A. Experimental design

For each of our experiments reported in Section 3.3, we recruited $N = 200$ participants from Prolific (Palan and Schitter, 2018). 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 (Cusumano-Towner et al., 2019; Anderson et al., 2017; Pharr et al., 2016). 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 4. 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.**

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.

C.1. Food trucks (joint belief/desire inference)

The food trucks domain, taken from the cognitive science literature (Baker et al., 2017), is a Partially Observable Markov Decision Process (POMDP). It consists of a 5×10 grid-world 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 5, 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 6). 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.

C.2. Heist (multi-agent domain)

In this multi-agent domain inspired by classic stimuli in cognitive science (Baker et al., 2008; Southgate and Csibra, 2009; Heider and Simmel, 1944), two agents—blue and pink—occupy a 7×7 gridworld representing an art museum.

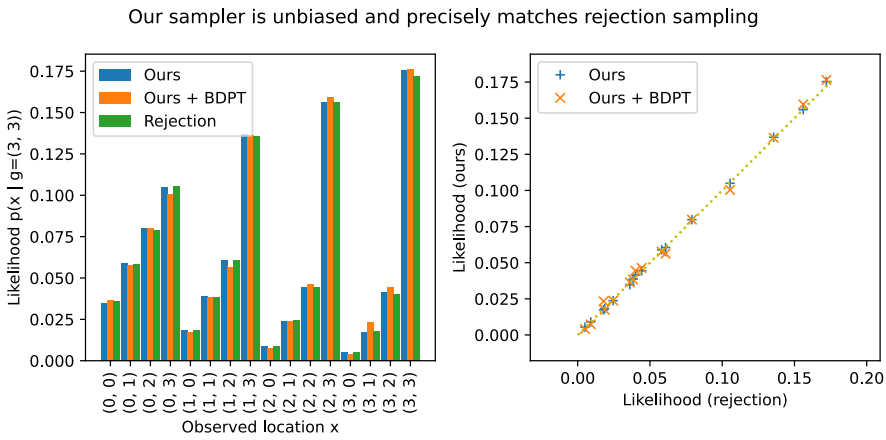

Figure 4. Our sampler's likelihoods precisely match rejection sampling, with and without bidirectional path tracing, giving a strong numerical check of our method's correctness (Appendix B).

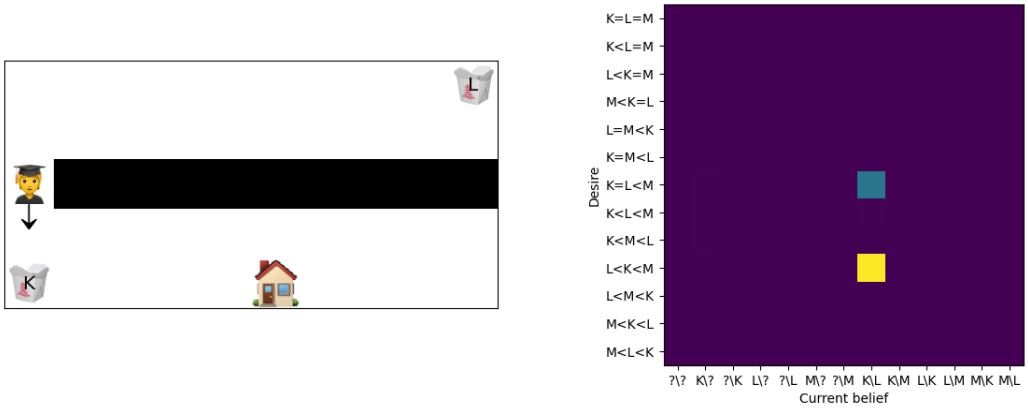

Figure 5. The student is observed heading south around the wall. A rational inference is that the student started at home, and went around the wall to check what the far food truck was. Seeing that it was Lebanese and not Mexican (their favorite), the student disappointedly turns around to make peace 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 "$K \setminus ?$" 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 C.1.

