathsf{post})$. Suppose that the precondition is satisfied, i.e., $\mathcal{M}, w \models \mathsf{pre}(e)$. Then the *product update* results in a new state $(\mathcal{M}', (w, e))$, where the updated epistemic model $\mathcal{M}' = (W', R', V')$ is defined as follows:

- $W' = \{(w', e') \in W \times E \mid \mathcal{M}, w' \models \mathsf{pre}(e')\}$;
- For each $a \in \mathcal{A}$, $R'_a = \{((w', e'), (v', f')) \in W' \times W' \mid w'R_av' \wedge e'Q_af'\}$;
- $(w', e') \in V'(p)$ iff $\mathsf{post}(e') \models p$ or $(\mathcal{M}, w' \models p \wedge \mathsf{post}(e') \not\models \neg p)$, for each $p \in \mathcal{P}$.

Applying DEL to ToM Reasoning. We illustrate States 4–6 in Figure 2. In State 4, both Mary and Alice are present and observe that the chocolate is on the table, so $\mathcal{M}, w_4 \models table$ and $R_M = R_A = \{(w_4, w_4)\}$. After Action 5, Mary exits the kitchen, $\mathsf{pre}(e_5) = \top$, $\mathsf{post}(e_5) = table$, so facts remain unchanged but Mary will not observe subsequent actions. In Action 6, Alice moves the chocolate to the cupboard with $\mathsf{pre}(e_6) = \top$ and $\mathsf{post}(e_6) = cupboard \wedge \neg table$. After the product updates, the actual state $(\mathcal{M}, w_6)$ satisfies

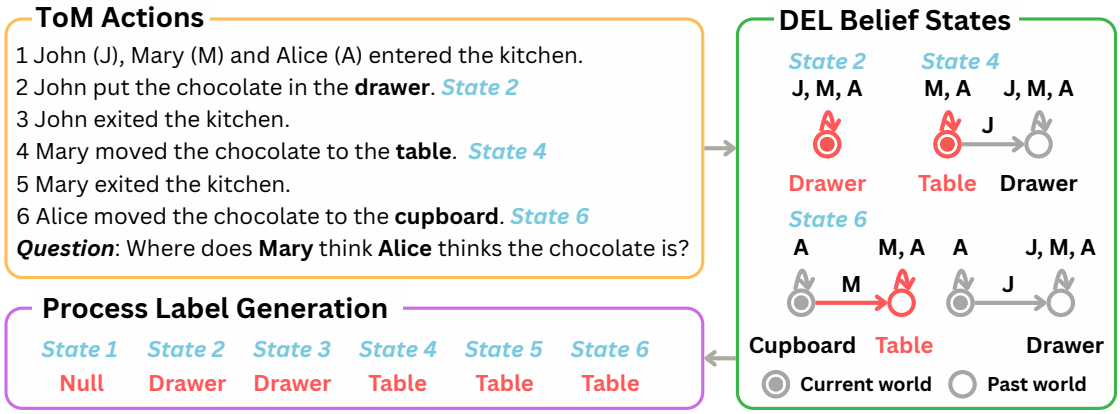


Figure 2: Training data synthesis for PBM. The right part illustrates the accessibility relations generated by the DEL simulator.

$\mathcal{M}, w_6 \models \mathit{cupboard}$; Alice's accessibility relation R'_A points to cupboard-worlds, while Mary's relation R'_M still reaches the table-world. Hence

$$\begin{aligned}
&\mathcal{M}, w_6 \models B_{\text{Mary}}\, B_{\text{Alice}}\, \varphi \\
&\iff \forall v\,(w_6 R'_M v \Rightarrow \mathcal{M}, v \models B_A \varphi) \\
&\iff \mathcal{M}, w_4 \models B_{\text{Alice}}\, \varphi \\
&\iff \forall u\,(w_4 R'_A u \Rightarrow \mathcal{M}, u \models \varphi) \\
&\iff \mathcal{M}, w_4 \models \varphi,
\end{aligned}$$

where φ denotes "the chocolate is on the table." Thus, Mary believes that Alice believes it is on the table. This illustrates that the core of DEL reasoning lies in constructing the **accessibility relations** R at each state, finding the worlds that are compatible with an agent's belief (Hintikka and B. P. Hintikka, 1989).

3.2 Building the PBM with DEL

Generating Process-Level Labels via DEL. We integrate a DEL simulator into the Hi-ToM generators[1] (Wu et al., 2023) and synthesize 20,000 ToM stories with process labels. For each story, we build process-level traces across different orders of belief: at each action, we update the accessibility relations R based on the action's semantics and whether the observation is public or private, then update R accordingly and record the belief state in the trace set. All process-level label generation code is integrated into the Hi-ToM generators and included in our released codebase.

Dataset Assembly. For each synthesized story, we prompt GPT-4o-mini (Hurst et al., 2024) to produce step-by-step belief updates in a DEL format (the prompt is provided in Appendix A). We pair each LLM trace with the DEL per-step labels to form training instances, yielding both positive and negative supervision for process-level reward modeling.

Training the PBM. PBM is a scoring function $f : \mathcal{Q} \times \mathcal{S} \to \mathbb{R}^+$ that assigns a score to each step s_i in a GPT-4o-mini-generated belief trace s, given a ToM problem q. We treat this as a binary classification task: each step is labeled as either correct or incorrect according to the DEL-generated belief trace. The model is trained using the following binary cross-entropy loss:

$$\begin{aligned}
\mathcal{L}_{\text{PBM}} = &-\sum_{i=1}^{K} y_{s_i} \log f(s_i) \\
&-\sum_{i=1}^{K} (1 - y_{s_i}) \log(1 - f(s_i)),
\end{aligned}$$

where K is the number of steps, y_{s_i} is the binary label, and $f(s_i)$ is the predicted score. The training code is adopted from the RLHF-Reward-Modeling codebase[2].

