Infinite Ends from Finite Samples: Open-Ended Goal Inference as Top-Down Bayesian Filtering of Bottom-Up Proposals

Tan Zhi-Xuan, Gloria Kang, Vikash Mansinghka, Joshua B. Tenenbaum
Department of Brain and Cognitive Sciences, MIT
Correspondence to xuan@mit.edu

Abstract

The space of human goals is tremendously vast; and yet, from just a few moments of watching a scene or reading a story, we seem to spontaneously infer a range of plausible motivations for the people and characters involved. What explains this remarkable capacity for intuiting other agents' goals, despite the infinitude of ends they might pursue? And how does this cohere with our understanding of other people as approximately rational agents? In this paper, we introduce a sequential Monte Carlo model of *open-ended goal inference*, which combines top-down Bayesian inverse planning with bottom-up sampling based on the statistics of co-occurring subgoals. By proposing goal hypotheses related to the subgoals achieved by an agent, our model rapidly generates plausible goals without exhaustive search, then filters out goals that would be irrational given the actions taken so far. We validate this model in a goal inference task called Block Words, where participants try to guess the word that someone is stacking out of lettered blocks. In comparison to both heuristic bottom-up guessing and exact Bayesian inference over hundreds of goals, our model better predicts the mean, variance, efficiency, and resource rationality of human goal inferences, achieving similar accuracy to the exact model at a fraction of the cognitive cost, while also explaining garden-path effects that arise from misleading bottom-up cues. Our experiments thus highlight the importance of uniting top-down and bottom-up models for explaining the speed, accuracy, and generality of human theory-of-mind.

Keywords: theory-of-mind, goal inference, open-endedness, bottom-up heuristics, sampling, resource rationality

Introduction

Whether one is watching a play, reading a novel, or spending time with a friend at their house, inferences about others' goals and motivations often arise spontaneously and unbidden (Moskowitz & Olcaysoy Okten, 2016): Is the person crouching behind a tree trying to hide from, or spy on someone? Does the strange warrior who has just entered the fray of battle intend to kill the protagonist, or save them? When your friend gets up from the couch and walks to the kitchen, are they getting a snack, or making some tea? Despite the seemingly infinite space of possible goals, we have little trouble in coming up with plausible hypotheses, and then — as the story unfolds — filtering out those that fail to explain our observations. If the warrior defends our protagonist from a stray arrow, they are likely an ally. If your friend opens the fridge, they are probably having a snack. What computational mechanisms underlie this ability to both hypothesize and evaluate the goals that explain others' behavior, even when the set of possibilities is vast and open-ended?

While psychologists have long studied how people both *generate* (Heider & Simmel, 1944; Graesser et al., 1994; Hassin et al., 2005; Van Overwalle et al., 2012) and *evaluate* hypotheses about the goals that other agents have (Gergely & Csibra, 2003; Jara-Ettinger et al., 2015; Liu et al., 2017), computational models of human goal inference have focused on the latter, assuming a small and *fixed* set of possible goals, then modeling how people infer their relative likelihoods (Baker et al., 2009; Ullman et al., 2009; Kleiman-Weiner et al., 2016; Vered et al., 2016; Jara-Ettinger et al., 2019). This leaves open how people come up with plausible goals in the first place, especially in large hypothesis spaces where enumeration over all possibilities makes inference intractable (Kwisthout & Van Rooij, 2013; Blokpoel et al., 2013). How then are people solving this seemingly intractable problem (if they do so at all)? Even though recent advances in Bayesian inverse planning have shown how modeling the plans of other agents (Zhi-Xuan et al., 2020; Alanqary et al., 2021) and inferring goals from static scenes (Chandra et al., 2023) can be made orders of magnitude more efficient, they do not address the key challenge posed by open-ended settings: Efficiently generating plausible goal hypotheses.

In this paper, we develop an algorithmic account of open-ended goal inference, which combines top-down Bayesian inverse planning and bottom-up sampling in a sequential Monte Carlo (SMC) algorithm (Del Moral et al., 2006). Instead of exhaustively enumerating the space of goals, our model assumes that humans are familiar with the statistics of their environments (Griffiths & Tenenbaum, 2006), and can rapidly generate relevant hypotheses based on contextual, data-driven cues (Schulz, 2012; Phillips et al., 2019). In particular, we assume familiarity with the statistics of co-occurring subgoals, such that complete goals can rapidly be generated once some subgoals have been achieved. Our model then filters these goals according to the principle of rational action (Gergely & Csibra, 2003; Baker et al., 2009), keeping those that best explain the agent's actions. We evaluate this model in Block Words, a game where observers have to guess the word that someone is stacking out of lettered blocks (Ramírez & Geffner, 2010; Alanqary et al., 2021). Subgoals correspond to partial words, so observers can generate plausible goals by "auto-completion". However, this bottom-up strategy is insufficient in general — some goals may be irrational given the actions observed so far, necessitating inverse planning.

To test the predictions of our model, we conduct an experiment where human participants play a series of rounds in Block Words. Each round is carefully designed so as to elicit various patterns of inference — some where bottom-up guessing is sufficient, some where inverse planning is required to filter out irrational goals, and some intended to produce garden-path inferences that exact Bayesian reasoning should avoid. As alternatives to our model, we test pure bottom-up sampling, as well as an exact Bayesian baseline that performs enumerative inference over all English words that can be spelled in each round. We compare these models by measuring their similarity to human responses in terms of their mean, variance, sample efficiency, and computational cost, allowing us to determine their fidelity to human goal inference in both behavioral and algorithmic terms.

Computational Model

Building upon prior accounts of human goal inference (Baker et al., 2009; Zhi-Xuan et al., 2020; Alanqary et al., 2021), we assume that observers perform approximately Bayesian inference over a generative model of how other agents plan and act to achieve their goals:

$$\textit{Goal Prior:} \quad g \sim P(g) \tag{1}$$

$$\textit{Online Planning:} \quad \pi_t \sim P(\pi_t | s_{t-1}, \pi_{t-1}, g) \tag{2}$$

$$\textit{Action Selection:} \quad a_t \sim P(a_t | s_{t-1}, \pi_t) \tag{3}$$

$$\textit{State Transition:} \quad s_t \sim P(s_t | s_{t-1}, a_t) \tag{4}$$

Here g is the agent's goal, and at each step t, π_t is the agent's current plan or policy, a_t is the agent's action, and s_t is the state of the environment. Given a sequence of states $s_{0:T}$ and actions $a_{1:T}$, the observer's task is to infer the goal g by approximating the posterior $P(g|s_{0:T}, a_{1:T})$. Approximating this posterior presents numerous computational challenges. Among these, our focus is on the challenge posed by *open-ended* settings, where the set of possible goals $g \in \mathcal{G}$ is large or potentially infinite. In this section, we first review recent advances that render goal inference over *fixed* spaces algorithmically tractable, before explaining how we can extend these ideas to open-ended spaces.

Modeling Boundedly Rational Plans and Actions

Since computing the posterior requires simulating the plans π_t that an agent might follow to each goal g, this process is also known as *Bayesian inverse planning*. In general this is a difficult problem, because planning itself is a complicated and often intractable task. However, as Zhi-Xuan et al. (2020) show, this difficulty can be alleviated by treating agents as *boundedly rational planners*, who spend only limited computation at each step t on planning. We adopt a more recent version of this architecture (Zhi-Xuan et al., 2024; Ying et al., 2023), modeling agents that update a *policy* π_t (i.e. a conditional plan) that defines a Boltzmann distribution over actions a_t that can be taken at state s_{t-1}:

$$P(a_t | s_{t-1}, \pi_t) = \frac{\exp\left(-\beta \hat{Q}_{\pi_t}(s_{t-1}, a_t)\right)}{\sum_a \exp\left(-\beta \hat{Q}_{\pi_t}(s_{t-1}, a)\right)} \tag{5}$$

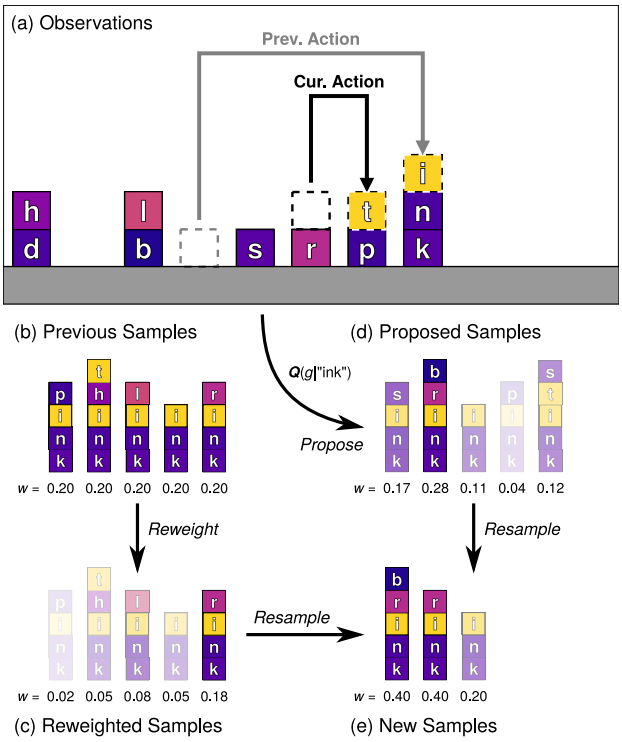


Figure 1: Illustration of open-ended goal inference in Block Words via particle filtering. Initially, **i** is stacked on **n**, leading **pink** to be proposed as a goal. In the next step, however, **t** is stacked on **p**. This makes **pink** much less likely after reweighting, and hence removed after resampling.

