A Predicate-based View on Theory of Mind Benchmarks

Anonymous EMNLP submission

Abstract

Common diagnosis of model errors on Theory-of-Mind benchmarks separates such tasks into three components: Parsing the narrative to locate relevant parts; tracking how the world changes; and reasoning about minds within that changing world. This separation is meaningful only if the narrative's state-transition rules can be recovered and executed independently. This paper argues that, for a broad class of textual ToM benchmarks, these rules are largely determined by the benchmark's verb inventory. A benchmark's verbs define its state-transitions: preconditions and effects of verb events. We propose PDDL-MIND, which recovers these transitions by parsing text narratives into the Planning Domain Definition Language and verifying candidate actions against a predefined predicate domain. The verified state sequence provides an executable and modular diagnostic for whether a benchmark's distinction between physical state tracking and mentalistic reasoning is shallow. On MMToM-QA, 17 of 24 of the benchmark's verbs functionally map onto three predicates. Conditioned on the verified states, a language model reaches roughly 88% accuracy on MMToM-QA and MuMa-ToM using three model calls per example.

1 Introduction

A Theory-of-Mind(ToM) benchmark is built to measure whether a system can reason about the minds of others. ToM benchmarks typically take one of two forms: descriptions of *physical actions* in an environment, or *dialogue transcripts* between agents. These benchmarks serve as compact tests of whether a reasoner distinguishes the true world state from an agent's belief state (Wimmer and Perner, 1983; Baron-Cohen et al., 1985). Since 2023, researchers have adapted these false-belief tests to large language models (LLMs). Early results suggested that newer LLMs could solve many classic false-belief tasks (Kosinski, 2024), but follow-up studies found that performance could be brittle under small task variations and prompt changes (Ullman, 2023; Ma et al., 2023).

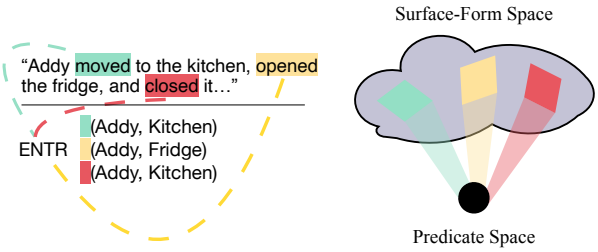

Figure 1: L: Linguistically distinct event descriptions, such as *moved*, *opened*, and *closed*, can induce a small set of executable predicate updates. R: Predicate collapse from surface forms to state transitions.

To test their belief and counterfactual reasoning ability, recent benchmarks add longer event chains, richer environments, more agents, dialogue, multimodal inputs, and more varied questions (Sileo and Lernould, 2023; Jin et al., 2024; Kim et al., 2025). On these newer tasks, even frontier LLMs remain substantially below human level (Sileo and Lernould, 2023; Jin et al., 2024; Kim et al., 2025). Various scaffolds were proposed to address this gap (Jin et al., 2024; Kim et al., 2025).

Broadly, multiple competing hypotheses for the cause of this gap have emerged. One hypothesis attributes errors to *environment and event parsing*: the model may fail because it misreads who is present, which object moved, or which utterance was heard. The second attributes errors to *physical state tracking*: given a fixed set of transition rules, models fail to internally maintain persistent representations of world states as events accumulate over time. The third attributes errors to *downstream 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). All hypotheses implicitly target the

model. We wish to nudge the community to turn instead to the benchmark, an instrument that can be examined on its own terms. We ask a retroactive question: **Given the taxonomy of hypothetical failure modes, to what extent do ToM benchmarks separate the capacities the community purports them to measure?**

If state tracking or parsing are model abilities that can succeed or fail on in isolation assumes the function carrying out the state transitions is fixed, with scope of this fixation ranging from timesteps within an instance to entire benchmarks. For theory of mind, there is a further assumption that this function can be extracted from the narrative and executed on its own, apart from any inference about belief. Previous literature established this abstraction: each ToM narrative is fundamentally a sequence of *events* that induces a sequence of world states (Huang et al., 2024a; Xu et al., 2025).

However, current literature leaves the notion of an *event* unanalyzed beyond surface segmentation. We argue that a deeper understanding on the nature of events permits proper extraction of modular and executable state transition functions. 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 stable abstraction that diverse surface forms can converge on.

We introduce PDDL-MIND, a framework based on predicate definition that provides reliable and explicit state-tracking mechanisms for ToM narratives. Starting from a predefined *Planning Domain Definition Language* (PDDL; McDermott 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.

PDDL execution then verifies which actions are valid and produces an explicit state trace. 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.