3.3 Inference-Time Scaling Pipeline

Beam Search. Beam search is a decoding method that maintains multiple partial belief traces during generation (Figure 1): at each action, the LLM observes the trace so far and proposes multiple candidate belief updates for the current state. The PBM scores these candidates, and a high-scoring subset is selected to continue reasoning. This process repeats until all actions are processed. Formally, the procedure is as follows:

[1] https://github.com/ying-hui-he/Hi-ToM_dataset

[2] https://github.com/RLHFlow/RLHF-Reward-Modeling

- Initialize k beams with candidate first-step updates sampled from the model.
- Expand each beam with b next-step candidates, yielding $k \times b$ partial paths.
- Score each path with the PBM, ranking by the score of the most recent step.
- Retain the top k paths and iterate until reaching an end-of-sequence or the maximum depth.

Best-of-N (BoN). Alternatively, instead of updating step by step, the LLM may generate N complete belief traces after reading the entire story. The PBM scores each step in these traces, aggregates the step-wise scores into a process-level reward, and reranks the candidates to identify the most reliable trace as the final output. We experiment with different aggregation rules for computing the trace-level score:

- Last: Use the PBM score of the final step.
- Min: Use the lowest score across all steps.
- Avg: Use the average score across the trace.
- Prod: Multiply the scores of all steps.
- Majority: Select the final answer by simple majority voting across traces, without using PBM.

Based on the aggregated scores, we consider two ranking strategies:

- Vanilla BoN: Select the single trace with the highest PBM score.
- Weighted BoN: Group traces by their final answers, yielding a candidate set $\mathcal{Y} = \{y_1, y_2, \dots\}$. We then sum PBM scores within each group and select the answer $\hat{y}$ with the highest total:

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} \sum_{i=1}^{N} \mathbb{1}(y_i = y) \cdot \mathrm{PBM}(p, t_i),$$

where t_i is the i-th trace, y_i denotes the trace's final answer, and $\mathrm{PBM}(p, t_i)$ is its score.

4 Experiments

4.1 Experimental Setup

Platform. All experiments are conducted on a single NVIDIA GH200 GPU node. We use the vLLM (Kwon et al., 2023) framework for efficient batched inference and large-scale decoding.

PBM Training. We fine-tune a PBM model based on Llama3.1-8B-Instruct (Grattafiori et al., 2024). The model is trained for 1 epoch using our synthesized dataset.

Test Models. We evaluate our methods on both the Qwen3 series (0.6B, 1.7B, 4B, 8B) (Yang et al., 2025) and the Llama3.2 series (1B, 3B) (Grattafiori et al., 2024), as well as closed-source models including gpt-4.1, gpt-4o, gpt-4.1-mini, and gpt-4o-mini. For comparison, we also report results from baselines such as o4-mini, gpt-4.1-nano, Qwen3-235B-A22B (Yang et al., 2025), DeepSeek-V3 (Liu et al., 2024), and OLMo-2-0325-32B (Walsh et al., 2025). All models are evaluated under their default generation settings.

Datasets. We conduct evaluations on two datasets: Hi-ToM (Wu et al., 2023) and the ToM tasks introduced by Kosinski (Kosinski, 2024). For Hi-ToM, we only evaluate one-chapter stories, and for Kosinski's dataset we restrict evaluation to the unexpected transfer task.

Metrics and Prompt Format. We report final answer accuracy as the main evaluation metric. All models are evaluated using a consistent prompting format, as detailed in Appendix A.

4.2 Results on Hi-ToM Dataset

For BoN, we scale N up to 1024 and apply the weighted strategy, selecting the best aggregation rule for each instance. For beam search, we evaluate Qwen3-4B and Qwen3-8B with beam sizes from 4 to 256, excluding smaller models since they cannot generate valid intermediate reasoning steps.

Main Results. As shown in Table 1, incorporating PBM consistently improves ToM reasoning across both BoN and beam search. For example, Llama3.2-3B gains 33.6 points in average accuracy, while Qwen3-4B improves by 9.4 points in the BoN setting. Similarly, with beam search, Qwen3-8B, whose baseline underperforms Qwen3-4B, achieves the highest accuracy of 87.0 once guided by PBM. Moreover, our method generalizes to both open- and closed-source models, as the gpt series also shows clear gains with PBM.

Comparison with SOTA LLMs. As shown in Table 2, smaller open-source models can match or surpass much larger LLMs. For example, Qwen3-4B+PBM achieves higher average accuracy than gpt-4.1, DeepSeek-V3, and OLMo-32B, while Llama3.2-3B+PBM performs on par with gpt-4.1-mini. These findings highlight the effectiveness of PBM in scaling ToM reasoning.

Scaling Test-Time Compute for ToM Reasoning. As shown in Figure 3, increasing the number of sampled belief traces N improves ToM performance only when guided by PBM. Among aggregation strategies, min and prod are the most reliable, while avg and last often degrade under weighted aggregation. In contrast, majority voting fails to

Table 1: Inference-time scaling across belief orders in the Hi-ToM dataset using BoN and Beam Search. "Ori" denotes baseline accuracy, and "+PBM" denotes accuracy with inference-time scaling.

