Inferring the Future by Imagining the Past

Author Names Omitted for Anonymous Review.

***Abstract*—A single panel of a comic book can say a lot: it shows not only where characters currently are, but also where they came from, what their motivations are, and what might happen next. More generally, humans can often infer a complex sequence of past and future events from a *single snapshot image* of an intelligent agent.**

Building on recent work in cognitive science, we offer a Monte Carlo algorithm for making such inferences. Drawing a connection to Monte Carlo path tracing in computer graphics, we borrow ideas that help us dramatically improve upon prior work in sample efficiency. This allows us to scale to a wide variety of challenging inference problems with only a handful of samples. It also suggests some degree of cognitive plausibility, and indeed we present human subject studies showing that our algorithm matches human intuitions in a variety of domains that previous methods could not scale to.

I. Introduction

Hemingway's shortest short story simply reads "For sale: baby shoes, never worn." There is no action in this sentence—however, readers nonetheless infer a complex and tragic backstory from the single static snapshot Hemingway provides. This remarkable ability comes naturally to humans: we routinely reconstruct motives from evidence (e.g. at a crime scene), recognize intentions from unfinished tasks (e.g. grading incomplete homework), and enjoy artistic depictions of dynamic action in static drawings (e.g. a Renaissance "tableau" or a comic book panel).

How do we do it? Decades of work in both AI and cognitive science (see Section IV) has successfully addressed the simpler problem of inferring an agent's goal from a *trajectory* of observed actions. These methods infer $P(\text{goal} \mid \text{actions}) \propto P(\text{actions} \mid \text{goal})P(\text{goal})$, where $P(\text{actions} \mid \text{goal})$ is modeled by comparing the observed actions to the optimal actions a rational agent would take towards that goal.

But if we only observe a single state snapshot, this method breaks down—there are simply no actions to condition on. Instead, we must jointly infer not only where the agent might be going, but also where it came from. Recently, Lopez-Brau et al. [28, 29] performed this inference by rejection-sampling possible paths taken by the agent. Their model's predictions are remarkably close to human judgements. However, rejection sampling is extremely inefficient—it is slow even on simple problems, and simply does not scale to more sophisticated problems, suggesting that there is more to how humans perform such inference.

In this paper, we propose a solution: inspired by the wealth of Monte Carlo sampling algorithms for path tracing in computer graphics, we consider sampling paths *bidirectionally*. This leads to a dramatically more efficient sampling scheme that scales to more sophisticated problems. Specifically, we make the following contributions:

1) In Section II, we review how the problem is formalized and present our Monte Carlo algorithm for sampling approximate solutions. Our algorithm is up to $30,000\times$ more efficient than prior work, and lends itself to a natural cognitively-plausible implementation.
2) We extend prior work to support not only Markov Decision Processes (MDPs) as in prior work, but also on-line (classical) planning domains where possible, in order to avoid expensive pre-computation of policies (Section II-C).
3) Via three behavioral studies, we demonstrate that our model's predictions match human judgements on new, scaled-up tasks inaccessible to prior work (Section III-B and Appendix C).

II. Proposed algorithm

Consider an agent who begins in some initial state $s \sim p(s)$ and acts rationally to reach some goal g. For now, let us follow prior work in taking the agent's domain to be a Markov Decision Process (MDP), though we will later relax this assumption. In an MDP, the goal g might be modeled as a terminal state that the agent receives high reward for reaching.

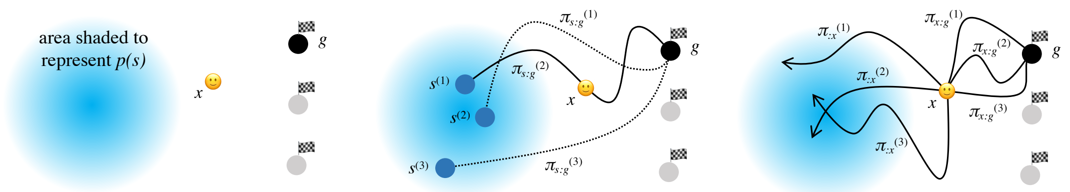
While the agent is on its trajectory from s to g, we observe a "snapshot" of the agent in some state x. Given only x (and not s!), our goal is to infer $p(g \mid x)$. Applying Bayes' rule, we have $p(g \mid x) \propto p(x \mid g)p(g)$. To evaluate the likelihood $p(x \mid g)$, we apply the Law of Total Probability over possible start states s, and then again over state sequences (or "paths") $\pi_{s:g}$ from s to g.

$$
\begin{aligned}
p(x \mid g) &= \int_s p(x \mid s, g)p(s \mid g)ds \\
&= \int_s \int_{\pi_{s:g}} p(x \mid \pi_{s:g}, s, g)p(\pi_{s:g} \mid s, g)p(s)d\pi_{s:g}ds
\end{aligned}
$$

(Note that $p(s \mid g) = p(s)$ because we assume s and g are independent.)

To evaluate the likelihood of a snapshot $p(x \mid \pi_{s:g}, s, g)$, we apply the *size principle* [46, 47, 19], analogous to the *generic viewpoint assumption* in computer vision [16, 1]. In this case, the principle states that the snapshot was equally likely to have been taken anywhere along the path, and therefore the likelihood of a snapshot conditioned on a path is inversely proportional to the length of the path. If $\delta(x \in \pi)$ indicates whether path π passes through x, and $|\pi|$ indicates the length of π, then $p(x \mid \pi_{s:g}, s, g)$ is given by $\delta(x \in \pi_{s:g})|\pi_{s:g}|^{-1}$.

To evaluate the likelihood of a path $p(\pi \mid s, g)$, we apply the *principle of rational action:* agents are likelier to take actions that maximize their utility [14, 23]. We formalize this intuition by saying that at each step, the agent chooses an



Suppose you observe an agent at x. If they started from some $s \sim p(s)$, what is the likelihood they are going to goal g?

Lopez-Brau et al. sample $s \sim p(s)$ and then paths from s to g. However, most paths do not pass through x…

We instead sample two kinds of paths: forward from x to g, and backwards from x through the space of possible s's.

Fig. 1. How can we infer what an agent is trying to do, based on a snapshot of its current state?

action with probability proportional to the softmax over its Q-values at its current state, with some temperature β. That is, $p(x \to x' \mid g) \propto \sum_a \exp(\beta Q_g(x,a))\mathrm{Tr}(x,a,x')$, where $\mathrm{Tr}(x,a,x')$ is the transition probability from x to x' if action a is taken, and $p(\pi \mid g) \propto \Pi_t p(x_t \to x_{t+1} \mid g)$.

We now have all the ingredients we need to evaluate $p(x \mid g)$. However, to compute it exactly we would need to integrate over all possible initial states s, and the set of paths $\pi_{s:g}$, which could be infinite (agents might wander for arbitrarily long, albeit with vanishingly low probability). To approximate the likelihood in finite time, Lopez-Brau et al. turn to Monte Carlo sampling (Algorithm 1). They rejection-sample paths $\pi_{s:g}$ by sampling a candidate start state $s^{(i)} \sim p(s)$, simulating a rollout of the agent to sample a path $\pi_{s:g}^{(i)} \sim p(\pi_{s:g}^{(i)} \mid s^{(i)}, g)$, and then averaging the integrand over these samples. With N samples, their unbiased likelihood estimator is given by $\hat{p}(x \mid g) = \frac{1}{N}\sum_{i=1}^{N} \delta(x \in \pi_{s:g}^{(i)})|\pi_{s:g}^{(i)}|^{-1}$.

