

PDDL-MIND: Reliable State Tracking is All You Need for Theory-of-Mind Benchmarks

Anonymous ACL submission

Abstract

Large language models (LLMs) perform substantially below human level on existing theory-of-mind (ToM) benchmarks, even when augmented with chain-of-thought prompting or probabilistic belief updates. We argue that these failures primarily arise from unreliable implicit state tracking rather than limitations in high-level reasoning. We introduce **PDDL-MIND**, a neuro-symbolic framework that decouples environment state evolution from belief inference. By translating narrative descriptions into explicit states and actions expressed in *Planning Domain Definition Language (PDDL)*, and by verifying action-induced state transitions against a predefined domain, PDDL-MIND provides LLMs with a logically consistent and explicit representation of world states for ToM tasks. Experiments on MMTOM-QA and MuMa-ToM show that PDDL-MIND achieves over 5% absolute accuracy gain over the best existing state-of-the-art method on ToM benchmark questions.

1 Introduction

Theory-of-Mind (ToM) benchmarks present narratives in which agents act in the world and observers must infer what those agents believe, know, or intend. These narratives typically take one of two forms: descriptions of *physical actions* in an environment, or *dialogue transcripts* between agents. These variations share a common underlying structure: each narrative is fundamentally a sequence of events that induces a sequence of world states (Huang et al., 2024a; Xu et al., 2025).

Frontier large language models (LLMs) continue to perform substantially below human level on these ToM tasks (Jin et al., 2024; Kim et al., 2025). There is no consensus on the root cause of these failures. Broadly, two competing hypotheses have emerged. The first attributes errors to *state tracking*: models fail to maintain accurate representa-

tions of world states as events accumulate over time. The second attributes errors to *reasoning*: models track the progression of events adequately, but fail during the subsequent inference of beliefs or intentions from those states (Huang et al., 2024a). Works from both isles admit ToM benchmark instances can be partitioned into chains of events, yet leave the notion of an event unanalyzed beyond surface segmentation. We argue that a deeper understanding the nature of events permits proper disentanglement of these hypotheses.

We draw insight from neo-Davidsonian event semantics. In the neo-Davidsonian view: “Mary lifted the cup to the kitchen,” “Mary walked with the cup,” and “Mary moved something” may all be true of a single occurrence (Schein, 2012; Kratzer, 2023). Defining a verb means specifying the event’s preconditions and effects. Events thus provide a level of abstraction above linguistic variation, a stable referent that diverse verb phrases can converge on.

Motivated by this observation, a programming language that places emphasis on predicate definition is a natural substrate to augment LLMs’ event perception and tracking. We introduce PDDL-MIND, a framework that provides a reliable and explicit state-tracking mechanism for ToM narratives. Starting from a predefined *Planning Domain Definition Language (PDDL)* (Ghallab et al., 1998) domain file with 7 general action types, we use an LLM to parse the initial environment description into a PDDL problem file and to translate narrative events into a sequence of PDDL actions. We then verify each action by checking its preconditions and executing valid transitions within the domain, filtering out invalid actions. As a result, we obtain a logically consistent sequence of world states, which we manually verified to be correct in 90% of cases. Finally, we provide this verified state sequence to the LLM and evaluate it solely on ToM inference questions.

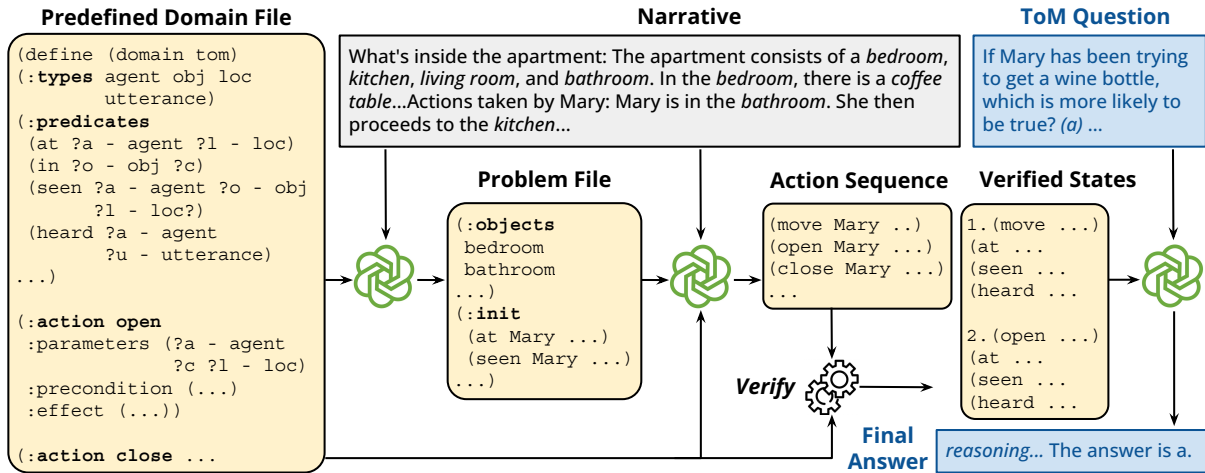


Figure 1: Starting from a predefined PDDL domain file, PDDL-MIND use a LLM to parse the narrative description into a PDDL problem file describing an initial world state, and then a PDDL action sequence. After verifying actions by induced state transitions, the LLM takes in a ToM question and verified actions and state transitions to produce a final answer.

PDDL-MIND with GPT-4o achieves 88.3% accuracy on MMTOM-QA and 88.8% accuracy on MuMa-ToM, using only 3 LLM calls per example. Both outperforms the previous state-of-the-art method, AutoToM (Zhang et al., 2025), which requires an average of 27 LLM calls per example, by more than 5%. These results provide strong evidence that unreliable state tracking, rather than insufficient reasoning capacity, is the primary bottleneck in current ToM evaluations of LLMs.

2 PDDL-MIND

We introduce PDDL-MIND, a neuro-symbolic framework that decouples environment state evolution from belief inference (Figure 1).

Predefined domain file. We begin with a predefined PDDL domain file. The design is motivated by the structure of existing textual ToM benchmarks, which pose ToM questions after presenting narratives in one of two forms: descriptions of *physical actions* in an environment, or *dialogue transcripts* between agents.