Model	0-th Order		1-th Order		2-th Order		3-th Order		4-th Order		Average	
	Ori	+PBM	Ori	+PBM	Ori	+PBM	Ori	+PBM	Ori	+PBM	Ori	+PBM
BoN ($N = 1024$)												
Qwen3-4B	100.0	100.0	79.8	85.0	79.3	90.0	70.2	82.5	46.0	65.0	75.1	84.5
Qwen3-1.7B	78.0	82.5	59.7	65.0	45.2	55.0	47.0	62.5	47.8	57.5	55.5	64.5
Qwen3-0.6B	69.2	80.0	52.0	72.5	35.0	47.5	31.5	52.5	34.0	47.5	44.3	60.0
Llama3.2-3B	68.2	85.0	52.0	80.0	43.2	82.5	37.0	82.5	36.8	75.0	47.4	81.0
Llama3.2-1B	41.5	46.2	40.0	53.8	28.5	61.5	41.5	84.6	29.2	58.3	36.1	60.9
BoN ($N = 4$)												
gpt-4.1	95.0	97.5	85.0	87.5	85.0	92.5	82.5	95.0	70.0	77.5	83.5	90.0
gpt-4.1-mini	77.5	70.0	90.0	85.0	70.0	75.0	75.0	92.5	77.5	92.5	78.0	83.0
gpt-4o	100.0	100.0	85.0	90.0	82.5	92.5	90.0	97.5	77.5	85.0	87.0	93.0
gpt-4o-mini	90.0	100.0	75.0	87.5	77.5	95.0	77.5	100.0	55.0	85.0	75.0	93.5
Beam Search ($N = 256$)												
Qwen3-8B	96.5	80.0	53.3	80.0	38.8	85.0	55.8	95.0	57.8	95.0	60.4	87.0
Qwen3-4B	100.0	100.0	79.8	85.0	79.3	97.5	70.2	82.5	46.0	60.0	75.1	85.0

Table 2: Comparison with SOTA LLMs on Hi-ToM (BoN, $N = 1024$). "+PBM" denotes accuracy with inference-time scaling.

Model	0-th	1-th	2-th	3-th	4-th	Avg.
o4-mini	97.5	95.0	77.5	87.5	85.0	88.5
gpt-4o	100.0	85.0	82.5	90.0	77.5	87.0
Qwen3-4B+PBM	**100.0**	**85.0**	**90.0**	**82.5**	**65.0**	**84.5**
Qwen3-235B-A22B	100.0	75.0	85.0	85.0	75.0	84.0
gpt-4.1	95.0	85.0	85.0	82.5	70.0	83.5
DeepSeek-V3	100.0	80.0	90.0	70.0	72.5	82.5
Llama3.2-3B+PBM	**85.0**	**80.0**	**82.5**	**82.5**	**75.0**	**81.0**
gpt-4.1-mini	77.5	90.0	70.0	75.0	77.5	78.0
gpt-4o-mini	90.0	75.0	77.5	77.5	55.0	75.0
Qwen3-1.7B+PBM	**82.5**	**65.0**	**55.0**	**62.5**	**57.5**	**64.5**
OLMo-32B	77.5	60.0	60.0	65.0	52.5	63.0
Llama3.2-1B+PBM	**46.2**	**53.8**	**61.5**	**84.6**	**58.3**	**60.9**
Qwen3-0.6B+PBM	**80.0**	**72.5**	**47.5**	**52.5**	**47.5**	**60.0**
gpt-4.1-nano	22.5	32.5	42.5	27.5	30.0	31.0

improve accuracy, since ToM requires evaluating intermediate belief states rather than aggregating final answers. A theoretical analysis of this limitation is provided in Appendix B.

BoN vs. Beam Search. Our experiments show that these two inference-time strategies achieve comparable accuracy. However, beam search rollouts often fail on smaller or weaker models that cannot reliably produce valid intermediate states, making PBM evaluation infeasible. In contrast, BoN generates full belief traces in one shot, where PBM remains effective even when some steps are noisy, and large candidate sets can be produced efficiently using high-throughput backends such as vLLM. We therefore recommend BoN as the preferred inference-time scaling method for ToM reasoning.

4.3 Results on Out-of-Distribution ToM Data

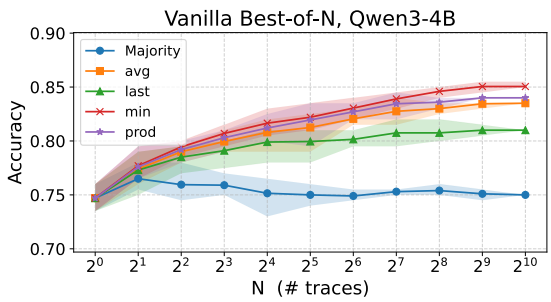
Our PBM is trained on Hi-ToM-style synthetic data, but we ask: *Can it generalize to ToM tasks from a different distribution?* To test this, we evaluate it on the dataset from Kosinski (Kosinski, 2024), which contains hand-written scenarios with false-belief and true-belief controls. We experiment with the Qwen3 series, following the same inference-time scaling and PBM-based selection procedure as before.

Main Results. As shown in Table 3, PBM improves accuracy across all models, confirming its ability to generalize beyond synthetic Hi-ToM scenarios. This shows that PBM functions as a genuine verifier of whether a ToM reasoning process is justified, rather than overfitting to the training distribution, and highlights its robustness on out-of-domain ToM tasks.