This decomposition lets us test the three hypotheses directly. If parsing is enough, then a model given structured actions but no verified state trace should already reach high performance. If state tracking is the main bottleneck, then PDDL verification should create a large gain. If belief reasoning is the main bottleneck, then large errors should remain even after state tracking is verified. The predefined domain is crucial for this diagnosis because it isolates the transition rules to be modular and executable. The model is restrained from inventing the world model it will later use

Our experiments support both the parsing and state-tracking hypotheses. On MMToM-QA, MuMa-ToM, and FanToM, valid parsing alone brings performance close to several agentic ToM pipelines. However, verified state tracking yields another large improvement, moving GPT-4o to 88.3%, 88.8%, and 80.0% accuracy, respectively, and approaching human performance on the former two benchmarks.

These results provide strong evidence that benchmarks tested potentially emits shallow distinction between state tracking and downstream reasoning. We argue that PDDL-MIND should be used as a self-diagnostic protocol for future ToM benchmarks. If proper state tracking scaffolds alone provides strong performance gains, the benchmark tested potentially conflates physical state tracking with general ToM capabilities. If errors remain under verified states, this is stronger evidence of belief or counterfactual reasoning failures. This distinction can help benchmark creators design tasks that target the capability they intend to measure.

2 PDDL-MIND

We introduce PDDL-MIND, a neuro-symbolic framework that decouples environment state tracking from belief inference by explicitly modeling state transition functions (Figure 2).

Predefined domain file. We begin with a generic, predefined PDDL domain file meant to capture the transition types that recur across current ToM benchmarks, not to model every detail of a possible world. Physical-action benchmarks require agents, locations, objects, containers, visibility, and manipulation. Dialogue benchmarks require agents, utterances, conversational presence, and memory.

We therefore define four object types: `agent`, representing people mentioned in the narrative; `loc`, representing locations; `obj`, representing physical items; and `utterance`, representing spoken messages. We define predicates such as `at`, `in`, and `holding` for physical state. We also define `seen`, which records whether an agent has observed an

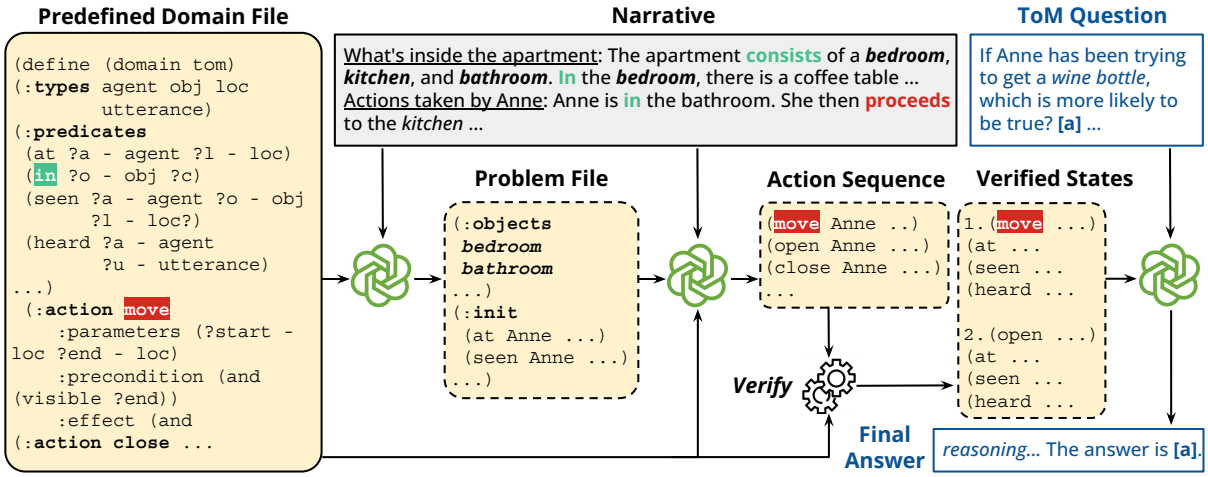

Figure 2: Starting from a highly generic, predefined PDDL domain file, PDDL-MIND parses the narrative into an initial world state and a sequence of candidate actions. PDDL verification executes only actions whose preconditions hold, producing a reliable state trace for final Theory-of-Mind question answering.