We define four object types in the domain: agent, representing people mentioned in the narrative (e.g., “Mary”); loc, representing locations in the environment (e.g., “bathroom”); obj, representing physical items (e.g., “apple”); and utterance, representing spoken messages in the narrative (e.g., “Mary has a golden retriever”). We then define a set of predicates to describe domain states, such as *at*, *in*, and *holding*. Crucially, we introduce the predicate *seen* to track whether an object is observed

by an agent, and *heard* to track whether an utterance is heard by an agent. Finally, we define seven actions: *move*, *open*, *close*, *grab*, *drop*, *ask*, and *tell*, each with explicitly specified parameters, preconditions, and postconditions. The complete PDDL domain file is provided in Appendix C.

State and action extraction. Given the domain file, we first prompt an LLM to parse the narrative into a PDDL problem file, which specifies the objects present in the environment and the initial state s_0 . We define a state s as a set of grounded predicates, such as (at Mary bathroom). We find this step to be relatively straightforward for frontier LLMs, because most ToM benchmarks describe simple, small-scale environments.

Next, using both the domain file and the generated problem file, the LLM parses the narrative into a sequence of candidate actions (a_1, \dots, a_t) . Because narratives may include events that cannot be mapped to PDDL actions (e.g., “steps near the sink”), we filter the generated action sequence post-hoc to retain only valid actions. For each candidate action a_i , if the current state s_{i-1} satisfies the action’s preconditions, we accept the action and apply its postconditions to produce the next state s_i . Otherwise, the action is ignored and not executed. This process yields a verified sequence of actions and corresponding world states.

ToM question answering. The LLM is provided with the verified state sequence and a ToM question, and is asked to perform inference in a chain-of-thought manner. By conditioning inference on ex-

Method	MMToM-QA	MuMa-ToM
BIP-ALM*	56.2	33.9
LIMP*	55.3	76.6
TT*	69.0	-
AutoToM*	83.0	81.4
Human*	82.5	93.5
PDDL-MIND	88.3	88.8

Table 1: PDDL-MIND outperforms existing methods on MMToM-QA and MuMa-ToM. * denotes results from previous papers.

PLICITLY verified states, PDDL-MIND isolates ToM reasoning from errors in implicit state tracking.

3 Experiment results

We evaluate PDDL-MIND on two ToM benchmarks: MMToM-QA (Jin et al., 2024), which contains 600 examples focusing on long-context belief and goal inference, and MuMa-ToM (Shi et al., 2024), which contains 900 examples targeting multi-agent belief and goal inference. Each ToM task is a multi-choice question answering evaluated by accuracy. We compare against four representative baselines: **BIP-ALM** (Jin et al., 2024) and **LIMP** (Shi et al., 2024), which combine Bayesian Inverse Planning with LLMs, with BIP-ALM performing explicit belief updates while LIMP does not; **TT** (Kim et al., 2025), which traces agents’ mental states by generating and weighting hypotheses based on observations; and **AutoToM** (Zhang et al., 2025), which iteratively constructs and refines agent models tailored to each ToM task.

PDDL-MIND provides reliable state tracking.

We report average accuracy for all methods evaluated with GPT-4o (OpenAI, 2024) in Table 1. PDDL-MIND outperforms all baselines on both benchmarks, achieving accuracy improvements of more than 5.3% on MMToM-QA and 7.4% on MuMa-ToM. To directly assess state-tracking quality, we manually inspect 20 verified action sequences per benchmark and find that 36 out of 40 (90%) are correct and fully consistent with the narratives. While all baselines incorporate some form of state tracking, these results indicate that their implicit or heuristic mechanisms are substantially less reliable than the explicit verification used in PDDL-MIND.

PDDL-MIND is cost-efficient. PDDL-MIND requires only three LLM calls per example, whereas all baseline methods in Table 1 require an order of magnitude more calls. For instance, AutoToM reports an average of 17.6 LLM calls per example on MMToM-QA and 35.1 calls on MuMa-ToM. These results demonstrate that reliable state tracking can be achieved without increased inference cost, and in fact leads to significantly more efficient evaluation.

Remaining errors reflect both parsing and reasoning limitations.

Despite its strong performance, PDDL-MIND still exhibits an error rate of approximately 12% when evaluated with GPT-4o. We manually analyze 20 failure cases per benchmark and find that 13 cases (33%) stem from parsing errors that produce incorrect action sequences, while the remaining 67% are due to genuine reasoning failures given correct states. This analysis suggests that, under perfect state tracking, the error rate would drop to approximately 8%, further supporting the claim that state tracking is the dominant bottleneck in current ToM evaluations. We provide qualitative examples in Appendix D.2.

PDDL-MIND yields consistent gains across models and categories.

We further compare PDDL-MIND against more cost-efficient ToM approaches, including vanilla chain-of-thought (CoT), which uses a single LLM call per instance, and DWM (Huang et al., 2024a), which partitions narratives into chunks and queries mental states after each segment. As shown in Table 2, PDDL-MIND consistently outperforms these baselines across nearly all sub-categories of MMToM-QA and MuMa-ToM and across different model backbones. The sole exception is the Belief category on MuMa-ToM when using Claude-Sonnet-4.5 (Anthropic, 2024), where both PDDL-MIND and DWM exceed 95% accuracy, with DWM outperforming by a modest margin of 3.3%. Further manual inspection reveals that, at such high accuracy levels, residual parsing errors disproportionately limit gains.

4 Related Work and Conclusion

Riemer et al. (2025) distinguishes *literal* ToM (accurately verbalizing another agent’s beliefs given context) from *functional* ToM (adapting one’s policy based on such inferences). We address literal ToM.

Model	Method	# Calls	MMToM-QA			MuMa-ToM			
			Avg.	Belief	Goal	Avg.	Belief	Goal(S)	Goal(B)
Kimi-K2-Instruct	CoT	1	54.2	70.6	42.0	58.9	63.3	66.7	48.0
	DWM	4	31.7	35.3	29.0	54.6	58.9	57.8	58.9
	PDDL-MIND	3	74.5	80.3	68.7	77.3	67.0	85.2	81.8
GPT-4o	CoT	1	61.4	85.3	43.5	41.8	57.8	31.1	37.0
	DWM	4	36.0	46.0	28.5	63.2	76.7	53.3	60.0
	PDDL-MIND	3	88.3	91.0	85.7	88.8	88.7	92.0	83.3
Claude-Sonnet-4.5	CoT	1	69.4	93.3	51.5	58.6	66.7	44.4	64.0
	DWM	4	66.6	76.6	59.0	86.1	98.9	72.2	87.0
	PDDL-MIND	3	90.9	92.7	89.0	88.6	95.6	82.2	87.5

Table 2: PDDL-MIND *v.s.* cost-efficient methods. PDDL-MIND consistently outperform on each sub-category of MMToM-QA and MuMa-ToM across models.