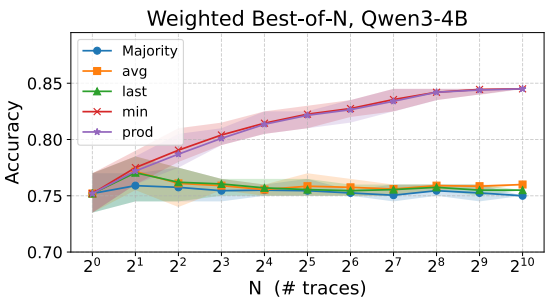
4.4 Benchmarking the PBM

To assess the PBM's standalone reliability, we construct a held-out test set of 2,000 multi-step reasoning examples generated by gpt-4o-mini and spanning all belief orders. Each step is labeled by the DEL simulator as either correct or incorrect, and the PBM is evaluated by its step-level classification accuracy. We benchmark two PBMs trained on different base models: Llama3.1-8B-Instruct and Llama3.2-3B-Instruct.

Evaluating PBM. As shown in Table 4, the larger PBM achieves consistently higher accuracy, and performance decreases as the belief order increases.



(a) Vanilla BoN decoding on Qwen3-4B.



(b) Weighted BoN decoding on Qwen3-4B.

Figure 3: Accuracy of BoN decoding on Qwen3-4B across different budgets N in the Hi-ToM dataset. Results are shown for (a) Vanilla and (b) Weighted aggregation strategies.

Table 3: BoN ($N = 1024$) inference-time scaling on the dataset from Kosinski (Kosinski, 2024), evaluated across different belief types. "Ori" denotes baseline accuracy; "+PBM" denotes accuracy with inference-time scaling.

Model	False Belief		Informed Protagonist		No Transfer		Present Protagonist		Average	
	Ori	+PBM	Ori	+PBM	Ori	+PBM	Ori	+PBM	Ori	+PBM
Qwen3-8B	83.3	87.5	83.8	85.0	92.8	97.5	79.5	85.0	84.8	88.8
Qwen3-4B	70.2	80.0	86.2	90.0	93.2	95.0	88.0	92.5	84.4	89.4
Qwen3-1.7B	18.2	35.0	15.5	37.5	24.8	60.0	13.8	30.0	18.1	40.6
Qwen3-0.6B	14.5	12.5	23.5	30.0	25.0	35.0	21.0	32.5	21.0	27.5

Table 4: PBM classification accuracy (%) across belief orders on the test set.

PBM	0-th	1-th	2-th	3-th	4-th	Avg.
Llama3.1-8B	99.2	94.6	89.0	87.0	79.9	90.0
Llama3.2-3B	99.1	91.9	84.9	83.8	73.8	86.7

Table 5: BoN inference-time scaling accuracy (%) on Hi-ToM using different PBMs.

Model+PBM	0-th	1-th	2-th	3-th	4-th	Avg.
Qwen3-4B + 8B	100.0	85.0	90.0	82.5	65.0	84.5
Qwen3-4B + 3B	100.0	77.5	77.5	72.5	47.5	75.0
Qwen3-1.7B + 8B	82.5	65.0	55.0	62.5	57.5	64.5
Qwen3-1.7B + 3B	82.5	60.0	45.0	47.5	50.0	57.0
Qwen3-0.6B + 8B	80.0	72.5	47.5	52.5	47.5	60.0
Qwen3-0.6B + 3B	77.5	55.0	27.5	35.0	32.5	45.5

This suggests that stronger models can better verify reasoning steps, while evaluating deeper recursive beliefs is inherently more challenging.

Impact of PBM Quality on Task Accuracy. We further test how the quality of the PBM affects end-task performance. Specifically, we run BoN inference-time scaling on the Hi-ToM dataset using different base models, guided either by a strong PBM (Llama3.1-8B-Instruct) or a weaker one (Llama3.2-3B-Instruct). As shown in Table 5, replacing the strong PBM with a weaker one consistently reduces accuracy across all base models and belief orders. This establishes a clear link between verifier quality and final task performance: a stronger PBM leads to better inference-time scaling outcomes.

Qualitative Analysis of PBM Behavior. To better understand when PBM succeeds or fails, we examine its behavior on reasoning traces. Below are two steps predicted by the Llama3.2-3B-Instruct PBM.

Scenario: Initially, everyone knows that the asparagus is in the `blue_cupboard`. At the current moment, Charlotte and Elizabeth are present in the room, while Alexander has just left. Charlotte holds a second-order belief about Alexander's belief regarding Elizabeth.

Step n:

- **Action:** `Elizabeth likes the red_box.`
- **State:** Irrelevant. Charlotte thinks Alexander thinks Elizabeth thinks the asparagus is in `blue_cupboard`.
- **Prediction:** $+$ **Ground Truth:** $+$
- **Annotation:** This step is correct. The statement is unrelated to the asparagus; no beliefs update. PBM correctly captures this invariance.

Step $n+1$:

- **Action:** `Elizabeth moved the asparagus to the green_bucket.`
- **State:** Only Elizabeth and Charlotte are present when this happens. Charlotte sees this move. Charlotte thinks Alexander thinks Elizabeth

Table 6: API price per 1M tokens.

Model	Input	Cached Input	Output	Total
gpt-4.1	$2.00	$0.50	$8.00	$10.50
gpt-4.1-mini	$0.40	$0.10	$1.60	$2.10
gpt-4o	$2.50	$1.25	$10.00	$13.75
gpt-4o-mini	$0.15	$0.075	$0.60	$0.825

thinks the asparagus is in `green_bucket`.

- **Prediction:** + **Ground Truth:** −
- **Annotation:** This step is incorrect. Since Alexander is not present, he cannot observe Elizabeth's action. Therefore, his beliefs (as perceived by Charlotte) should not change. PBM overgeneralizes belief update based on partial presence.

This example shows that while PBM handles simple irrelevant statements, it can fail on nested, perspective-sensitive updates, revealing a key challenge in verifying multi-agent reasoning.