Here $\hat{Q}_{\pi_t}(s_{t-1}, a_t)$ denotes the estimated cost of the shortest path to goal g from s_{t-1} that starts with action a_t. As such, π_t assigns higher probability to actions along more optimal paths to the goal, with β controlling the degree of optimality. To compute $\hat{Q}_{\pi_t}(s_{t-1}, a_t)$ efficiently, we use real-time heuristic search (RTHS), which updates the $\hat{Q}_{\pi_{t-1}}$ values computed for π_{t-1} via a search process guided by the Q-values themselves (Korf, 1990; Barto et al., 1995; Koenig & Likhachev, 2006). This process is deterministic, and limited to a computational budget of up to size B (e.g. the number of search iterations). As such, given $G = |\mathcal{G}|$ possible goals, the computational cost of simulating the plans of an agent for T steps (and hence exactly inferring their goals) is $O(GBT)$.

Bottom-Up Sampling of Plausible Goals

While the analysis above suggests that Bayesian inverse planning is not only tractable, but *linear* in computational complexity, it neglects the fact that the number of goals G can grow very large. As suggested by Blokpoel et al. (2013), this might be because G itself grows exponentially with some other natural parameter — in Block Words, for example, just 9-11 lettered blocks can be used to spell anywhere from 150 to 800 English words. But even without this exponential dependence, a large value of G can quickly render (exact) goal inference too costly to be algorithmically plausible.

How might people manage the complexity of their inferences in these open-ended settings? We posit that in sufficiently familiar contexts, people are familiar with the statistics of co-occurring subgoals, such that given some subgoal γ_i, they can rapidly sample a complementary subgoal $\gamma_j \sim P(\gamma_j|\gamma_i)$. This means that once an agent achieves some subgoal γ_i — say, boiling a kettle of water, or stacking the letter [n] on top of [g] — an observer can rapidly generate a complete goal $g = \gamma_i \wedge \gamma_j$ — perhaps adding tea to the boiled water, or spelling the word [s] [o] [n] [g]. These conditional distributions can be efficiently learned using either neural networks or classical sequence models such as n-grams. Since the goals in our study are English words, we use a character-level n-gram model. Regardless of what sequence model is used, a key property is that retrieval and sampling can occur in essentially *constant* time (Guthrie & Hepple, 2010), providing a plausible mechanism for relevance-guided hypothesis generation (Phillips et al., 2019; Schulz, 2012).

Bayesian Filtering of Bottom-Up Samples

Given the ability to rapidly generate plausible goals, it is tempting to forgo inverse planning altogether. As Figure 1 illustrates, however, this strategy can go awry. Suppose you see someone stack the block [i] on top of [n] [k], and the word [p] [i] [n] [k] comes to mind. But then you see [t] stacked on top of [p]. Is [p] [i] [n] [k] still a plausible goal? From a bottom-up perspective, [p] is still a likely completion of [i] [n] [k]. But if we understand agents as *rational* planners, this no longer seems likely. If [p] [i] [n] [k] had been the agent's goal, stacking [t] on [p] would be quite suboptimal.

If humans actually engage in the reasoning above, then modeling their inferences requires uniting top-down Bayesian inverse planning with bottom-up cues. Following other sampling-based accounts of sequential human inferences (Daw & Courville, 2008; Vul et al., 2009; Thaker et al., 2017), we model this integration with a sequential Monte Carlo (SMC) algorithm (Algorithm 1), extending the Sequential Inverse Plan Search (SIPS) algorithm of Zhi-Xuan et al. (2020). SMC algorithms are also known as particle filters, which approximate Bayesian posteriors by maintaining a weighted set of hypotheses or particles, then updating the weights of those particles as observations arrive (Del Moral et al., 2006). At each step, they may also *resample* particles according to their weights, or *rejuvenate* the particles, making perturbations to the sample collection to increase hypothesis diversity (Chopin, 2002; Lew, Matheos, et al., 2023).

A variant of this rejuvenation phase is where our bottom-up samplers come in: As Algorithm 1 shows, after observing each state s_t and action a_t at step t, we use these samplers as *proposal distributions* over goals $Q(g|s_t, a_t)$, generating N new goal hypotheses g (L4) based on bottom-up cues. These new hypotheses assigned an importance weight $P(g, \pi_{1:t}, s_{0:t}, a_{1:t})/Q(g|s_t, a_t)$, where the numerator $P(g, \pi_{1:t}, s_{0:t}, a_{1:t})$ accounts for how well the plans $\pi_{1:t}$ that lead to g explain the actions $a_{1:t}$, and the denominator $Q(g|s_t, a_t)$ compensates for g having been sampled from the proposal (L5–7). Open-ended SIPS also *reweights* previous samples based on how well they explain the current action a_t (L7–8). Finally, we resample the particle collection back down to N samples (L9), coalescing identical samples by summing their weights (L10). Our algorithm thus implements the high-level logic described earlier: Incrementally generate plausible hypotheses, evaluate them, then filter out those that do not make sense. This can be viewed as an epistemic analogue to recent accounts of open-ended decision making (Morris et al., 2021; Phillips et al., 2019).[1]

Algorithm 1 Open-Ended SIPS for Goal Inference

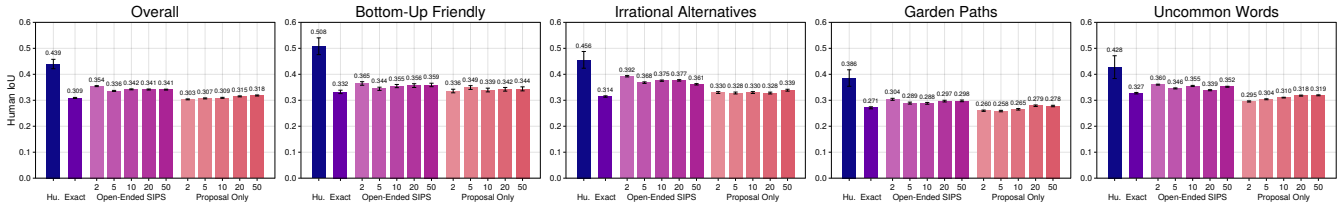
1: **Procedure** OPEN-ENDED-SIPS($s_{0:T}, a_{1:T}, N$)
2: **Using**: $Q(g|s_t, a_t)$, a bottom-up goal proposal.
3: **For** each step t from 1 to T **do**
4: Propose N new goals g from $Q(g|s_t, a_t)$.
5: Simulate policies $\pi_{1:t}$ for each new goal.
6: Compute weights $\frac{P(g, \pi_{1:t}, s_{0:t}, a_{1:t})}{Q(g|s_t, a_t)}$ for new particles.
7: Update policies π_t for previous goal samples g.
8: Multiply their weights by $P(a_t|s_{t-1}, \pi_t)$.
9: Resample full collection down to N particles.
10: Coalesce identical particles.
11: **End**
12: **Return** weighted collection of $\leq N$ goal hypotheses.
13: **End**

Experiments

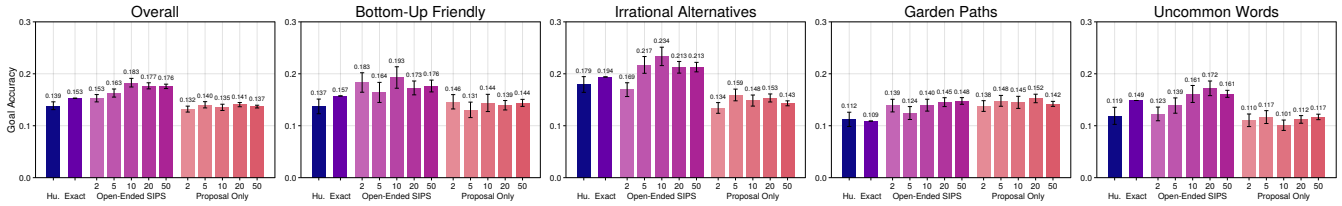
We evaluated open-ended SIPS as a model of human goal inference on a set of 16 scenarios in Block Words, a variant of the classic Blocksworld domain where the goal is to *infer* the word that an agent is spelling by stacking a tower of lettered blocks. In contrast to previous Block Words tasks (Ramírez & Geffner, 2009, 2010; Alanqary et al., 2021; Chandra et al., 2023), we did not specify a fixed set of 5 to 20 goal words. Instead, we told participants that the goal might be any English word between 3 to 8 letters long, with the implied restriction that the word had to be spelled out of the available blocks.