Dynamics of implicit wagers in inference-time augmentation. Inference-time augmentation methods that do not modify model weights share a common incentive: eliciting structure the model is assumed to already possess. This division of labor places an implicit wager that the language model has capability C : if it does, performance improves; if not, no scaffold compensates, since there is no more-intelligent system to fall back on. Our results suggest a more favorable wager: separate state tracking from reasoning. Language models reliably parse into constrained symbolic forms and perform variable binding (Wu et al., 2025; Prakash et al., 2025), particularly for PDDL (Liu et al., 2023; Zhu et al., 2025); we situate our approach in this regime.

Probabilistic methods. One family wagers on numerical confidence estimates. AutoToM (Zhang et al., 2025) and BIP-ALM (Jin et al., 2024) require LM logits over actions conditioned on hypothesized goals and beliefs, then aggregate via Bayes’ rule across timesteps. ThoughtTracing (Kim et al., 2025) assigns each hypothesis a numerical weight by mapping LLM categorical likelihood judgments to fixed scores, tuning favorability under random resampling. These approaches hinge on LLMs inducing a coherent probability measure over action–goal–belief triples, calibrated to track the diagnosticity of evidence; such measures remain implicit in the training distribution (Huang et al., 2024b). We hypothesize a second architectural hazard: when scalar comparison is the sole intermediate arbiter, upstream linguistic richness cannot overcome a numerical bottleneck at the decision

point—information that does not separate the scalar cannot affect the outcome, regardless of how richly it was represented moments before. PDDL-MIND avoids this bottleneck by preserving linguistic structure throughout inference, aligning with discrete methods.

Discrete state methods. Partially sidestepping these hazards, DWM (Huang et al., 2024a), EnigmaToM (Xu et al., 2025), and PDDL-MIND embrace discreteness by parsing natural language into symbolic representations while assuming deterministic transition rules. DWM and EnigmaToM place two wagers: that neural models can identify state-relevant events **and** correctly infer the transition rules. Our work shows the second wager is unnecessary. The transition semantics of ToM benchmarks are fully specified by their verb inventories, with explicit preconditions and effects; asking a model to “discover” what a benchmark already defines invites hallucination and inconsistency.

Conclusion. We show that current LLM failures on ToM tasks primarily reflect unreliable implicit state tracking rather than deficiencies in high-level reasoning, and that PDDL untangling isolates this bottleneck. We therefore argue that future ToM benchmarks should treat accurate, verifiable state tracking as a prerequisite for meaningful assessment. Broader philosophical implications of this disentanglement are discussed in Appendix B.2.

Limitations

First, our evaluation does not cover purely conversational theory-of-mind settings in which mental states must be inferred solely from dialogue, with-

out reference to an underlying physical environment. Although our PDDL-based representation supports basic utterances through ask and tell actions, richer conversational phenomena—such as implicature, deception, and pragmatic reasoning—are not explicitly modeled. In principle, this limitation can be addressed by extending the domain file with additional conversation-specific actions and predicates; we discuss such extensions in Appendix B.1.

Second, PDDL-MIND operates entirely on textual inputs and symbolic abstractions, and therefore does not address multimodal ToM scenarios involving visual, spatial, or embodied perception. Extending the framework to grounded or multimodal environments would require integrating perception modules and aligning their outputs with symbolic state representations, which is beyond the scope of this work.

Finally, our approach relies on a predefined PDDL domain, which introduces brittleness for weaker models that struggle to correctly parse narratives into symbolic actions and states. While frontier models handle this parsing reliably in our experiments, improving robustness for smaller or less capable models remains an open challenge.

References

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziemis, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alex Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. 2025. [GEPA: Reflective prompt evolution can outperform reinforcement learning](#). In *First Workshop on Foundations of Reasoning in Language Models*.

Anthropic. 2024. [System card: Claude sonnet 4.5](#).

Chunkit Chan, Cheng Jiayang, Yauwai Yim, Zheyue Deng, Wei Fan, Haoran Li, Xin Liu, Hongming Zhang, Weiqi Wang, and Yangqiu Song. 2024. [NegotiationToM: A benchmark for stress-testing machine theory of mind on negotiation surrounding](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4211–4241, Miami, Florida, USA. Association for Computational Linguistics.

Kanishk Gandhi, Jan-Philipp Fränken, Tobias Gerstenberg, and Noah Goodman. 2023. [Understanding social reasoning in language models with language models](#). In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Malik Ghallab, Adele Howe, Craig Knoblock, Drew McDermott, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins. 1998. PDDL - the planning domain definition language. *Technical Report, Tech. Rep.*

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.

X. Angelo Huang, Emanuele La Malfa, Samuele Marro, Andrea Asperti, Anthony G. Cohn, and Michael J. Wooldridge. 2024a. [A notion of complexity for theory of mind via discrete world models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2964–2983, Miami, Florida, USA. Association for Computational Linguistics.

Yukun Huang, Yixin Liu, Raghuvver Thirukovalluru, Arman Cohan, and Bhuwan Dhingra. 2024b. [Calibrating long-form generations from large language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13441–13460, Miami, Florida, USA. Association for Computational Linguistics.

Chuanyang Jin, Yutong Wu, Jing Cao, Jiannan Xiang, Yen-Ling Kuo, Zhiting Hu, Tomer Ullman, Antonio Torralba, Joshua B Tenenbaum, and Tianmin Shu. 2024. [Mmtom-qa: Multimodal theory of mind question answering](#). *arXiv preprint arXiv:2401.08743*.

Hyunwoo Kim, Melanie Sclar, Tan Zhi-Xuan, Lance Ying, Sydney Levine, Yang Liu, Joshua B. Tenenbaum, and Yejin Choi. 2025. [Hypothesis-driven theory-of-mind reasoning for large language models](#). In *COLM*.