Unfortunately, in practice this scheme is extremely slow: even in a 7×7 gridworld with fewer than 49 states (only 2 of which were possible initial states), Lopez-Brau et al. report taking over 300,000 trajectory samples per goal to perform inference. In the rest of this section, we will describe a series of algorithmic enhancements that allow for comparable inference quality with just 10 samples per goal (i.e. $30,000\times$ fewer). We will develop our algorithm (Alg 2) through three insights.

A. First insight: only sample paths through the observed state

Our first insight is that $\delta(x \in \pi)$ is extremely sparse—most paths likely do not pass through x, and so most naïve path samples contribute zero to the estimator. We would like to only sample paths that pass through x. Any such path can be partitioned at x into two portions, $\pi_{s:x}$ and $\pi_{x:g}$. Let us integrate separately over those portions.

$$p(x \mid g) = \int_s \int_{\pi_{s:x}} \int_{\pi_{x:g}} \frac{p(\pi_{s:x} \mid g)\ p(\pi_{x:g} \mid g)}{|\pi_{s:x}| + |\pi_{x:g}|} p(s)\ d\pi_{x:g}\ d\pi_{s:x}\ ds$$

This already suggests a more efficient Monte Carlo sampling scheme: rather than rejection-sampling paths $\pi_{s:g}^{(i)}$ from s to g, we can independently sample two paths: a "past" path $\pi_{s:x}^{(i)}$ from s to x, and a "future" path $\pi_{x:g}^{(i)}$ from x to g. Any such path is guaranteed to pass through x; no samples are wasted.

However, we now have two new problems. First, it is not clear how to sample paths $\pi_{s:x}^{(i)}$ from s to x, because rollouts of a simulated agent are unlikely to pass through x on their way to g. We could imagine using a *second* planner just to chart paths from s to x, but this would require a lot of additional planning work. Second, we still have to sample $s^{(i)}$. If the space of initial states is small (e.g. a room only has one or two doors), then this is no issue. However, in practice this space might be very large or even infinite. For example, if you observe someone driving to work in the morning, their home could be anywhere in the city. Furthermore, most of these states might be inaccessible or otherwise implausible, and it would be a waste of computational resources to consider them. In the next section, we show how to solve both of these problems by tracing paths *backwards in time*.

B. Second insight: sample backwards in time

Our second insight is that we can collapse the first two integrals by jointly integrating over the domain of all paths $\pi_{:x}$ that terminate at x, no matter where they started from. Say a path $\pi_{:x}$ begins at $\pi_{:x}[0]$. Then, we can rewrite our likelihood as below.

$$p(x \mid g) = \int_{\pi_{:x}} \int_{\pi_{x:g}} \frac{p(\pi_{:x} \mid g)\ p(\pi_{x:g} \mid g)}{|\pi_{:x}| + |\pi_{x:g}|} p(\pi_{:x}[0])\ d\pi_{x:g}\ d\pi_{:x}$$

This suggests that we should sample $\pi_{:x}^{(i)}$ *backwards* through time, starting from x. No matter how we extend this path, we obtain a valid path from $\pi_{:x}^{(i)}[0]$ to x.

An analogy to path tracers in computer graphics may be helpful. When rendering a 3D scene, a renderer must integrate over all paths of light that begin at a light source in the scene and end at a pixel on the camera's film—a problem formalized by the rendering equation [24]. Of course, these paths may be reflected and refracted by several surface interactions along the way. Rather than starting at one of the millions of light sources in the scene and tracing a ray hoping to eventually reach the camera film, renderers instead start at the camera and trace rays *backward into the scene* until they reach a light source. Similarly, here we trace paths backwards from x into the past—s corresponds to a light source, each action taken by the agent corresponds to a surface interaction, and x corresponds to a camera pixel. Indeed, our integral is analogous to the rendering equation, bringing to our disposal the entire Monte Carlo light transport toolbox—a toolbox the rendering community has spent decades developing [50, 37]. The particular ideas we borrow are importance sampling, Russian roulette path termination [9, 5], and bidirectional path tracing [26, 51].

C. Third insight: plan on-line via incremental A-star search

One last dissatisfying aspect of this algorithm is that it requires an expensive pre-computation of Q-functions for all possible goals and states. It seems implausible that humans do this, because we make judgements so quickly, even in new domains. Thus, we extend our algorithm to classical planning domains, where algorithms such as A-star search provide a lightweight on-line source of information. We can compute $p(x \to x' \mid g)$ by taking a softmax over the difference in path costs between x and x' to g as given by a planner: $p(x \to x' \mid g) \propto \sum_a \exp\left(\beta(C(x \to g) - C(x' \to g)\right)$, so that the agent is more likely to move to states that will bring them closer to the goal. To avoid re-planning from scratch for every evaluation of $p(x_{t-1} \to x_t \mid g)$, we run A-star backwards from the goal. This lets us re-use intermediate computations (known distances, evaluations of the heuristic, etc.) between queries.

III. Experiments

To evaluate our sampling algorithm, we chose a suite of benchmark domains from prior work.

Simple gridworld: We re-implement the 7×7 gridworld domain from Lopez-Brau et al. The agent seeks one of three gems in the gridworld and can move north, south, east or west. The inference task is to look at a snapshot image and determine which gem the agent seeks. While Lopez-Brau et al. fix two possible starting-points ("entryways") for the agent, we optionally relax this constraint and instead have a uniform prior over the start state (see Figure 2).

Doors, keys, and gems (multi-stage planning): This is a more advanced 8×8 gridworld, inspired by Zhi-Xuan et al. [54]. The agent is blocked from its gem by *doors*, which can only be opened if the agent is carrying the correct *keys*. The inference task is to look at a snapshot image and determine which gem the agent seeks (see Figure 2).

Word blocks (non-spatial): In this domain, the agent spells a word out of the six letter blocks by picking and placing them in stacks on a table. However, they are interrupted (e.g. by a fire alarm) and have to leave the room before finishing. The inference tasks are to look at the blocks left behind and determine (a) which word the agent was trying to spell, and (b) which blocks the agent touched.

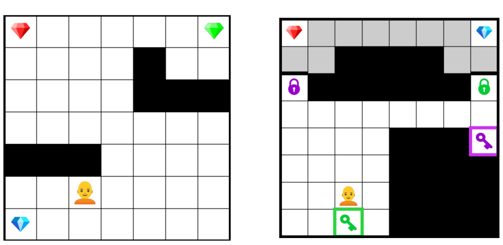


Fig. 2. **(left)** In this example of the "grid" domain, we observe an agent near the blue gem. Even though we do not know where the agent started from, our intuition says that the agent wants the blue gem. **(right)** In this example of the "keys" domain, we observe an agent right next to the green key. Humans infer that the agent is heading towards the green key because it wants the blue gem. Our algorithm replicates both of these inferences with only 10 samples.

A. Results

Table I shows some example inferences made by our algorithm. With just 10 samples, our method's posterior inferences are near-convergent and align well with human responses. In comparison, with 10 samples rejection sampling typically produces extremely noisy predictions, and often simply fails to produce any non-rejected samples at all.