Structure & Design Participants were first shown the initial layout of the blocks. They could then advance the scenario, watching several actions play out as an animated video. The video would then pause at a judgment point, giving participants time to guess the word being spelled via text box entry. Participants could *add* as many guesses as they liked, and also *remove* any previous guesses that they no longer considered likely. They could then advance to the next judgment point, continuing in this way until the end of the scenario. Each participant was presented 8 out of the 16 scenarios, after first completing a tutorial and a comprehension quiz. To incentivize high quality responses, we paid participants a reward based on the accuracy of their guesses ($\$0.1/n$ for every correct answer out of n guesses), and presented the bonus point breakdown after they completed each scenario.

[1] While this algorithm has $O(T^2)$ runtime due to rejuvenation, in practice runtime is closer to $O(T)$ due to reuse of likelihood computations in L6. A slight variant can guarantee $O(T)$ runtime by forgetting previous observations (Beronov et al., 2021).



(a) Intersection over Union (IoU) between model inferences and the mean human distribution of guesses.



(b) Average goal accuracy (posterior probability of true goal) across humans, models, and conditions.

Figure 2: Human similarity and accuracy of goal inference models, measured in terms of **(a)** IoU with mean human inferences, and **(b)** average goal accuracy. Each bar corresponds to a model (and sample size N), while each column is a condition. We computed human-human IoU through repeated sampling of 50-50 splits. Error bars denote 95% confidence intervals.

Experimental Conditions To tease apart the predictions of our model from those that would be made by either exact Bayesian inference or pure bottom-up proposals, we designed our 16 scenarios to fall into one of four conditions:

Bottom-Up Friendly. Words are stacked more-or-less linearly, such that it is sufficient to guess words that complete either the most recently stacked tower, or any partial word.

Irrational Alternatives. Blocks are stacked so that some bottom-up guesses are made irrational, like our p i n k example from Figure 1.

Garden Paths. Cases where bottom-up guessing suggests a plausible but misleading interpretation of the first few actions, which turn out to be merely instrumental for the true goal.

Uncommon Words. The true goal is either a longer or more uncommon word (aft, chump, wizard, banish), which people and bottom-up proposals might find difficult to initially guess. Otherwise similar to the first condition.

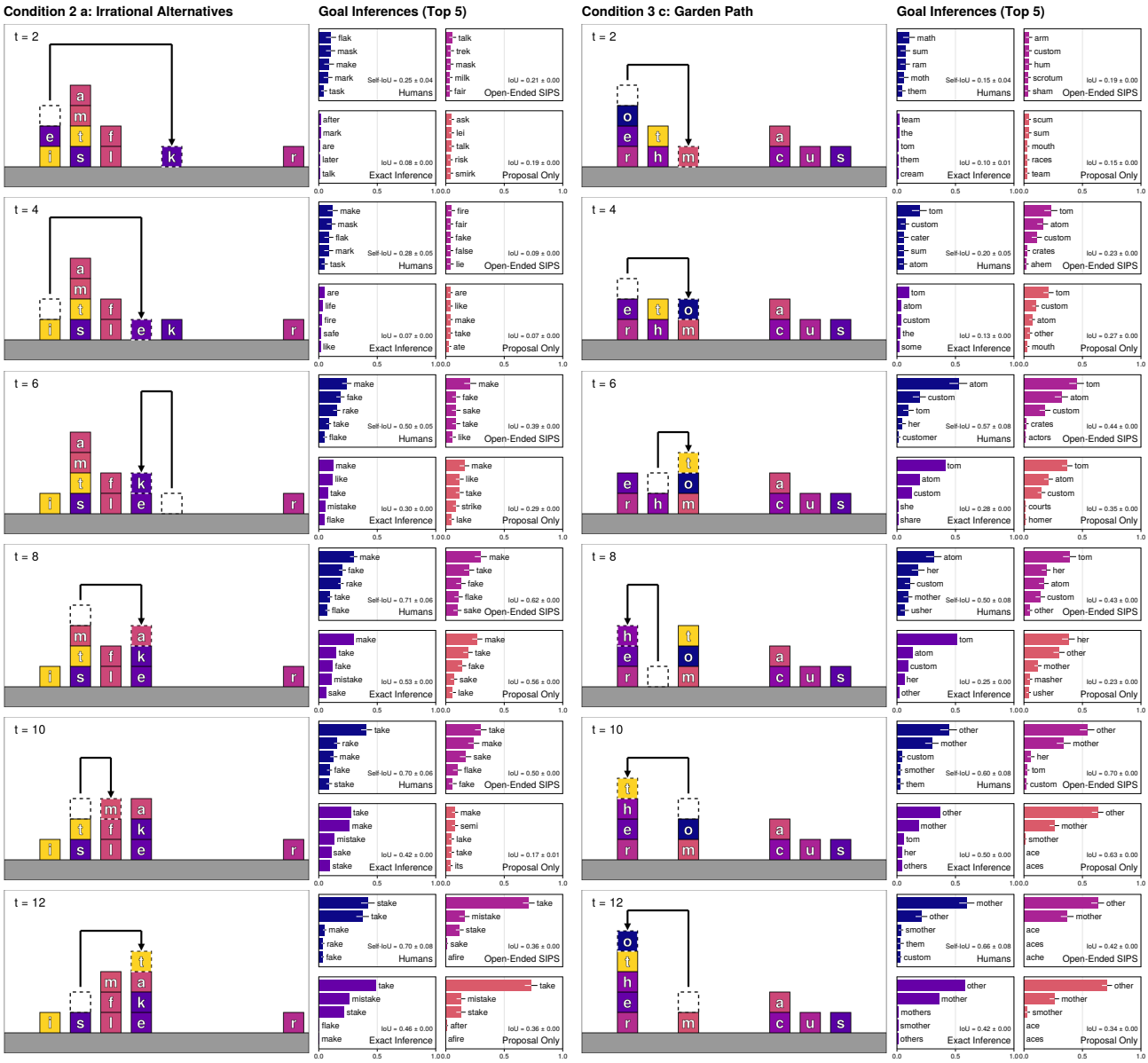
Participants We recruited 100 US participants fluent in English via Prolific (mean age 39.4; 44 women, 54 men, 2 non-binary), such that every scenario was completed by 50 individuals. Participants were paid US$15/hr along with the bonus described earlier. Familiarity with word games varied, with 17 reporting that they played word games daily, 22 weekly, 30 every 1-2 months, 11 yearly, and 20 almost never.

Despite comprehension checks, a subset of participants did not follow instructions correctly, either because they never updated their guesses (36 out of 800 scenario responses), or only added guesses without removing them (139 out of 800). As such, we excluded such responses from our analysis.

Model Configuration We implemented open-ended SIPS using the particle filtering extension of the Gen.jl probabilistic programming framework (Zhi-Xuan, 2020; Cusumano-Towner et al., 2019), and the Blocksworld domain in the Planning Domain Definition Language (Zhi-Xuan, 2022; McDermott et al., 1998). We fit parameters via grid search to improve model similarity with humans as measured by the intersection over union (IoU) between distributions (see Appendix), which gave an inverse temperature of $\beta = 1.0$, planning budget of $B = 100$, and prior $P(g)$ fitted to tempered word frequencies from the `wordfreq` library (Speer, 2016), using the `3of6game` word list as our dictionary (Beale, 2016). We ran open-ended SIPS with $N \in \{2, 5, 10, 20, 50\}$ particles, taking the mean and variance over $M = \max(10, 200/N)$ trials. We describe the proposals $Q(g|s_t, a_t)$ in the next section.

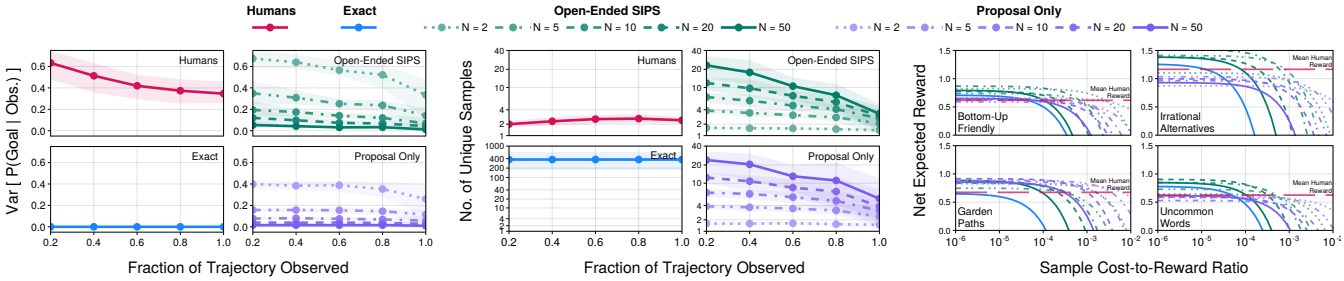
Alternative Models As alternatives to our hybrid SMC model, we tested both (i) exact inference via fully enumerative Bayesian inverse planning over all valid English goal words (145–807 words, depending on the scenario) and (ii) pure bottom-up sampling using a subgoal-conditioned proposal. Exact inference was implemented using the same model parameters as open-ended SIPS, except that all goals were considered as hypotheses from the outset.