4.5 Discussion

Cost Efficiency for API-based Usage. As shown in Table 1, applying PBM narrows the gap between small and large models: gpt-4.1-mini approaches gpt-4.1, while gpt-4o-mini gains +18.5 points, surpassing gpt-4o. Despite sampling $N{=}4$ outputs, mini models remain more cost-efficient, with per-million-token costs of only 2.10 and 0.825 compared to 10.50 and 13.75 for the larger models (Table 6). Furthermore, because all N samples share the same input prompt, the input cost is paid only once, and only the output tokens scale with N. This makes PBM-guided small-batch inference-time scaling a cheaper alternative to using larger models.

Scaling with Model Size. Figure 4 shows how ToM accuracy changes with model size. PBM consistently improves performance and strengthens the scaling trend. For Llama 3.2, the accuracy curve becomes steeper when equipped with PBM, suggesting that larger models benefit more and generalize better under our inference-time intervention. Interestingly, Qwen3-8B performs worse than Qwen3-4B under the vanilla setting, but becomes the best-performing variant once PBM is applied. This indicates that PBM not only boosts accuracy but can also unlock higher-order reasoning abilities that remain latent in the base model.

Comparison with RL-based Methods. Recent work (Lu et al., 2025) has explored fine-tuning LLMs with ToM supervision using GRPO (Shao et al., 2024) to enhance their ToM abilities. However, GRPO requires substantial compute and is notoriously difficult to optimize. In contrast, our PBM is lightweight and efficient: it trains in under three hours on a single GH200 GPU and can be applied to any target model without retraining. GRPO must be re-trained for each model and may even degrade performance on unrelated tasks such as GSM8K (Lu et al., 2025). Our method avoids this issue entirely by leaving model parameters unchanged. PBM thus offers a practical, generalizable, and non-invasive alternative for improving ToM reasoning.

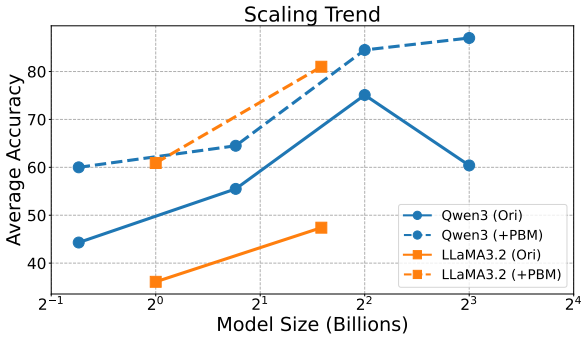


Figure 4: Scaling trend of average accuracy before and after applying PBM across different LLMs on Hi-ToM. "Ori" denotes baseline accuracy; "+PBM" denotes accuracy with inference-time scaling.

5 Related Work

DEL and Its Connections to ToM. DEL builds on a line of work in epistemic logic, tracing back to Hintikka's possible-world model of knowledge and belief (Hintikka, 1962) and Kripke's formal semantics (Kripke, 1963), and later evolving through studies on information change (Baltag et al., 1998). It was later formalized as a unified framework of epistemic and event models with product updates (Van Ditmarsch et al., 2007) for representing and updating agents' beliefs. This aligns naturally with the core of ToM, which concerns reasoning about others' beliefs. Early cognitive models used DEL to simulate belief change in multi-agent settings (Bolander and Andersen, 2011), showing its suitability for structured belief reasoning. More recent work uses logic-based simulators to supply symbolic supervision for belief updates (Bolander, 2014; Hansen and Bolander, 2020). Building on this line, we use DEL not only as a formalism for modeling beliefs but also as a scaffold for inference-time scaling, enabling compositional and verifiable reasoning in ToM tasks.

Inference-Time Scaling of LLMs. Recent work investigates inference-time scaling as an alternative to increasing model size for improving reasoning capabilities (Beeching et al., 2024; Muennighoff et al., 2025). Two main paradigms have been studied. One is single-trace scaling, which encourages deeper reasoning within a single inference path, often via reinforcement learning (Guo et al., 2025a; Cheng et al., 2025) or distillation from a stronger teacher (Li et al., 2025). The other is multi-trace scaling (Brown et al., 2024; Snell et al., 2025; Schaeffer et al., 2025), which generates multiple reasoning traces in parallel and selects the best outcome using voting (Wang et al., 2023, 2025) or external verifiers (Wang et al., 2024; Sun et al., 2024; Guo et al., 2025b; Saad-Falcon et al., 2025). Recent work further combines multi-trace generation with search algorithms such as tree search and beam search to refine reasoning step by step (Zhang et al., 2024; Lin et al., 2025). Our approach follows the multi-trace paradigm and introduces PBM-guided selection, extending inference-time scaling to ToM tasks.

6 Conclusion

This work introduces DEL-ToM, a framework that enhances Theory-of-Mind (ToM) reasoning in LLMs through inference-time scaling. By modeling belief updates with Dynamic Epistemic Logic (DEL) and training a verifier using DEL-generated labels, our approach enables structured and verifiable dynamic logical reasoning. DEL-ToM improves ToM performance across models and datasets, demonstrating that logical reasoning can be strengthened through formal logic and inference-time supervision. This opens new avenues for deploying ToM-capable LLMs in resource-constrained settings without retraining.

Limitation

Our approach depends on accurate belief supervision from a formal-logic-based simulator. Such supervision may not generalize to all types of reasoning or real-world language use. Additionally, beam search is less effective for models with weak instruction-following capabilities, limiting their practical deployment. Future work could explore more efficient trace selection methods and extend our approach to broader domains beyond ToM.