Hyunwoo Kim, Melanie Sclar, Xuhui Zhou, Ronan Le 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*.

Angelika Kratzer. 2023. [Situations in Natural Language Semantics](#). In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*, Fall 2023 edition. Metaphysics Research Lab, Stanford University.

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, Hong Kong, China. Association for Computational Linguistics.

Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. 2023.

406	LLM+P: Empowering large language models with optimal planning proficiency. <i>ArXiv preprint</i> .	463
407		464
408	OpenAI. 2024. Gpt-4o system card . <i>Preprint</i> , arXiv:2410.21276.	465
409		466
410	Nikhil Prakash, Natalie Shapira, Arnab Sen Sharma, Christoph Riedl, Yonatan Belinkov, Tamar Rott Shaham, David Bau, and Atticus Geiger. 2025. Language models use lookbacks to track beliefs . <i>Preprint</i> , arXiv:2505.14685.	467
411		468
412		469
413		470
414		
415	Matthew Riemer, Zahra Ashktorab, Djallel Bouneffouf, Payel Das, Miao Liu, Justin D. Weisz, and Murray Campbell. 2025. Position: Theory of mind benchmarks are broken for large language models . In <i>Forty-second International Conference on Machine Learning Position Paper Track</i> .	471
416		472
417		473
418		474
419		
420		
421	Barry Schein. 2012. <i>Event Semantics</i> . In Gillian Russell, Delia Graff Fara, ed., <i>The Routledge Companion to Philosophy of Language</i> , pages 280–294.	475
422		476
423		477
424	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 <i>Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)</i> , pages 13960–13980, Toronto, Canada. Association for Computational Linguistics.	478
425		479
426		480
427		481
428		482
429		483
430		
431		
432	Haojun Shi, Suyu Ye, Xinyu Fang, Chuanyang Jin, Layla Isik, Yen-Ling Kuo, and Tianmin Shu. 2024. Muma-tom: Multi-modal multi-agent theory of mind . <i>Preprint</i> , arXiv:2408.12574.	475
433		476
434		477
435		478
436	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 <i>Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)</i> , pages 8292–8308, Bangkok, Thailand. Association for Computational Linguistics.	479
437		480
438		481
439		482
440		483
441		
442		
443		
444	Yiwei Wu, Atticus Geiger, and Raphaël Millière. 2025. How do transformers learn variable binding in symbolic programs? In <i>Forty-second International Conference on Machine Learning</i> .	475
445		476
446		477
447		478
448	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 <i>Findings of the Association for Computational Linguistics: EMNLP 2023</i> , pages 10691–10706, Singapore. Association for Computational Linguistics.	479
449		480
450		481
451		482
452		483
453		
454		
455	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 <i>Findings of the Association for Computational Linguistics: ACL 2025</i> , pages 13598–13622, Vienna, Austria. Association for Computational Linguistics.	475
456		476
457		477
458		478
459		479
460		480
461		481
462		482
		483

484	Appendix	
485	A More Related Works	7
486	B Dissecting the Structure of ToM Benchmarks	7
487	B.1 Tree Structure	7
488	B.2 Philosophical Implications	8
489		
490	C Prompts	10
491	D Experiment Details	12
492	D.1 Implementation Details	12
493	D.2 Error Analysis	12
494	E Checklist Information	13

495 **A More Related Works**

496 **Textual ToM benchmarks.** A large body of
 497 theory-of-mind benchmarks are purely textual, in-
 498 cluding ToMi (Le et al., 2019), BigToM (Gandhi
 499 et al., 2023), Hi-ToM (Wu et al., 2023), Fan-
 500 ToM (Kim et al., 2023), OpenToM (Xu et al., 2024),
 501 and NegotiationToM (Chan et al., 2024). We did
 502 not test on all of these benchmarks, as they all share
 503 a common narrative structure: each presents a se-
 504 quence of events in one of two forms—descriptions
 505 of *physical actions* within an environment, or *di-*
 506 *alogue transcripts* between agents. Moreover, re-
 507 sults from prior methods (e.g., AutoToM) indicate
 508 that several benchmarks, such as ToMi and Big-
 509 ToM, are already saturated, limiting their utility for
 510 distinguishing model capabilities.

511 **Prompting-based ToM methods.** Prior
 512 prompting-based approaches to ToM emphasize
 513 the apparent limitations of LLMs in mental-state
 514 reasoning. For instance, SymbolicToM (Sclar
 515 et al., 2023) constructs explicit belief graphs for
 516 each character, SimToM (Wilf et al., 2024) filters
 517 narratives to retain only events observable to a
 518 target agent before prompting, and TimeToM (Hou
 519 et al., 2024) augments stories with explicit
 520 temporal structure. Although superficially similar,
 521 PDDL-MIND is grounded in a fundamentally
 522 different perspective: LLMs are capable of
 523 theory-of-mind reasoning, but are primarily limited
 524 by unreliable state tracking. This perspective finds
 525 theoretical grounding in GEPA’s (Agrawal et al.,
 526 2025) principle of competence-aligned scaffolding.
 527 Effective augmentation identifies what the model
 528 performs reliably and externalizes the rest. LLMs

interpret linguistic feedback productively but learn
 529 poorly from scalar gradients; GEPA accordingly
 530 shifts learning to language-based reflection.
 531 LLMs reason over explicit states adequately but
 532 recover states from narrative variation unreliably;
 533 PDDL-MIND accordingly shifts state tracking to
 534 verified PDDL transitions. Enabling substantially
 535 improved ToM inference.
 536

B Dissecting the Structure of ToM Benchmarks 537
 538

B.1 Tree Structure 539

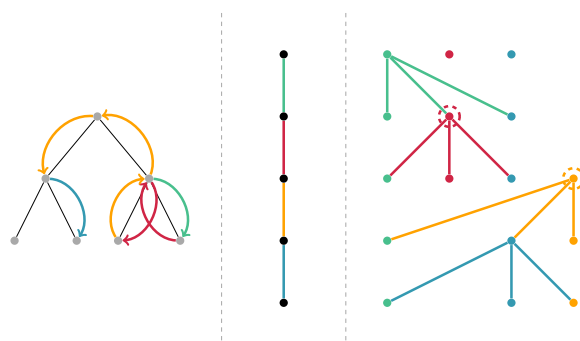


Figure 2: L: Actions M: Events R: Conversations

Figure 2 presents three schematic views of ToM
 540 benchmark instances.
 541

Single-agent action sequences (left panel). The
 542 left panel illustrates a typical single-agent Sally-
 543 Anne instance. Consider the narrative:
 544

545 Alice is in the kitchen and opens the
 546 fridge, moves to the cabinet and opens
 547 it, she moves to the living room, and
 548 opens the cabinet there.