For the bottom-up proposal, we sampled complete words g by conditioning an n-gram model on some partial word that can be stacked from blocks in the current state s_t. We used $n = 5$, fitting the n-gram on the same tempered word frequencies used for the prior $P(g)$. To decide which partial word to complete, the proposal first considers the tower stacked by the last action a_t, sampling a completion if it is sufficiently word-like (as determined by the n-gram model). If not, it considers if the last block was moved because the agent intended to reach some block *underneath* it, and tries to form a word with one of those blocks. If no way of using those blocks is sufficiently word-like, the proposal samples a random tower in state s_t (weighted by how word-like it is), then samples a completion. Other proposals are discussed in the Appendix.



(a) Irrational Alternatives scenario: **stake** **(b)** Garden Path scenario: **mother**

Figure 3: Step-by-step inference results on two illustrative Block Words scenarios. On the left, we show the sequence of actions, and on the right, the 5 most probable goals at each step for humans and our models ($N = 2$ for the sampling-based methods), averaged across humans and algorithm runs (error bars reflect the standard error). In **(a)**, only the bottom-up proposal fails to infer that **m** being stacked on **f** at $t = 10$ implies that **make** and **fake** are less likely than **take**. In **(b)**, both open-ended SIPS and humans exhibit sticky inferences at $t = 8$, assigning high weight to **atom** and **custom** as guesses as a result of the garden path trajectory. In contrast, the bottom-up proposal displays a recency bias since it does not store previous guesses.



(a) Total variance of the goal probability estimates produced by each method vs. humans, averaged over scenarios.

(b) Sample efficiency of inference methods over time, measured by the number of *unique* tracked hypotheses (averaged over scenarios).

(c) Expected reward minus cognitive cost, as a function of the ratio of sample cost to the reward of accurate goal inference.

Figure 4: Response variance, sample efficiency, and cognitive cost trade-offs vs. humans. Ribbons show 10th-90th quantiles.

Results

We analyzed human responses and model outputs by comparing them in terms of distribution similarity (Fig. 2a), average accuracy (Fig. 2b), step-by-step inferences (Fig. 3), response variance (Fig. 4a), sample efficiency (Fig. 4b), and resource rationality (Fig. 4c). Additional results (e.g. accuracy vs. runtime) and sensitivity analyses are in the Appendix.

Open-ended SIPS is most similar to human inferences across all conditions. As we predicted, human inferences showed the highest similarity with open-ended SIPS (IoU = 0.33–0.36 for all N) compared to exact inference (IoU = 0.31) or bottom-up guessing (IoU = 0.30–0.32), with $N = 2$ samples being the most similar. Notably, open-ended SIPS was more similar to humans in the *Irrational Alternatives* condition, with both achieving considerably higher accuracy than the bottom-up only heuristic, indicating that humans indeed engage in inverse planning. Our model was also more similar to humans than exact inference, especially in the *Bottom-Up Friendly* and *Garden Path* conditions, consistent with the hypothesis that humans engage in bottom-up sampling.

Step-by-step human inferences are best matched by open-ended SIPS. The step-by-step comparisons in Figure 3 help to elucidate these aggregate findings. On one hand, open-ended SIPS and the proposal-only model make initial guesses that are biased towards words that complete the first few stacked letters, whereas the exact posterior is much more uncertain. On the other hand, humans account for the rationality of the observed actions when drawing inferences (e.g. Figure 3(a), $t = 10$), just like our exact and approximate Bayesian inverse planning algorithms.

Our model's algorithmic properties best explain human variance and guess counts. In Figure 4, we compare the *algorithmic* properties of the models. Human variance was best matched by open-ended SIPS with $N = 2$. Bottom-up proposals had lower variance, and did not prune samples as effectively as sample-matched counterparts. Exact inference is zero-variance, but at the cost of tracking drastically more hypotheses. As such, it was dominated by open-ended SIPS in terms of net reward when accounting for cognitive costs (Fig. 4c). The comparison with pure bottom-up proposals was more nuanced. If reweighting a sample via inverse planning is costly enough, pure bottom-up guessing can be more resource-rational (Lieder & Griffiths, 2020). However, there is a large range of cost-ratios where it pays to do inverse planning. Since humans attained more reward than all proposal-only baselines in the *Irrational Alternatives* condition, this suggests that they indeed find inverse planning worthwhile.

Discussion

In comparison to alternative models, our sampling-based account of open-ended goal inference is best supported on both empirical and theoretical grounds, providing an algorithmically plausible explanation for the speed and flexibility of human goal inference. Still, our experiments find that humans remain more similar to themselves (IoU = 0.44) than our best-fitting model (IoU = 0.35). Part of this might be explained by the discrepancy between the statistics of how humans guess in word games versus the text corpus frequencies that inform our model. This could be addressed by deriving a prior and proposal from human guesses. Humans also appear to exhibit *stickier* inferences in garden path cases, whereas open-ended SIPS tends to avoid them when run with larger values of N by proposing new goals at every step. This suggests that humans may be *adaptive* in deciding when to rejuvenate their hypotheses (Del Moral et al., 2012; Elvira et al., 2016). Finally, unlike our model, humans might forget older observations, becoming more inaccurate, but also more efficient at inference. SMC algorithms that selectively forget past observations could mimic this (Beronov et al., 2021).

Another open question is how bottom-up sampling can be made more general. In future work, we plan to explore how the statistics of co-occurring subgoals can be distilled from web-scale language models (West et al., 2022) into domain-specific models for rapid hypothesis generation. These statistics might be augmented by static analysis of environment models, automatically determining which subgoals are instrumental for other goals (Blum & Furst, 1997). Such mechanisms for flexible domain adaptation could provide an even richer picture of how we contend with the infinitude of ends that others pursue, even in the face of our very finite means.

Acknowledgements

This work was funded in part by the DARPA Machine Common Sense, AFOSR, and ONR Science of AI programs, along with the MIT-IBM Watson AI Lab and gifts from the Aphorism Foundation and the Siegel Family Foundation. Tan Zhi-Xuan is supported by an Open Philanthropy AI Fellowship.

Code and Data Availability

Code and data for this paper can be found at the accompanying OSF repository: `https://osf.io/bygwm/`

References

Alanqary, A., Lin, G. Z., Le, J., Zhi-Xuan, T., Mansinghka, V., & Tenenbaum, J. (2021). Modeling the mistakes of boundedly rational agents within a Bayesian theory of mind. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 43).

Baker, C. L., Saxe, R., & Tenenbaum, J. B. (2009). Action understanding as inverse planning. *Cognition*, *113*(3), 329–349.

Barto, A. G., Bradtke, S. J., & Singh, S. P. (1995). Learning to act using real-time dynamic programming. *Artificial Intelligence*, *72*(1-2), 81–138.

Beale, A. (2016). *12dicts package.* Retrieved from `http://wordlist.aspell.net/12dicts-readme/`

Beronov, B., Weilbach, C., Wood, F., & Campbell, T. (2021). Sequential core-set Monte Carlo. In *Uncertainty in artificial intelligence* (pp. 2165–2175).

Blokpoel, M., Kwisthout, J., van der Weide, T. P., Wareham, T., & van Rooij, I. (2013). A computational-level explanation of the speed of goal inference. *Journal of Mathematical Psychology*, *57*(3-4), 117–133.

Blum, A. L., & Furst, M. L. (1997). Fast planning through planning graph analysis. *Artificial Intelligence*, *90*(1-2), 281–300.

Chandra, K., Chen, T., Li, T.-M., Ragan-Kelley, J., & Tenenbaum, J. B. (2023). Inferring the future by imagining the past. In *Thirty-seventh Conference on Neural Information Processing Systems.*

Chopin, N. (2002). A sequential particle filter method for static models. *Biometrika*, *89*(3), 539–552.

Cusumano-Towner, M. F., Saad, F. A., Lew, A. K., & Mansinghka, V. K. (2019). Gen: A general-purpose probabilistic programming system with programmable inference. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation* (pp. 221–236).

Daw, N., & Courville, A. (2008). The pigeon as particle filter. *Advances in Neural Information Processing Systems*, *20*, 369–376.

Del Moral, P., Doucet, A., & Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *68*(3), 411–436.

Del Moral, P., Doucet, A., & Jasra, A. (2012). On adaptive resampling strategies for sequential Monte Carlo methods. *Bernoulli*, *18*(1), 252–278.