Method	MMToM-QA	MuMa-ToM	FanToM
BIP-ALM*	56.2	33.9	-
LIMP*	55.3	76.6	-
TT*	69.0	-	54.7
AutoToM*	83.0	81.4	-
EnigmaToM*	-	-	55.3
S^{3}AP*	69.2	-	49.1
Human*	82.5	93.5	87.5
PDDL-MIND w/o predefined domain	74.9	78.9	69.7
PDDL-MIND w/o verification	77.4	79.6	74.7
PDDL-MIND	**88.3**	**88.8**	**78.5**

Table 1: Main diagnostic results. The no-domain ablation tests whether LLMs can induce the transition rules; the no-verification ablation tests structured parsing without verified state execution. * denotes results from previous papers. The S^{3}AP row reports GPT-4o results with the primary o3 parser from Social World Models.

object, and `heard`, which records whether an agent has heard an utterance. Finally, we define seven actions: `move`, `open`, `close`, `grab`, `drop`, `ask`, and `tell`. Each action has explicit parameters, preconditions, and postconditions. The complete domain file is provided in Appendix A. We choose a predefined domain for two reasons. First, it makes the benchmark's world model inspectable: a reader can see exactly which transition rules are being assumed. Second, it prevents the diagnostic from mixing two questions that should be separated: whether the LLM can parse a narrative into a known formal schema, and whether the LLM can invent a correct formal schema in the first place. The latter is an important planning problem, but it falls outside the scope of this paper, which focuses on benchmark diagnostics.

State and action extraction. Given the domain file, we prompt an LLM to parse the narrative into a PDDL problem file. The problem file specifies the scenario objects and initial state s_0. We define a state s as a set of grounded predicates, such as `(at Mary bathroom)`. This stage tests environment parsing: the model must identify the entities and initial relations that the story assumes. The LLM then parses the narrative into a sequence of candidate actions $(a_1, \ldots, a_t)$. This stage tests parsing: We allow the candidate sequence to contain invalid or irrelevant actions. The model must map language such as "Mary walks to the kitchen" or "John says that the key is in the drawer" into grounded PDDL actions.

Verified state tracking. For each candidate action a_i, we check whether the current state s_{i-1} satisfies the action's preconditions. If the preconditions hold, we execute the action's postconditions and produce s_i. If the preconditions fail, the action is ignored and the state remains unchanged. This

procedure gives us a verified sequence of actions and world states. The verification stage tests the state-tracking hypothesis: if state tracking is the bottleneck, providing the verified state trace should substantially improve ToM question answering.

ToM question answering. Finally, the LLM receives the verified state sequence and a ToM question, and answers with chain-of-thought reasoning. Residual errors given the correct states therefore provide a cleaner signal that seperates failure on belief reasoning, counterfactual reasoning, or answer selection.

Diagnostic variants. We evaluate two ablations to isolate the role of each component. PDDL-MIND without verification gives the LLM the parsed symbolic events but does not provide the verified intermediate states, measuring how far valid parsing goes without reliable state execution. PDDL-MIND without a predefined domain asks the model to construct its own domain before parsing, measuring whether failures come from inducing the world model rather than using it.

3 Experiment results

Existing ToM Benchmarks can be categorized by the amount of participants involved in a single narrative. Dialogue-based narratives are inherently multi-agent, while action-based narratives can be either multi and single agent. The three benchmarks examined in this paper were curated to represent a balanced set of heuristics. MMToM-QA (Jin et al., 2024) contains 600 questions about belief and goal inference from long household single agent action sequences. FanToM (Kim et al., 2023) is pure multi-agent dialogue-based, containing information-asymmetric conversations with questions about answerability and accessibility. While MuMa-ToM (Shi et al., 2025) is a mixture of multi-agent dialogue and action. All tasks are evaluated as multiple-choice question answering by accuracy.

We do not treat our coverage as exhaustive. Datasets such as ToMi and BigToM remain important, but prior work reports that several older or more templated ToM settings are close to saturated under chain-of-thought prompting (Xu et al., 2025; **?**). Such saturation leaves little headroom for diagnosing whether failures come from parsing, state tracking, or belief reasoning.

We compare against BIP-ALM (Jin et al., 2024) and LIMP (Shi et al., 2025), which combine Bayesian inverse planning with language models; ThoughtTracing (TT) (Kim et al., 2025), which generates and weights hypotheses about mental states; AutoToM (Zhang et al., 2026), which iteratively constructs agent models for ToM inference; and EnigmaToM (Xu et al., 2025), which tracks events with a neural knowledge base; S^3AP (Zhou et al., 2026), which implements stateful prompting with states derived by a Language Model. We use GPT-4o (OpenAI, 2024) for the main PDDL-MIND results.

Parsing helps, but does not solve ToM. Table 1 merges the main result table with the two diagnostic ablations. PDDL-MIND without verification reaches 77.4% on MMToM-QA, 79.6% on MuMa-ToM, and 74.7% on FanToM. This is already close to several agentic pipelines, despite using only structured parsing and no verified state trace. The result demonstrate that a large part of the improvement available to ToM systems comes from turning event narratives into structural representation.

