EnactToM: An Evolving Benchmark for Functional Theory of Mind in Embodied Agents

Gurusha Juneja [*1] **Dylan Lu** [*1] **Saaket Agashe** [1] **Parth Diwane** [1] **Edward Gunn** [2] **Jayanth Srinivasa** [3] **Gaowen Liu** [3] **William Yang Wang** [1] **Yali Du** [2] **Xin Eric Wang** [1]

Abstract

Theory of Mind (ToM), the ability to track others' epistemic state, makes humans efficient collaborators. AI agents need the same capacity in multi-agent settings, yet existing benchmarks mostly test *literal* ToM by asking direct belief questions. The ability act optimally on implicit beliefs in embodied environments, called *functional* ToM, remains largely untested. We introduce `EnactToM`, an evolving benchmark of 300 embodied multi-agent tasks set in a 3D household with partial observability, private information, and constrained communication. Each task is formally verified for solvability and required epistemic depth, and new tasks are generated increase difficulty as models improve. On the hard split, all seven evaluated frontier models score 0.0% Pass^3 on functional task completion, while averaging 45.0% on literal belief probes. Manual analysis traces 93% of sampled failures to epistemic coordination breakdowns such as withheld information, ignored partner constraints, and misallocated messages, providing a concrete target for future work.

1. Introduction

Language Models (LMs) are increasingly deployed as agents in shared embodied settings where effective collaboration requires Theory of Mind (ToM), the ability to model the beliefs, intentions, and observations of others (Premack & Woodruff, 1978; Wimmer & Perner, 1983; Tomasello et al., 2005). However, even when frontier LMs can *report* another agent's mental state when prompted (*literal* ToM (Riemer et al., 2025)), they routinely fail to *act* on that knowledge during grounded multi-agent coordination, demonstrating a lack of *functional* ToM.

An agent lacking functional ToM is prone to severe coordination failures: it may redundantly broadcast already-known information, fail to anticipate collaborators' actions, or remain idle awaiting unnecessary instructions. In the worst case, this inability to infer intent can lead to critically unsafe actions or actively disrupt the very human environments the system was designed to assist.

Existing benchmarks lack three properties needed to measure this gap. *Functional*: success must depend on actions that use partner beliefs, not by simply answering belief questions. *Grounded*: the evaluation must take place in environments where spatial constraints and information asymmetry arise naturally, reflecting the conditions under which deployed embodied agents will need to exhibit ToM in practice. *Saturation-resistant*: as models improve, the evaluation must generate new tasks against remaining failure modes, with guarantees of solvability and required epistemic depth.

We address these gaps with `EnactToM`, an evolving benchmark for functional Theory of Mind in embodied multi-agent settings. Our contributions are: (1) a **300-task benchmark** that scores functional task completion and literal belief probes on the same task instances, isolating the act-vs-report gap on a per-task basis; (2) an **evolving generation pipeline** in which an autonomous coding agent authors PDDL-verified tasks at a target epistemic depth and seeds successive rounds from current frontier-model failures, hardening the pool without modifying the generator; and (3) an **analysis of frontier-model behavior** across seven LMs that decomposes 93% of sampled failures into five recurring epistemic-coordination breakdowns.

On the hard split of `EnactToM`, every one of the seven frontier models we evaluate scores 0.0% Pass^3 for functional task completion. No model coordinates reproducibly across three independent runs, while the same models correctly answer up to 45.0% of literal belief probes on the same tasks. Figure 1 summarizes the full evaluation loop: private observations and constrained communication create the need for functional ToM, formal verification keeps generated tasks solvable and epistemically valid, and the final

[*]Equal contribution [1]University of California, Santa Barbara [2]King's College London [3]Cisco Research. Correspondence to: Gurusha Juneja <gurusha@ucsb.edu>, Xin Eric Wang <ericxwang@ucsb.edu>.

ICML 2026 Workshop on Trustworthy AI for Good.

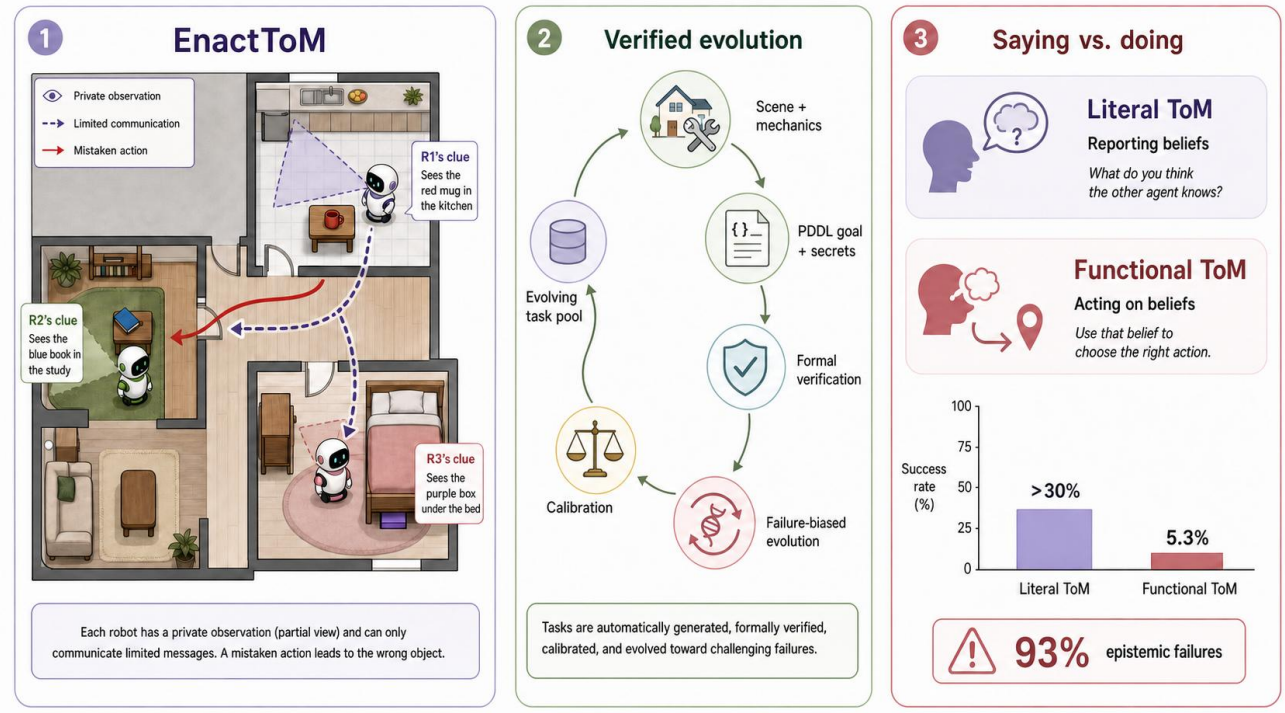

Figure 1. **EnactToM overview. (1) Embodied task:** agents operate in a shared 3D household environment but receive different private observations and can exchange only limited messages; success requires choosing actions that account for what teammates know and do not know. **(2) Verified evolution:** tasks are generated from scenes and mechanics, instantiated as PDDL goals with private secrets, checked for formal solvability and epistemic validity, calibrated, and then evolved from model failures to keep the benchmark difficult. **(3) Saying vs. doing:** the evaluation separates literal ToM (reporting another agent's belief when asked) from functional ToM (using that belief to act correctly), revealing that models can often say what others know while still failing to coordinate.

evaluation compares belief reporting against belief-guided action. Models *report* their partners' knowledge but cannot reliably *use* it to act. The five failure modes catalogued in Section 6.3 are withheld information, broken epistemic chains, ignored partner constraints, misallocated messages, and sabotaged private incentives. These give concrete targets for follow-up work.

2. Related Work

Existing ToM evaluations are text-based and test *literal* ToM only: they prompt agents with explicit questions about others' beliefs and measure whether the answer is correct (Le et al., 2019; Wu et al., 2023; Gandhi et al., 2023; Kim et al., 2023; Sclar et al., 2025; Xu et al., 2024; Jin et al., 2024; Shi et al., 2025; Riemer et al., 2025). FANToM (Kim et al., 2023) and OpenToM (Xu et al., 2024) scale to multi-party conversations and open-ended generation, but the agent is still an observer answering questions, never a participant who must act on inferred beliefs. Recent work confirms a gap between tracking beliefs and acting on them (Zhou et al., 2023; Gu et al., 2026): models that pass false-belief tests can still fail to use that information when choosing actions. No existing benchmark measures this gap in grounded settings where agents must physically coordinate.

Embodied benchmarks (Szot et al., 2021; Chang et al., 2025; Padmakumar et al., 2022; Bard et al., 2020; Zhou et al., 2024) test multi-agent coordination but none formally require epistemic reasoning or verify that task success depends on modeling what partners know. PARTNR (Chang et al., 2025) evaluates collaborative household tasks in Habitat but assumes shared observability and does not introduce information asymmetry. Hanabi (Bard et al., 2020) requires reasoning about hidden information but is a card game without spatial grounding or communication constraints.

Static benchmarks also saturate as models improve (Akhtar et al., 2026; Ott et al., 2022) and suffer contamination when tasks enter training data (Jacovi et al., 2023; Golchin & Surdeanu, 2024). Dynamic approaches like Dynabench (Kiela et al., 2021) and LiveBench (White et al., 2025) address staleness through human-in-the-loop or periodic refreshes, but neither provides formal solvability guarantees or epistemic depth verification. `EnactToM` addresses all three gaps: it is embodied, measures functional ToM through action rather than verbal report, and evolves its task pool to stay ahead of improving models. An extended discussion is in Appendix B; ToM studies from cognitive science are in

Appendix N.

3. Preliminaries

3.1. Functional and Literal Theory of Mind

Literal Theory of Mind evaluations probe an agent's ability to report the beliefs, desires, or intentions of others when asked directly. On the other hand, Functional Theory of Mind requires agents to *act* based on their understanding of others' mental states in order to succeed at a task (Riemer et al., 2025). In our setting, agents operate in a shared 3D environment with private information, limited communication, and physical constraints. Success in this setting depends not on answering questions about others' beliefs but on choosing actions that account for what others know, can observe, and intend to do.

3.2. Epistemic operators and ToM depth

Consider a task in which Agent A places a bowl on a table and Agent B must *confirm* the placement. B can only confirm what it has observed or been told, so the goal is not just that the bowl is on the table but that B *knows* it is. If a third agent C must verify that B knows then it must reason about B's knowledge. Each additional layer demands a deeper level of Theory of Mind.

We formalize this with the epistemic operator $\mathcal{K}$. $\mathcal{K}_i(\phi)$ asserts that agent a_i knows fact ϕ. The assertion is satisfied when a_i has directly observed ϕ or received it through communication. We say that a task requires depth $d-1$ ToM reasoning when the task success depends on the truth value of the operators nested to depth d:

$$\underbrace{\mathcal{K}_{a_1}(\mathcal{K}_{a_2}(\cdots \mathcal{K}_{a_d}(\phi)\cdots))}_{\text{depth } d} \tag{1}$$

At depth 1, an agent needs to learn a fact (zeroth-order). At depth 2, it must reason about what another agent knows (first-order). This connects to level-k reasoning from behavioral game theory (Stahl & Wilson, 1994; Camerer et al., 2004), where a level-k agent selects actions assuming others reason at level $k-1$. Table 1 summarizes the orders used throughout the benchmark.

3.3. Task goals in PDDL

Each task goal is specified in PDDL, combining physical predicates (`is_on_top`, `is_open`) with epistemic $\mathcal{K}$-operators. The $\mathcal{K}$-depth of the goal is computed from the nesting structure. The physical predicates determine task success; the $\mathcal{K}$-operators define literal ToM probes that are evaluated separately at the end of each episode (Section 5). An example goal and the full compilation procedure are in Appendix E.

Table 1. Orders of Theory of Mind used to author and validate `EnactToM` tasks. The reported benchmark caps generated tasks at depth 3.