Elvira, V., Míguez, J., & Djurić, P. M. (2016). Adapting the number of particles in sequential Monte Carlo methods through an online scheme for convergence assessment. *IEEE Transactions on Signal Processing*, *65*(7), 1781–1794.

Gergely, G., & Csibra, G. (2003). Teleological reasoning in infancy: The naıve theory of rational action. *Trends in cognitive sciences*, *7*(7), 287–292.

Graesser, A. C., Singer, M., & Trabasso, T. (1994). Constructing inferences during narrative text comprehension. *Psychological review*, *101*(3), 371.

Griffiths, T. L., & Tenenbaum, J. B. (2006). Optimal predictions in everyday cognition. *Psychological science*, *17*(9), 767–773.

Guthrie, D., & Hepple, M. (2010). Storing the web in memory: Space efficient language models with constant time retrieval. In *Proceedings of the 2010 conference on Empirical Methods in Natural Language Processing* (pp. 262–272).

Hassin, R. R., Aarts, H., & Ferguson, M. J. (2005). Automatic goal inferences. *Journal of Experimental Social Psychology*, *41*(2), 129–140.

Heider, F., & Simmel, M. (1944). An experimental study of apparent behavior. *The American Journal of Psychology*, *57*(2), 243–259.

Hernández, C., & Meseguer, P. (2007). Improving lrta*(k). In *Proceedings of the 20th International Joint Conference on Artifical Intelligence* (pp. 2312–2317).

Jara-Ettinger, J., Gweon, H., Tenenbaum, J. B., & Schulz, L. E. (2015). Children's understanding of the costs and rewards underlying rational action. *Cognition*, *140*, 14–23.

Jara-Ettinger, J., Schulz, L., & Tenenbaum, J. (2019). The naive utility calculus as a unified, quantitative framework for action understanding. *PsyArXiv*.

Kleiman-Weiner, M., Ho, M. K., Austerweil, J. L., Littman, M. L., & Tenenbaum, J. B. (2016). Coordinate to cooperate or compete: Abstract goals and joint intentions in social interaction. In *Proceedings of the Annual Meeting of the Cognitive Science Society.*

Koenig, S., & Likhachev, M. (2006). Real-Time Adaptive A*. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems* (pp. 281–288).

Koenig, S., & Sun, X. (2009). Comparing real-time and incremental heuristic search for real-time situated agents. *Autonomous Agents and Multi-Agent Systems*, *18*, 313–341.

Korf, R. E. (1990). Real-time heuristic search. *Artificial Intelligence*, *42*(2-3), 189–211.

Kwisthout, J., & Van Rooij, I. (2013). Bridging the gap between theory and practice of approximate bayesian inference. *Cognitive Systems Research*, *24*, 2–8.

Lew, A. K., Cusumano-Towner, M., & Mansinghka, V. K. (2022). Recursive monte carlo and variational inference

with auxiliary variables. In *Uncertainty in artificial intelligence* (pp. 1096–1106).

Lew, A. K., Ghavamizadeh, M., Rinard, M. C., & Mansinghka, V. K. (2023). Probabilistic programming with stochastic probabilities. *Proceedings of the ACM on Programming Languages*, *7*(PLDI), 1708–1732.

Lew, A. K., Matheos, G., Zhi-Xuan, T., Ghavamizadeh, M., Gothoskar, N., Russell, S., & Mansinghka, V. K. (2023). SMCP3: Sequential Monte Carlo with probabilistic program proposals. In *International Conference on Artificial Intelligence and Statistics* (pp. 7061–7088).

Lieder, F., & Griffiths, T. L. (2020). Resource-rational analysis: Understanding human cognition as the optimal use of limited computational resources. *Behavioral and Brain Sciences*, *43*, e1.

Liu, S., Ullman, T. D., Tenenbaum, J. B., & Spelke, E. S. (2017). Ten-month-old infants infer the value of goals from the costs of actions. *Science*, *358*(6366), 1038–1041.

McDermott, D., Ghallab, M., Howe, A., Knoblock, C., Ram, A., Veloso, M., ... Wilkins, D. (1998). *PDDL - the Planning Domain Definition Language* (Tech. Rep.). Yale Center for Computational Vision and Control.

Morris, A., Phillips, J., Huang, K., & Cushman, F. (2021). Generating options and choosing between them depend on distinct forms of value representation. *Psychological science*, *32*(11), 1731–1746.

Moskowitz, G. B., & Olcaysoy Okten, I. (2016). Spontaneous goal inference (sgi). *Social and Personality Psychology Compass*, *10*(1), 64–80.

Phillips, J., Morris, A., & Cushman, F. (2019). How we know what not to think. *Trends in cognitive sciences*, *23*(12), 1026–1040.

Ramírez, M., & Geffner, H. (2009). Plan recognition as planning. In *Twenty-first international joint conference on artificial intelligence.*

Ramírez, M., & Geffner, H. (2010). Probabilistic plan recognition using off-the-shelf classical planners. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 24).

Schulz, L. (2012). Finding new facts; thinking new thoughts. *Advances in child development and behavior*, *43*, 269–294.

Speer, R. (2016). *wordfreq: a Python library for looking up the frequencies of words in many languages.* Retrieved from `https://github.com/rspeer/wordfreq/`

Thaker, P., Tenenbaum, J. B., & Gershman, S. J. (2017). Online learning of symbolic concepts. *Journal of Mathematical Psychology*, *77*, 10–20.

Ullman, T., Baker, C., Macindoe, O., Evans, O., Goodman, N., & Tenenbaum, J. (2009). Help or hinder: Bayesian models of social goal inference. *Advances in Neural Information Processing Systems*, *22*.

Van Overwalle, F., Van Duynslaeger, M., Coomans, D., & Timmermans, B. (2012). Spontaneous goal inferences are often inferred faster than spontaneous trait inferences. *Journal of Experimental Social Psychology*, *48*(1), 13–18.

Vered, M., Kaminka, G. A., & Biham, S. (2016). Online goal recognition through mirroring: Humans and agents. In *Annual conference on advances in cognitive systems 2016.*

Vul, E., Alvarez, G., Tenenbaum, J., & Black, M. (2009). Explaining human multiple object tracking as resource-constrained approximate inference in a dynamic probabilistic model. *Advances in Neural Information Processing Systems*, *22*.

West, P., Bhagavatula, C., Hessel, J., Hwang, J., Jiang, L., Le Bras, R., ... Choi, Y. (2022). Symbolic Knowledge Distillation: from general language models to commonsense models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 4602–4625). Seattle, United States: Association for Computational Linguistics.

Ying, L., Zhi-Xuan, T., Mansinghka, V., & Tenenbaum, J. B. (2023). Inferring the goals of communicating agents from actions and instructions. *ICML 2023 Workshop on Theory of Mind in Communicating Agents*.

Zhi-Xuan, T. (2020). *GenParticleFilters.jl.* Retrieved from `https://github.com/probcomp/GenParticleFilters.jl`

Zhi-Xuan, T. (2022). *PDDL.jl: An extensible interpreter and compiler interface for fast and flexible AI planning*. Unpublished master's thesis, MIT.

Zhi-Xuan, T., Mann, J., Silver, T., Tenenbaum, J., & Mansinghka, V. (2020). Online Bayesian goal inference for boundedly rational planning agents. *Advances in Neural Information Processing Systems*, *33*.

Zhi-Xuan, T., Ying, L., Mansinghka, V., & Tenenbaum, J. B. (2024). Pragmatic instruction following and goal assistance via cooperative language guided inverse plan search. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multi-Agent Systems.*

Appendix

Experiment Interface

The web interface used by participants is shown in Figure A1. At each judgment point, participants typed their guesses into the text box, which validated whether the guess was between 3 and 8 characters and used only the letters that were available. Participants could also remove their guesses by clicking the ⊗ symbol next to each guess. The list of guesses was converted into a probability distribution by assigning a probability of $1/n$ to each word among the n guesses. Participants could rewatch the most recent segment of the animation by pressing the *Replay* button, or rewatch the whole animation up to the judgment point by pressing the *Replay All* button. This interface is accessible at `https://block-words.web.app/?local=true`.

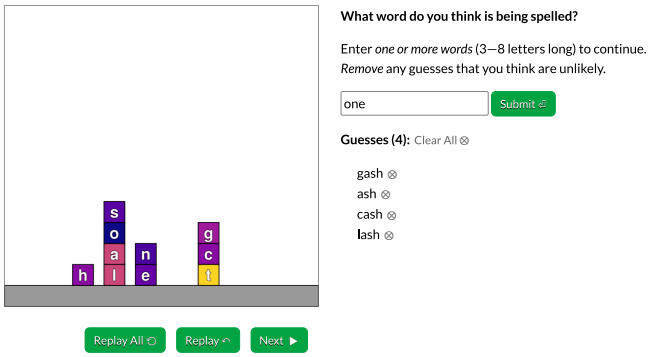


Figure A1: Interface for our open-ended goal inference task.