Acknowledgment

We thank the anonymous reviewers for their valuable feedback. We gratefully acknowledge the support of Lambda, Inc., for providing compute resources for this project. The work of Zhaozhuo Xu was supported by NSF grants 2451398 and 2450524.

References

Maryam Amirizaniani, Elias Martin, Maryna Sivachenko, Afra Mashhadi, and Chirag Shah. 2024. Do llms exhibit human-like reasoning? evaluating theory of mind in llms for open-ended responses. *arXiv preprint arXiv:2406.05659*.

Ian A Apperly and Stephen A Butterfill. 2009. Do humans have two systems to track beliefs and belief-like states? *Psychological review*, 116(4):953.

Guillaume Aucher and François Schwarzentruber. 2013. On the complexity of dynamic epistemic logic. In *Proceedings of the 14th Conference on Theoretical Aspects of Rationality and Knowledge*.

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*.

Simon Baron-Cohen. 1991. Precursors to a theory of mind: Understanding attention in others. *Natural theories of mind: Evolution, development and simulation of everyday mindreading*, 1(233-251):1.

Edward Beeching, Lewis Tunstall, and Sasha Rush. 2024. Scaling test-time compute with open models.

Thomas Bolander. 2014. Seeing is believing: Formalising false-belief tasks in dynamic epistemic logic. In *European conference on social intelligence (ECSI 2014)*, pages 87–107.

Thomas Bolander and Mikkel Birkegaard Andersen. 2011. Epistemic planning for single-and multi-agent systems. *Journal of Applied Non-Classical Logics*, 21(1):9–34.

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. *arXiv preprint arXiv:2407.21787*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Daniel C. Dennett. 1978. Beliefs about beliefs. *Behavioral and Brain Sciences*, 1:568.

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.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. 2025. Reasoning with exploration: An entropy perspective. *arXiv preprint arXiv:2506.14758*.

Gottlob Frege. 1879. Begriffsschrift, a formula language, modeled upon that of arithmetic, for pure thought. *From Frege to Gödel: A Source Book in Mathematical Logic*, 1931:1–82.

Alvin I. Goldman. 1979. What is justified belief? *Justification and Knowledge*, 17:1–23.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025a. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Jiaxin Guo, Zewen Chi, Li Dong, Qingxiu Dong, Xun Wu, Shaohan Huang, and Furu Wei. 2025b. Reward reasoning model. *arXiv preprint arXiv:2505.14674*.

Lasse Dissing Hansen and Thomas Bolander. 2020. Implementing theory of mind on a robot using dynamic epistemic logic. In *Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 1615–1621. International Joint Conference on Artificial Intelligence Organization.

Jaakko Hintikka. 1962. Knowledge and belief: An introduction to the logic of the two notions. *Studia Logica*, 16:119–122.

Jaakko Hintikka and Merrill B. P. Hintikka. 1989. *The Logic of Epistemology and the Epistemology of Logic: Selected Essays*. Springer Verlag, Dordrecht, Netherland.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Donald E Knuth. 1964. Backus normal form vs. backus naur form. *Communications of the ACM*, 7(12):735–736.

Michal Kosinski. 2024. Evaluating large language models in theory of mind tasks. *Proceedings of the National Academy of Sciences*, 121(45):e2405460121.

Saul Kripke. 1963. Semantical considerations on modal logic. *Acta Philosophica Fennica*, 16:83–94.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.

Dacheng Li, Shiyi Cao, Tyler Griggs, Shu Liu, Xiangxi Mo, Eric Tang, Sumanth Hegde, Kourosh Hakhamaneshi, Shishir G Patil, Matei Zaharia, Joseph E Gonzalez, and Ion Stoica. 2025. Llms can easily learn to reason from demonstrations structure, not content, is what matters! *arXiv preprint arXiv:2502.07374*.

Qingwen Lin, Boyan Xu, Zijian Li, Zhifeng Hao, Keli Zhang, and Ruichu Cai. 2025. Leveraging constrained monte carlo tree search to generate reliable long chain-of-thought for mathematical reasoning. *arXiv preprint arXiv:2502.11169*.

Zizheng Lin, Chunkit Chan, Yangqiu Song, and Xin Liu. 2024. Constrained reasoning chains for enhancing theory-of-mind in large language models. In *Pacific Rim International Conference on Artificial Intelligence*, pages 354–360. Springer.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

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

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*.

Jan Plaza. 2007. Logics of public communications. *Synthese*, 158:165–179.

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

Neil Rabinowitz, Frank Perbet, Francis Song, Chiyuan Zhang, SM Ali Eslami, and Matthew Botvinick. 2018. Machine theory of mind. In *Proceedings of*

the Thirty-Fifth International conference on machine learning, pages 4218–4227.

Bertrand Russell and Alfred North Whitehead. 1910. *Principia Mathematica Vol. I*. Cambridge University Press.

Jon Saad-Falcon, Buchanan E Kelly, Mayee F Chen, Tzu-Heng Huang, Brendan McLaughlin, Tanvir Bhathal, Shang Zhu, Ben Athiwaratkun, Frederic Sala, Scott Linderman, Azalia Mirhoseini, and Christopher Ré. 2025. Shrinking the generation-verification gap with weak verifiers. *arXiv preprint arXiv:2506.18203*.

Rylan Schaeffer, Joshua Kazdan, John Hughes, Jordan Juravsky, Sara Price, Aengus Lynch, Erik Jones, Robert Kirk, Azalia Mirhoseini, and Sanmi Koyejo. 2025. How do large language monkeys get their power (laws)? In *Proceedings of the Forty-second International Conference on Machine Learning*.

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*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

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*.