Order	Pattern	`EnactToM` meaning
0: no ToM	ϕ	Direct physical goal; no partner knowledge matters.
1: self-aware	$\mathcal{K}_a(\phi)$	Notice one's own information gap and obtain or communicate the missing fact.
2: other-aware	$\mathcal{K}_a(\mathcal{K}_b(\phi))$	Act from a model of what a partner knows and still needs to know.
3: recursive	$\mathcal{K}_a(\mathcal{K}_b(\mathcal{K}_c(\phi)))$	Sustain an epistemic relay over who knows that another agent knows.
4+: self-ref.	$\mathcal{K}_a(\mathcal{K}_b(\cdots))$	Deeper belief loops; excluded because coordination becomes brittle even for humans.

3.4. Mechanics

Mechanics model constraints that arise in real coordination. **Room restriction** confines agents to a subset of rooms, like teams operating in different physical spaces. **Limited bandwidth** caps each agent's messages, modeling channels with limited capacity. **Restricted communication** permits messages only along a fixed graph, modeling networks where not every pair of agents can communicate directly. **Remote control**, **state mirroring**, and **inverse state** all model effects the actor cannot observe directly: a trigger object actuating a target across rooms, two objects' states staying synchronized, and an affordance being reversed. The generation agent draws and composes any subset; Appendix J expands the groundings.

4. The EnactToM Framework

Evaluating functional Theory of Mind requires tasks where success depends on agents adapting their actions to the private knowledge, access, and intentions of other agents, not merely reporting beliefs when prompted. Hand-crafting such tasks does not scale, and as models improve, fixed benchmarks saturate. We propose an agentic task generation framework that addresses both problems. An autonomous coding agent authors multi-agent ToM tasks inside a sandboxed workspace, invoking verifiers that ensure each task is logically solvable, physically executable, and genuinely requires epistemic reasoning.

4.1. Task representation

Each task $\mathcal{T}$ is a tuple:

$$\mathcal{T} = \big(\mathcal{S},\ \mathcal{A},\ \varphi,\ \delta,\ \mathcal{M},\ \Sigma,\ \mathcal{C}\big) \tag{2}$$

Algorithm 1 `EnactToM` task generation.

Input: category $\mathcal{C}$, target depth d, seed pool $\mathcal{P}$, seed-task failure ratio ρ
Output: accepted task $\mathcal{T}$
$\mathcal{S} \leftarrow$ `new_scene()`
$\mathcal{P}_{\text{seed}} \leftarrow \textsc{SampleSeedTasks}(\mathcal{P}, \rho)$
$\mathcal{T} \leftarrow \textsc{AuthorTask}(\mathcal{S}, \mathcal{P}_{\text{seed}}, \mathcal{C}, d)$
while `judge`($\mathcal{T}$) or `test_task`($\mathcal{T}$) rejects **do**
  $\mathcal{T} \leftarrow \textsc{ReviseTask}(\mathcal{T})$
end while
return `submit_task`($\mathcal{T}$)

where $\mathcal{S}$ is an HSSD scene (Szot et al., 2021), $\mathcal{A} = \{a_1, \ldots, a_n\}$ is a set of agents with spawn positions, φ is a PDDL goal formula combining physical predicates with $\mathcal{K}$-operators (Section 3.2), δ is a natural-language task description shared with all agents, $\mathcal{M}$ is a set of mechanic bindings drawn from the registry (Section 3.4), Σ maps each agent to private secrets, and $\mathcal{C} \in \{\text{cooperative}, \text{mixed}\}$ is the task category.

Secrets. Each agent receives private secrets $\Sigma(a_i)$: natural-language statements of room restrictions, target object identities, and mechanic hints. Secrets state *what* each agent privately knows but never *how* to coordinate. Figuring out who to tell, what to say, and when to act is the epistemic coordination challenge the benchmark measures.

Categories. Each category targets a different facet of epistemic reasoning. A task is considered solved when all physical predicates in φ are satisfied by the end of the episode.

Cooperative tasks give all agents a shared φ. Agents have different room access and private knowledge. Success requires recognizing what partners cannot observe and communicating accordingly. This tests **epistemic perspective-taking**: can an agent infer what its partner does not know and act on that inference?

Mixed-motive tasks combine a shared cooperative goal with private per-agent side-objectives. The private objectives are not in direct conflict with the shared goal, but they introduce additional coordination demands: agents must allocate limited turns and messages across shared and private tasks, and must reason about whether partners are spending effort on private objectives that could delay the shared plan. This tests **strategic resource allocation under partial information about partner priorities**.

4.2. Task generation agent

We use the `mini-SWE-agent` (Yang et al., 2024) harness in a sandboxed workspace seeded with the scene graph, a blank task template, reference files listing available mechanics, predicates, and actions, and in-context seed tasks drawn from the existing pool with failure ratio ρ (Section 4.4; Appendix F). The agent edits the task file with `bash` and `jq` and invokes five tools: `new_scene`, `bash`, `judge`, `test_task`, and `submit_task`. The only human inputs are the target category $\mathcal{C} \in \{\text{cooperative}, \text{mixed}\}$ and ToM depth $d \in \{1, 2, 3\}$; authoring, evaluation, and revision are otherwise end-to-end.

Authoring order. The agent writes the formal PDDL goal φ first, then derives δ and per-agent secrets $\Sigma(a_i)$ from it. Anchoring the narrative to φ prevents drift between the description and the formal specification. Secrets carry only constraints, target IDs, and mechanic hints; the prompt forbids encoding coordination strategy, which the agents must work out themselves.

Iteration. The agent invokes `judge` and `test_task` in any order and may revise goals, swap mechanics, or call `new_scene` to restart. Tools return structured feedback: per-criterion scores with required fixes, planner diagnostics, or pass/fail traces. The agent uses this feedback to revise. `submit_task` requires both `judge` and `test_task` to have passed.

4.3. Verifiers

PDDL Parsing. The `judge` tool first runs two deterministic checks: structural validation that the PDDL goal is syntactically valid, all referenced objects and furniture exist in the scene, and mechanic bindings are well-formed; and a $\mathcal{K}$-depth check that rejects the task when the nesting depth of epistemic operators in φ (Section 3.3) does not match the requested target d.[1] These cheap checks gate the more expensive judge and calibration runs; physical executability is verified separately by the calibrator below.

LLM Judge Council. Two LLMs (Kimi-K2.5 and GPT-5.2) independently score each candidate on eight $[0, 1]$ criteria, and a task passes only when both agree. Seven criteria are shared: **agent necessity**, **secret quality**, **public/private grounding**, **narrative consistency**, **goal relevance**, **mechanic utilization**, and **formal goal quality**; the eighth is category-specific: **task interdependence** (cooperative) or **subgoal tension** (mixed). Acceptance requires mean $\bar{s} \geq 0.65$ with a per-criterion floor $s_c \geq 0.5$, both tuned on 50 manually rated tasks. On failure, the council returns per-criterion scores and concrete fixes (e.g., "add a secret for agent_1 explaining the limited_bandwidth mechanic") that drive the next revision. Appendices G and I give the full prompt, sample feedback, and rationale for each criterion.

Structural calibrator. Success on `EnactToM` depends on two factors: embodied reasoning (navigation, object recognition, manipulation) and Theory of Mind. To isolate ToM, the calibrator runs each candidate in a *baseline* condition with all secrets revealed to every agent. If agents

[1] See Appendix I for why the planner is not given to the evaluated agent as a tool.

fail with full knowledge, the gap is embodied and the task is rejected.[2] Only tasks that pass the baseline enter the benchmark, so scores reflect ToM.

4.4. Difficulty evolution

A fixed benchmark saturates as models improve. EnactToM addresses this by evolving the task pool. The generation agent receives seed tasks from the existing pool as in-context examples (Section 4.2), sampled by their performance against a family of target models (GPT-5.4, Sonnet, DeepSeek-v3.2): a fraction ρ are tasks these models failed, and the remainder are tasks they passed. We refer to ρ as the *seed-task failure ratio*; higher ρ pushes the generator toward the epistemic coordination patterns current frontier models fail on. As new tasks enter the pool and are benchmarked, the seed distribution shifts. Later generation runs see harder examples, creating evolutionary pressure without changing the generation infrastructure.

5. Experiments

Experimental setup. We evaluate seven frontier models on two EnactToM splits, *standard* and *hard*, each containing 150 tasks spanning cooperative and mixed-motive categories. The splits differ only in seed-task failure ratio during generation, $\rho = 0.8$ for standard, $\rho = 0.9$ for hard. The hard split concentrates on tasks current frontier models fail. Full pool composition is in Table 3.[3] Each episode is given double the calibration-baseline turn and message budget to account for coordination overhead of partial information.

Success criteria. A task is solved once all physical predicates in φ are completed; this is the *functional* measure. Separately, the $\mathcal{K}$-operators in φ define *literal ToM probes*: at episode end each agent is asked an explicit question about another agent's knowledge state (e.g., "what state does agent_0 think the fridge is in?"); the full probe prompt is in Appendix H. The probe accuracy, reported as "Literal" in Table 2, measures whether agents can *report* beliefs, independent of whether they *acted* on them.

Metrics. Each task is run n=3 times. **Avg** is the mean per-run pass rate, reported with binomial standard error over fixed attempts. **Pass@3** is a union metric: success if at least one of the three runs succeeds. **Pass^3** is an exact metric: success only if all three runs succeed. Avg shows single-attempt capability and Pass@3 the best-of-3 ceiling, but a one-in-three success is luck, not ToM. We emphasize Pass^3: coordination should be reproducible.

[2]This is just a baseline for eliminating environmental and embodiment failures. We do not see the solution of this problem to look like this.

[3]We limit to depth 3 because reasoning beyond d=3 is difficult even for humans.

6. Results

Table 2 gives the main comparison between functional task success and literal belief-probe accuracy on matched EnactToM-Standard and EnactToM-Hard tasks. The central pattern is a sharp act–report gap. On the hard split, all seven frontier models achieve 0.0% overall functional Pass^3, while their literal Avg scores average 45.0%. Even the strongest literal models, Gemini-Pro and O3, answer many belief probes correctly but do not convert those beliefs into reproducible coordination. Figure 2 summarizes the evolution, literal–functional gap, and depth analyses. We organize the results around four questions.

6.1. Does high literal ToM imply high functional ToM?

We find that it does not. On the hard split, every evaluated model scores 0.0% overall functional Pass^3, while literal Pass^3 reaches 37.5% for Gemini-Pro and 25.0% for O3. The same pattern appears in single-run averages: Gemini-Pro reaches $63.3_{\pm 4.4}$% literal Avg but only $6.7_{\pm 2.3}$% functional Avg on hard tasks, and O3 reaches $52.5_{\pm 4.6}$% literal Avg but only $5.0_{\pm 2.0}$% functional Avg. This gap shows significant separation between reporting a belief and acting on it.

Thus literal ToM probes can overestimate the ToM abilities of these models, suggesting that existing benchmarks (Shi et al., 2025; Jin et al., 2024; Wu et al., 2023) do not fully predict behavior in action-constrained settings. In EnactToM, the useful unit of ToM is the policy that decides when to communicate, whom to inform, and how to route action through constrained partners.

6.2. Does evolution make tasks more difficult?

We find that evolution does successfully create tasks where reproducible functional success is harder to achieve. The standard subset still admits some stable coordination: Gemini-Flash reaches 22.5% overall functional Pass^3, Gemini-Pro reaches 12.5%, and GPT-5.4 reaches 5.0%. On the hard subset, every model drops to 0.0% overall functional Pass^3. Average functional pass rates also fall from standard to hard for Gemini-Pro ($39.2_{\pm 4.5}$% to $6.7_{\pm 2.3}$%), Gemini-Flash ($42.5_{\pm 4.5}$% to $4.2_{\pm 1.8}$%), GPT-5.4 ($17.5_{\pm 3.5}$% to $3.3_{\pm 1.6}$%), O3 ($12.5_{\pm 3.0}$% to $5.0_{\pm 2.0}$%), and GPT-5.4-mini ($10.8_{\pm 2.8}$% to $3.3_{\pm 1.6}$%). Kimi-K2.5 and DeepSeek-v3.2 also have zero hard functional Pass^3.