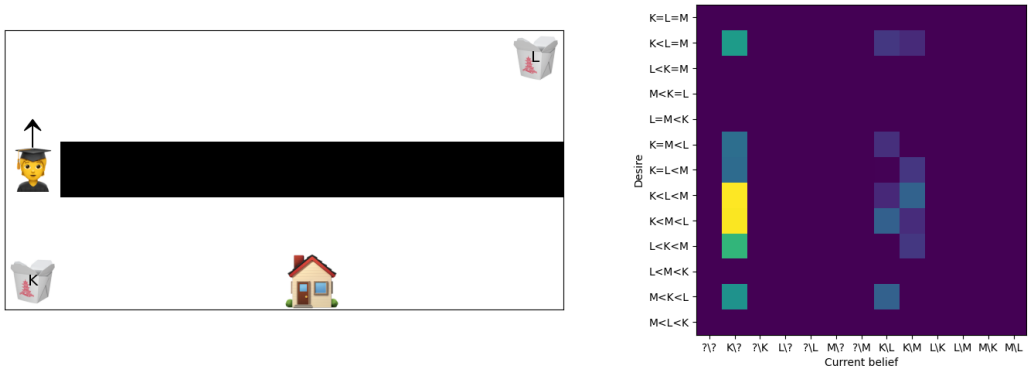

Figure 6. Here, the student is observed going north instead of south. A more sophisticated inference emerges, showing that the student is likely uncertain about which truck is parked behind the wall. See Appendix C.1

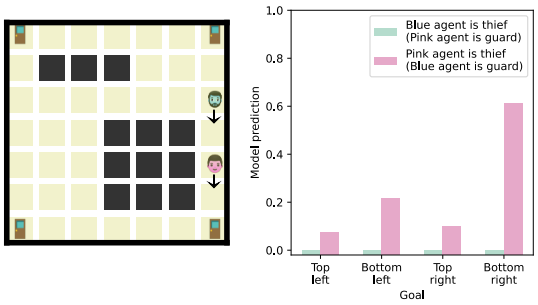

Figure 7. Two agents are observed by a security camera in an art museum. Who is the guard, who is the thief, and where is the thief trying to escape to? **Our model predicts that the guard is the blue agent, the thief is the pink agent, and that the exit is in the bottom right.** See Appendix C.2.

One of the agents is a "thief," whose objective is to escape the museum by reaching the exit, and the other 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 7, 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.

The next two examples (Figure 8) 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 9), 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.

C.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

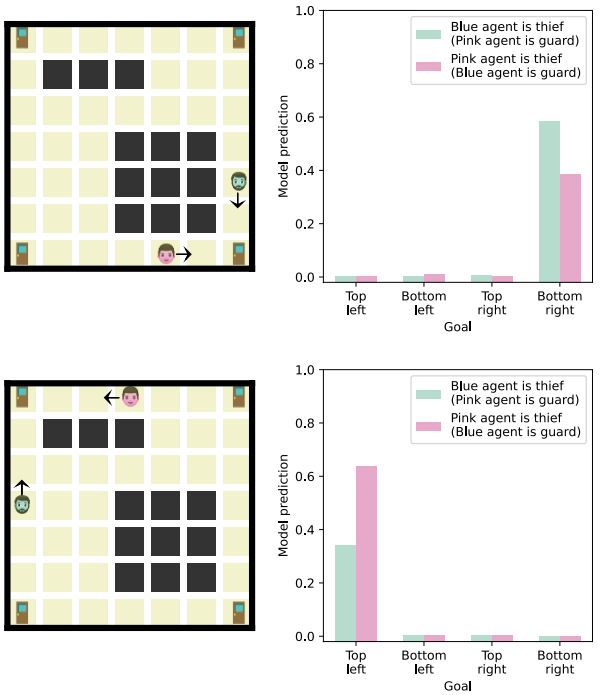

Figure 8. In these examples, it is unclear who the guard and thief are—however, it is clear where the exit is. **The model reproduces this uncertainty as desired.** See Appendix C.2.