The event colors correspond to the arrows in the
 549 left panel of Figure 2. This setting admits a tree
 550 representation: the task implicitly induces a con-
 551 tainment hierarchy over entities in the scene. The
 552 house contains rooms; rooms contain containers
 553 (e.g., fridges, cabinets); and containers may them-
 554 selves contain objects. Each colored arrow denotes
 555 an action that updates either the agent’s location
 556 within this hierarchy or their perceptual access at
 557 a node (e.g., by opening a container). State track-
 558 ing in such benchmarks reduces to maintaining the
 559 agent’s position in the tree and recording which
 560 containers have been opened at each node. Com-
 561 putationally, this corresponds to tree traversal with
 562 local state updates.
 563

Multi-agent conversation (right panel). The right panel extends this analysis to multi-agent conversation. We model conversation as a closed broadcast environment. Agents occupy a figurative room; when one agent speaks, all agents currently present append the utterance to memory, regardless of its truth value. Agents may leave, after which they no longer receive subsequent utterances. Agents may also join mid-conversation and begin receiving updates from that point onward.

Concretely, the right panel can be read as:

Alice: “I hope everyone is doing well!”

Billy: “I need to leave now, talk to you later!”

Cliff: “Hope I’m not interrupting!”

David: “Of course you’re not!”

Read the right panel row by row (top to bottom), where each row is a timestep and each column corresponds to an agent who is present at that time. At $t = 0$, agents **A**, **B**, and **C** are present. At $t = 1$, **A** speaks; edges from the speaker to all present listeners represent the broadcast relation. At $t = 2$, **B** speaks, and the dotted circle indicates that **B** is announcing departure; **B** is removed at the next timestep. At $t = 3$, **C** speaks while (per the schematic) joining late; **C** is then added to the set of present agents, and **D** subsequently addresses the remaining agents at $t = 4$.

From the schematic, the update structure induced by multi-agent conversation—once preconditions and effects are applied—can be viewed as a collection of disjoint trees. Benchmarks with this organization can, in principle, be modeled as PDDL states with predicates corresponding to SPEAK, HEAR, PRESENT, JOIN, and LEAVE.

A minimal event-state view (middle panel). The middle panel depicts a minimal structure common to ToM benchmark instances: a sequence of states (black nodes) connected by events (colored edges), read from top to bottom. This abstraction factors out linguistic variation and isolates the dynamics that a ToM system must track. With a universal PDDL file, one event may correspond to a physical action while another corresponds to a conversational update. Under PDDL-MIND, preconditions and effects are defined uniformly across event types, yielding a representation that is inherently multi-agent.

Implications for benchmark structure. Across these scenarios, tree structure arises frequently

as an abstraction for logging event effects. This prevalence reflects inductive biases that many ToM benchmarks implicitly adopt: (i) Sally–Anne-style action narratives assume a tree-structured world (containment hierarchies), and (ii) conversational benchmarks assume a fixed arrow of time that induces an acyclic update structure. Recognizing these biases, together with the fact that trees form a strict subset of graphs, allows us to place a stronger structural bound on ToM state-tracking causal models than the graph-based commitments in Xu et al. (2025), where graphs are used to track entity states. We also encourage future benchmark design to explore beyond these biases in order to obtain evaluations that are more resistant to purely symbolic state-tracking strategies.

B.2 Philosophical Implications

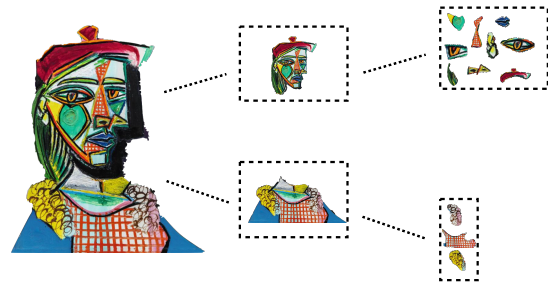


Figure 3: The Picasso Thesis

This section discusses philosophical implications of disentangling state tracking from belief inference.

Complexity frameworks such as Huang et al. (2024a) propose that the complexity of theory-of-mind tasks can be characterized by the number of states an observer must track. This view presupposes that a task comes with a natural partition into discrete states. We argue that such partitions exist only under specific structural conditions, and that making these conditions explicit clarifies both the scope and the limits of rigorous ToM evaluation.

Inferring artistic intention from a painting is itself a theory-of-mind task. Consider Picasso’s *Lover in a Beret* in Figure 3. If asked to interpret Picasso’s intentions, a typical human viewer may first recognize a human face in the center of the frame. This recognition constitutes immediate progress toward attributing an intention (to depict a face), after which the viewer can further decompose the face into subcomponents (e.g., nose, eyes) and stylistic distortions characteristic of cubist portraiture.

653 However, human face’s physiology is shaped by
654 millennia of selection pressures under Earth’s con-
655 ditions. Now imagine presenting the same painting
656 to an alien with human-comparable intelligence
657 but whose evolution under a distinct environment
658 yields no facial similarity to humans. If the alien
659 has never encountered a human face, it is unreason-
660 able to expect even the shallow interpretations that
661 humans with the requisite background knowledge
662 achieve effortlessly. Instead, the alien may rely on a
663 more basic representation: geometric properties of
664 the image, trajectories of lines and curves, enclosed
665 shapes, and relations among regions. The alien is
666 likely to form a partition of the painting starkly dif-
667 ferent from Figure 3. One might argue that humans
668 also use geometric information—perhaps passing
669 through a geometric stage before recognizing that a
670 face is a face and a nose is a nose. If so, the earlier
671 depiction of interpretation omitted several inter-
672 mediate steps: humans compress many geometric
673 primitives into a single semantic unit. The curves
674 and angles undergo a gestalt transformation and
675 was assigned contingent hierarchies by a viewer
676 who possesses the concept “face.”