6.3. What specific model behavior explains the gap?

In qualitative analysis, we find that models fail at the operational steps that make beliefs useful. The strongest standard functional model is Gemini-Flash (42.5% Avg, 22.5% Pass^3); the strongest hard literal model is Gemini-Pro (63.3% Avg, 37.5% Pass^3). This shows that models that

Table 2. EnactToM results on matched standard and hard subsets. Each model is evaluated over cooperative, mixed, and overall scopes; each split reports functional task success and literal ToM probe success. Definitions of Avg, Pass@3, and Pass^3 are given in the experimental setup. Rose cells emphasize low performance, darker rose marks exact zero, and green marks the best overall exact Pass^3 in each split and metric family. Asterisks mark partial API runs; missing attempts are counted as non-passes under fixed n=3 accounting.

Model	Scope	Standard						Hard					
		Functional			Literal			Functional			Literal		
		Avg	Pass@3	Pass^3	Avg	Pass@3	Pass^3	Avg	Pass@3	Pass^3	Avg	Pass@3	Pass^3
Gemini-Pro	Coop	$53.0_{\pm 6.1}$	77.3	22.7	$34.8_{\pm 5.9}$	59.1	13.6	$7.6_{\pm 3.3}$	22.7	**0.0**	$60.6_{\pm 6.0}$	86.4	36.4
	Mixed	$22.2_{\pm 5.7}$	44.4	**0.0**	$18.5_{\pm 5.3}$	33.3	11.1	$5.6_{\pm 3.1}$	16.7	**0.0**	$66.7_{\pm 6.4}$	88.9	38.9
	Overall	$39.2_{\pm 4.5}$	62.5	12.5	$27.5_{\pm 4.1}$	47.5	12.5	$6.7_{\pm 2.3}$	20.0	**0.0**	$63.3_{\pm 4.4}$	87.5	**37.5**
Gemini-Flash	Coop	$43.9_{\pm 6.1}$	72.7	22.7	$39.4_{\pm 6.0}$	72.7	18.2	$3.0_{\pm 2.1}$	9.1	**0.0**	$45.5_{\pm 6.1}$	72.7	13.6
	Mixed	$40.7_{\pm 6.7}$	66.7	22.2	$13.0_{\pm 4.6}$	27.8	5.6	$5.6_{\pm 3.1}$	16.7	**0.0**	$38.9_{\pm 6.6}$	72.2	11.1
	Overall	$42.5_{\pm 4.5}$	70.0	**22.5**	$27.5_{\pm 4.1}$	52.5	12.5	$4.2_{\pm 1.8}$	12.5	**0.0**	$42.5_{\pm 4.5}$	72.5	12.5
GPT-5.4	Coop	$21.2_{\pm 5.0}$	36.4	9.1	$22.7_{\pm 5.2}$	45.5	**0.0**	$3.0_{\pm 2.1}$	9.1	**0.0**	$45.5_{\pm 6.1}$	72.7	13.6
	Mixed	$13.0_{\pm 4.6}$	27.8	**0.0**	$11.1_{\pm 4.3}$	22.2	**0.0**	$3.7_{\pm 2.6}$	11.1	**0.0**	$42.6_{\pm 6.7}$	83.3	16.7
	Overall	$17.5_{\pm 3.5}$	32.5	5.0	$17.5_{\pm 3.5}$	35.0	**0.0**	$3.3_{\pm 1.6}$	10.0	**0.0**	$44.2_{\pm 4.5}$	77.5	15.0
O3	Coop	$13.6_{\pm 4.2}$	31.8	**0.0**	$33.3_{\pm 5.8}$	59.1	13.6	$7.6_{\pm 3.3}$	22.7	**0.0**	$56.1_{\pm 6.1}$	86.4	31.8
	Mixed	$11.1_{\pm 4.3}$	22.2	**0.0**	$35.2_{\pm 6.5}$	50.0	16.7	$1.9_{\pm 1.8}$	5.6	**0.0**	$48.1_{\pm 6.8}$	88.9	16.7
	Overall	$12.5_{\pm 3.0}$	27.5	**0.0**	$34.2_{\pm 4.3}$	55.0	**15.0**	$5.0_{\pm 2.0}$	15.0	**0.0**	$52.5_{\pm 4.6}$	87.5	25.0
Kimi-K2.5*	Coop	$6.1_{\pm 2.9}$	13.6	**0.0**	$6.1_{\pm 2.9}$	9.1	**0.0**	$3.0_{\pm 2.1}$	4.5	**0.0**	$42.4_{\pm 6.1}$	68.2	18.2
	Mixed	$5.6_{\pm 3.1}$	16.7	**0.0**	$3.7_{\pm 2.6}$	11.1	**0.0**	$9.3_{\pm 3.9}$	27.8	**0.0**	$46.3_{\pm 6.8}$	72.2	16.7
	Overall	$5.8_{\pm 2.1}$	15.0	**0.0**	$5.0_{\pm 2.0}$	10.0	**0.0**	$5.8_{\pm 2.1}$	15.0	**0.0**	$44.2_{\pm 4.5}$	70.0	17.5
GPT-5.4-mini*	Coop	$10.6_{\pm 3.8}$	18.2	**0.0**	$24.2_{\pm 5.3}$	54.5	4.5	$4.5_{\pm 2.6}$	13.6	**0.0**	$36.4_{\pm 5.9}$	63.6	4.5
	Mixed	$11.1_{\pm 4.3}$	27.8	**0.0**	$18.5_{\pm 5.3}$	38.9	**0.0**	$1.9_{\pm 1.8}$	5.6	**0.0**	$25.9_{\pm 6.0}$	61.1	**0.0**
	Overall	$10.8_{\pm 2.8}$	22.5	**0.0**	$21.7_{\pm 3.8}$	47.5	2.5	$3.3_{\pm 1.6}$	10.0	**0.0**	$31.7_{\pm 4.2}$	62.5	2.5
DeepSeek-v3.2*	Coop	$3.0_{\pm 2.1}$	9.1	**0.0**	**0.0**$_{\pm 0.0}$	**0.0**	**0.0**	$7.6_{\pm 3.3}$	18.2	**0.0**	$34.8_{\pm 5.9}$	63.6	4.5
	Mixed	$1.9_{\pm 1.8}$	5.6	**0.0**	**0.0**$_{\pm 0.0}$	**0.0**	**0.0**	$9.3_{\pm 3.9}$	27.8	**0.0**	$38.9_{\pm 6.6}$	66.7	5.6
	Overall	$2.5_{\pm 1.4}$	7.5	**0.0**	**0.0**$_{\pm 0.0}$	**0.0**	**0.0**	$8.3_{\pm 2.5}$	22.5	**0.0**	$36.7_{\pm 4.4}$	65.0	5.0

Table 3. EnactToM dataset statistics across the 300-task pool.

Composition	**Count**
Total tasks	300
Cooperative / Mixed	150 / 150
2 agents	78 (26.0%)
3 agents	220 (73.3%)
4+ agents	2 (0.7%)
$\mathcal{K}$-depth 1	112 (37.3%)
$\mathcal{K}$-depth 2	79 (26.3%)
$\mathcal{K}$-depth 3	109 (36.3%)
Mechanic coverage	**Count**
Room restriction	300 (100%)
Limited bandwidth	300 (100%)
Restricted communication	172 (57.3%)
Remote control	48 (16.0%)
Inverse state	9 (3.0%)
State mirroring	5 (1.7%)
Baseline turns (mean)	9.6
Standard turns (mean)	19.2 (2×)

can answer belief probes still fail to prioritize information relay, reason over partner constraints, or preserve message budget. A manual audit of 40 sampled failures found that 37 were epistemic coordination breakdowns.

Table 4 shows that the failures are ToM failures not random simulator mistakes. Agents possess decisive facts but communicate them too late, complete actions without ensuring partners know, or spend scarce messages on low-priority recipients. Current models reason about these ingredients locally, but they do not maintain them as a global coordination state.

6.4. How do models behave in cooperative vs. strategic settings?

Interestingly, we find that mixed-motive tasks do not uniformly reduce performance relative to cooperative tasks. On hard functional Avg, Kimi-K2.5 is higher on mixed than cooperative tasks (9.3% vs. 3.0%), as are Gemini-Flash (5.6% vs. 3.0%), GPT-5.4 (3.7% vs. 3.0%), and DeepSeek-v3.2 (9.3% vs. 7.6%). Gemini-Pro, O3, and GPT-5.4-mini show the opposite pattern. Strategic private objectives can either structure action or create new opportunities for premature disclosure and sabotage.

The traces suggest that mixed-motive tasks fail in a different way from cooperative tasks. In cooperative failures, agents usually lose because a useful fact never reaches the right partner. In mixed-motive failures, the model also has to decide whether a fact should be shared at all. Some agents reveal their private goal immediately, making it easy for partners to block or undo it; others overcommit to the private goal and damage the shared objective before the team has established the necessary state. This makes the higher mixed scores for some models informative: private incentives can give the model a clearer local plan, but success requires controlling disclosure while still preserving enough trust

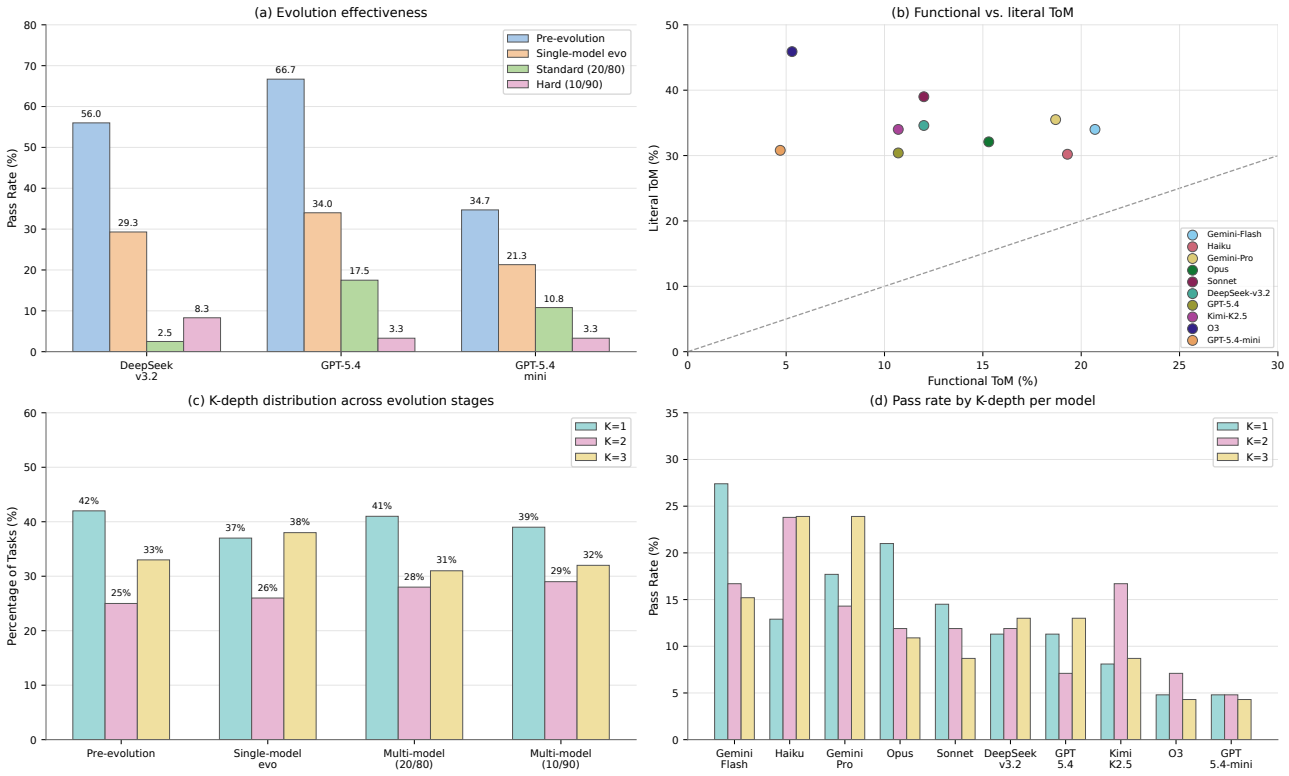

Figure 2. **(a)** Functional Avg single-run pass rate for the three seed models across pre-evolution, single-model evolution, and multi-model 20/80 and 10/90 pools. **(b)** Functional Avg vs. literal Avg, showing belief probes exceed embodied task success. **(c)** Task percentage at each $\mathcal{K}$ depth across evolution stages, showing hardness is not just deeper nesting. **(d)** Functional Avg pass rate by $\mathcal{K}$ depth for each model, showing brittleness at every depth. Panels (a), (b), and (d) report Avg, not Pass@k or Pass^k.

and coordination to finish the common task.