Quantitatively, we report the total variation $\mathrm{TV}(x) = \frac{1}{2}\sum_{g_i} |\hat{p}(g_i \mid x) - p(g_i \mid x)|$ between the true posterior and inferences made using 10 samples of both our method and rejection sampling, averaged for 100 trials and across all of the inference tasks in the benchmark. We take the true posterior to be our method's estimate with 1,000 samples. Our results are shown in Table II. Across all domains, our algorithm substantially outperforms rejection sampling.

B. Comparison to human judgements

We recruited $N = 200$ participants and collected judgements for a variety of "snapshots" in each domain. Our model predicts human intuitions quite well (see Table I, right). Showing that findings of previous work continue to hold in domains that previous algorithms could not scale to.

IV. Related work

Human social cognition and "theory of mind" are well-modeled by **Bayesian inverse planning** [7, 22, 8, 25, 53, 49, 32, 34, 54], inferring an agent's goals from its observed actions. Lopez-Brau et al. [28, 29], building on past work [43, 18, 27, 36, 21], ask how people make inferences about agents from static evidence. We extend their work in this paper.

Our work provides a method for **plan recognition** [39, 40, 44, 12] from a single snapshot. Relatedly, Shah et al. [42] recently propose **inverse reinforcement learning** [33, 4, 55] from a single state. We build on their work in three ways: **(1)** Their work assumes paths of a fixed length, whereas we integrate over trajectories of all possible lengths. **(2)** We do not assume the snapshot was taken at the end of the agent's journey. **(3)** Our method scales to larger domains because we do not integrate exhaustively over trajectories.

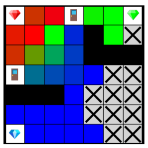
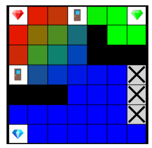
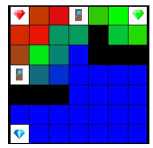
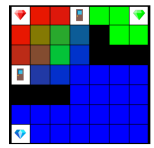
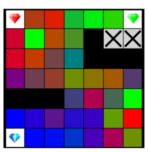
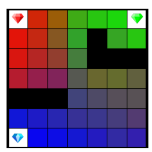
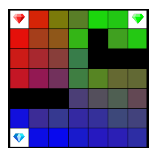
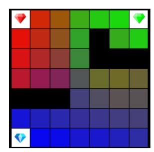
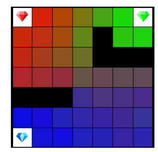
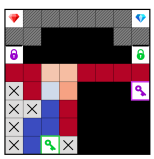
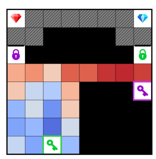
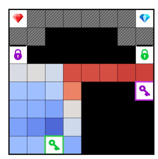
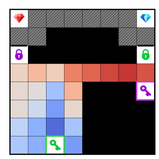
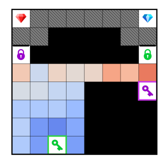
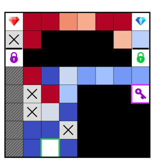
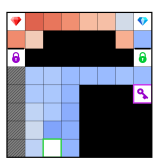
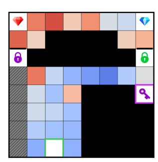
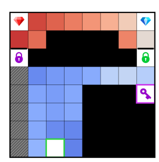
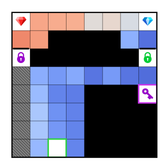
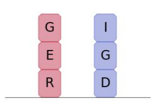
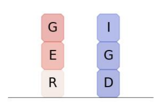
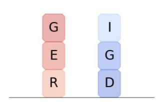
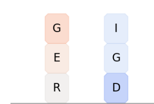
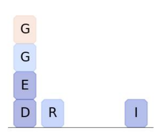
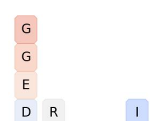
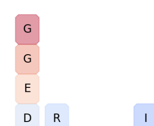
V. Limitations and future work

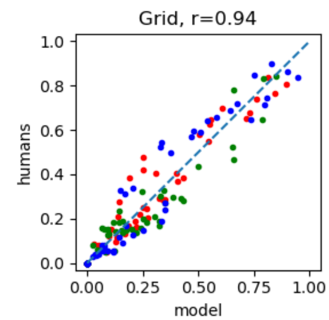
Sampling over goals: We currently compute posteriors by enumeration over all goals, which takes linear time in the size of the goal space. We hope to scale to larger or continuous goal spaces via pseudo-marginal Monte Carlo methods [3].

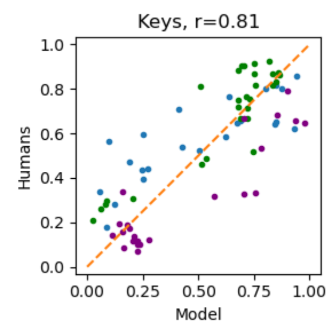
Cognitive plausibility: Our method's sample efficiency suggests that it may resemble how humans do this task [52]. Following previous work [17], we hope to use eye-tracking to compare humans to our algorithm. In the language of Marr [30], this would allow us to go beyond a *computational* account and towards an *algorithmic* account.

TABLE I

QUALITATIVE COMPARISON OF INFERENCE ALGORITHMS. (LEFT) FOR "GRIDWORLD" AND "KEYS," CELLS ARE COLORED BASED ON THE POSTERIOR DISTRIBUTION OVER GOALS IF THE AGENT IS OBSERVED IN THAT CELL. CELLS MARKED × HAD ALL SAMPLES REJECTED. GRAY CELLS WERE EXCLUDED FROM ANALYSIS BECAUSE IT WOULD BE IRRATIONAL FOR THE AGENT TO BE THERE FOR ANY GOAL. FOR "BLOCKS," EACH BLOCK IS COLORED ACCORDING TO INFERRED PROBABILITY OF IT HAVING BEEN TOUCHED (RED HIGH, BLUE LOW). WE SHOW RESULTS FOR 10 SAMPLES AND 1,000 (1K) SAMPLES, COMPARING REJECTION SAMPLING, OUR METHOD, AND HUMAN SUBJECTS. **WE PRODUCE NEAR-CONVERGENT INFERENCES WITH ONLY 10 SAMPLES,** AND OUR METHOD QUALITATIVELY MATCHES HUMANS. IN COMPARISON, REJECTION SAMPLING IS OFTEN UNABLE TO MAKE ANY INFERENCE WITH 10 SAMPLES, AND SOMETIMES EVEN FAILS WITH 1,000 SAMPLES. WHEN IT SUCCEEDS, ITS PREDICTIONS ARE HIGH-VARIANCE AND OVERCONFIDENT. (RIGHT) OUR MODEL INFERENCES MATCH HUMAN RESPONSES ON ALL THREE TASKS.