Verified state tracking closes most of the gap. Full PDDL-MIND reaches 88.3% on MMToM-QA, 88.8% on MuMa-ToM, and 80.0% on FanToM. This moves performance near human accuracy on MMToM-QA and MuMa-ToM and exceeds the strongest prior method reported for each benchmark in Table 1. The jump from parsed actions to verified states is especially diagnostic as the model is benefiting from an executable state trace whose transitions have been checked against preconditions and postconditions.

FanToM's All metric reveals a separate consistency capability. On FanToM, the 80.0% we report is a per-question average over the Accessibility and Answerability subtypes. The benchmark also defines an All metric that requires a model to answer *every* question type about the same conversation correctly before that context counts as solved (Kim et al., 2023). The All metric is stricter than the per-question average because it tests answer consistency across question types in addition to per-question correctness: a model might fail the All metric while scoring well on the average. Humans reach 87.5% on the per-question average but score noticeably lower under All, and the residual gap between PDDL-MIND and human accuracy on FanToM should be read in this light. Reliable

Model	Method	# Calls	MMToM-QA			MuMa-ToM				FanToM		
			Avg.	Belief	Goal	Avg.	Belief	Goal(S)	Goal(B)	Avg.	Accessibility	Answerability
Kimi-K2-Instruct	CoT	1	54.2	70.6	42.0	58.9	63.3	66.7	48.0	46.4	43.8	26.1
	DWM	4	31.7	35.3	29.0	54.6	58.9	57.8	58.9	69.8	73.4	61.6
	PDDL-MIND	3	**74.5**	**80.3**	**68.7**	**77.3**	**67.0**	**85.2**	**81.8**	**80.0**	**73.4**	**77.8**
GPT-4o	CoT	1	61.4	85.3	43.5	41.8	57.8	31.1	37.0	45.3	27.1	25.9
	DWM	4	36.0	46.0	28.5	63.2	76.7	53.3	60.0	70.9	78.5	70.6
	PDDL-MIND	3	**88.3**	**91.0**	**85.7**	**88.8**	**88.7**	**92.0**	**83.3**	**78.5**	72.9	**76.6**
Claude-Sonnet-4.5	CoT	1	69.4	93.3	51.5	58.6	66.7	44.4	64.0	40.5	27.1	26.0
	DWM	4	66.6	76.6	59.0	86.1	**98.9**	72.2	87.0	61.9	56.7	55.1
	PDDL-MIND	3	**90.9**	**92.7**	**89.0**	**88.6**	95.6	**82.2**	**87.5**	**79.3**	**73.2**	**73.8**

Table 2: PDDL-MIND *v.s* cost-efficient methods. PDDL-MIND consistently outperform on each sub-category of MMToM-QA, MuMa-ToM and FanToM across models, with only 3 LLM calls per instance

state tracking removes most of the per-question gap, but the All metric measures a different capability: maintaining a coherent answer policy across multiple questions about the same mental-state configuration. This is closer to belief-reasoning consistency than to state tracking, and is precisely the kind of residual capability that our diagnostic framing aims to expose.

Inducing the world model is a separate bottleneck. PDDL-MIND without a predefined domain performs worse than the no-verification variant on all three benchmarks. This means that asking the LLM to invent the PDDL transition system is harder than asking it to parse into a fixed transition system. We interpret this as a world-model induction failure and not direct evidence of a ToM reasoning failure. For benchmark analysis, this distinction matters as a task may be hard because the relevant transition rules are under-specified.

Manual analysis separates parsing/state errors from reasoning errors. We manually inspect 20 failure cases on each of MMToM-QA and MuMa-ToM for the full GPT-4o pipeline. Following the diagnostic framing, we merge parsing and state-tracking failures into one category: these are cases where the final answer was wrong because the symbolic trace was wrong or incomplete. The second category contains belief and counterfactual reasoning failures given a correct trace. Out of these 40 inspected failures, 13 cases (33%) are parsing/state-tracking errors and 27 cases (67%) are belief or counterfactual reasoning errors. Because the total error rate of full PDDL-MIND is about 12%, the reasoning-only gap after perfect parsing and tracking would be roughly 8% on these benchmarks. The two failure modes have distinct signatures. A representative parsing failure on MMToM-QA (Listing 6) involves a narrative in which David moves to the fourth kitchen cabinet, opens it, closes it, and then begins opening the third cabinet. The parser misclassifies `cabinet_kitchen_4` as a location rather than an object, asserting `(is-loc cabinet_kitchen_4)` in the initial state. This violates the preconditions of `open` on a container, so the verifier silently halts the trace after the `move` step. The downstream model then answers the goal-inference question from an impoverished state. A representative reasoning failure (Listing 5) shows the opposite pattern: the trace is correct — Karen opens the fridge and microwave, and the verifier correctly registers the chips revealed inside the microwave — but the model infers from "saw chips but did not take them" that Karen *was* searching for chips, when the standard goal-inference reading is that an agent who passes over a visible object is unlikely to want it. Together, these examples illustrate the two error modes whose ratio our diagnostic aims to isolate. Further cases in Appendix B.2.