K-depth validity. We manually verify 50 randomly sampled tasks; 49 (98%) have valid $\mathcal{K}$-depth (Appendix K).

Ablations. We ablate the generation pipeline (Table 9, Appendix M). Without **baseline calibration**, 51% of tasks are physically unsolvable even with full information. Without **ICL seed examples**, acceptance drops to 50% and depth decreases. **Relaxing the secret quality check** inflates pass rate from 26.7% to 43.5% because agents follow prescribed coordination rather than modeling partner knowledge.

7. Discussion

Limitations and future directions. `EnactToM` caps ToM depth at $d = 3$, uses 2–4 agents in HSSD household scenes, and covers cooperative and mixed-motive settings rather than fully adversarial ones. Future work should scale to larger teams, other domains, and deeper recursive beliefs, while further separating embodied skill from epistemic coordination for weaker models.

Interpreting progress. A high score on `EnactToM` should mean that an agent can maintain and update who-knows-what, route facts to the right partner, and act before the information becomes stale. We report literal and functional scores because belief statements alone are insufficient: literal probes test whether a belief can be stated, while functional success tests whether it is used under action and communication constraints.

Using the evolving pool. As models improve, old tasks remain useful for historical comparability and regression checks. New rounds should be added against current frontier failures, preserving a ladder from solved to unsolved ToM behaviors. This makes `EnactToM` a measurement instrument rather than a fixed leaderboard: it tracks which epistemic operations have become reliable and which still break under embodiment.

Conclusion. We introduce `EnactToM`, an evolving benchmark for functional Theory of Mind in embodied multi-agent settings. Frontier models can often *state* what partners know but cannot reliably *use* that knowledge during coordination. The dominant failure is epistemic coordination breakdown: withheld information, ignored partner constraints, and misallocated messages.

Failure pattern	Behavior in the audit	Representative evidence
Withholding critical information	In 7 of 40 cases, an agent holds a target, room, or object fact that a partner needs but communicates it only after the partner has already acted on a wrong guess.	*Inspection Staging with a Hidden Target Cabinet*: the target cabinet ID is sent after partners have spent their messages and placed the object on the wrong cabinet (Appendix C.1).
Epistemic chain breakdown	In 8 of 40 cases, an agent completes the physical action but never establishes that the teammate who needs the fact knows it.	*Staging + Fridge Verification*: the fridge is opened, but the opening is never relayed to the agents whose success depends on knowing it (Appendix C.2).
Private objective sabotage or disclosure	In mixed-motive episodes, agents either damage the shared plan for private gain or reveal private objectives so early that partners can block them.	*Safety Staging with Conflicting Incentives*: one agent announces and executes a fridge-state conflict rather than modeling how teammates will respond (Appendix C.3).
Misallocating scarce messages	In 4 of 40 cases, agents spend limited messages on the wrong recipient, an unreachable recipient, or low-priority content.	*One-Shot Relay*: the only agent with the critical table ID first messages a blocked recipient, then sends an irrelevant fact to the reachable partner (Appendix C.4).
Ignoring partner constraints	Agents delegate actions to partners who are barred from the relevant room or already constrained by object possession.	*Inspection Prep with Nested Confirmation*: a partner is repeatedly asked to act in a room they cannot enter, wasting two messages before the delegation is corrected (Appendix C.5).

Table 4. Failure-analysis summary. Full trajectories and quoted actions are in Appendix C.

References

Akhtar, M., Reuel, A., Soni, P., Ahuja, S., Ammanamanchi, P. S., Rawal, R., Zouhar, V., Yadav, S., Whitehouse, C., Ki, D., Mickel, J., Choshen, L., Šuppa, M., Batzner, J., Chim, J., Sania, J., Long, Y., Rahmani, H. A., Knight, C., Nan, Y., Raj, J., Fan, Y., Singh, S., Sahoo, S., Habba, E., Gohar, U., Pawar, S., Scholz, R., Subramonian, A., Ni, J., Kochenderfer, M., Koyejo, S., Sachan, M., Biderman, S., Talat, Z., Ghosh, A., and Solaiman, I. When ai benchmarks plateau: A systematic study of benchmark saturation, 2026. URL https://arxiv.org/abs/2602.16763.

Apperly, I. A. and Butterfill, S. A. Do humans have two systems to track beliefs and belief-like states? *Psychological Review*, 116(4):953–970, 2009. doi: 10.1037/a0016923. URL https://doi.org/10.1037/a0016923.

Aumann, R. J. Agreeing to disagree. *The Annals of Statistics*, 4(6):1236–1239, 1976. ISSN 00905364, 21688966. URL http://www.jstor.org/stable/2958591.

Aumann, R. J. Interactive epistemology II: Probability. *International Journal of Game Theory*, 28(3):301–314, 1999. doi: 10.1007/s001820050112. URL https://doi.org/10.1007/s001820050112.

Baker, C. L., Saxe, R., and Tenenbaum, J. B. Action understanding as inverse planning. *Cognition*, 113(3):329–349, 2009. ISSN 0010-0277. doi: 10.1016/j.cognition.2009.07.005.

Baker, C. L., Jara-Ettinger, J., Saxe, R., and Tenenbaum, J. B. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nature Human Behaviour*, 1:0064, 2017. doi: 10.1038/s41562-017-0064. URL https://doi.org/10.1038/s41562-017-0064.

Bard, N., Foerster, J. N., Chandar, S., Burch, N., Lanctot, M., Song, H. F., Parisotto, E., Dumoulin, V., Moitra, S., Hughes, E., Dunning, I., Mourad, S., Larochelle, H., Bellemare, M. G., and Bowling, M. The Hanabi challenge: A new frontier for AI research. *Artificial Intelligence*, 280:103216, 2020. ISSN 0004-3702. doi: 10.1016/j.artint.2019.103216.

Baron-Cohen, S., Leslie, A. M., and Frith, U. Does the autistic child have a "theory of mind" ? *Cognition*, 21(1):37–46, 1985. ISSN 0010-0277. doi: 10.1016/0010-0277(85)90022-8.

Bratman, M. E. Shared cooperative activity. *The Philosophical Review*, 101(2):327–341, 1992.

Camerer, C. F., Ho, T.-H., and Chong, J.-K. A cognitive hierarchy model of games. *The Quarterly Journal of Economics*, 119(3):861–898, 2004.

Carroll, M., Shah, R., Ho, M. K., Griffiths, T. L., Seshia, S. A., Abbeel, P., and Dragan, A. On the utility of learning about humans for human-ai coordination, 2020. URL https://arxiv.org/abs/1910.05789.

Chang, M., Chhablani, G., Clegg, A., Cote, M. D., Desai, R., Hlavac, M., Karashchuk, V., Krantz, J., Mottaghi, R., Parashar, P., Patki, S., Prasad, I., Puig, X., Rai, A., Ramrakhya, R., Tran, D., Truong, J., Turner, J. M., Undersander, E., and Yang, T.-Y. PARTNR: A

benchmark for planning and reasoning in embodied multi-agent tasks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=T5QLRRHyL1.

Crawford, V. P. and Iriberri, N. Fatal attraction: Salience, naïveté, and sophistication in experimental "hide-and-seek" games. *American Economic Review*, 97(5):1731–1750, 2007. doi: 10.1257/aer.97.5.1731. URL https://doi.org/10.1257/aer.97.5.1731.

Dennett, D. C. Beliefs about beliefs. *Behavioral and Brain Sciences*, 1(4):568–570, 1978. doi: 10.1017/S0140525X00076664.

Dennett, D. C. *The Intentional Stance*. MIT Press, Cambridge, MA, 1987. URL https://mitpress.mit.edu/9780262540537/the-intentional-stance/.

Flavell, J. H. The development of knowledge about visual perception. In *Nebraska Symposium on Motivation*, volume 25, pp. 43–76. University of Nebraska Press, Lincoln, NE, 1977.

Flavell, J. H. Perspectives on perspective taking. In Beilin, H. and Pufall, P. B. (eds.), *Piaget's Theory: Prospects and Possibilities*, pp. 107–139. Lawrence Erlbaum Associates, Hillsdale, NJ, 1992.

Flavell, J. H., Shipstead, S. G., and Croft, K. Young children's knowledge about visual perception: Hiding objects from others. *Child Development*, 49(4):1208–1211, 1978. ISSN 00093920, 14678624. URL http://www.jstor.org/stable/1128761.

Flavell, J. H., Everett, B. A., Croft, K., and Flavell, E. R. Young children's knowledge about visual perception: Further evidence for the Level 1–Level 2 distinction. *Developmental Psychology*, 17(1):99–103, 1981. doi: 10.1037/0012-1649.17.1.99. URL https://doi.org/10.1037/0012-1649.17.1.99.

Gandhi, K., Fraenken, J.-P., Gerstenberg, T., and Goodman, N. Understanding social reasoning in language models with language models. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 13518–13529. Curran Associates, Inc., 2023. URL https://neurips.cc/virtual/2023/poster/73680.

Golchin, S. and Surdeanu, M. Time travel in LLMs: Tracing data contamination in large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=2Rwq6c3tvr.

Gu, Y., Tafjord, O., Kim, H., Moore, J., Bras, R. L., Clark, P., and Choi, Y. Simpletom: Exposing the gap between explicit tom inference and implicit tom application in llms, 2026. URL https://arxiv.org/abs/2410.13648.

Jacovi, A., Caciularu, A., Goldman, O., and Goldberg, Y. Stop uploading test data in plain text: Practical strategies for mitigating data contamination by evaluation benchmarks. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5075–5084, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.308. URL https://aclanthology.org/2023.emnlp-main.308/.

Jin, C., Wu, Y., Cao, J., Xiang, J., Kuo, Y.-L., Hu, Z., Ullman, T., Torralba, A., Tenenbaum, J., and Shu, T. MMToM-QA: Multimodal theory of mind question answering. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16077–16102, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.851. URL https://aclanthology.org/2024.acl-long.851/.

Keysar, B., Lin, S., and Barr, D. J. Limits on theory of mind use in adults. *Cognition*, 89(1):25–41, 2003. ISSN 0010-0277. doi: 10.1016/S0010-0277(03)00064-7.

Kiela, D., Bartolo, M., Nie, Y., Kaushik, D., Geiger, A., Wu, Z., Vidgen, B., Prasad, G., Singh, A., Ringshia, P., Ma, Z., Thrush, T., Riedel, S., Waseem, Z., Stenetorp, P., Jia, R., Bansal, M., Potts, C., and Williams, A. Dynabench: Rethinking benchmarking in NLP. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y. (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4110–4124, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.324. URL https://aclanthology.org/2021.naacl-main.324/.

Kim, H., Sclar, M., Zhou, X., Bras, R. L., Kim, G., Choi, Y., and Sap, M. FANToM: A benchmark for stress-testing machine theory of mind in interactions. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 14397–14413, 2023.

Kolve, E., Mottaghi, R., Han, W., VanderBilt, E., Weihs, L., Herrasti, A., Deitke, M., Ehsani, K., Gordon, D., Zhu, Y., Kembhavi, A., Gupta, A., and Farhadi, A. Ai2-thor:

An interactive 3d environment for visual ai, 2017. URL https://arxiv.org/abs/1712.05474.