Task	Rejection (10)	Rejection (1k)	Ours (10)	Ours (1k)	Humans
Grid (two doors, as in [28])					Previously shown to be matched by model [29]
Grid (start anywhere)					
Keys (holding no key)					
Keys (holding green key)					
Blocks 1 (spelling "DIGGER")	(all samples rejected)				
Blocks 2 (spelling "RIGGED")	(all samples rejected)	(all samples rejected)			





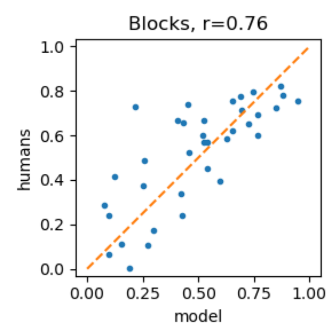


TABLE II

QUANTITATIVE COMPARISON OF INFERENCE ALGORITHMS. WE SHOW THE TOTAL VARIATION DISTANCE (TV) OF A 10-SAMPLE POSTERIOR ESTIMATE, AVERAGED OVER 100 TRIALS. **LOWER IS BETTER.**

Benchmark	Rejection TV	Ours TV
Grid (two doors, as in [28])	0.063	0.0257
Grid (starting anywhere)	0.159	0.0538
Keys (observed holding no key)	0.818	0.215
Keys (observed holding pink key)	0.777	0.314
Keys (observed holding green key)	0.762	0.239
Blocks	0.985	0.358

Beyond inference: Artists have long represented dynamic action in static scenes [31]. We hope to consider the inverse problem of designing evocative scenes by optimizing *over* inference [10, 11, 15].

VI. CONCLUSION

We offered an algorithm for making inferences about the past and future of an intelligent agent based on an observed present. Building on prior work from cognitive science and AI, and drawing inspiration from Monte Carlo rendering, we presented a sample-efficient algorithm and showed that it matches human intuitions on a variety of challenging tasks.

Note: supplementary appendix attached after references.

References

[1] Marc K Albert and Donald D Hoffman. The generic-viewpoint assumption and illusory contours. *Perception*, 29(3):303–312, 2000. URL https://journals.sagepub.com/doi/pdf/10.1068/p3016.

[2] Luke Anderson, Tzu-Mao Li, Jaakko Lehtinen, and Frédo Durand. Aether: An embedded domain specific sampling language for monte carlo rendering. *ACM Transactions on Graphics (TOG)*, 36(4):1–16, 2017. URL https://dl.acm.org/doi/pdf/10.1145/3072959.3073704.

[3] Christophe Andrieu and Gareth O Roberts. The pseudo-marginal approach for efficient monte carlo computations. 2009. URL https://www.jstor.org/stable/30243645.

[4] Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, 297:103500, 2021. URL https://www.sciencedirect.com/science/article/am/pii/S0004370221000515.

[5] James Arvo and David Kirk. Particle transport and image synthesis. In *Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 63–66, 1990. URL https://dl.acm.org/doi/10.1145/97880.97886.

[6] Chris L Baker, Noah D Goodman, and Joshua B Tenenbaum. Theory-based social goal inference. In *Proceedings of the thirtieth annual conference of the cognitive science society*, pages 1447–1452. Cognitive Science Society Austin, TX, 2008. URL https://cocolab.stanford.edu/papers/BakerEtAl2008-Cogsci.pdf.

[7] Chris L Baker, Rebecca Saxe, and Joshua B Tenenbaum. Action understanding as inverse planning. *Cognition*, 113 (3):329–349, 2009. URL https://www.sciencedirect.com/science/article/pii/S0010027709001607.

[8] Chris L Baker, Julian Jara-Ettinger, Rebecca Saxe, and Joshua B Tenenbaum. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nature Human Behaviour*, 1(4):0064, 2017. URL https://www.nature.com/articles/s41562-017-0064.

[9] Leland Lavele Carter and Edmond Darrell Cashwell. Particle-transport simulation with the monte carlo method. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 1975.

[10] Kartik Chandra, Tzu-Mao Li, Joshua Tenenbaum, and Jonathan Ragan-Kelley. Acting as inverse inverse planning. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Proceedings (SIGGRAPH '23 Conference Proceedings)*, aug 2023. doi: 10.1145/3588432.3591510.

[11] Kartik Chandra, Tzu-Mao Li, Joshua Tenenbaum, and Jonathan Ragan-Kelley. Storytelling as inverse inverse planning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 45, 2023.

[12] Eugene Charniak and Robert P Goldman. A bayesian model of plan recognition. *Artificial Intelligence*, 64(1):53–79, 1993. URL https://www.researchgate.net/profile/Robert-Goldman-5/publication/222330914_A_Bayesian_model_of_plan_recognition/links/5c7d77d0299bf1268d390c71/A-Bayesian-model-of-plan-recognition.pdf.

[13] Marco F Cusumano-Towner, Feras A Saad, Alexander K Lew, and Vikash K Mansinghka. Gen: a general-purpose probabilistic programming system with programmable inference. In *Proceedings of the 40th acm sigplan conference on programming language design and implementation*, pages 221–236, 2019.

[14] Daniel C Dennett. *The intentional stance*. MIT press, 1989.

[15] Anca D Dragan, Kenton CT Lee, and Siddhartha S Srinivasa. Legibility and predictability of robot motion. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 301–308. IEEE, 2013. URL https://ieeexplore.ieee.org/iel7/6476064/6483487/06483603.pdf.

[16] William T Freeman. The generic viewpoint assumption in a framework for visual perception. *Nature*, 368(6471): 542–545, 1994. URL https://www.nature.com/articles/368542a0.pdf.

[17] Tobias Gerstenberg, Matthew F Peterson, Noah D Goodman, David A Lagnado, and Joshua B Tenenbaum. Eye-tracking causality. *Psychological science*, 28(12):1731–1744, 2017.

[18] Tobias Gerstenberg, Max Siegel, and Joshua Tenenbaum. What happened? reconstructing the past through vision and sound. 2021. URL https://psyarxiv.com/tfjdk/.

[19] Thomas L Griffiths and Joshua B Tenenbaum. Optimal predictions in everyday cognition. *Psychological science*, 17(9):767–773, 2006. URL https://journals.sagepub.com/doi/pdf/10.1111/j.1467-9280.2006.01780.x.

[20] Fritz Heider and Marianne Simmel. An experimental study of apparent behavior. *The American journal of psychology*, 57(2):243–259, 1944.

[21] Colin Jacobs, Michael Lopez-Brau, and Julian Jara-Ettinger. What happened here? children integrate physical reasoning to infer actions from indirect evidence. In *Proceedings of the 43rd Annual Meeting of the Cognitive Science Society*, volume 43, 2021.

[22] Julian Jara-Ettinger. Theory of mind as inverse reinforcement learning. *Current Opinion in Behavioral Sciences*, 29:105–110, 2019.

[23] Julian Jara-Ettinger, Hyowon Gweon, Laura E Schulz, and Joshua B Tenenbaum. The naïve utility calculus: Computational principles underlying commonsense psychology. *Trends in cognitive sciences*, 20(8):589–604, 2016. URL https://www.sciencedirect.com/science/article/pii/S1364661316300535.

[24] James T Kajiya. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 143–150, 1986. URL https://dl.acm.org/doi/pdf/10.1145/15922.15902.

[25] Max Kleiman-Weiner, Mark K Ho, Joseph L Austerweil, Michael L Littman, and Joshua B Tenenbaum. Coordinate

to cooperate or compete: abstract goals and joint intentions in social interaction. In *CogSci*, 2016.

[26] Eric P Lafortune and Yves Willems. Bi-directional path tracing. In *Compugraphics' 93*, pages 145–153, 1993. URL https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=c480a06ab16aa18673cc872f811e5177a74fb790.