Generalizability across models and categories. Table 2 compares PDDL-MIND with cost-efficient baselines across model backbones and question categories. PDDL-MIND consistently improves over chain-of-thought and DWM across MMToM-QA, MuMa-ToM, and FanToM. The gains hold for Kimi-K2-Instruct, GPT-4o, and Claude-Sonnet-4.5, although weaker models show more parsing errors and stronger models leave less room for improvement in saturated categories. The main exception is MuMa-ToM belief questions with Claude-Sonnet-4.5, where both DWM and PDDL-MIND exceed 95% accuracy and DWM is modestly higher. A small number of parsing errors can dominate the residual difference.

4 Inductive-biases of Benchmarks make PDDL diagnostics effective

Figure 3 summarizes the underlying structure of ToM benchmark instances through three complementary views, with colors denoting actions.

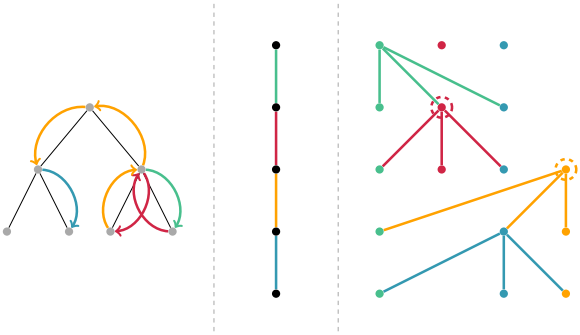

Figure 3: L: Actions M: Events R: Conversations(dashed circle means join/leave)

Single-agent action sequences (left panel). The left panel illustrates a typical single-agent Sally–Anne scenario. The colored events correspond to transitions in the left panel of Figure 3. Such instances admit a hierarchical (tree-structured) interpretation: the environment induces a containment structure. Each action updates either agent's location within this tree world and their perceptual access.

By this view, opening a fridge is functionally identical to entering a room, as state tracking reduces to maintaining the agent's position in this structure and recording visibility at each node. This computationally corresponds to tree traversal.

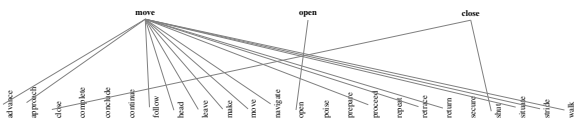

Figure 4: 17 out of 24 verbs that appeared in MMToM are functionally identical to a set of three predicates

This finding identifies a phenomenon demonstrated in Figure 4: *Predicate Collapse*, different verbs functionally mapping onto a single predicate. equipped with this view, we identified significant functional redundancy in MMToM.

Multi-agent conversation (right panel). The right panel extends this structural view to multi-agent settings. We model conversation as broadcast within a shared environment: when an agent speaks, all currently present agents receive and store the utterance. State transitions are therefore defined by how memories evolve over time, independent of the truth value of the utterances. Agents may leave (ceasing to receive updates) or join mid-conversation (receiving only subsequent information).

Conversational interactions are strictly hierarchical given a fixed arrow of time. Thus it can be encoded using predicate-based updates such as SPEAK, HEAR, PRESENT, JOIN, and LEAVE.

A unified event–state abstraction (middle panel). The middle panel merges a chain of states connected by typed events, read from top to bottom. Both physical actions and conversational updates are events with preconditions (who must be present, what must be true) and effects (what predicates change). By committing to a predefined PDDL domain, PDDL-MIND inherits a single executable and modular representation that covers single-agent action sequences, multi-agent action sequences, and multi-agent conversation.

Preconditions and Effects Verify State Tracking. PDDL actions are modular transition functions with explicit preconditions and effects. This lets PDDL-MIND check each candidate event before committing it to the state trace.

If a parser proposes an impossible action, PDDL execution rejects it rather than letting the error silently contaminate downstream reasoning. If an action is valid, its effects are applied deterministically, so the resulting state trace is internally consistent. Without the predefined domain, the model must infer which event occurred and what state transition the event ellicits. Our no-domain ablation shows that this world-model induction step is unreliable. That failure deviates from the target of benchmark diagnostics.