Kosinski, M. Evaluating large language models in theory of mind tasks. *Proceedings of the National Academy of Sciences*, 121(45):e2405460121, 2024. doi: 10.1073/pnas.2405460121. URL https://www.pnas.org/doi/abs/10.1073/pnas.2405460121.

Le, M., Boureau, Y.-L., and Nickel, M. Revisiting the evaluation of theory of mind through question answering. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pp. 5872–5877, 2019. doi: 10.18653/v1/D19-1598.

Masangkay, Z. S., McCluskey, K. A., McIntyre, C. W., Sims-Knight, J., Vaughn, B. E., and Flavell, J. H. The early development of inferences about the visual percepts of others. *Child Development*, 45(2):357–366, 1974. ISSN 00093920, 14678624. doi: 10.2307/1127956. URL http://www.jstor.org/stable/1127956.

Nagel, R. Unraveling in guessing games: An experimental study. *The American Economic Review*, 85(5):1313–1326, 1995. ISSN 00028282. URL http://www.jstor.org/stable/2950991.

Onishi, K. H. and Baillargeon, R. Do 15-month-old infants understand false beliefs? *Science*, 308 (5719):255–258, 2005. doi: 10.1126/science.1107621. URL https://www.science.org/doi/abs/10.1126/science.1107621.

Ott, S., Barbosa-Silva, A., Blagec, K., Brauner, J., and Samwald, M. Mapping global dynamics of benchmark creation and saturation in artificial intelligence. *Nature Communications*, 13(1):6793, November 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-34591-0.

Padmakumar, A., Thomason, J., Shrivastava, A., Lange, P., Narayan-Chen, A., Gella, S., Piramuthu, R., Tur, G., and Hakkani-Tur, D. TEACh: Task-driven embodied agents that chat. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(2):2017–2025, June 2022. doi: 10.1609/aaai.v36i2.20097.

Park, J. S., O'Brien, J., Cai, C. J., Morris, M. R., Liang, P., and Bernstein, M. S. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701320. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

Perner, J. and Wimmer, H. "John thinks that Mary thinks that..." attribution of second-order beliefs by 5- to 10-year-old children. *Journal of Experimental Child Psychology*, 39(3):437–471, 1985. ISSN 0022-0965. doi: 10.1016/0022-0965(85)90051-7.

Premack, D. and Woodruff, G. Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences*, 1(4): 515–526, 1978.

Puig, X., Ra, K., Boben, M., Li, J., Wang, T., Fidler, S., and Torralba, A. VirtualHome: Simulating Household Activities Via Programs . In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8494–8502, Los Alamitos, CA, USA, June 2018. IEEE Computer Society. doi: 10.1109/CVPR.2018.00886. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00886.

Riemer, M., Ashktorab, Z., Bouneffouf, D., Das, P., Liu, M., Weisz, J. D., and Campbell, M. Position: Theory of mind benchmarks are broken for large language models. In Singh, A., Fazel, M., Hsu, D., Lacoste-Julien, S., Berkenkamp, F., Maharaj, T., Wagstaff, K., and Zhu, J. (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 82091–82130. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/riemer25a.html.

Sap, M., Le Bras, R., Fried, D., and Choi, Y. Neural theory-of-mind? on the limits of social intelligence in large LMs. In Goldberg, Y., Kozareva, Z., and Zhang, Y. (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3762–3780, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.248. URL https://aclanthology.org/2022.emnlp-main.248/.

Sclar, M., Dwivedi-Yu, J., Fazel-Zarandi, M., Tsvetkov, Y., Bisk, Y., Choi, Y., and Celikyilmaz, A. Explore theory of mind: program-guided adversarial data generation for theory of mind reasoning. In Yue, Y., Garg, A., Peng, N., Sha, F., and Yu, R. (eds.), *International Conference on Learning Representations*, volume 2025, pp. 67635–67660, 2025. URL https://openreview.net/forum?id=246rHKUnnf.

Shapira, N., Levy, M., Alavi, S. H., Zhou, X., Choi, Y., Goldberg, Y., Sap, M., and Shwartz, V. Clever hans or neural theory of mind? stress testing social reasoning in large language models. In Graham, Y. and Purver, M. (eds.), *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2257–2273, St.

Julian's, Malta, March 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.eacl-long.138. URL https://aclanthology.org/2024.eacl-long.138/.

Shi, H., Ye, S., Fang, X., Jin, C., Isik, L., Kuo, Y.-L., and Shu, T. Muma-tom: Multi-modal multi-agent theory of mind. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(2):1510–1519, Apr. 2025. doi: 10.1609/aaai.v39i2.32142. URL https://ojs.aaai.org/index.php/AAAI/article/view/32142.

Shridhar, M., Thomason, J., Gordon, D., Bisk, Y., Han, W., Mottaghi, R., Zettlemoyer, L., and Fox, D. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Stahl, D. O. and Wilson, P. W. Experimental evidence on players' models of other players. *Journal of Economic Behavior & Organization*, 25(3):309–327, 1994.

Stahl, D. O. and Wilson, P. W. On players' models of other players: Theory and experimental evidence. *Games and Economic Behavior*, 10(1):218–254, 1995. ISSN 0899-8256. doi: 10.1006/game.1995.1031.

Szot, A., Clegg, A., Undersander, E., Wijmans, E., Zhao, Y., Turner, J., Maestre, N., Mukadam, M., Chaplot, D. S., Maksymets, O., et al. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems*, volume 34, pp. 251–266, 2021.

Tomasello, M., Carpenter, M., Call, J., Behne, T., and Moll, H. Understanding and sharing intentions: The origins of cultural cognition. *Behavioral and Brain Sciences*, 28(5):675–691, 2005.

Ullman, T. Large language models fail on trivial alterations to theory-of-mind tasks, 2023. URL https://arxiv.org/abs/2302.08399.

Vesper, C., Butterfill, S., Knoblich, G., and Sebanz, N. A minimal architecture for joint action. *Neural Networks*, 23(8):998–1003, 2010. ISSN 0893-6080. doi: 10.1016/j.neunet.2010.06.002.

Wellman, H. M. and Liu, D. Scaling of theory-of-mind tasks. *Child Development*, 75(2):523–541, March 2004. ISSN 0009-3920. doi: 10.1111/j.1467-8624.2004.00691.x.

Wellman, H. M., Cross, D., and Watson, J. Meta-analysis of theory-of-mind development: The truth about false belief. *Child Development*, 72(3):655–684, 05 2001. ISSN 0009-3920. doi: 10.1111/1467-8624.00304. URL https://doi.org/10.1111/1467-8624.00304.

White, C., Dooley, S., Roberts, M., Pal, A., Feuer, B., Jain, S., Shwartz-Ziv, R., Jain, N., Saifullah, K., Dey, S., Shubh-Agrawal, Sandha, S. S., Naidu, S. V., Hegde, C., LeCun, Y., Goldstein, T., Neiswanger, W., and Goldblum, M. Livebench: A challenging, contamination-limited LLM benchmark. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=sKYHBTAxVa.

Wimmer, H. and Perner, J. Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children's understanding of deception. *Cognition*, 13(1):103–128, 1983. doi: 10.1016/0010-0277(83)90004-5.

Wu, Y., He, Y., Jia, Y., Mihalcea, R., Chen, Y., and Deng, N. Hi-ToM: A benchmark for evaluating higher-order theory of mind reasoning in large language models. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10691–10706, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.717. URL https://aclanthology.org/2023.findings-emnlp.717/.

Xu, H., Zhao, R., Zhu, L., Du, J., and He, Y. OpenToM: A comprehensive benchmark for evaluating theory-of-mind reasoning capabilities of large language models. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8593–8623, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.466. URL https://aclanthology.org/2024.acl-long.466/.

Yang, J., Jimenez, C., Wettig, A., Lieret, K., Yao, S., Narasimhan, K., and Press, O. Swe-agent: Agent-computer interfaces enable automated software engineering. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 50528–50652. Curran Associates, Inc., 2024. doi: 10.52202/079017-1601.

Zhou, P., Madaan, A., Potharaju, S. P., Gupta, A., McKee, K. R., Holtzman, A., Pujara, J., Ren, X., Mishra, S., Nematzadeh, A., Upadhyay, S., and Faruqui, M. How far are large language models from agents with theory-of-mind?, 2023. URL https://arxiv.org/abs/2310.03051.

Zhou, X., Zhu, H., Mathur, L., Zhang, R., Yu, H., Qi, Z., Morency, L.-P., Bisk, Y., Fried, D., Neubig, G., and Sap, M. SOTOPIA: Interactive evaluation for social intelligence in language agents. In *The Twelfth International Conference on Learning Representations*,

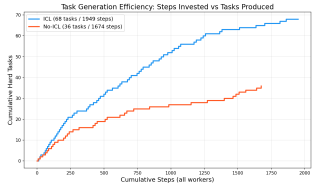

Figure 3. Cumulative tasks generated with and without ICL seed examples. With ICL (seed tasks biased toward failures), the generation agent produces accepted tasks faster and at higher epistemic depth.

2024. URL https://openreview.net/forum?id=mM7VurbA4r.

A. Use of LLMs in this work

We used large language models as assistive tools during the development of this work, for both coding and writing. We document this usage for transparency.

Coding assistance. We used Claude Code (Anthropic) as a coding assistant throughout the development of the `EnactToM` pipeline. This included writing and debugging Python code for the task generation agent (mini SWE agent), and the visualization server, which will be released publicly with the benchmark. The assistant helped with code refactoring and shell scripting. All generated code was reviewed and tested by the authors before integration.

Writing assistance. We used Claude Code as a writing assistant for this paper. The first drafts were manually written by the authors. The assistant helped with revising sections, shortening sentences, resolving inline comments, formatting LaTeX tables and figures, and structuring appendices. All generated text was reviewed, edited, and approved by the authors. The scientific content, experimental design, analysis, and conclusions are the authors' own.

B. Extended related work

ToM in LLMs. Early claims that LLMs possess ToM (Kosinski, 2024) have been challenged: trivial surface changes break performance (Ullman, 2023), and apparent competence reduces to shallow heuristics (Shapira et al., 2024; Sap et al., 2022). This has motivated text-based benchmarks, including ToMi (Le et al., 2019), HiToM (Wu et al., 2023), BigToM (Gandhi et al., 2023), FANToM (Kim et al., 2023), ExploreToM (Sclar et al., 2025), OpenToM (Xu et al., 2024), and multimodal variants (Jin et al., 2024; Shi et al., 2025). All evaluate *literal* ToM (Riemer et al., 2025): the model reads a scenario and reports beliefs. Recent work recognizes this gap: T4D (Zhou et al., 2023) shows LLMs track beliefs but fail to act on them, and SimpleToM (Gu et al., 2026) confirms a stark gap between explicit inference and implicit application. These evaluations remain text-based and single-agent.

Embodied multi-agent benchmarks. Habitat (Szot et al., 2021), AI2-THOR (Kolve et al., 2017), and VirtualHome (Puig et al., 2018) provide 3D household simulators. PARTNR (Chang et al., 2025), TEACh (Padmakumar et al., 2022), and ALFRED (Shridhar et al., 2020) evaluate instruction following within them. Hanabi (Bard et al., 2020) and Overcooked-AI (Carroll et al., 2020) test coordination under hidden information. SOTOPIA (Zhou et al., 2024) and Generative Agents (Park et al., 2023) evaluate social intelligence through dialogue. None formally require epistemic state reasoning: PARTNR tests instruction following, Hanabi's asymmetry is over card values not spatial beliefs, and SOTOPIA lacks embodiment.

Benchmark saturation. Static benchmarks saturate as models improve (Akhtar et al., 2026; Ott et al., 2022), and contamination compounds the problem (Jacovi et al., 2023; Golchin & Surdeanu, 2024). Dynamic approaches like Dynabench (Kiela et al., 2021) and LiveBench (White et al., 2025) use human-in-the-loop or monthly refreshes, but neither provides formal solvability or epistemic depth guarantees.