[27] Michael Lopez-Brau and Julian Jara-Ettinger. Physical pragmatics: Inferring the social meaning of objects. 2020. URL https://psyarxiv.com/mnf4y/.

[28] Michael Lopez-Brau, Joseph Kwon, and Julian Jara-Ettinger. Mental state inference from indirect evidence through bayesian event reconstruction. In *CogSci*, 2020. URL https://cognitivesciencesociety.org/cogsci20/papers/0085/0085.pdf.

[29] Michael Lopez-Brau, Joseph Kwon, and Julian Jara-Ettinger. Social inferences from physical evidence via bayesian event reconstruction. *Journal of Experimental Psychology: General*, 2022. URL https://psyarxiv.com/4zu9n.

[30] David Marr. *Vision: A computational investigation into the human representation and processing of visual information*. MIT press, 1982.

[31] Scott McCloud. Understanding comics: The invisible art. *Northampton, Mass*, 7:4, 1993.

[32] Aviv Netanyahu, Tianmin Shu, Boris Katz, Andrei Barbu, and Joshua B Tenenbaum. Phase: Physically-grounded abstract social events for machine social perception. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 845–853, 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/16167/15974.

[33] Andrew Y Ng, Stuart Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2, 2000. URL https://www.eecs.harvard.edu/cs286r/courses/spring06/papers/ngruss_irl00.pdf.

[34] Desmond C Ong, Jamil Zaki, and Noah D Goodman. Affective cognition: Exploring lay theories of emotion. *Cognition*, 143:141–162, 2015.

[35] Stefan Palan and Christian Schitter. Prolific. ac—a subject pool for online experiments. *Journal of Behavioral and Experimental Finance*, 17:22–27, 2018.

[36] Madeline Pelz, Laura Schulz, and Julian Jara-Ettinger. The signature of all things: Children infer knowledge states from static images. In *Proceedings of the 42nd Annual Meeting of the Cognitive Science Society*, 2020. URL https://psyarxiv.com/f692k/.

[37] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016. URL https://pbr-book.org/3ed-2018/Monte_Carlo_Integration.

[38] Antonin Raffin, Ashley Hill, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, and Noah Dormann. Stable baselines3, 2019.

[39] Miquel Ramırez and Hector Geffner. Plan recognition as planning. In *Proceedings of the 21st international joint conference on Artifical intelligence. Morgan Kaufmann Publishers Inc*, pages 1778–1783. Citeseer, 2009. URL https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=0de92a30c8a92a693705b53eaf602bbf258b4f39.

[40] Miquel Ramırez and Hector Geffner. Probabilistic plan recognition using off-the-shelf classical planners. In *Proceedings of the Conference of the Association for the Advancement of Artificial Intelligence (AAAI 2010)*, pages 1121–1126. Citeseer, 2010. URL https://cdn.aaai.org/ojs/7745/7745-13-11274-1-2-20201228.pdf.

[41] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[42] Rohin Shah, Dmitrii Krasheninnikov, Jordan Alexander, Pieter Abbeel, and Anca Dragan. Preferences implicit in the state of the world. *International Conference on Learning Representations (ICLR)*, 2019. URL https://arxiv.org/pdf/1902.04198.

[43] Kevin Smith and Edward Vul. Looking forwards and backwards: Similarities and differences in prediction and retrodiction. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 36, 2014.

[44] Shirin Sohrabi, Anton V Riabov, and Octavian Udrea. Plan recognition as planning revisited. In *IJCAI*, pages 3258–3264. New York, NY, 2016. URL http://www.cs.toronto.edu/~shirin/Sohrabi-IJCAI-16.pdf.

[45] Victoria Southgate and Gergely Csibra. Inferring the outcome of an ongoing novel action at 13 months. *Developmental psychology*, 45(6):1794, 2009.

[46] Joshua Tenenbaum. Bayesian modeling of human concept learning. *Advances in neural information processing systems*, 11, 1998. URL https://proceedings.neurips.cc/paper/1998/file/d010396ca8abf6ead8cacc2c2f2f26c7-Paper.pdf.

[47] Joshua Tenenbaum. Rules and similarity in concept learning. *Advances in neural information processing systems*, 12, 1999. URL https://proceedings.neurips.cc/paper/1999/file/86d7c8a08b4aaa1bc7c599473f5dddda-Paper.pdf.

[48] Christopher D Twigg and Doug L James. Backward steps in rigid body simulation. In *ACM SIGGRAPH 2008 papers*, pages 1–10. 2008. URL https://dl.acm.org/doi/pdf/10.1145/1399504.1360624.

[49] Tomer Ullman, Chris Baker, Owen Macindoe, Owain Evans, Noah Goodman, and Joshua Tenenbaum. Help or hinder: Bayesian models of social goal inference. *Advances in neural information processing systems*, 22, 2009.

[50] Eric Veach. *Robust Monte Carlo methods for light transport simulation*. PhD thesis, Stanford University, 1998. URL https://www.proquest.com/docview/304456010?pq-origsite=gscholar&fromopenview=true.

[51] Eric Veach and Leonidas Guibas. Bidirectional estimators for light transport. In *Photorealistic Rendering Techniques*, pages 145–167. Springer, 1995. URL https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=3c9921cf30576edffd4d611e8f5d05e94421b57d.

[52] Edward Vul, Noah Goodman, Thomas L Griffiths, and Joshua B Tenenbaum. One and done? optimal decisions from very few samples. *Cognitive science*, 38(4):599–637, 2014. URL https://onlinelibrary.wiley.com/doi/pdfdirect/10.1111/cogs.12101.

[53] Sarah A Wu, Rose E Wang, James A Evans, Joshua B Tenenbaum, David C Parkes, and Max Kleiman-Weiner. Too many cooks: Bayesian inference for coordinating multi-agent collaboration. *Topics in Cognitive Science*, 13(2):414–432, 2021.

[54] Tan Zhi-Xuan, Jordyn Mann, Tom Silver, Josh Tenenbaum, and Vikash Mansinghka. Online bayesian goal inference for boundedly rational planning agents. *Advances in neural information processing systems*, 33:19238–19250, 2020. URL https://proceedings.neurips.cc/paper/2020/file/df3aebc649f9e3b674eeb790a4da224e-Paper.pdf.

[55] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

APPENDIX

A. *Experimental design*

For each of our experiments reported in Section III-B, we recruited $N = 200$ participants from Prolific [35]. Participants were paid \$15 per hour (\$1.25 total for blocks and grid domains, and \$2.00 for the keys domain), and our experiments were conducted with IRB approval.

Participants were first familiarized with the environment, through both text instructions and a sample video of an agent performing the task in the domain. Then, they were told that their objective was to infer the agent's goal from a single snapshot. They answered several questions to check their comprehension of both the domain and the task they were asked to perform, and were not allowed to continue unless they answered the comprehension questions correctly. The full experimental design is available in HTML format upon request. No data was excluded from our analyses.