Model Fitting and Sensitivity Analysis

Our model of open-ended goal inference is characterized by two sets of parameters: The parameters of the *generative model* $P(g, \pi_{1:t}, s_{0:t}, a_{1:t})$, and the parameters of the *inference algorithm* which approximates $P(g|s_{0:t}, a_{1:t})$. We fit the parameters of the generative model across the following ranges:

- Goal prior word temperature $T_w \in \{1, 2, 4, 8, 16\}$
- Inverse temperature $\beta \in \{\frac{1}{4}, \frac{1}{2}, 1, 2, 4\}$
- Planning budget $B \in \{5, 10, 20, 50, 100, 200, 500\}$
- Replanning cadence $\Delta t \in \{1, 2\}$
- RTHS search strategy $\sigma =$ A* or BFS

T_w controls tempering of the `wordfreq`-derived word frequencies used for the goal prior $P(g)$, and β controls the optimality of action selection. B is the planning budget for real-time heuristic search (RTHS) algorithm, Δt is the number of timesteps between each call to RTHS that updates the policy π_t, and σ controls how nodes are expanded by RTHS, which is done either via A* search around each neighbor of the current state s_t (guided by the FF heuristic as the default $\hat{Q}_{\pi_t}$ value) as in LSS-LRTA* (Koenig & Sun, 2009), or via breadth-first search (BFS) around the current state s_t, as in LRTA*-LS (Hernández & Meseguer, 2007).

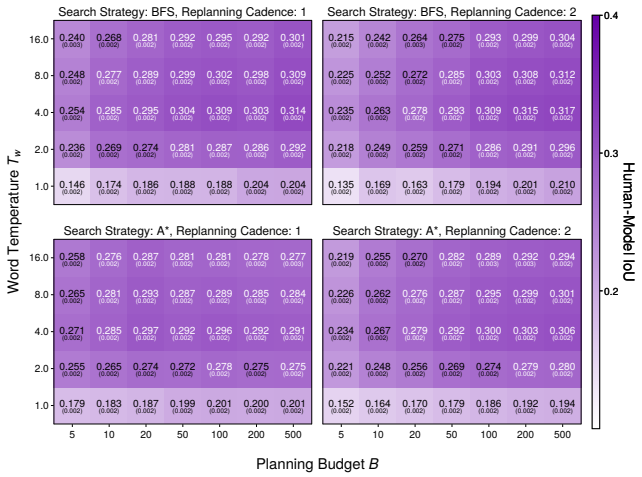


Figure A2: Human-model similarity (IoU) across generative model parameters when using exact Bayesian inference.

For the inference algorithm, we fit these parameters:

- n-gram word temperature $T_w \in \{1, 2, 4, 8, 16\}$
- n-gram termination bias $\varepsilon \in \{0, 0.05, 0.1, 0.15, 0.2, 0.25\}$
- Bottom-up proposal strategy $Q \in \{\text{LAST-AND-NEXT}, \ldots\}$
- Number of samples $N \in \{2, 5, 10, 20, 50\}$

T_w tempers the word frequencies used to fit the n-gram model for the bottom-up proposal Q, and is matched to be the same value used for the goal prior $P(g)$. To capture the difficulty of guessing longer words, we modified the n-gram to have an additional ε probability of terminating after each character. For simplicity, we fixed the context length of the n-gram model to $n = 5$. Various ways of implementing the bottom-up proposal Q are discussed in the next section. We also vary the number of particles N used by open-ended SIPS.

Fitting procedure. Model fitting proceeded in two stages. We first fit the generative model parameters to improve similarity with humans, using *exact* inference to factor out stochasticity or performance issues in the inference algorithm from the quality of the generative model itself. Instead of Pearson's correlation coefficient (commonly used in other BToM studies), we used the intersection-over-union between human and model distributions (i.e. the Jaccard index) as our similarity metric, since it does not consider two probability vectors similar just because they both contain many zeros. Having determined values of $B = 100$, $\Delta t = 2$ and $\sigma =$ BFS that led to the most similarity with humans under the constraint of a reasonable runtime, we then fit the parameters of the open-ended SIPS algorithm. The best fitting inference parameters were $T_w = 4$ (which was matched with the goal prior's T_w), $\varepsilon = 0.05$, $Q =$ LAST-AND-NEXT, and $N = 2$.

Generative model sensitivity analysis. Figure A2 shows how similarity with humans varies across generative model parameters when using exact Bayesian inference. A higher planning budget B leads to a stronger fit, showing the importance of computing a good estimate of the agent's policy via

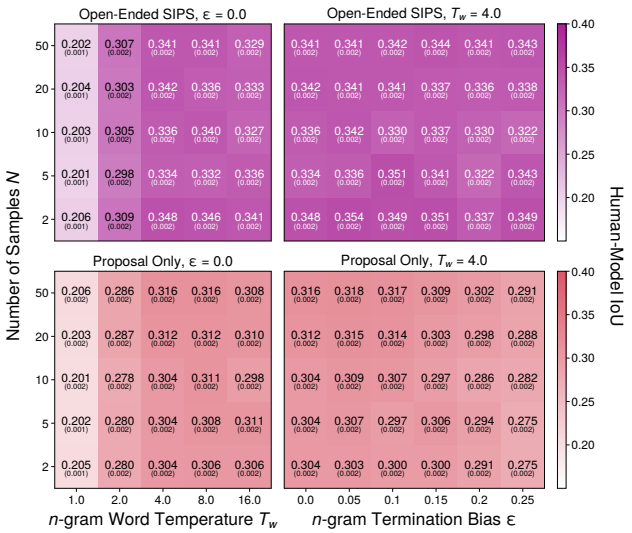


Figure A3: Human-model similarity (IoU) across inference parameters for open-ended SIPS and the bottom-up proposal.

planning. Interestingly however, the more informed search strategy, A*, led to slightly worse fits, suggesting that humans may not be explicitly modeling other's detailed search processes when performing inverse planning over large numbers of goals. Human similarity also improved when using a tempered word prior with $T_w = 4.0$, whereas using the raw corpus frequencies as the prior ($T_w = 1.0$) led to substantially poorer fits. In other words, people's intuitions for what words are likely in a word game are substantially broader than everyday word usage statistics.

Inference algorithm sensitivity analysis. Figure A3 shows how human-model IoU varies with different inference parameters for both open-ended SIPS and the proposal-only baseline. Again, untempered word frequencies ($T_w = 1$) lead to a poor fit, which cannot be overcome even with a highe sample count N. Intermediate tempering ($T_w = 4$) leads to the best fit, capturing the broader distribution of human word guesses. The termination bias ε has a less pronounced effect, with the best value ($\varepsilon = 0.05$) capturing the additional difficulty of proposing long words. Open-ended SIPS dominates the bottom-up proposal for almost all parameter settings.

Bottom-Up Proposals

For bottom-up sampling, we experimented with proposals $Q(g|s_t, a_t)$ of varying degrees of sophistication. In all of these proposals, we sample complete words g by conditioning an n-gram language model on some partial word that can be stacked from the blocks in the current state s_t. However, this still leaves undetermined *which* partial words, or subgoals, to consider. We implemented the following strategies for selecting partial words to complete:

ANY-TOWER samples a random block tower τ in state s_t, then tries to complete it into a full word. The probability of sampling a tower is proportional to how word-like the tower is — i.e., how probable it is according the n-gram model.

Open-Ended SIPS	**Human-Model Similarity (IoU)**				
Proposal Strategy Q	$N = 2$	$N = 5$	$N = 10$	$N = 20$	$N = 50$
ANY-TOWER	0.294	0.310	0.297	0.314	0.319
LAST-TOWER	0.348	0.340	0.317	0.339	0.344
NEXT-TOWER	0.302	0.304	0.306	0.309	0.329
LAST-AND-NEXT	**0.354**	0.336	0.342	0.341	0.341
	Goal Accuracy				
ANY-TOWER	0.115	0.162	0.156	0.159	0.160
LAST-TOWER	0.163	0.171	**0.184**	0.174	0.176
NEXT-TOWER	0.118	0.157	0.160	0.170	0.165
LAST-AND-NEXT	0.153	0.163	**0.183**	0.177	0.176

(a) Open-Ended SIPS

Bottom-Up	**Human-Model Similarity (IoU)**				
Proposal Strategy Q	$N = 2$	$N = 5$	$N = 10$	$N = 20$	$N = 50$
ANY-TOWER	0.211	0.218	0.217	0.224	0.228
LAST-TOWER	0.300	0.301	0.301	0.306	0.310
NEXT-TOWER	0.226	0.223	0.221	0.231	0.237
LAST-AND-NEXT	0.303	0.307	0.309	0.315	**0.318**
	Goal Accuracy				
ANY-TOWER	0.067	0.071	0.072	0.069	0.070
LAST-TOWER	**0.141**	0.134	0.139	0.138	0.134
NEXT-TOWER	0.073	0.074	0.068	0.072	0.074
LAST-AND-NEXT	0.132	0.140	0.135	**0.141**	0.137

(b) Proposal-Only Baseline

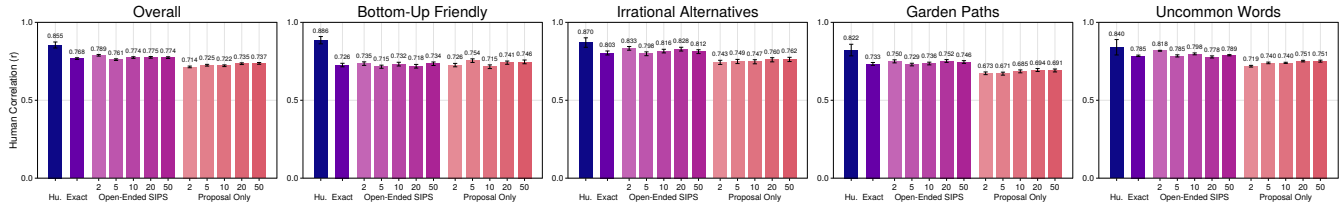
Table A1: Effect of bottom-up proposal strategy Q on human-model similarity (IoU) and goal accuracy.