James WA Strachan, Dalila Albergo, Giulia Borghini, Oriana Pansardi, Eugenio Scaliti, Saurabh Gupta, Krati Saxena, Alessandro Rufo, Stefano Panzeri, Guido Manzi, and 1 others. 2024. Testing theory of mind in large language models and humans. *Nature Human Behaviour*, 8(7):1285–1295.

Winnie Street, John Oliver Siy, Geoff Keeling, Adrien Baranes, Benjamin Barnett, Michael McKibben, Tatenda Kanyere, Alison Lentz, Robin IM Dunbar, and 1 others. 2024. Llms achieve adult human performance on higher-order theory of mind tasks. *arXiv preprint arXiv:2405.18870*.

Zhiqing Sun, Longhui Yu, Yikang Shen, Weiyang Liu, Yiming Yang, Sean Welleck, and Chuang Gan. 2024. Easy-to-hard generalization: Scalable alignment beyond human supervision. *Advances in Neural Information Processing Systems*, 37:51118–51168.

Alfred Tarski. 1956. The concept of truth in formalized languages. In *Logic, semantics, metamathematics*, pages 152–278. Clarendon Press.

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

Johan Van Benthem. 2001. Games in dynamic-epistemic logic. *Bulletin of Economic Research*, 53(4):219–248.

Hans Van Ditmarsch, Wiebe van Der Hoek, and Barteld Kooi. 2007. *Dynamic epistemic logic*, volume 337. Springer Science & Business Media.

Evan Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, and 1 others. 2025. 2 olmo 2 furious (colm's version). In *Second Conference on Language Modeling*.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439.

Weiqin Wang, Yile Wang, and Hui Huang. 2025. Ranked voting based self-consistency of large language models. *arXiv preprint arXiv:2505.10772*.

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*.

Ludwig Wittgenstein. 1922. Tractatus logico-philosophicus. *Filosoficky Casopis*, 52:336–341.

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, Wentao Guo, Zirui Liu, Heng Ji, Zhaozhuo Xu, and Denghui Zhang. 2025. How large language models encode theory-of-mind: a study on sparse parameter patterns. *npj Artificial Intelligence*, 1(1):20.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024. Rest-mcts*: Llm self-training via process reward guided tree search. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, pages 64735–64772.

Appendix

A Prompt Templates

The full prompt templates are shown in Figures 5, 6, and 7.

B What Makes PBM Different from Majority Voting?

Here, we provide a analysis to compare PBM with majority voting.

Problem Setup and Notation. Given an input x, the language model π_θ must perform K sequential belief-updating steps, generating a trajectory

$$(z_1, z_2, \ldots, z_K), \quad z_i \sim \pi_\theta(\cdot \mid x, z_{<i}).$$

After the trajectory is complete, it outputs a final answer y, chosen from a set of L candidates (typically $L \approx 5-6$ in HiToM). We assume:

- Each step is independently correct with probability q.
- A trajectory is *Good* if all K steps are correct: $\Pr[\text{Good}] = q^K$.
- Otherwise, it is *Bad*: $\Pr[\text{Bad}] = 1 - q^K$.

Majority Voting. We sample N i.i.d. trajectories and return the most frequent final answer. Let:

- $G \sim \text{Binomial}(N, q^K)$: number of Good trajectories.
- $R = N - G$: number of Bad trajectories.

Under the uniform scattering assumption, Bad votes are evenly spread over $L-1$ wrong answers:

$$B_j \mid R \sim \text{Binomial}\left(R, \frac{1-q^K}{L-1}\right), \quad j = 1, \ldots, L-1.$$

Majority voting succeeds iff

$$E_{\text{maj}} = \{G \geq 1 \text{ and } G > \max_j B_j\}.$$

PBM Reranking. We sample N trajectories and assign each a stepwise score:

$$s(z) = \frac{1}{K} \sum_{k=1}^{K} \mathbf{1}\{z_k \text{ correct}\}.$$

Good trajectories receive score 1, and Bad trajectories score at most $1 - \frac{1}{K}$. We select the trajectory with the highest score. PBM thus succeeds iff

$$E_{\text{pbm}} = \{G \geq 1\}.$$

PBM Success Rate. By independence:

$$A_{\text{pbm}} = \Pr(E_{\text{pbm}}) = 1 - (1 - q^K)^N.$$

Majority Voting Success Rate. We have

$$\begin{aligned} A_{\text{maj}} &= \Pr(E_{\text{maj}}) \leq \Pr(G \geq 1) \\ &= 1 - (1 - q^K)^N = A_{\text{pbm}}. \end{aligned}$$

Hence, PBM always outperforms majority voting.

Remarks. For majority voting, divide both sides of $G > \max_j B_j$ by N and take the limit as $N \to \infty$. By the law of large numbers:

$$\frac{G}{N} \to q^K, \qquad \frac{\max_j B_j}{N} \to \frac{1-q^K}{L-1}.$$

A necessary condition for success is therefore:

$$q^K > \frac{1-q^K}{L-1} \quad \Leftrightarrow \quad q^K > \frac{1}{L}.$$

If $q^K \leq \frac{1}{L}$ (typical for small models or hard ToM question), then

$$\lim_{N \to \infty} A_{\text{maj}} = 0.$$

In conclusion, majority voting is vulnerable to *vote dilution*: if $q^K \leq \frac{1}{L}$, Bad trajectories cluster on wrong answers and can dominate.

This analysis explains why PBM offers more reliable inference than majority voting, especially in complex ToM settings.