B. *Numerical test of correctness*

Programming sophisticated importance sampling routines is a challenging and bug-prone engineering effort [13, 2, 37]. To test that our algorithm is unbiased, i.e. that it produces correct likelihoods in expectation, we compared likelihoods computed by rejection sampling and our sampler using converged estimates (25,000 samples each). For this experiment we used a uniform 4×4 grid-world, with the prior on start states being uniform along the first row ($x = 0$) and the goal being the far corner $(3, 3)$. The results of this experiment are shown in Figure 3. Our estimator has a dramatically different implementation than rejection sampling (compare Algorithms 1 and 2). However, the computed likelihoods are indistinguishable at every cell in the grid, even in "corner-case" cells such as the goal cell itself. **This provides a strong check that our algorithm and its implementation are both indeed correct.**

Algorithm 1 Rejection sampling, as in prior work. Compare to our proposed method, Algorithm 2.

Require: x, the agent's current state (e.g. position in grid)
g, the hypothesized goal
$P(s \to s' \mid g)$, the probability the agent will move to s' from s
$P_{\text{start}}(s)$, the prior over the agent starting at s

$t \leftarrow 0, n \leftarrow 0$, sample x_{current} with probability $\propto P_{\text{start}}(\cdot)$
while x_{current} is not an end state **do**
  if $x_{\text{current}} = x$ **then**
    $n \leftarrow n + 1$
  sample x_{next} w.p. $p_{\text{choice}} \propto P(x_{\text{current}} \to \cdot \mid g)$
  $x_{\text{current}} \leftarrow x_{\text{next}}$ and $t \leftarrow t + 1$
return $n \;/\; \ell$

C. *Additional domains*

We used our algorithm to perform inferences in three additional domains. The purpose of these domains is to show the remarkable flexibility of our method: how it can make interesting inferences in a wide variety of settings. Though we did not collect human subject data for these domains, we show results for cases where the inference task is relatively straightforward.

1) Food trucks (joint belief/desire inference): The food trucks domain, taken from the cognitive science literature [8], is a Partially Observable Markov Decision Process (POMDP). It consists of a 5×10 gridworld with an opaque wall in the middle. A hungry graduate student wakes up at home (one side of the wall) and wishes to eat at a food truck. There are two parking spots where food trucks usually park, and three kinds of food trucks that could be parked at each of those spots: Korean, Lebanese, and Mexican (K, L, and M). The graduate student might have preferences among the cuisines, but might also be uncertain about which trucks are parked at each spot today. Thus, they might engage in information-seeking behavior by looking behind the opaque wall, and then choosing a food truck to walk to based on their preferences. **The inference task is to determine (a) the student's preferences over food trucks, and (b) the student's (current) belief state about which truck is at each parking spot.**

Using this domain, Baker et al.'s inverse planning model was able to jointly infer the student's beliefs and desires from an observed trajectory; those inferences closely matched responses from human subjects. Here, we perform the same type of inference, but from a single observed snapshot.

For example, in the example in Figure 4, the student is observed moving south next to the wall. A Korean food truck is parked in the southwest parking spot, and a Lebanese food truck is parked in the northeast spot. Seeing this scene, a reasonable inference is that the student went looking around the wall to see if the Mexican food truck (their favorite) was parked on the other side. Seeing that it was Lebanese food instead, the student turns around and makes peace with the nearby Korean food. Indeed, our model captures this inference: in the joint posterior distribution over both beliefs and desires, our model is confident that the student now knows that the northeast truck has Lebanese food, and furthermore that the student's favorite food is Mexican.

A more sophisticated inference emerges if the student is observed moving *north* instead of south (Figure 5). Now, a reasonable inference is that the student dislikes Korean food, and is going around the wall to check what is at the other truck. The model captures this: it favors the hypothesis that the student is unsure what is at the northeast truck, and also places high weight on Korean being the least favorite food option.

However, as is visible on the right half of the heatmap, the model also places some weight on the possibility that the student knows that there is Lebanese food and prefers it, or that the student (mistakenly) believes there is Mexican food and prefers that.

2) Heist (multi-agent domain): In this multi-agent domain inspired by classic stimuli in cognitive science [6, 45, 20], two agents—blue and pink—occupy a 7×7 gridworld representing an art museum. One of the agents is a "thief," whose objective is to escape the museum by reaching the exit, and the other

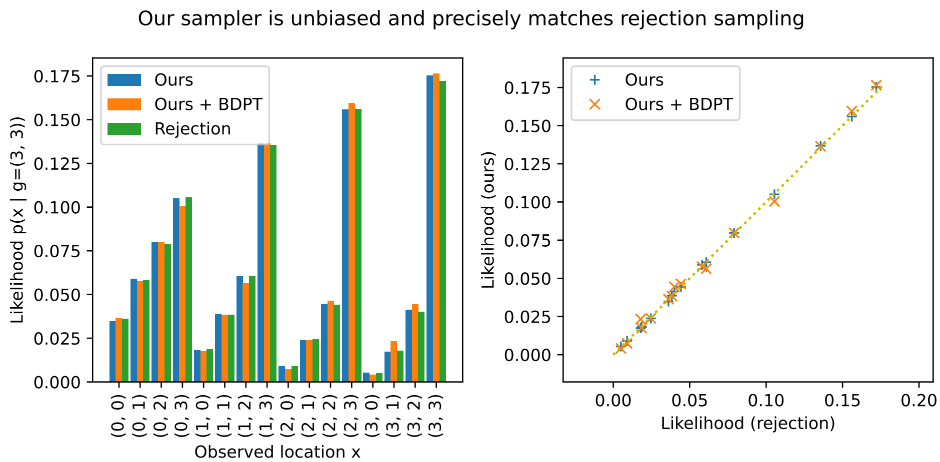


Fig. 3. Our sampler's likelihoods precisely match rejection sampling, with and without bidirectional path tracing, giving a strong numerical check of our method's correctness (Appendix B).

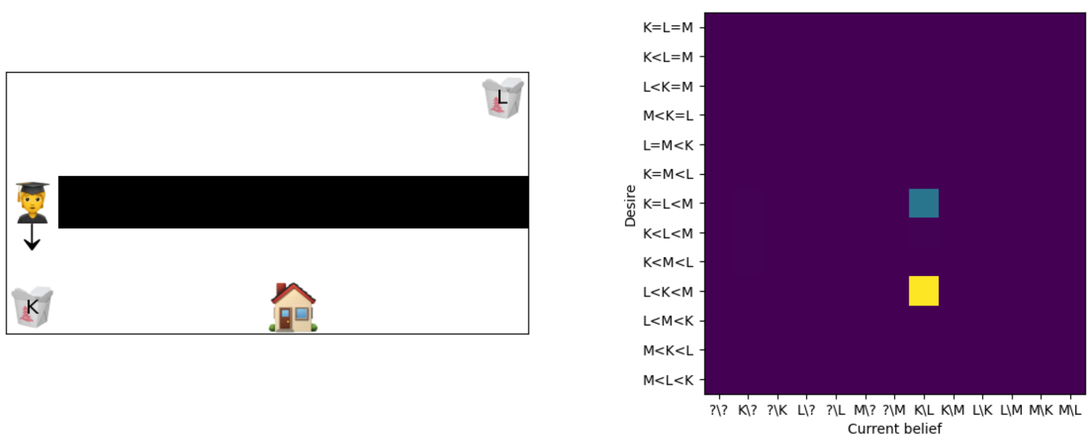


Fig. 4. The student is observed heading south around the wall. A rational inference is that the student started at home, and went around the wall to check what the far food truck was. Seeing that it was Lebanese and not Mexican (their favorite), the student disappointedly turns around to make peace with the nearby Korean food. **As shown on the heatmap to the right, our model captures this joint belief-desire inference, predicting that the student now knows what is at both trucks, and reconstructing the student's likely preference ordering over the three cuisines.** *Note: the belief label "K \ ?" means that the student thinks the south-west parking spot has a Korean food truck parked, but is unsure about the north-east parking spot.* See Appendix C1.