677 This thought experiment poses a challenge for
678 state-counting approaches to ToM complexity. The
679 same painting induces markedly different state rep-
680 resentations depending on the observer’s concep-
681 tual vocabulary. For a human, the painting may
682 involve tracking a small number of semantic units
683 (face, expression, background). For the alien, it
684 may require tracking many more geometric prim-
685 itives. If the number of states depends on the ob-
686 server in this way, state counts cannot be treated
687 as an intrinsic property of the task. Instead, the
688 partitioning procedure that produces those states—
689 and its upstream epistemological commitments—
690 becomes the central object of analysis. Complex-
691 ity derived from state statistics is thus observer-
692 relative, and cross-observer comparisons lose a
693 clear meaning. Consequently, any benchmark in-
694 tended to be solvable by state-tracking algorithms
695 alone must eliminate this relativity by fixing the vo-
696 cabulary relative to which states are individuated.

697 Fixing a vocabulary and its update rules effec-
698 tively encodes an algorithm. Each benchmark spec-
699 ifies, through its verb inventory and the semantics
700 of those verbs, a procedure for computing state
701 transitions. Under such designs, success reduces
702 to recovering this algorithm from narrative input
703 and executing it correctly. ToM evaluation in these
704 settings becomes a form of algorithm recovery.

705 The extent to which world knowledge partici-
706 pates in this recovery process admits a hierarchy.
707 At the minimal end, the algorithm is recoverable
708 from syntax alone. For example, “Mary moves the
709 ball to the basket” encodes agent, object, and desti-
710 nation in its argument structure; a syntactic parse
711 can populate the slots of an update function with-
712 out requiring knowledge of what moving, balls, or
713 baskets are. Sally–Anne benchmarks often operate
714 near this regime: verb inventories are constrained,
715 syntactic frames are uniform, and the algorithm
716 reduces to maintaining a record of which agents
717 were present when which verbs applied to which
718 arguments.

719 One step beyond syntax, the algorithm requires
720 lexical semantics. Many subjective predicates im-
721 plicitly parameterize a continuous internal degree
722 whose value is mediated by external context. Mod-
723 eling such verbs may therefore require represent-
724 ing latent scalar state variables and updating them
725 when relevant external events occur. Because exter-
726 nal events can continuously modulate these latent
727 degrees, reliable inference requires explicit track-
728 ing of their event-conditioned updates rather than
729 recovering them from narrative surface form alone.
730 More broadly, lexical semantics can introduce ad-
731 ditional latent mental variables. For instance, to
732 properly compute *hide*-related conditions, one may
733 need to track an implicit variable capturing an
734 agent’s aversion to being discovered, maybe even
735 by a specific person. Such transition semantics are
736 not recoverable from syntax alone and often require
737 simulating a world model that includes variables
738 not explicitly present in the predicate parameters.
739 Selecting such variables depends on inductive bi-
740 ases that are not resolved by the more rudimentary
741 structural biases discussed above (tree structure,
742 fixed arrow of time, syntax).

743 Further along the hierarchy, the algorithm re-
744 quires compositional integration of long-tail world
745 knowledge embedded in narrative form. As with in-
746 terpreting another agent’s interpretation of Picasso,
747 the same syntactic frame—with identical grammat-
748 ical roles—can demand different transition seman-
749 tics depending on what the model knows about
750 domains not explicitly mentioned in the narrative.
751 A character’s reaction may depend on biological
752 inheritance, professional norms, or cultural and
753 historical conventions that no verb encodes. Com-
754 puting the correct state transition then requires rec-
755 ognizing the relevance of such facts and retrieving
756 them from outside the text.

This hierarchy suggests two directions for benchmark evolution.

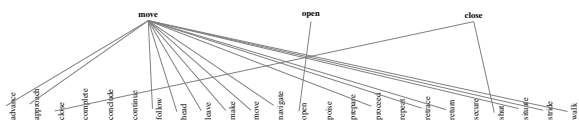


Figure 4: 17 out of 24 verbs that appeared in MMTom are functionally equivalent to 1 of the 3 PDDL actions defined for the benchmark

First, benchmarks can vary the algorithm associated with a verb across instances. In Sally–Anne-style action sequences, verbs such as “move” typically correspond to a single, uniform update rule (e.g., updating spatial containment). Moreover, many benchmarks admit substantial lexical variation without introducing new transition dynamics: for example, although “leave” differs from “move” in surface form, both effectively denote a change in an agent’s occupied location and therefore reduce to the same underlying state update. Figure 4 illustrates this collapse in MMTom, where 17 of 24 verbs ($\approx 70\%$) are functionally equivalent to one of three predefined PDDL actions, while the rest can be safely ignored without compromising the causal fidelity of the parsed action chain. This kind of synonymy lowers the burden of algorithm recovery once a small set of transition rules is learned. A more resistant design would break this uniformity by allowing the same surface verb to realize different transition functions across contexts: “move” in one scenario could update spatial containment, while “move” in another (e.g., chess) updates strategic position. Under identical syntax, the correct algorithm would then be context-dependent, and a system that has learned one interpretation of “move” would not transfer without first identifying which interpretation the current scenario requires.

Second, benchmarks can expand the world model so that only systems with human-level integration of long-tail knowledge can answer the questions posed. Certain benchmarks are solvable by symbolic methods because the relevant domain can be fully specified in advance: a PDDL file enumerates predicates, verbs, and update rules. A benchmark that intentionally exceeds this closure would be less amenable to symbolic assistance. BigToM (Gandhi et al., 2023) is one example, where language models generate novel instances using long-tail knowledge. However, BigToM is now saturated, with Gemini 2.0 scoring 86.33%

and GPT-4o scoring 82.42% (Kim et al., 2025). In benchmarks of this kind, correct transitions may depend on facts about genetics, economics, or social convention that cannot be exhaustively enumerated in a domain file. Only a system that has acquired broad world knowledge—a language model or a human—can reliably recognize the relevance of such facts and retrieve them appropriately.

C Prompts

We list the domain file (Listing 1) and the prompt for LLMs (Listing 2, 3, 4) here. Note that for the domain file, we write specific simulator for the “forall” operation as effects, which is not supported even by the latest PDDL.