C. ToM failure trajectory evidence

This appendix provides trajectory evidence for each ToM failure mode identified in Section 6.3. Examples are from the manual failure audit sample and were selected from failed tasks exhibiting each pattern. Quotes are extracted directly from agent action and message fields in the benchmark logs.

C.1. Failure to relay critical information

Example 1. *Inspection Staging with a Hidden Target Cabinet.* Cooperative, 3 agents, $\mathcal{K}$-depth 3. Mechanics: room restriction, restricted communication, limited bandwidth (1 message per agent).

Agent secrets:

- agent_0: barred from kitchen_1. Knows the correct target for picture_frame_5 is cabinet_43 in kitchen_1. Knows fridge_36 starts closed.
- agent_1: barred from closet_1. Knows picture_frame_5 is on stool_29 in closet_1.
- agent_2: no private knowledge about the correct cabinet or object locations.

What should happen: agent_0 should immediately tell a partner that the correct cabinet is cabinet_43. agent_1 or

agent_2 then retrieves the picture frame from closet_1 and places it on cabinet_43 in kitchen_1.

What actually happens:

> **T1–4** agent_0 searches for objects (`FindObjectTool[picture_frame_5]`, `FindObjectTool[phone_stand]`) and navigates between rooms. Does not send any message.
> **T5** agent_1 → agent_2: "Please get picture_frame_5 from stool_29 in closet_1 and place it on **cabinet_40** in kitchen." Agent_1 guesses cabinet_40 because it does not know the correct target.
> **T6–9** agent_2 follows agent_1's instruction: navigates to closet_1, picks up picture_frame_5, navigates to cabinet_40. agent_0 continues working on other subtasks (opening chest_of_drawers_34).
> **T10** agent_2: `Place[picture_frame_5, on, cabinet_40]` → Frame placed on **wrong cabinet**.
> **T12** agent_2 → agent_1: "Picture frame placed in cabinet_40. Confirm other tasks complete?"
> **T13** agent_1 → agent_2: "All goals complete: picture frame on cabinet_40, fridge_36 open." Agent_1 exhausts its message confirming wrong placement.
> **T16** agent_0 → agent_2: "Can you pick it up and place it on cabinet_43 in kitchen_1?" **Too late**: agent_2 has exhausted its messages and the episode is ending.

agent_0 held the only piece of information that determined task success (cabinet_43) but spent 15 turns on its own physical subtasks before sending it. Agent_0 did not model that agent_1 would guess the cabinet and direct agent_2 to act on that guess.

Example 2. *Cabinet Confirmation Chain.* agent_0 knows the correct cabinet is cabinet_32. agent_0 has one message. Instead of sharing this critical info, agent_0 uses the message to ask about the glass location. agent_1, lacking the cabinet info, opens cabinet_31 (wrong) and reports it as correct.

C.2. Epistemic chain breakdown

Example 1. *Staging + Fridge Verification with Hidden Incentives.* 3 agents. Goal includes $\mathcal{K}_{a_0}$(`is_open(fridge_34)`). agent_1 opens fridge_34 at turn 3.

> **T3** agent_1: `Open[fridge_34]` → Successful.
> **T19** agent_1 → all: "Found bowl_0 on counter, plate_1 in cabinet_26 . . . Need to find bottle." → Mentions items but **never mentions fridge state**.
> **T21** all agents: `Done[None]` → No agent confirmed fridge is open. K-chain not established.

agent_1 opened the fridge but never told anyone. The physical goal (fridge open) is met; the epistemic goal (agent_0 knows it is open) is not. agent_1 does not distinguish "the fridge is open" from "my partners know the fridge is open."

Example 2. *Relay Inspection with Restricted Access.* agent_1 holds spectacles_1 and tells agent_0 to place them on stand_50 in living_room_1. But agent_1 is barred from living_room_1 and is the one holding the item. agent_1 never puts the spectacles down for handoff; instead places them on the wrong surface in the dining room. The physical chain breaks before the epistemic chain is even attempted.

C.3. Private objective sabotage and disclosure

Example 1: Sabotage. *Safety Staging with Conflicting Incentives.* Shared goal: fridge_14 open. agent_1's private goal: fridge_14 closed.

> **T14** agent_1 → all: "The toy fire truck is on table_9. Can you move it to couch_10? **I'll close the fridge after**."
> **T15** agent_1: `Close[fridge_14]` → Directly sabotages shared goal.

agent_1 announces the sabotage before executing it. The agent does not model that teammates will read this message and could preemptively re-open the fridge.

Example 2: Disclosure. *Inspection Prep with Limited Messaging.* agent_1's private goal: fridge closed (shared goal: open).

> **T1** agent_1 → all: "Cup_4 is on cabinet_48 in bathroom_1. Also: **my personal objective requires the fridge CLOSED at end, conflicting with shared objective**."
> **T2** agent_2 → all: "I can get cup_4. But conflict: **my personal objective needs fridge CLOSED too**."

Both agents reveal conflicting private objectives in their first messages, eliminating any strategic advantage the private information could provide. Both patterns reflect a failure to reason about how one's actions change what others believe and how they will respond. The sabotaging agent does not model that teammates will detect defection. The disclosing agent does not model that revealing private information lets others preemptively block the private goal (a second- or third-order ToM failure).

C.4. Misallocating scarce messages

Effective communication under bandwidth constraints requires modeling who can receive a message and what each recipient needs most. When agents fail at this, they waste their limited budget on the wrong recipient or on low-priority content.

Example. *One-Shot Relay (Third-Order).* 4 agents. agent_3 knows the correct table is table_18. agent_3 can only message agent_1.

> **T8** agent_3 → agent_0: "glass_6 needs to go to table_18" → **Blocked**: "You can only send messages to: agent_1."
> **T10** agent_3 → agent_1: "bedroom_2 chest needs to be closed" → Successful, but uses message on **less critical** info.
> **T12** agent_3 → agent_0: "verify chest is closed" → **Blocked again**.

agent_3 tries to reach agent_0 twice (blocked both times) but never sends the table_18 info to agent_1 (the only reachable partner). After the first block, agent_3 does not update its model of the communication graph. The agent neither models who it can reach nor prioritizes what its reachable partner needs most.

C.5. Not modeling partner constraints

Example. *Inspection Prep with Nested Confirmation.* agent_0 is barred from kitchen_1.

> **T8** agent_1 → agent_0: "Place bowl_0 on counter_29 so I can verify it's there."
> **T9** agent_1 → agent_0: "Agent_0, please place bowl_0 down on counter_29 now."
> **T10** agent_0 → agent_1: "**I'm barred from kitchen.** You must place bowl_0 on counter_29."

agent_1 asks agent_0 to act in the kitchen twice before agent_0 corrects them. agent_1 does not model agent_0's room restriction when delegating, wasting two scarce messages on impossible requests. In other instances of this failure mode, agents ask partners to pick up objects that the partner is already holding, or delegate tasks to agents who lack the required room access. In all cases, the constraint information is available in the shared context but the delegating agent does not consult it.

D. Task Examples

D.1. Cooperative: depth-4 epistemic chain

Four agents coordinate a "silent inspection" across a household: open the kitchen cabinet, close the bedroom wardrobe, and place items in designated locations. Communication follows a linear chain ($a_0 \leftrightarrow a_1 \leftrightarrow a_2 \leftrightarrow a_3$) with one message per agent. Room restrictions prevent any single agent from accessing all locations; only a_1 can enter the kitchen.

The PDDL goal includes:

$$\mathcal{K}_{a_0}\big(\mathcal{K}_{a_1}\big(\mathcal{K}_{a_2}\big(\mathcal{K}_{a_3}(\texttt{is\_on\_top}(\texttt{box},\texttt{cabinet}))\big)\big)\big) \tag{3}$$

This requires a_0 to know that a_1 knows that a_2 knows that a_3 knows the box is on the cabinet. The box is in the kitchen, visible only to a_1. For this chain to hold, a_1 must relay the observation through a_2 to a_3, consuming two of the chain's four message slots. Agent a_0 must infer that a_1 relayed rightward (the only viable direction), that a_2 forwarded it, and that a_1 anticipated this relay. This is third-order ToM.

Table 5. Agent configuration for the cooperative silent inspection task. Each agent has 1 message. The linear chain and room restrictions force depth-4 epistemic reasoning.

Agent	Can message	Restricted from	Key private knowledge
a_0	a_1	kitchen	Cabinet holds a cardboard box
a_1	a_0 or a_2	entryway, bedroom, living room	Only agent with kitchen access
a_2	a_1 or a_3	entryway, bathroom, kitchen	Must relay between a_1 and a_3
a_3	a_2	kitchen	Can access bathroom (target surface)

E. Epistemic compilation example

We walk through the full compilation of an epistemic goal. The task has two agents (`agent_0` and `agent_1`), where `agent_1` is in the same room as `bowl_1` and `table_22`, and `agent_1` can communicate with `agent_0` with a message budget of 2.

Original goal (with epistemic operators).

```
(and
  (is_on_top bowl_1 table_22)
  (K agent_0 (K agent_1
    (is_on_top bowl_1 table_22)))
  (is_open cabinet_34)
)
```

The planner needs to verify that (1) the bowl can be placed on the table, (2) the cabinet can be opened, and (3) `agent_0` can come to know that `agent_1` knows the bowl has been placed. The physical subgoals (1) and (2) are already classical PDDL. The epistemic subgoal (3) is not. The compiler transforms it as follows.

Step 1: Create knowledge predicates. For the leaf fact `(is_on_top bowl_1 table_22)`, the compiler creates a first-layer knowledge predicate for each agent:

```
(knows_agent_0_a1b2c3d4) ; a0 knows fact
(knows_agent_1_a1b2c3d4) ; a1 knows fact
```

Because the goal nests two $\mathcal{K}$-operators, the compiler also creates a second-layer predicate for the outer agent:

```
(knows_agent_0_e5f6g7h8) ; a0 knows a1
    knows
```

Step 2: Create observe operators. `Agent_1` is co-located with the bowl and table, so the compiler generates an observe operator that sets `agent_1`'s knowledge predicate when the physical fact holds:

```
(:action observe_knows_agent_1_a1b2c3d4
  :parameters ()
  :precondition (is_on_top bowl_1
   table_22)
  :effect (knows_agent_1_a1b2c3d4))
```

This operator says: if the bowl is on the table and `agent_1` is in the room (encoded in the precondition via the observability model), then `agent_1` can be marked as knowing this fact. No real observation happens; the planner is simply checking whether a valid sequence of such operators exists.

Step 3: Create inform operators. `Agent_1` can communicate with `agent_0`, so the compiler generates inform operators that propagate first-layer knowledge. Each operator consumes one message token:

```
(:action inform_a0_fact_from_a1_tok1
  :parameters ()
  :precondition (and
    (knows_agent_1_a1b2c3d4)
    (can_communicate agent_1 agent_0)
    (msg_tok_agent_1_1))
  :effect (and
    (knows_agent_0_a1b2c3d4)
    (not (msg_tok_agent_1_1))))
```

A second copy uses `msg_tok_agent_1_2`, giving the planner two opportunities (matching the budget of 2). The `(not (msg_tok_...))` effect consumes the token, preventing reuse.

Step 4: Create nested-knowledge inform operators. For the outer $\mathcal{K}$-goal, the compiler generates an operator that lets `agent_1` inform `agent_0` about its own knowledge state:

```
(:action inform_a0_nested_from_a1_tok2
  :parameters ()
  :precondition (and
    (knows_agent_1_a1b2c3d4)
    (can_communicate agent_1 agent_0)
    (msg_tok_agent_1_2))
  :effect (and
    (knows_agent_0_e5f6g7h8)
    (not (msg_tok_agent_1_2))))
```

The precondition requires that `agent_1` already knows the fact (first layer). The effect sets the second-layer predicate, establishing that `agent_0` now knows `agent_1` knows.