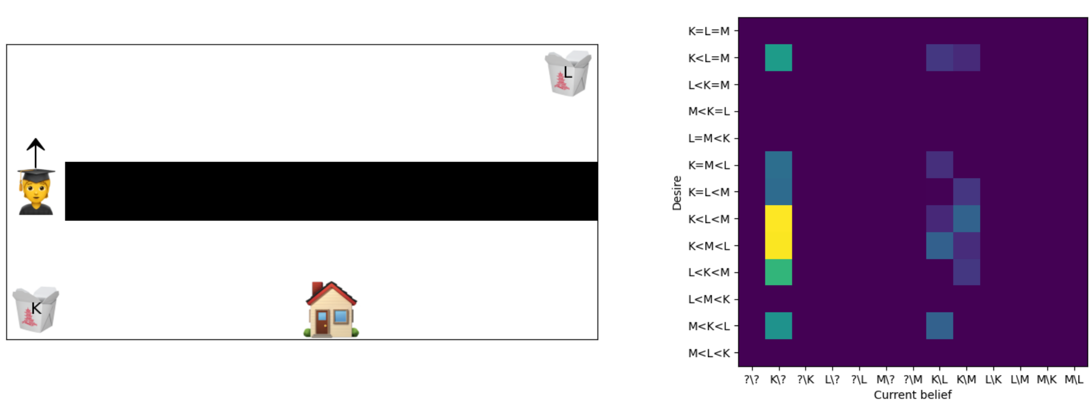


Fig. 5. Here, the student is observed going north instead of south. A more sophisticated inference emerges, showing that the student is likely uncertain about which truck is parked behind the wall. See Appendix C1

Algorithm 2 Our bidirectional likelihood sampler

Require: x, g, $P(s \to s' \mid g)$, $P_{\text{start}}(s)$ as in Algorithm 1
α, the strength of importance sampling
d, an average termination depth for Russian roulette
C, an optional bidirectional path tracing cache (see Algorithm 3)

$\ell \leftarrow 0$
$t_{\text{next}} \leftarrow 0$, $x_{\text{current}} \leftarrow x$ ▷ *Sample forward*
while x_{current} is not an end state **do**
  sample successor state x_{next} with probability $p_{\text{choice}} \propto P(x_{\text{current}} \to \cdot \mid g)$
  $x_{\text{current}} \leftarrow x_{\text{next}}$ and $t_{\text{next}} \leftarrow t_{\text{next}} + 1$
$t_{\text{prev}} \leftarrow 1$, $x_{\text{current}} \leftarrow x$, $p_\pi \leftarrow 1$ ▷ *Sample backwards*
while true **do**
  if $x_{\text{current}} \in C$ **then** ▷ *Check BDPT cache*
    sample (t_{cache}, w) from $C[x_{\text{current}}]$
    return $w \cdot (\#C[x_{\text{current}}]/\#C) \cdot p_\pi/(t_{\text{cache}} + t_{\text{prev}} + t_{\text{next}})$
  if flip() $< 1/d$ **then** ▷ *Russian roulette termination*
    return $P_{\text{start}}(x_{\text{current}}) \cdot p_\pi/(t_{\text{prev}} + t_{\text{next}}) \cdot 1/(1/d)$
  $p_\pi \leftarrow p_\pi \;/\; (1 - 1/d)$ ▷ *Russian roulette weight*
  sample predecessor state x_{prev} with probability $p_{\text{choice}} \propto \exp(\alpha \cdot P(\cdot \to x_{\text{current}} \mid g))$
  $p_\pi \leftarrow p_\pi \cdot P(x_{\text{prev}} \to x_{\text{current}} \mid g) \;/\; p_{\text{choice}}$
  $x_{\text{current}} \leftarrow x_{\text{prev}}$ and $t_{\text{prev}} \leftarrow t_{\text{prev}} + 1$
return ℓ

Algorithm 3 Grow the bidirectional path tracer's cache (to be called repeatedly)

Require: g, $P(s \to s' \mid g)$, $P_{\text{start}}(s)$ as in Algorithm 1, d as in Algorithm 2, and C, a cache

$t \leftarrow 0$, $w \leftarrow 1$, sample x_{current} with prob. $\propto P_{\text{start}}(\cdot)$
while x_{current} is not an end state **do**
  add $(t, d \cdot w)$ to $C[x_{\text{current}}]$
  sample x_{next} with prob. $\propto P(x_{\text{current}} \to \cdot \mid g)$
  if flip() $< 1/d$ **then**
    break
  $w \leftarrow w \;/\; (1 - 1/d)$
  $x_{\text{current}} \leftarrow x_{\text{next}}$ and $t \leftarrow t + 1$

is a "guard," whose objective is to catch the thief. There are four doors in the room, only one of which is an exit, and the rest of which are dead ends. Both agents know which door is the exit, but this information is *not* visible to the observer (all doors are rendered identically). **The inference tasks are to look at a snapshot of the two agents and jointly infer (a) which agent is the thief and which is the guard, and (b) which door is the exit.**

In the example in Figure 6, it is clear from the snapshot that the blue agent is the guard and is chasing the pink agent, the thief, to the bottom-right corner. The model reproduces this inference, though also acknowledges the possibility that the thief might actually be heading onward past the bottom-right, to the bottom-left corner instead.

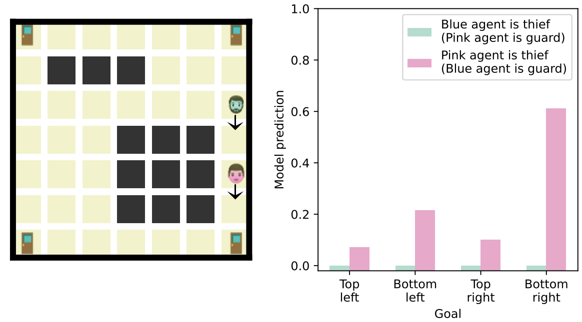


Fig. 6. Two agents are observed by a security camera in an art museum. Who is the guard, who is the thief, and where is the thief trying to escape to? **Our model predicts that the guard is the blue agent, the thief is the pink agent, and that the exit is in the bottom right.** See Appendix C2.