Listing 1: PDDL domain used in PDDL-MIND

```
(define (domain tom_general)
  (:requirements :strips)
  (:types agent obj loc utterance) ; agent,
  object, location and utterance

  (:predicates
    ;; Required for all scenarios, if for
    communication data, create an abstract
    location (e.g. conversation) for everyone
    present
    (at ?a - agent ?l - loc) ; agent a
    is at location l.

    ;; Physical movement, ONLY use when the data
    is about MOVING agents
    (is-open ?c) ; c is
    open, c is a container, which can be an
    object or a location.
    (is-loc ?c) ;
    container c is a location (e.g. a room).
    (in ?o - obj ?c) ; object
    o is in container c, which can be an object
    or a location, do NOT declare object X on
    some other object Y, instead, declare object
    X "in" the same container as object Y.
    (holding ?a - agent ?o - obj) ; agent
    is holding object o.
    (seen ?a - agent ?o - obj ?l - loc?)
    ; agent a has seen object o at location l.

    ;; Communication tracking, ONLY use when
    agents have active conversations
    (said ?a - agent ?u - utterance) ; agent
    a said utterance u.
    (heard ?a - agent ?u - utterance) ; agent
    a heard utterance u.
  )

  ;; Physical movement, ONLY use when the data
  is about MOVING agents
  ;; agent moves from a start room to a dest
  room, sees all objects visible in the
  destination room
  (:action move
  :parameters (?a - agent ?start - loc ?dest -
  loc)
  :precondition (at ?a ?start)
```

```

862 :effect (and
863   (forall (?o - obj)
864     when (in ?o ?dest)
865       (seen ?a ?o ?dest)
866   )
867 )
868 )
869
870 ;; Physical movement, ONLY use when the data
871 is about MOVING agents
872 ;; open a container c at location l, the
873 container can be an object or a location.
874 (:action open
875 :parameters (?a - agent ?c ?l - loc)
876 :precondition (and
877   (at ?a ?l)
878   or
879   (in ?c ?l)
880   (is-loc ?c)
881 )
882 )
883 :effect (and
884   (is-open ?c)
885   (forall (?o - obj)
886     when (in ?o ?c)
887       (seen ?a ?o ?l)
888   )
889 )
890 )
891
892 ;; Physical movement, ONLY use when the data
893 is about MOVING agents
894 ;; close a container c at location l, the
895 container can be an object or a location.
896 (:action close
897 :parameters (?a - agent ?c ?l - loc)
898 :precondition (and
899   (at ?a ?l)
900   (is-open ?c)
901   or
902   (in ?c ?l)
903   (is-loc ?c)
904 )
905 )
906 :effect (not (is-open ?c))
907 )
908
909 ;; Physical movement, ONLY use when the data
910 is about MOVING agents
911 ;; grab an object o at location l
912 (:action grab
913 :parameters (?a - agent ?o - obj ?l - loc)
914 :precondition (and
915   (at ?a ?l)
916   (in ?o ?l)
917   (not (holding ?a ?o))
918 )
919 :effect (and
920   (holding ?a ?o)
921   (not (in ?o ?l))
922   (seen ?a ?o ?l)
923 )
924 )
925
926 ;; Physical movement, ONLY use when the data
927 is about MOVING agents
928 ;; drop an object o at location l
929 (:action drop
930 :parameters (?a - agent ?o - obj ?l - loc)
931 :precondition (and

```

```

(at ?a ?l)
(holding ?a ?o)
)
)
:effect (and
(not (holding ?a ?o))
(in ?o ?l)
)
)
)
)
;; Communication tracking, ONLY use when
agents have active conversations
;; agent a tells others an utterance u
(:action tell
:parameters (?a - agent ?u - utterance)
:precondition ()
:effect (and
(said ?a ?u)
;; Hear of other agents will be handled in
simulation
)
)
)
)
;; Communication tracking, ONLY use when
agents have active conversations
;; agent hears an utterance u at location l (
while present), can be triggered by action
tell
(:action hear
:parameters (?a - agent ?l - loc ?u -
utterance)
:precondition (and (at ?a ?l))
:effect (and (heard ?a ?u))
)
)
)
)

```

932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965

Listing 2: Prompt template used for PDDL problem file generation

```

The domain types and predicates definition for
the current environment:
[[domain_type_pred]]
Given the above domain types and predicates,
please write the PDDL problem file for the
following environment.
All rooms (locations) are seen.
All objects are NOT seen.
Please list the objects (with their types) and
the initial state faithfully, but only a
dummy goal state like (:goal ).
Environment:
[[environment]]

```

967
968
969
970
971
972
973
974
975
976
977
978
979
980
981

Listing 3: Prompt template used for PDDL action extraction

```

Domain:
[[domain]]
Initially, the environment states are
[[init_state]]
Parse the following actions including the
ongoing actions into PDDL format given the
domain PDDL.
Only take the single-agent perspective actions:
move, open and close in the domain PDDL.
Only use the objects provided.
Format the final PDDL action list as

```

983
984
985
986
987
988
989
990
991
992
993
994
995
996

```

997 "Executed Actions:
998 (action1 ...)
999 ...
1000 Ongoing Actions:
1001 (action2 ...)
1002 ..."
1003
1004 [[action]]

```

Listing 4: Prompt template used for ToM question answering

```

1006 [[names]] was/were in an environment with these
1007 actions allowed: [[domain_action_name]]. [[
1008 action]]
1009
1010 We break down the actions taken by [[names]]
1011 with location, the seen objects lists, and
1012 heard utterance lists.
1013 [[act_state_seq]][[ongoing_prompt]]
1014
1015 [[question]]
1016 [[notes]]
1017
1018 Please answer based on the above information.
1019 [[final_format]]
1020

```

1022 D Experiment Details

1023 D.1 Implementation Details

1024 For all models and all benchmarks, we use a temper-
1025 ature of 0 and a max token of 4,096 for LLM infer-
1026 ence. To avoid data leakage, we stratified the sam-
1027 ple 50 examples per sub-category on each bench-
1028 mark, such that we exclude the examples used to
1029 design PDDL domain file and prompt templates.