One-Shot Prompt - Part 1

```
Here is a story that unfolds in chronological order.

You will be asked a question about the story, which may involve either:
(1) Locating an object, or
(2) Inferring an agent's mental state (e.g., what A thinks B thinks C thinks).

To solve it, think step-by-step. At each step, repeat the current line from
    the story, then explain its effect on beliefs. Use [Null] if someone does
    not yet have knowledge. If a belief chain cannot be formed (e.g., some
    agent exited too early), freeze belief at the last available step.

<Note>
{note}

In public or private communication:
- The speaker believes the listener will believe the claim.
- If the listener exited the room earlier than the speaker, they will believe
    it.

If the question is zero-order (e.g., "Where is X really?"), then in each step,
     only track the actual location of the object (e.g., "X is in [Y]"). You do
     not need to track nested beliefs.

Here is an example:
<Story>
1 Amelia, Chloe, Liam, Owen and Benjamin entered the TV_room.
2 The celery is in the red_envelope.
3 Amelia made no movements and stayed in the TV_room for 1 minute.
4 Chloe lost his watch.
5 Amelia exited the TV_room.
6 Chloe moved the celery to the green_bucket.
7 Chloe exited the TV_room.
8 Liam moved the celery to the red_bathtub.
9 Liam exited the TV_room.
10 Owen made no movements and stayed in the TV_room for 1 minute.
11 Owen exited the TV_room.
12 Benjamin made no movements and stayed in the TV_room for 1 minute.
13 Benjamin exited the TV_room.
14 Amelia, Chloe, Liam, Owen and Benjamin entered the waiting_room.
15 Liam publicly claimed that celery is in the white_bathtub now.
16 Benjamin privately told Liam that the celery is in the blue_drawer now.

<Question>
Where does Owen think Liam thinks Chloe thinks the celery is?

<Trace>
## Step 1 ##
Amelia, Chloe, Liam, Owen and Benjamin entered the TV_room.
Everyone is present, but the celery's location is still unknown.
Owen thinks Liam thinks Chloe thinks the celery is in [Null]

## Step 2 ##
The celery is in the red_envelope.
Everyone observes this.
Owen thinks Liam thinks Chloe thinks the celery is in [red_envelope]

## Step 3 ##
Amelia made no movements and stayed in the TV_room for 1 minute.
No effect.
Owen thinks Liam thinks Chloe thinks the celery is in [red_envelope]
```

Figure 5: One-Shot Prompt - Part 1.

One-Shot Prompt - Part 2

```
## Step 4 ##
Chloe lost his watch.
Irrelevant.
Owen thinks Liam thinks Chloe thinks the celery is in [red_envelope]

## Step 5 ##
Amelia exited the TV_room.
Irrelevant.
Owen thinks Liam thinks Chloe thinks the celery is in [red_envelope]

## Step 6 ##
Chloe moved the celery to the green_bucket.
Only Chloe, Liam, Owen, Benjamin are present. They all see this move.
Owen thinks Liam thinks Chloe thinks the celery is in [green_bucket]

## Step 7 ##
Chloe exited the TV_room.
Chloe's belief frozen; still [green_bucket]
Owen thinks Liam thinks Chloe thinks the celery is in [green_bucket]

## Step 8 ##
Liam moved the celery to the red_bathtub.
Only Liam, Owen, Benjamin present. They observe the move. Chloe not present,
    so her belief unchanged.
Owen thinks Liam thinks Chloe thinks the celery is in [green_bucket]

## Step 9 ##
Liam exited the TV_room.
No change.
Owen thinks Liam thinks Chloe thinks the celery is in [green_bucket]

## Step 10 ##
Owen made no movements and stayed in the TV_room for 1 minute.
Irrelevant.
Owen thinks Liam thinks Chloe thinks the celery is in [green_bucket]

## Step 11 ##
Owen exited the TV_room.
Owen's belief frozen.
Owen thinks Liam thinks Chloe thinks the celery is in [green_bucket]

## Step 12 ##
Benjamin made no movements and stayed in the TV_room for 1 minute.
Irrelevant.
Owen thinks Liam thinks Chloe thinks the celery is in [green_bucket]

## Step 13 ##
Benjamin exited the TV_room.
No change.
Owen thinks Liam thinks Chloe thinks the celery is in [green_bucket]

## Step 14 ##
Everyone entered the waiting_room.
No effect on beliefs.
Owen thinks Liam thinks Chloe thinks the celery is in [green_bucket]
```

Figure 6: One-Shot Prompt - Part 2.

One-Shot Prompt - Part 3

```
## Step 15 ##
Liam publicly claimed that celery is in the white_bathtub now.
Owen hears this statement. However, public speech only affects first- and
    second-order beliefs (e.g., what Liam believes, what Owen thinks Liam
    believes, and what Liam thinks Owen believes). It does not change Owen's
    belief about what Liam thinks Chloe thinks.
Owen thinks Liam thinks Chloe thinks the celery is in [green_bucket]

## Step 16 ##
Benjamin privately told Liam that the celery is in the blue_drawer now.
Owen does not hear this, but more importantly, private communication only
    affects beliefs between the speaker and the listener. It can change what
    Liam believes (based on exit order), or what Liam thinks Benjamin believes
    (based on exit order), or what Benjamin thinks Liam believes (always change
    ) - but it cannot affect higher-order beliefs. So this does not change Owen
    's belief about what Liam thinks Chloe thinks.

Owen thinks Liam thinks Chloe thinks the celery is in [green_bucket]

Final Answer: [green_bucket]

Now it's your turn.

<Story>
{story}

<Question>
{question}

Give a step-by-step trace as in the example. Then, give the final answer in
    one line like:
Final Answer: [your choice]

<trace>
```

Figure 7: One-Shot Prompt - Part 3.