Implications for benchmark structure. Across existing ToM benchmarks, tree-like structure frequently emerges as a convenient abstraction for tracking event effects. This prevalence reflects implicit inductive biases: (i) Sally–Anne-style action narratives assume a hierarchical, containment-based world, and (ii) conversational benchmarks impose a fixed temporal order that yields acyclic physical state transitions. Recognizing these biases, and the fact that trees are a strict subset of graphs, allows us to impose a stronger structural constraint on ToM state-tracking models than prior graph-based formulations such as Xu et al. (2025).

5 Related Work

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: separating state tracking from reasoning. Language models reliably parse into constrained symbolic forms and perform variable binding (Wu et al., 2025; Prakash et al., 2026), 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., 2026) and BIP-ALM (**?**) require LM logits over actions conditioned on hypothesized goals and beliefs, then aggregate via Bayes' rule across timesteps. ThoughtTracing (**?**) 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. Shrinking room for error permits proper capacity disentanglement.

Discrete state methods. Partially sidestepping these hazards, DWM (Huang et al., 2024a), EnigmaToM (Xu et al., 2025), SWM (Zhou et al., 2026) and PDDL-MIND embrace discreteness by parsing natural language into symbolic representations while assuming deterministic transition rules. DWM, EnigmaToM, and SWM place two wagers: that neural models can identify state-relevant events **and** correctly embody the transition rules. Our work shows the second wager can be stripped away by invoking proper event-based inductive biases. Asking a model to "discover" what a benchmark already defines invites hallucination and inconsistency. Aforementioned frameworks also track both explicit physical state updates and implicit mental state updates, which does not properly separate the introduced failure modes. We make this separation precise. Predicates like `at` and `is-open` are physical. Predicates like `seen` and `heard` are quasi-physical: agent-indexed and epistemic, yet updated by a deterministic record of events, since an agent present when a container opens observes its contents, a heard dialogue too gets memorized.

ToM benchmarks. Textual and multimodal ToM benchmarks now test longer and more diverse narratives, including ToMi (Le et al., 2019), BigToM (Gandhi et al., 2023), Hi-ToM (Wu et al., 2023), FanToM (Kim et al., 2023), OpenToM (Xu et al., 2024), NegotiationToM (Chan et al., 2024), MMToM-QA (Jin et al., 2024), and MuMa-ToM (Shi et al., 2025). These benchmarks differ in modality, number of agents, and question types, but they commonly present an ordered sequence of events that changes what agents can observe or remember. This shared structure makes them a natural target for diagnostic state-tracking interventions. At the same time, benchmark saturation matters for diagnosis. Prior work reports near-ceiling performance on several older or more templated ToM datasets with chain-of-thought prompting alone (Xu et al., 2025; Kim et al., 2025), leaving little headroom for identifying which component fails. We therefore evaluate on three benchmarks that still exhibit substantial gaps while spanning the principal task categories: MMToM-QA for single-agent action sequences, MuMa-ToM for multi-agent action and conversation, and FanToM for multi-agent conversation. Our goal is to test a failure-mode hypothesis across representative settings, not to exhaustively cover every ToM dataset.

6 Conclusion

We show that current ToM benchmarks emit a shallow distinction between physical state tracking and mentalistic reasoning. LLM failures on ToM tasks primarily reflect unreliable implicit state tracking rather than deficiencies in downstream reasoning, and that PDDL untangling isolates this bottleneck. We therefore encourage future ToM benchmark designers to treat accurate, verifiable physical state tracking as a prerequisite for meaningful assessment.

Adopting a predicate based view admits further diagnostics. 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.

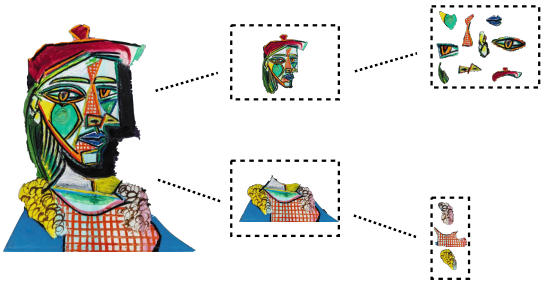

Figure 5: Sample of conceptual partition of *Lover in a Beret*

Inferring artistic intention from a painting is a theory-of-mind task. Consider Picasso's *Lover in a Beret* in Figure 5. If asked to interpret Picasso's intentions, a typical human viewer may first recognize a human face on the painting. This recognition constitutes immediate progress towards attributing Picasso's intention, after which the viewer can further decompose the face into sub-components (e.g., nose, eyes). Now imagine presenting the same painting to an alien with human-comparable intelligence but whose evolution yields no similarity to any face-ful being. If the alien has never encountered a face, it is unreasonable to expect even the shallow interpretations that humans achieve effortlessly. The alien is likely to form a partition of the painting starkly different from Figure 5, relying more on geometric primitives. For humans, the painting has fewer states to track because the viewer's equipment of contingent knowledge compresses geometric objects into a single token.