1030 D.2 Error Analysis

Listing 5: Example on failed example in MMToM-QA with reasoning errors.

```

1031 Narratives:
1032 What's inside the apartment: The apartment
1033 consists of a bedroom, a bathroom, a living
1034 room, and a kitchen.
1035 In the bedroom, there is a coffee table with a
1036 dish bowl, a remote control, a plate, and a
1037 water glass on it.
1038 The bathroom houses a cabinet, which is
1039 currently empty.
1040 The living room is furnished with a cabinet, a
1041 coffee table, a sofa, and a desk. The
1042 cabinet contains three apples, a condiment
1043 bottle, three water glasses, a remote
1044 control, a wine glass, and a dish bowl. A
1045 water glass is placed on the coffee table.
1046 The kitchen is equipped with a fridge, a stove,
1047 a kitchen table, and a microwave. Inside the
1048 fridge, there are two dish bowls and a
1049 bottle of wine. The stove contains four
1050 plates, a salmon, and a cupcake. The
1051 microwave holds a salmon, a condiment bottle
1052 , and a bag of chips.
1053

```

```

Actions taken by Karen: Karen is situated in the
kitchen. She strides towards the
refrigerator, opens it, and subsequently
closes it. Following this, she opens the
microwave and then securely shuts it.
-----
Predicted action sequence:
(move Karen kitchen fridge)
(open Karen fridge kitchen)
(close Karen fridge kitchen)
(open Karen microwave kitchen)
(close Karen microwave kitchen)
-----
Verified action sequence:
Initially, Karen is situated in the kitchen
Seen objects: fridge, kitchen_table, microwave
Heard utterance: N/A
1. (open Karen fridge kitchen)
Seen objects: fridge, kitchen_table, microwave,
dish_bowl_3, dish_bowl_4, wine_bottle
Heard utterance: N/A
2. (close Karen fridge kitchen)
Seen objects: fridge, kitchen_table, microwave,
dish_bowl_3, dish_bowl_4, wine_bottle
Heard utterance: N/A
3. (open Karen microwave kitchen)
Seen objects: fridge, kitchen_table, microwave,
dish_bowl_3, dish_bowl_4, wine_bottle,
salmon_2, condiment_bottle_2, chips.
Heard utterance: N/A
4. (close Karen microwave kitchen)
Seen objects: fridge, kitchen_table, microwave,
dish_bowl_3, dish_bowl_4, wine_bottle,
salmon_2, condiment_bottle_2, chips.
Heard utterance: N/A
-----
Question:
Which one of the following statements is more
likely to be true? (a) Karen has been trying
to get a bag of chips. (b) Karen has been
trying to get a remote control.
-----
Reasoning & Answer:
Karen interacted with the fridge and the
microwave, both of which revealed the
presence of chips upon opening the microwave
. Since she did not pick up the chips after
seeing them, it suggests she might be
looking for something else. However, she did
not interact with any other objects or
containers that might suggest a search for a
remote control. Therefore, the presence of
chips and the lack of any interaction with
objects related to a remote control make

```

option (a) more likely. The answer is a.

Listing 6: Example on failed example in MMToM-QA with parsing errors.

Narratives:
What's inside the apartment: The apartment consists of a bedroom, kitchen, living room, and bathroom.
In the bedroom, there is a coffee table and a desk
The kitchen is equipped with four cabinets, a fridge, a microwave, a kitchen table, and a stove. The fourth cabinet from the left houses two water glasses, while the second and third cabinets are empty. The first cabinet contains a water glass, a dish bowl, and a condiment bottle. Inside the fridge, there is a plate and an apple, and a cupcake is in the microwave. The stove contains a plate and a piece of salmon.
The living room features a cabinet, a coffee table, a sofa, and a desk. The cabinet is filled with two books, a water glass, a bag of chips, a bottle of wine, two remote controls, two apples, a wine glass, and a condiment bottle. The coffee table holds two dish bowls and a water glass.
The bathroom has a cabinet, which is currently empty.
Actions taken by David: David is situated in the kitchen. He strides towards the fourth kitchen cabinet, opens it, subsequently closes it, and is now preparing to open the third kitchen cabinet.

Predicted action sequence:
(move David kitchen cabinet_kitchen_4)
(open David cabinet_kitchen_4 kitchen)
(close David cabinet_kitchen_4 kitchen)

Ongoing action sequence:
(open David cabinet_kitchen_3 kitchen)

Verified action sequence:
Initially, David is situated in the kitchen
Seen objects: stove
Heard utterance: N/A

1. (move David kitchen cabinet_kitchen_4)
Seen objects: stove, water_glass_1, water_glass_2
Heard utterance: N/A

[[Note that cabinet_kitchen_4 is parsed as a location with (is-loc cabinet_kitchen_4), making the action sequence stops at 'move', also cabinet_kitchen_1, cabinet_kitchen_2, cabinet_kitchen_3, cabinet_kitchen_4 are not listed are seen objects due to this.]]

Question:
Which one of the following statements is more likely to be true? (a) Karen has been trying to get a bag of chips. (b) Karen has been trying to get a remote control.

Reasoning & Answer:
David is searching through the cabinets, which suggests he believes the bottle of wine might be in one of them. The answer is a.

E Checklist Information

Potential risks. Our method conditions LLMs on a predefined PDDL domain file that specifies permissible actions and state transitions. While this design improves state-tracking reliability, it also introduces potential risks. Broadly, providing structured domain knowledge to LLMs raises the risk that reported performance improvements reflect alignment with the predefined schema rather than intrinsic gains in reasoning ability. To mitigate this, we release the full domain specification and prompts, and encourage future work to evaluate robustness under alternative or extended domain definitions.

Data and model use. We use public dataset MMToM-QA and MuMa-ToM, which are both under MIT license. Both datasets are anonymized and contain no human private data. GPT-4o (OpenAI) and Claude-4.5-Sonnet (Anthropic) are accessible exclusively via commercial APIs. These systems are governed by restrictive Terms of Service that prohibit model weight extraction, reverse engineering, and competitive distillation, serving to protect their respective architectural innovations and agentic harnesses. Kimi-K2-Instruct operates under a Modified MIT License; this variant permits general commercial use but mandates strictly visible attribution for entities exceeding 100 million monthly active users or \$20 million in monthly revenue.

LLM use. We use LLMs to generate PDDL problem files and action sequences, to answer theory-of-mind questions in our experiments, and to improve writing fluency and correct grammatical errors.