LAST-TOWER tries to sample a word that completes the tower *most recently* stacked by action a_t. However, if this tower τ is not sufficiently word-like (or if a_t is not a stacking action), the proposal falls back to ANY-TOWER instead. This is implemented by comparing the probability p_{last} of the most recently stacked tower under the n-gram against the probability p_{rand} of an equally tall tower of random blocks, then deciding to complete the last tower with probability $\frac{p_{\text{last}}}{p_{\text{last}}+p_{\text{rand}}}$.

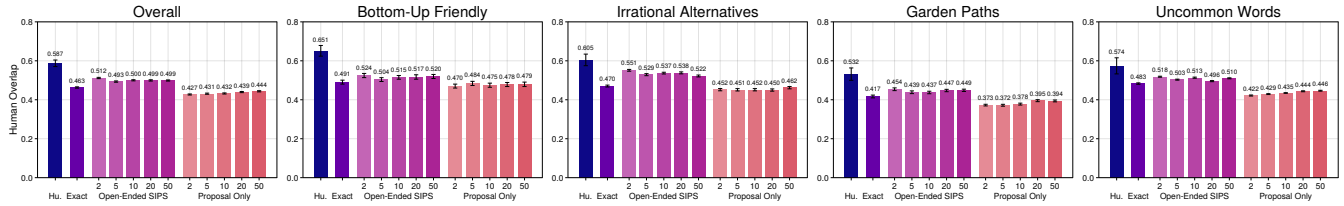
NEXT-TOWER focuses on cases where the agent is unstacking a tower of blocks in order to reach a block in that tower. The proposal considers all ways of using a block in the most recently unstacked tower to complete some other block tower. One of these candidate towers τ is selected with probability proportional to how word-like it is, and then a completion is sampled from to the n-gram model. If no candidate is word-like enough compared to the probability of selecting a random block, then the proposal defaults to ANY-TOWER.

LAST-AND-NEXT is the most sophisticated of our proposals, which we describe and use in the main text. It is equivalent to LAST-TOWER, except it defaults to NEXT-TOWER if the last action is not a stacking action, or if the last stacked tower is not sufficiently word-like.

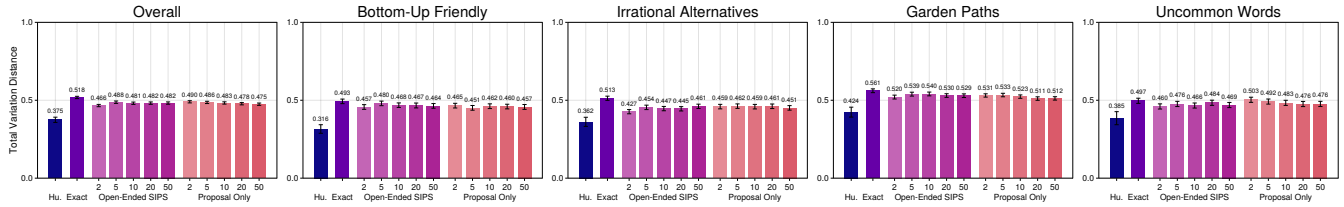
Table A1 shows how these different proposal strategies compare in terms of both human similarity and goal inference accuracy. As expected, the LAST-AND-NEXT proposal best matches human inferences whether it is incorporated into open-ended SIPS (Table A1(a)) or used on its own (Table A1(b)), while achieving (close to) the highest accuracy. The LAST-TOWER also performs well in this regard, albeit with slightly lower human similarity. In contrast, both ANY-TOWER and LAST-TOWER fare poorly. Open-ended SIPS is able to make up for their weakness to some degree, showing the value of inverse planning even with a weak proposal.



(a) Pearson's correlation r between human inferences and model inferences across conditions.



(b) Distribution overlap between human inferences and model inferences across conditions.



(c) Total variation distance between human inferences and model inferences across conditions (lower is better).

Figure A4: Similarity of average human and model goal inferences measured in terms of **(a)** Pearson's correlation coefficient, **(b)** distribution overlap ($\sum_g \min[P(g), Q(g)]$ for distributions P and Q), and **(c)** total variation distance ($\frac{1}{2}\sum_g |P(g), Q(g)|$). As in Figure 2, error bars denote 95% CIs, calculated from 1000 bootstrap samples of the distribution of human responses.

Handling auxiliary randomness. Note that all of our bottom-up proposals make use of *auxiliary randomness* (Lew et al., 2022) when sampling a tower τ to complete into a full word g. This means that even though we can *sample* from $g \sim Q(g|s_t, a_t)$, we cannot exactly *evaluate* the probability $Q(g|s_t, a_t)$ used in the importance weight. Instead, we can only evaluate $Q(g|s_t, a_t, \tau)$ using the n-gram model, which is conditional on the choice of tower $\tau \sim Q(\tau)$. In the context of an SMC algorithm like open-ended SIPS, however, we can use an *unbiased density sampler* of $Q(g|s_t, a_t)$ (Lew, Ghavamizadeh, et al., 2023; Lew et al., 2022), which returns both g and a weight w such that $\mathbb{E}_Q[w^{-1}|g] = Q(g|s_t, a_t)^{-1}$. This weight w can then be used as the denominator when computing importance weights in L6 of Algorithm 1. Using $w = Q(g|s_t, a_t, \tau)$ satisfies this property.

Additional Similarity Metrics

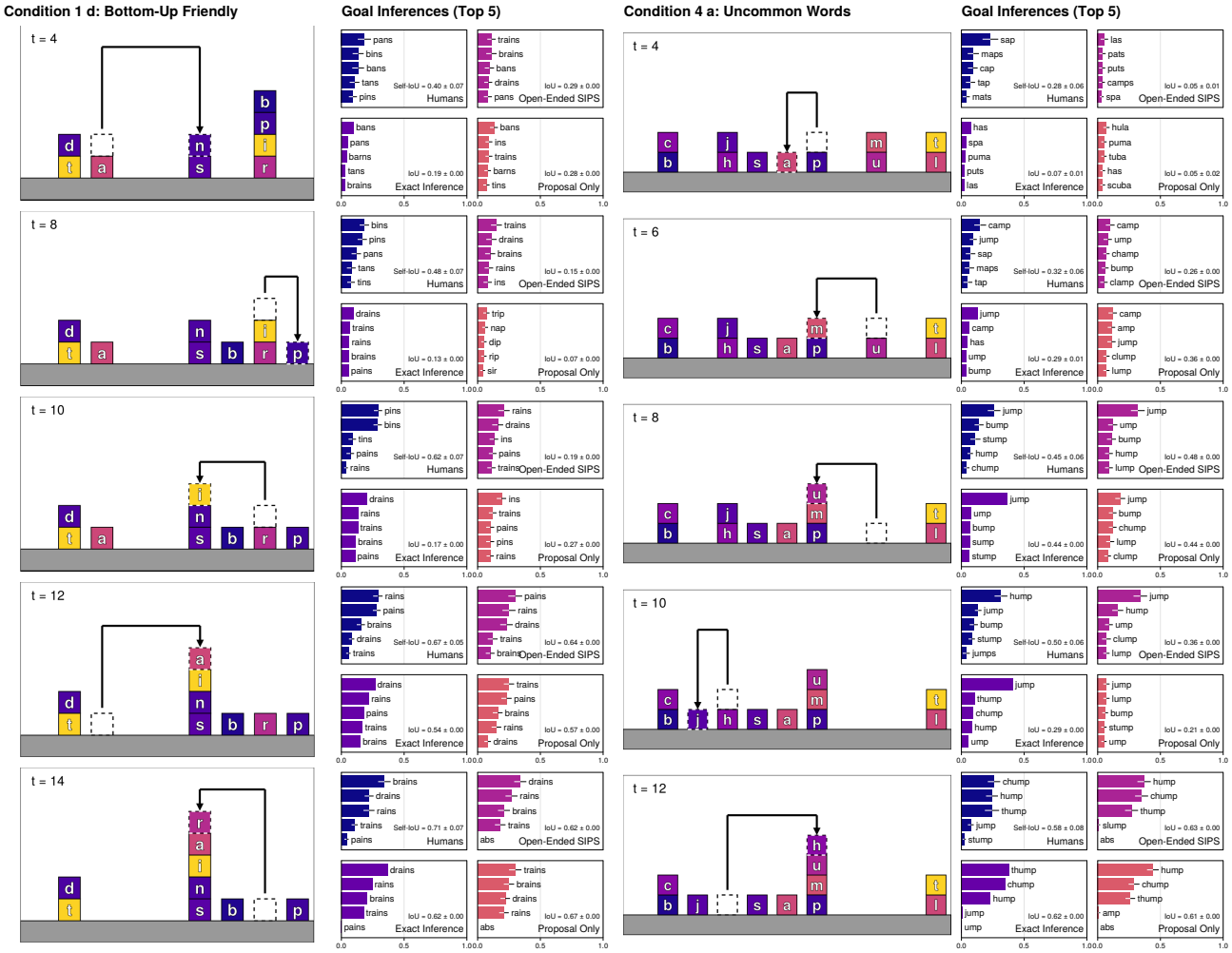
In Figure A4, we report additional measures of similarity between human goal inferences and model outputs: (a) Pearson's correlation r, (b) the overlap coefficient between distributions (a generalized version of recall), and (c) total variation distance (which captures the *maximum* difference in the probability of any event under two distributions). Regardless of the metric, open-ended SIPS is more similar to humans than exact inference or the bottom-up proposal. This is most pronounced in the *Irrational Alternatives* condition.