This thought experiment exposes a limit for all state-counting perspectives on ToM complexity. If the number of states depends on the observer restricted by the long tail of contingent knowledge. Complexity derived from state statistics is thus observer-relative and cannot be treated as an identical property across every ToM task. Instead, the partitioning procedure that produces those states and its upstream epistemological commitments becomes the central object of analysis. **Any benchmark that emits shallow separation between state tracking and mentalistic reasoning eliminates this relativity by fixing the semantics of vocabularies relative to which states are tokenized.**

Fixing a vocabulary and its update rules effectively encodes an algorithm. The aforementioned subset of benchmarks specify, through its verb inventory and the semantics of those verbs, a procedure for computing state transitions. Under such designs, ToM evaluation in these settings becomes a form of algorithm recovery. The extent to which world knowledge participates in this recovery admits a hierarchy. At the minimal end, the algorithm is recoverable from surface forms alone. For example, "Mary moves the ball to the basket" encodes agent, object, and destination in its argument structure. Sally–Anne benchmarks operate near this regime.

One step beyond syntax, many subjective predicates implicitly parameterize a continuous, sometimes probabilistic internal degree whose value is mediated by context beyond surface forms. For instance, to properly compute *hide*-related conditions, one may need to track an implicit variable capturing an agent's aversion to being discovered. including such predicates increases benchmark expressivity. Further along the hierarchy, recovery requires compositional integration of long-tail world knowledge embedded in narrative form. A character's reaction may depend on biological inheritance, professional norms, or cultural and historical conventions. Computing the correct state transition then requires recognizing the relevance of such facts and retrieving them from outside the text.

The hierarchy suggests two directions for benchmark evolution. First, benchmarks can vary the algorithm associated with the same verb across instances. Second, benchmarks can expand their world model to align with systems of human-level integration of knowledge. BigToM (Gandhi et al., 2023) is one example, where language models generate novel narratives 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, only a system that has acquired broad world knowledge—a language model or a human—can possibly recognize the relevance of such knowledge and retrieve them appropriately.

Limitations

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.

Our approach also 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

Simon Baron-Cohen, Alan M. Leslie, and Uta Frith. 1985. Does the autistic child have a "theory of mind"? *Cognition*, 21(1):37–46.

Chunkit Chan, Cheng Jiayang, Yauwai Yim, Zheye 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*.

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, Raghuveer 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 Tenenbaum, and Tianmin Shu. 2024. MMToM-QA: Multimodal theory of mind question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16077–16102, Bangkok, Thailand. Association for Computational Linguistics.

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 *Second Conference on Language Modeling*.

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*, pages 14397–14413.

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

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. Llm+p: Empowering large language models with optimal planning proficiency. *Preprint*, arXiv:2304.11477.

Xiaomeng Ma, Lingyu Gao, and Qihui Xu. 2023. ToM-Challenges: A principle-guided dataset and diverse evaluation tasks for exploring theory of mind. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pages 15–26, Singapore. Association for Computational Linguistics.

Drew McDermott, Malik Ghallab, Adele Howe, Craig Knoblock, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins. 1998. PDDL – the planning domain definition language. Technical Report CVC TR-98-003, Yale Center for Computational Vision and Control.

OpenAI. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Nikhil Prakash, Natalie Shapira, Arnab Sen Sharma, Christoph Riedl, Yonatan Belinkov, Tamar Rott Shaham, David Bau, and Atticus Geiger. 2026. Language models use lookbacks to track beliefs. In *The Fourteenth International Conference on Learning Representations*.

Barry Schein. 2012. *Event Semantics. In Gillian Russell, Delia Graff Fara, ed., The Routledge Companion to Philosophy of Language*, pages 280–294.

Haojun Shi, Suyu Ye, Xinyu Fang, Chuanyang Jin, Leyla Isik, Yen-Ling Kuo, and Tianmin Shu. 2025. Muma-tom: multi-modal multi-agent theory of mind. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press.

Damien Sileo and Antoine Lernould. 2023. Mindgames: Targeting theory of mind in large language models with dynamic epistemic modal logic. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

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

Heinz Wimmer and Josef Perner. 1983. Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children's understanding of deception. *Cognition*, 13(1):103–128.

Yiwei Wu, Atticus Geiger, and Raphaël Millière. 2025. How do transformers learn variable binding in symbolic programs? In *Forty-second International Conference on Machine Learning*.

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.