Step 5: Replace the goal. The original epistemic goal is replaced with a conjunction of classical predicates:

```
(and
  (is_on_top bowl_1 table_22)
  (knows_agent_1_a1b2c3d4)
  (knows_agent_0_e5f6g7h8)
  (is_open cabinet_34)
)
```

Step 6: Add budget tokens to the initial state. The problem's `:init` section is augmented with tokens representing `agent_1`'s message budget:

```
(msg_tok_agent_1_1)
(msg_tok_agent_1_2)
```

Result. The compiled problem is entirely classical PDDL. Fast Downward searches over the physical actions (place, open, navigate) together with the observe and inform operators. A valid plan might be:

1. `agent_1` places `bowl_1` on `table_22`.
2. `observe_knows_agent_1_a1b2c3d4` fires (agent_1 sees the bowl is placed).
3. `inform_knows_agent_0_a1b2c3d4_from_agent_1_tok1` fires (agent_1 tells agent_0 the fact, consuming token 1).
4. `inform_knows_agent_0_e5f6g7h8_from_agent_1_tok2` fires (agent_1 tells agent_0 that it knows, consuming token 2).
5. Another agent opens `cabinet_34`.

If Fast Downward finds this plan (or any valid alternative), the task is provably solvable. The $\mathcal{K}$-depth of 2 is read directly from the nesting structure during Step 1.

F. Task generation agent workspace and prompt

The generation agent operates in an isolated workspace directory:

```
workspace/
  working_task.json      # task being
    authored
  template.json          # blank task
    skeleton
  current_scene.json     # rooms,
    furniture, objects, spawns
  sampled_tasks/         # seed tasks (
    biased toward failures)
  submitted_tasks/       # accepted tasks
  available_mechanics.md # mechanic
    registry
  available_predicates.md
  available_actions.md
```

Example scene graph. The `current_scene.json` file describes the loaded Habitat scene. Below is an abridged example for a 2-agent, 5-room scene:

```
{
  "scene_id": "102344280",
  "episode_id": "885",
  "rooms": ["office_1", "dining_room_1",
            "laundryroom_1", "kitchen_2",
            "entryway_1"],
  "furniture": ["table_25", "cabinet_33",
       "couch_9", "counter_30", "
    cabinet_31",
       "table_18"],
  "objects": ["cushion_2", "bowl_4"],
  "articulated_furniture": ["cabinet_33",
                            "cabinet_31"],

  "furniture_in_rooms": {
    "office_1":      ["table_25", "
    cabinet_33"],
    "dining_room_1": ["couch_9"],
    "laundryroom_1": ["counter_30"],
    "kitchen_2":     ["cabinet_31"],
    "entryway_1":    ["table_18"]
  },
  "objects_on_furniture": {
    "table_25": ["cushion_2"],
    "chair_10": ["bowl_4"]
  },
  "agent_spawns": {
    "agent_0": {"position": [1.2, 0.1,
    3.4],
                "room": "office_1"},
    "agent_1": {"position": [5.6, 0.1,
    2.1],
                "room": "dining_room_1"}
  }
}
```

The agent uses this to select goal-relevant objects and furniture, configure room restrictions, and ground the PDDL goal in actual scene entities. Only articulated furniture (cabinets, fridges, drawers) can appear in `is_open`/`is_closed` goals.

System prompt. The agent receives the following system prompt (abridged; full prompt is ∼400 lines):

Generation Agent System Prompt (abridged)

```
You are a puzzle designer creating multi-agent
collaboration challenges.

Response format. Each turn: Thought: [reasoning]
then Action: tool_name[argument].

Tools.
new_scene[N] - load HSSD scene with N agents, reset
task.
bash[cmd] - run shell commands (jq, cat, python3,
...).
judge[] - PDDL and K-depth verification + LLM
quality evaluation.
test_task[] - physically simulate the
all-secrets-public baseline.
submit_task[] - save task (requires judge +
test_task).

Workflow.
1. new_scene[N] → load scene.
2. Inspect seed tasks in sampled_tasks/ for
inspiration.
3. Edit working_task.json: author the
problem_pddl :goal FIRST, then write task,
agent_secrets, and mechanic bindings to match.
Do NOT hand-author :objects or :init.
4. judge[] → fix → repeat until pass.
5. test_task[] → reject tasks that fail with full
information.
6. submit_task[].

Core rules.
- Author the PDDL goal as the source of truth; write
narrative to match it.
- Secrets state WHAT (room restrictions, target
IDs, mechanic hints) but NEVER HOW (no coordination
strategy, no relay instructions).
- Every agent must make a distinct,
non-substitutable contribution.
- At least one physical action must be
information-dependent: an agent cannot determine
what to do without information held by another
agent.
- Do not prescribe coordination strategy in secrets.
The agent must figure out how to communicate.

Secret examples.
BAD (leaks coordination strategy):
agent_0: "Wait for agent_3 to tell you whether
stand_34 is open, then forward that to agent_0."
GOOD (states constraints, agents figure out
coordination):
agent_0: ["You cannot enter hallway_2.", "You can
only message agent_1. You can send 2 messages.",
"By the end, you must be confident a teammate knows
stand_34 is open."]
Functional ToM patterns (use at least one):
1. Delegation choice - agent must choose which teammate
to inform; only one can act on the info.
2. Sequencing choice - correct action order depends on
what a teammate already knows.
3. Relay choice - sender cannot reach the actor
directly; must pick a relay path.
4. Information-gated action - agent's correct action
depends on a fact only another agent can observe.
5. Mixed-motive cooperation - private objectives change
how useful or reliable a teammate is.

Self-tests before submitting.
- Remove all Communicate actions from the golden
trajectory. Can agents still achieve all physical
goals independently? If yes, the task does not
test functional ToM.
- Can the outermost K() agent directly walk to the
room and observe the fact? If yes, the K-goal is
trivial.
```

G. Judge council prompt and feedback

The judge council (Kimi-K2.5 and GPT-5.2) receives the following prompt (abridged):

Judge Council Prompt (abridged)

```
You are an expert evaluator for multi-agent tasks.

Context provided to the judge:
- Task category and category-specific rules
- Available actions, mechanics, items, and
predicates
- Scene objects (rooms, furniture, objects on
furniture)
- The compiled formal view (PDDL after mechanic
compilation)
- Derived runtime semantics (functional goal +
literal-ToM probes)
- The full task JSON (description, secrets,
mechanics, PDDL goal)

Key checks:
- task is global and must not leak secret targets
- Secrets must be actionable (exact IDs) and must
not prescribe strategy
- Every K() goal must be backed by a mechanic
preventing direct observation
- The functional projection (after dropping K-goals)
must remain non-trivial
- K() probes should test who knows functionally
relevant facts under real asymmetry
- Reward tasks where the best action depends on a
partner-specific model; penalize pure fact-relay
tasks

Scoring: Each criterion scored 0.0-1.0.
Output: JSON with per-criterion scores, reasoning
(under 15 words each), and required fixes.
```

Example judge feedback. Below is an actual judge response for a task that failed on secret quality and mechanic utilization (task selected at random from the generation logs):


```
{
"agent_necessity": {"score": 0.7, "reasoning":
"Both agents have distinct room access"},
"secret_quality": {"score": 0.3, "reasoning":
"Agent_0 secret prescribes relay strategy"},
"task_naturalness": {"score": 0.8, "reasoning":
"Clean high-level description"},
"narrative_consistency": {"score": 0.7,
"reasoning": "Matches PDDL goal"},
"goal_relevance": {"score": 0.8, "reasoning":
"All conjuncts needed"},
"mechanic_utilization": {"score": 0.4,
"reasoning": "room_restriction is decorative"},
"pddl_solvability": {"score": 0.9, "reasoning":
"Valid and solvable at K-2"},
"task_interdependence": {"score": 0.6,
"reasoning": "Some parallel execution possible"},
"overall_reasoning": "Secret leaks strategy; room
restriction does not block any goal.",
"required_fixes": [
"Remove strategy hints from agent_0 secret",
"Restrict agent_0 from kitchen_1 where cabinet_28
is"
]
}
```


H. Literal ToM probe prompt

At the end of every episode, each agent receives the following prompt to elicit literal ToM probe answers. The probe identifiers (`k_probe_X`) and predicate vocabulary mirror the planner predicates introduced in Section 3.1; one per-probe specification line is appended for each $\mathcal{K}$-operator extracted from φ.

Literal ToM Probe Prompt (abridged)

```
The episode is over. Do not propose any more
actions.
Using only the episode context above, provide the
requested structured report.
Report one structured answer per probe.
Respond with JSON only:
{"answers":[
{"probe_id":"k_probe_X",
 "predicate":"<predicate_name>|unknown",
 "holds":true|false|null,
 "args":["entity_or_target", ...]},
...
]}
Use predicate "unknown" with holds null and empty
args if the agent does not know.
```

Allowed benchmark predicates and signatures.

Spatial / Relational

```
- (is_on_top x:object y:furniture) - object is on
top of furniture
- (is_inside x:object y:furniture) - object is
inside furniture (container)
- (is_in_room x:object r:room) - object is located
in room
- (is_on_floor x:object) - object is on the floor
- (is_next_to x:object y:object) - object is
adjacent to another object
```

Unary State

```
- (is_open f:furniture), (is_closed f:furniture) -
furniture open/closed
- (is_clean x:object), (is_dirty x:object) - object
clean/dirty
- (is_filled x:object), (is_empty x:object) - object
filled with liquid / empty
- (is_powered_on x:object) - object is powered on
- (is_locked f:furniture) - furniture is locked
```

Agent

```
- (is_held_by x:object a:agent) - object is held by
agent
- (agent_in_room a:agent r:room) - agent is in room
- (has_item a:agent i:item) - agent has item in
inventory
- (has_at_least a:agent i:item) - agent has at least
N of item
- (has_most a:agent i:item) - agent has the most of
item among all agents
- (item_in_container i:item f:furniture) - (planner)
item is hidden inside furniture until opened
```

Mechanic (init-only, do NOT use in pddl_goal)

```
- (is_inverse f:furniture) - inverted open/close
- (mirrors f1 f2), (mirrors_closed f1 f2) - f1
mirrors f2's state / open-close toggle
- (controls f1 f2), (controls_unlocked f1 f2),
(controls_closed f1 f2), (controls_locks f1 f2) -
remote control of state, unlock, open, lock
- (is_restricted a:agent r:room) - agent cannot
enter room
- (is_locked_permanent f:furniture), (requires_item
f:furniture i:item),
(unlocks x:object f:furniture) - key-gated access
- (irreversible_enabled x:object),
(interaction_locked x:object) - one-shot
interactions
- (can_communicate from:agent to:agent) -
directional messaging permitted
```

```
For every answer, use the exact predicate name
and the exact argument order required by that
predicate.

Per-probe specifications (one example shown).
- k_probe_1:  Predict what agent_2 would report
about "cabinet 31 is open".  Use ordered entities
[cabinet_31] and the benchmark predicate vocabulary
above.
```

I. Key design decisions

Several design choices in the generation pipeline were informed by failure modes observed during early development. We document each decision and the problem it addresses.

Why agent necessity. Early generated tasks often included agents that contributed nothing: an agent would be assigned to a room but have no goal-relevant object there, or two agents would have identical access and capabilities, making one redundant. When benchmarked, the redundant agent would simply idle (Wait actions for the entire episode) while the other completed the task alone. The *agent necessity* criterion rejects tasks where any agent can be removed without breaking the intended solution. This forces the generator to design tasks where each agent holds unique access, information, or physical capability.

Why secrets must not prescribe strategy. In early experiments, secrets contained instructions like "ask agent_1 to open the fridge and relay the result to agent_2." Agents followed these instructions verbatim and achieved near-perfect pass rates, but the task reduced to instruction following, not epistemic reasoning. The agent never needed to model what others know or can do; the secret told it exactly what to communicate and to whom. The *secret quality* criterion now rejects any secret that leaks coordination strategy. Secrets state constraints ("you cannot enter kitchen_1"), targets ("cabinet_28 must end open"), and mechanic hints ("the handle is reversed"), but never the plan.

Why public/private grounding. If the shared task description δ contains exact object IDs and target locations, all agents receive the same complete information and there is no reason to communicate. Information asymmetry arises only when δ stays high-level ("reset the house for inspection") and the actionable specifics (which cabinet, which room, which object) are distributed across private secrets $\Sigma(a_i)$. This split is what creates the need for agents to share information selectively.