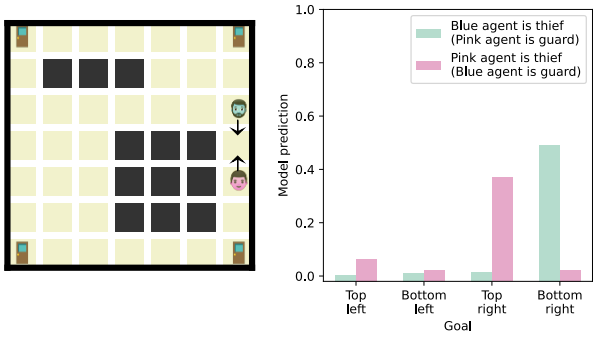

Figure 9. In this example, it is unclear whether a blue guard is blocking a pink thief from heading to the top-right corner, or whether a pink guard is blocking the blue thief from heading the the bottom-right corner. **The model reproduces this joint uncertainty as desired.** See Appendix C.2.

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 (Schulman et al., 2017) from stable-baselines3 (Raffin et al., 2019) 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 (Twigg and James, 2008), 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 10). 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.

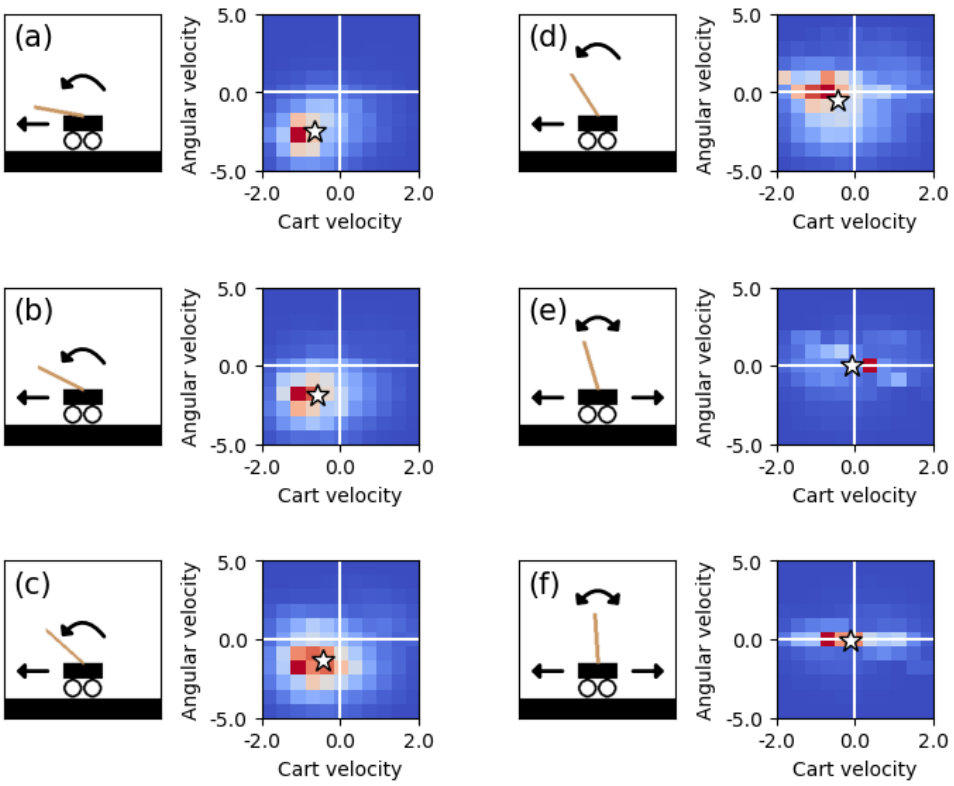

Figure 10. In each pair, the left image shows the cart-pole snapshot given to the algorithm, and the overlaid arrows summarize the model's predictions about how the system might evolve. The right heatmap shows our model's full joint distribution of inferred cart velocity (positive means moving to the 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.**