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

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

Zhining Zhang, Chuanyang Jin, Mung Yao Jia, Shunchi Zhang, and Tianmin Shu. 2026. Autotom: Scaling model-based mental inference via automated agent modeling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Xuhui Zhou, Jiarui Liu, Akhila Yerukola, Hyunwoo Kim, and Maarten Sap. 2026. Social world models. *Preprint*, arXiv:2509.00559.

Wang Bill Zhu, Ishika Singh, Robin Jia, and Jesse Thomason. 2025. Language models can infer action semantics for symbolic planners from environment feedback. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8751–8773, Albuquerque, New Mexico. Association for Computational Linguistics.

Appendix

A Prompts 11

B Experiment Details 12

B.1 Implementation Details 12

B.2 Error Analysis 12

C Ethics and Risks 14

A Prompts

We list the domain file (Listing 1) and the prompt for LLMs (Listing 2, 3, 4) here. We use an ADL-style PDDL action-schema notation as an inspectable transition language. Candidate actions are executed by a deterministic finite-state interpreter that grounds each action, checks types and preconditions, expands quantified conditional effects over the finite object set, applies add/delete effects, and records invalid actions without applying their effects. We use PDDL notation for transparency and reproducibility, but the reported traces are produced by this executor rather than by an off-the-shelf planner.

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)
    :effect (and
      (forall (?o - obj)
        when (in ?o ?dest)
        (seen ?a ?o ?dest)
      )
    )
  )

  ;; Physical movement, ONLY use when the data
    is about MOVING agents
  ;; open a container c at location l, the
    container can be an object or a location.
  (:action open
    :parameters  (?a - agent ?c ?l - loc)
    :precondition (and
      (at ?a ?l)
      (or
        (in ?c ?l)
        (is-loc ?c)
      )
    )
    :effect (and
      (is-open ?c)
      (forall (?o - obj)
        when (in ?o ?c)
        (seen ?a ?o ?l)
      )
    )
  )

  ;; Physical movement, ONLY use when the data
    is about MOVING agents
  ;; close a container c at location l, the
    container can be an object or a location.
  (:action close
    :parameters (?a - agent ?c ?l - loc)
    :precondition (and
      (at ?a ?l)
      (is-open ?c)
      (or
        (in ?c ?l)
        (is-loc ?c)
      )
    )
    :effect (not (is-open ?c))
  )

  ;; Physical movement, ONLY use when the data
    is about MOVING agents
  ;; grab an object o at location l
  (:action grab
    :parameters (?a - agent ?o - obj ?l - loc)
    :precondition (and
      (at ?a ?l)
      (in ?o ?l)
      (not (holding ?a ?o))
    )
```

```
    :effect (and
      (holding ?a ?o)
      (not (in ?o ?l))
      (seen ?a ?o ?l)
    )
  )

  ;; Physical movement, ONLY use when the data
     is about MOVING agents
  ;; drop an object o at location l
  (:action drop
    :parameters (?a - agent ?o - obj ?l - loc)
    :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))
  )
)
```

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]]
```

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
"Executed Actions:
(action1 ...)
...
Ongoing Actions:
(action2 ...)
..."

[[action]]
```

Listing 4: Prompt template used for ToM question answering

```
[[names]] was/were in an environment with these
     actions allowed: [[domain_action_name]]. [[
     action]]

We break down the actions taken by [[names]]
     with location, the seen objects lists, and
     heard utterance lists.
[[act_state_seq]][[ongoing_prompt]]

[[question]]
[[notes]]

Please answer based on the above information.
[[final_format]]
```

B Experiment Details

B.1 Implementation Details

For all models and all benchmarks, we use a temperature of 0 and a max token of 4,096 for LLM inference. To avoid data leakage, we stratified the sample 50 examples per sub-category on each benchmark, such that we exclude the examples used to design PDDL domain file and prompt templates.

B.2 Error Analysis

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

```
Narratives:
What's inside the apartment: The apartment
     consists of a bedroom, a bathroom, a living
     room, and a kitchen.
In the bedroom, there is a coffee table with a
     dish bowl, a remote control, a plate, and a
     water glass on it.
The bathroom houses a cabinet, which is
     currently empty.
```

```
The living room is furnished with a cabinet, a
     coffee table, a sofa, and a desk. The
     cabinet contains three apples, a condiment
     bottle, three water glasses, a remote
     control, a wine glass, and a dish bowl. A
     water glass is placed on the coffee table.
The kitchen is equipped with a fridge, a stove,
     a kitchen table, and a microwave. Inside the
      fridge, there are two dish bowls and a
     bottle of wine. The stove contains four
     plates, a salmon, and a cupcake. The
     microwave holds a salmon, a condiment bottle
     , and a bag of chips.
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.
```

C Ethics and Risks

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.