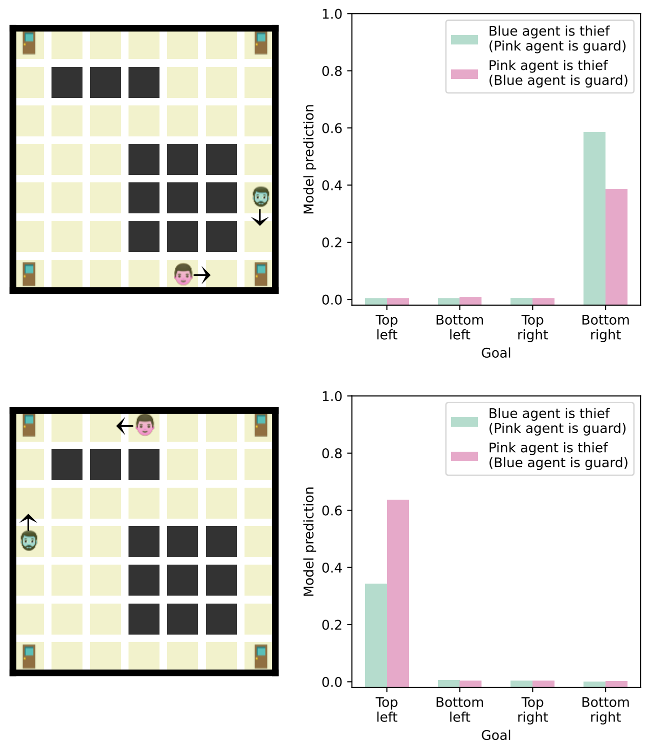


Fig. 7. In these examples, it is unclear who the guard and thief are—however, it is clear where the exit is. **The model reproduces this uncertainty as desired.** See Appendix C2.

The next two examples (Figure 7) are ambiguous cases: the two agents are in symmetric positions, so it is unclear who is who. Here, the model can determine with high confidence where the exit is, but remains uncertain about who is the thief and who is the guard.

Finally, in the last example (Figure 8), it is unclear whether a blue guard is blocking a pink thief from heading to the top-right corner, or whether a pink guard is blocking the blue thief from heading the the bottom-right corner. Indeed, the model reproduces this ambiguity.

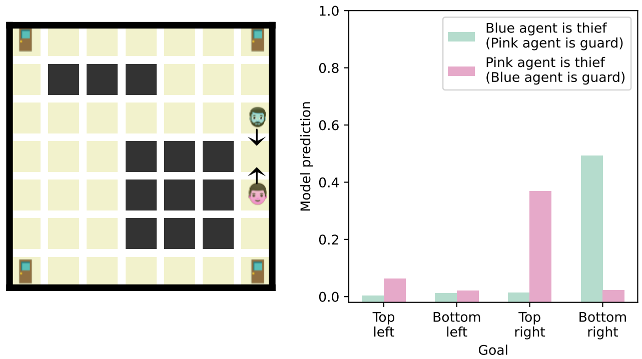


Fig. 8. In this example, it is unclear whether a blue guard is blocking a pink thief from heading to the top-right corner, or whether a pink guard is blocking the blue thief from heading the the bottom-right corner. **The model reproduces this joint uncertainty as desired.** See Appendix C2.

3) Cart-pole (continuous state space with physical dynamics): The cart-pole domain is a classic problem in reinforcement learning and optimal control. The goal is to balance a pole in an upright position, by moving the cart left or right. The state space of this domain consists of four continuous numbers: the horizontal position of the cart and its velocity, and the angle of the pole along with its angular velocity. **The inference tasks are to look at a snapshot image—which only shows the cart position and the pole angle—and determine the velocity of the cart and the angular velocity of the pole.** Note that rejection sampling cannot solve this task because the probability of a randomly-sampled trace passing through the observed state is zero.

We use an off-the-shelf pre-trained Proximal Policy Optimization (PPO) controller [41] from stable-baselines3 [38] to compute a probability distribution over actions. Inference in this domain is complicated by the fact that computing backward dynamics in physical simulation is challenging and often ill-posed. While previous work has proposed analytic approaches [48], we instead train a neural network to approximate the reverse physical dynamics. We place a unit Gaussian prior over the velocities, and use a Von-Mises distribution as a prior over the initial pole angle. We infer the velocities of the system by sampling candidate pairs of cart and pole velocities (stratified in an 11×11 grid) and computing likelihoods using our algorithm.

The inferred posteriors are intuitive and track the relative stability of the position in each snapshot (Figure 9). For example, in part (a), the pole has almost completely fallen over, and so our method infers that the pole has a large negative angular velocity, and is falling fast towards the ground. At the same time, it infers that the cart is moving fast to the left, in an attempt to re-balance. In comparison, for part (f), the pole is nearly upright, so the model predicts that the pole is not rotating, and that the cart might be moving left or right to keep the pole balanced.

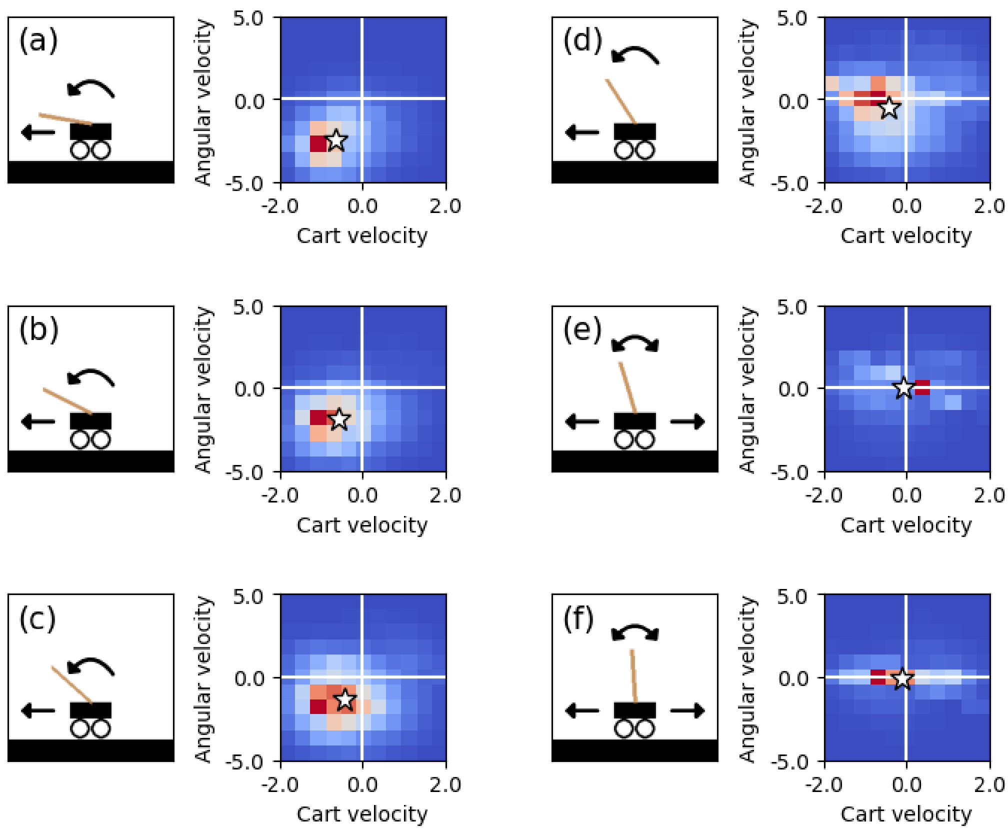


Fig. 9. In each pair, the left image shows the cart-pole snapshot given to the algorithm, and the overlaid arrows summarize the model's predictions about how the system might evolve. The right heatmap shows our model's full joint distribution of inferred cart velocity (positive means moving to the right) and pole angular velocity (positive means clockwise), and the white stars mark posterior expectations. **When the pole is near-horizontal, our algorithm infers that the pole is falling, and the cart is moving left to re-balance. When the pole is near-vertical, the algorithm infers that the pole is stationary, and the cart is making minor adjustments to keep the pole balanced.**