Why PDDL goal before narrative. When the generation agent wrote the natural-language description first, it frequently invented requirements not expressible in PDDL (e.g., "agents should feel satisfied with the arrangement") or omitted requirements that were in the formal goal. Writing φ first and deriving δ and Σ from it eliminated this class of inconsistencies.

Why a two-model council. A single judge model exhibited systematic biases: GPT-5.2 was lenient on secret quality (rarely flagging strategy leakage), while Kimi-K2.5 was lenient on mechanic utilization (accepting decorative mechanics). Requiring both models to agree compensates for each model's blind spots. Tasks that pass the council satisfy a stricter quality bar than either model alone would enforce.

Why baseline calibration. The baseline condition (all secrets public) serves two purposes. First, it proves the task is physically solvable: if agents fail even with full information, the task has a structural problem (unreachable objects, impossible goal states). Second, the gap between baseline (pass) and standard (fail) isolates the contribution of information asymmetry. Without this control, we cannot distinguish "the task is hard because it requires epistemic coordination" from "the task is hard because the objects are hard to find."

Why not give the planner to the evaluated agent. The epistemic planner (Section 4.3) has access to all agents' secrets, room restrictions, and the complete goal formula simultaneously. It solves the task as an omniscient coordinator. If we gave this planner as a tool to the evaluated agent, the agent could query it to determine what every other agent knows, what information to communicate, and in what order. This would bypass epistemic reasoning entirely. The whole point of the benchmark is that the agent must *infer* what its partners know from their room access, communication history, and behavior. Handing the agent an oracle that answers "what does agent_1 know?" would reduce the task to tool calling, not Theory of Mind.

J. Real-world grounding of mechanics

Each mechanic in `EnactToM` models a constraint that commonly arises in real-world multi-agent systems.

Room restriction. A warehouse fulfillment center assigns robots to specific zones. A robot on the packing floor cannot observe inventory levels on the storage floor. To coordinate a restock, the packing robot must communicate its needs to a storage robot that can verify shelf state directly.

Limited bandwidth. A search-and-rescue team operates on a shared radio frequency with limited airtime per responder. Each transmission must carry the most critical information first. A responder who wastes a transmission asking for confirmation of already-known facts may not have airtime left to relay a newly discovered survivor location.

Restricted communication. In a hospital, a nurse reports to the attending physician, not directly to the specialist in another department. If the specialist needs information from the nurse, the physician must relay it. The communication topology determines who can inform whom and how many

hops a piece of knowledge must travel.

Remote control. A smart-home system links a wall switch in the hallway to a heater in the bedroom. The person operating the switch cannot see whether the heater actually turned on. They must either walk to the bedroom to verify, or ask someone already in the bedroom to confirm.

State mirroring. Two networked thermostats in different rooms are synchronized: adjusting one changes the setting on both. An occupant in one room who lowers the temperature may not realize they also lowered it in a room where someone else prefers it warm. Coordination requires knowing who else is affected by the shared state.

Inverse state. A pressure release valve works opposite to intuition: turning it clockwise releases pressure rather than increasing it. An operator unfamiliar with this mapping will produce the wrong effect. In a team setting, the operator who knows the mapping must communicate it to others before they interact with the valve.

K. $\mathcal{K}$-depth validity study

We manually verify whether the stated $\mathcal{K}$-depth in the PDDL goal matches the actual epistemic reasoning required. For each of 50 randomly sampled tasks from the mixture-optimized pool, we read the goal, agent instructions, room restrictions, and communication topology. For each $\mathcal{K}$-goal, we check: (a) is the outermost agent barred from the room where the fact holds? (b) can the inner agent directly message the outer agent, or must knowledge relay through intermediaries?

A task's $\mathcal{K}$-depth is **valid** if the outermost agent cannot directly observe the fact and must rely on communication or inference to learn about the inner agent's knowledge state. A task is **inflated** if the outermost agent can directly observe the inner agent performing the action (e.g., both agents are in the same room with no restriction).

Table 6. $\mathcal{K}$-depth validity on 50 randomly sampled tasks. Valid: the outermost agent is barred from the fact room and must rely on communication to satisfy the $\mathcal{K}$-goal. Inflated: the outer agent can directly observe the fact, making the $\mathcal{K}$-goal trivially satisfiable without epistemic reasoning.

$\mathcal{K}$-level	Total	Valid	Inflated
$\mathcal{K}$-1	20	20 (100%)	0
$\mathcal{K}$-2	13	12 (92%)	1 (8%)
$\mathcal{K}$-3	17	17 (100%)	0
All	50	49 (98%)	1 (2%)

49 of 50 tasks (98%) have valid $\mathcal{K}$-depth. All $\mathcal{K}$-1 and $\mathcal{K}$-3 tasks are valid. The single inflated task is a $\mathcal{K}$-2 task where neither agent is restricted from the room containing the target furniture, so both can observe the state directly.

Table 7. Task-generation cost. "Final benchmark" counts tasks that survived the full pipeline (PDDL verification, ToM scoring, calibration, judge curation).

Measure	Cost
Per worker attempt (win or lose)	∼\$0.52
Per task that passes generation	∼\$0.74
Per task kept in the final benchmark	∼\$9.70
Wall-clock per attempt (mean)	∼13 min (range 6–40 min)
Parallel workers	∼24

Among the valid tasks, 33 (67%) require genuine multi-hop reasoning due to restricted communication (the inner agent cannot directly message the outer agent), while 16 (33%) are achievable via a single direct message. In both cases, the outermost agent is barred from the fact room and must model whether the inner agent has observed the fact. The stated $\mathcal{K}$-depth is correct in both cases; the difference is in the communication complexity, not the epistemic depth.

L. Compute and API cost

All experiments run via LLM APIs, so we report wall-clock time and dollar cost in lieu of GPU-hours. Figures cover the runs reported in the paper.

Task generation. Generating one task runs the full pipeline described in Section 4.2: the generation agent proposes a candidate, which then passes through PDDL verification, ToM scoring, baseline calibration, and judge curation. Most candidates are filtered out, so per-accepted-task cost reflects ∼13× more attempts than tasks kept. Table 7 reports cost at three accounting granularities; the relevant one depends on the question being asked.

The \$0.74 figure is the raw generation cost; \$9.70 reflects the full pipeline cost, since most generated tasks are filtered out by PDDL verification, ToM scoring, calibration, and curation. As a rough rule of thumb, producing one benchmark-quality task costs \$1–10 depending on how strict the acceptance criteria are, and takes 1–2 hours of single-threaded wall-clock time end-to-end.

Evaluation. Each benchmark task is run n=3 times per model. Per-task evaluation cost varies by two orders of magnitude across the models in Table 2, ranging from ∼\$0.0020 (Gemini-Flash) to ∼\$0.2560 (GPT-5.4). Wall-clock per task averages ∼16 min.

Table 8. Per-task evaluation cost. Each task is run n=3 times per model; cost is per single run.

Measure	Value
Cheapest model (Gemini-Flash) per task	∼\$0.0020
Most expensive model (GPT-5.4) per task	∼\$0.2560
Wall-clock per task (mean)	∼16 min

Table 9. Effect of removing each pipeline component.

Ablation	Failure Rate	Avg. Pass Rate
Full pipeline	– –	26.7%
w/o baseline calibration	51%	– –
w/o ICL seed examples	50%	– –
w/o secret quality check	– –	43.5%

M. Detailed ablation studies

Without baseline calibration. The benchmark gate normally runs the target model in two modes: *standard* (partial information) and *baseline* (full information), accepting only tasks where standard fails but baseline succeeds. This ensures difficulty comes from information asymmetry, not from the task being fundamentally unsolvable. Removing the baseline run, 51% of generated tasks are unsolvable even with full information.

Without LLM council judge. The judge scores each task on 8 quality criteria using a two-model council. A task passes only if its overall score is ≥ 0.65 and every individual criterion scores ≥ 0.5. Without the judge, 50–90% of tasks fail post-hoc quality checks depending on the council model strength. A stronger judge model catches more issues: mechanics that are present but not load-bearing, secrets that leak exact object IDs, and mixed tasks whose private goals duplicate rather than conflict with shared goals.

Without ICL seed examples. Without in-context seed tasks biased toward frontier model failures, the generation agent defaults to simpler coordination patterns. The acceptance rate drops to roughly 50% and accepted tasks have lower epistemic depth. Figure 3 shows the cumulative task yield with and without ICL.

Without secret quality check. Relaxing the secret quality constraint allows secrets to include coordination instructions (e.g., "ask agent_1 to open the fridge and relay the result to agent_2"). When secrets prescribe how to coordinate, the average pass rate rises from 26.7% to 43.5%. Agents follow instructions verbatim without modeling partner knowledge, confirming the check is essential for measuring epistemic coordination rather than instruction following.

N. Theory of Mind in Cognitive Science

Theory of Mind was first posed as an empirical question by (Premack & Woodruff, 1978), who asked whether chimpanzees attribute mental states to others. The subsequent decades of research have produced a rich decomposition of the construct that informs how we evaluate it in artificial agents.

The earliest ToM precursors concern visual perspective-taking. (Masangkay et al., 1974) showed that children as young as two can judge what another person sees, while (Flavell, 1977; Flavell et al., 1981; Flavell, 1992) introduced the Level 1 / Level 2 distinction: knowing *that* someone can see something versus knowing *how* it appears to them. (Flavell et al., 1978) demonstrated that young children can reason about hiding objects from others — an early form of modelling informational access. These findings ground EnactToM's use of room restrictions and partial observability as the basic mechanism for creating epistemic asymmetry.

The false-belief task (Wimmer & Perner, 1983; Baron-Cohen et al., 1985) became the gold standard for assessing whether an agent can represent a belief diverging from reality. (Wellman et al., 2001) established that children reliably pass around age four, and (Wellman & Liu, 2004) showed that ToM capacities form a Guttman scale — each level a prerequisite for the next, not a continuous variation — motivating EnactToM's discrete epistemic depth levels. Second-order false belief emerges later and is substantially harder even for adults (Perner & Wimmer, 1985). A major theoretical development is the two-systems account of (Apperly & Butterfill, 2009): a fast, automatic system for tracking belief-like states (Onishi & Baillargeon, 2005) and a slower system for deliberate propositional reasoning. Critically, these dissociate in adults — (Keysar et al., 2003) showed that people with full access to a partner's perspective still default to egocentric interpretations under load. The dissociation between implicit tracking and explicit report that (Apperly & Butterfill, 2009) describe in humans is consistent with the gap between functional and literal scores observed in our evaluation 2, where several models score substantially higher on literal ToM probes than on functional task completion.

(Dennett, 1978) argued that the critical test of ToM is attribution of false beliefs, and (Dennett, 1987) formalised the "intentional stance" — predicting behaviour by attributing beliefs and rational agency. (Baker et al., 2009) computationally formalised this as Bayesian inverse planning, later extended to joint inference over beliefs, desires, and percepts (Baker et al., 2017). On the coordination side, (Bratman, 1992) and (Tomasello et al., 2005) characterised shared cooperative activity as requiring mutual responsiveness and shared intentionality, while (Vesper et al., 2010) proposed a minimal joint-action architecture built on prediction, monitoring, and coordination smoothing that does not require full recursive mentalising. EnactToM's cooperative tasks are designed to require at least this minimal architecture.

Finally, the formal apparatus for nested knowledge traces to epistemic logic (Aumann, 1976; 1999). The connection to strategic reasoning runs through level-k models: (Stahl & Wilson, 1994; 1995) and (Camerer et al., 2004) showed that humans reason at finite, heterogeneous depths, (Nagel, 1995) demonstrated bounded iterated reasoning in guess-

ing games, and (Crawford & Iriberri, 2007) applied level-k analysis to spatial hide-and-seek — structurally similar to EnactToM's embodied coordination. These models predict heterogeneous and bounded depth of reasoning, which EnactToM's per-depth evaluation is designed to measure.