Additional Step-by-Step Comparisons

In Figure A5, we show step-by-step comparisons between human and model inferences for the two experimental conditions not covered in the main text.

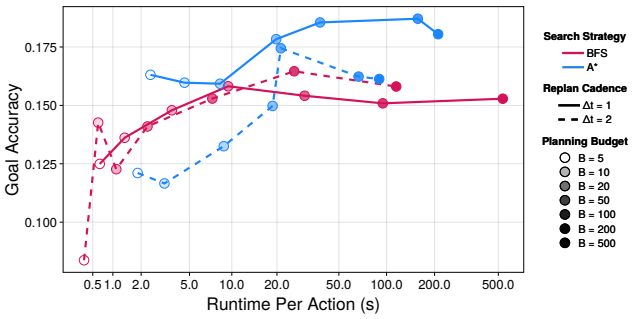
In the *Bottom-Up Friendly* scenario, limited inverse planning is necessary, and so the bottom-up proposal is as good an explanation for human goal inference as open-ended SIPS for all steps except $t = 8$. At this step, the bottom-up proposal generates words that end in **p** by following the LAST-TOWER proposal strategy. This fails to take into account the previous actions of stacking block **n** onto block **s**, highlighting the importance of inverse planning in even simple scenarios.

In the *Uncommon Words* scenario, we chose the uncommon word **chump** to be the goal, and designed actions to make more common distractors like **jump** and **hump** seem likely. As expected, most humans failed to think of **chump** as a possibility until the very last step ($t = 12$). Open-ended SIPS and the proposal only baseline with $N = 2$ particles reflected this tendency, demonstrating how inferring rare events is difficult with a small number of samples. In contrast, the exact inference baseline enumerates all possible words at every step, and hence assigns **chump** a significant probability at $t = 10$. This is the case even though **chump** is less likely under the goal prior $P(g)$: an event being rare under the proposal Q leads to qualitatively different behavior than exact inference over events with low prior probabilities.

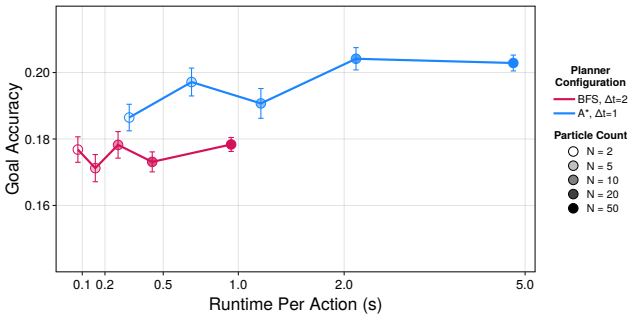


(a) Bottom-Up Friendly scenario: **drains** **(b)** Uncommon Words scenario: **chump**

Figure A5: Step-by-step inference results on scenarios from the *Bottom-Up Friendly* and *Uncommon Words* conditions. In **(a)** there are few qualitative differences between the inference methods, apart from $t = 8$, where the proposal only baseline generates words that end in **p** instead of taking into account the fact that **n** was previously stacked on **s**. In **(b)**, humans, open-ended SIPS, and the bottom-up proposal largely fail to guess the uncommon word **chump** until the very last timestep ($t = 12$), in contrast to the fully enumerative baseline, which assigns a non-trival probability to **chump** by $t = 10$.



(a) Exact Inference



(b) Open-Ended SIPS

Figure A6: Accuracy vs. runtime for **(a)** exact inference and **(b)** open-ended SIPS across planner configurations.

Accuracy and Runtime

While our analysis in this paper focused on open-ended SIPS as a rational process model of human goal inference, our algorithm can also be used as a practical tool for building AI systems that better infer people's goals in order to assist them (Zhi-Xuan et al., 2024). To that end, we compare the accuracy-runtime tradeoffs of both open-ended SIPS and the baseline methods. All experiments were conducted on a laptop with an i7-1370P 1.90 GHz CPU and 64 GB of RAM.

Effect of planner configuration. In Figure A6, we show how the accuracy of goal inference changes as a function of the planner configuration used in the generative model. Accuracy is plotted against algorithm runtime per observed action. We find that accuracy generally increases with planning budget, indicating the importance of spending enough computation on calculating good $\hat{Q}_{\pi_t}$ estimates, which improves the quality of the action likelihood $P(a_t|g)$. The effect of the search strategy is more subtle. Using A* search as the RTHS search strategy can *improve* accuracy when $\Delta t = 1$ (i.e. when the policy is updated at every timestep). However, this leads to an increased runtime compared to BFS for any given planning budget B, with A* being 2-5 times slower than BFS for low planning budgets. Still, using A* with $\Delta t = 1$ achieves higher accuracy than BFS ever does, indicating its value when accuracy is paramount. Unlike the planning budget, increasing the particle count N does not appear to substantially improve the average accuracy of open-ended SIPS, with changes in planner configuration dominating any accuracy improvement from additional particles.

Method	Search Strategy	Particle Count N	Runtime / Act. (s)	Accuracy $P(g_{\text{true}})$	Accuracy Std. Dev.
Exact Inference	BFS	—	**7.31**	0.153	***0.000***
	A*	—	37.95	**0.186**	***0.000***
Open-Ended SIPS	BFS	2	**0.08**	0.177	0.218
		5	0.16	0.171	0.140
		10	0.26	**0.178**	0.095
		20	0.44	0.173	0.069
		50	0.95	0.178	**0.049**
Open-Ended SIPS	A*	2	**0.32**	0.186	0.225
		5	0.67	0.197	0.146
		10	1.18	0.191	0.108
		20	2.14	***0.204***	0.080
		50	4.73	0.203	**0.055**
Proposal Only	—	2	***0.0004***	0.145	0.171
		5	0.0006	**0.147**	0.104
		10	0.0008	0.139	0.073
		20	0.001	**0.147**	0.052
		50	0.003	0.146	**0.035**

Table A2: Accuracy and runtime across inference methods, RTHS search strategies, and particle counts. **Bold** entries denote the best performance within each method. *Italicized* entries denote best performance across methods.

Overall comparison. In Table A2, we show accuracy and runtime measures across all inference methods and particle counts. For the inverse planning methods, we fixed the planning budget to $B = 100$, but compare both the accuracy-maximizing planner configuration ($\sigma = \text{A*}, \Delta t = 1$) and the runtime-minimizing configuration ($\sigma = \text{BFS}, \Delta t = 2$). While exact inference has zero variance, the cost of tracking all goal hypotheses leads to an unacceptably high runtime for many applications. The bottom-up proposal is on the opposite end of the spectrum, achieving millisecond or less runtimes but with a subpar accuracy that does not increase with particle count. In contrast, open-ended SIPS combines the best of both worlds, achieving the highest goal accuracies while maintaining real-time speed. Changing the particle count N trades off variance in inference results vs. runtime. By using $N = 20$, the variance in the probability estimate of the true goal can also be limited to less than 10% while still taking less than half a second process each new observation. These results demonstrate the promise of open-ended SIPS as a practical algorithm for real-time open-ended goal inference.