UC Merced

Proceedings of the Annual Meeting of the Cognitive Science Society

Title

Goal Inference using Reward-Producing Programs in a Novel Physics Environment

Permalink

https://escholarship.org/uc/item/6tb2m1zv

Journal

Proceedings of the Annual Meeting of the Cognitive Science Society, 47(0)

Authors

Davidson, Guy
Todd, Graham
Colas, CŽdric
et al.

Publication Date

2025

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at https://creativecommons.org/licenses/by/4.0/

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

Goal Inference using Reward-Producing Programs in a Novel Physics Environment

Guy Davidson
Center for Data Science, NYU
and FAIR at Meta

Graham Todd
Computer Science
and Engineering, NYU

Cédric Colas
Brain and Cognitive Sciences
MIT

Junyi Chu
Department of Psychology
Stanford University

Julian Togelius
Computer Science
and Engineering, NYU

Joshua B. Tenenbaum
Brain and Cognitive Sciences
MIT

Todd M. Gureckis
Department of Psychology
NYU

Brenden M. Lake
Department of Psychology and
Center for Data Science, NYU

Abstract

A child invents a game, describes its rules, and in an instant, we can play it, judge progress, and even suggest new variations. What mental representations enable such flexible reasoning? We build on recent work formalizing naturally expressed goals as a type of program, grounding linguistic descriptions into precise scoring systems. To support this notion, we study human-created objectives in a physics game environment. We leverage the formal representations to quantitatively analyze relationships between reward geometry, goal complexity, and perceived difficulty. We then propose a proof-of-concept of a computational goal inference method using these program representations and behavioral demonstrations, offering a concrete proposal of how humans reason about others' goals.

Keywords: goals, play, goal inference, program synthesis, domain-specific language, physics environment

Introduction

Goals fundamentally shape human cognition and behavior—organizing our actions, directing our attention, and structuring our learning (Dweck, 1992; Austin and Vancouver, 1996; Fishbach and Ferguson, 2007). Yet cognitive science has largely taken goals as given, studying how predetermined objectives influence behavior while leaving deeper questions unexplored: What exactly are goals, and how are they mentally represented? These questions carry particular weight given two factors. First, flexibility: our representations of goals facilitate planning toward them, evaluating progress toward them, suggesting variations, and inferring likely goals from behavior. Second, novelty: humans, unlike other animals whose goals are largely shaped by survival needs (Tomasello, 2022), display a remarkable capacity to generate and pursue novel objectives. This ability emerges early in development (Chu and Schulz, 2020) and persists throughout life, with humans constantly inventing new challenges and games for themselves. Indeed, this *autotelic* nature — our drive to create and pursue self-generated goals — appears central to human learning and exploration (Czikszentmihalyi, 1990; Ryan and Deci, 2000; Chu et al., 2024). Understanding the cognitive foundations of this core human capability requires us to identify what mental representations could support both the remarkable breadth of human goals and our fluid ability to generate new ones.

Recent work by Davidson et al. (2025) proposes modeling goals as *reward-producing programs* within the language of thought paradigm (Fodor, 1979; Goodman et al., 2008; Rule et al., 2020). Under this framework, goals are represented as symbolic expressions with well-defined syntax and semantics, enabling the composition of complex objectives from simpler components. This approach facilitates concept reuse and allows partial or complete goal achievement evaluation, acting as an abstraction over the reward functions typically used in reinforcement learning. Here, we extend their work, applying these goal representations in a new domain and demonstrating how they facilitate goal inference from behavior. To study goal inference, we begin by collecting a rich dataset of human-generated goals, demonstrations of their achievement, and self-reported difficulty ratings. We then demonstrate how these reward-producing program representations enable inferring an agent's goal from a few samples of their behavior.

We proceed by describing our experiment, the formal semantics of our reward-producing goal programs, and our approach to translating from natural language descriptions into these structured representations. By analyzing the goals people create, we find relationships between the geometry of a goal's reward landscape, its complexity, and its assessed difficulty. We observe, in particular, that post-gameplay assessed difficulty is associated with reward sensitivity (the degree of accuracy required for top scores) and with actual participant-achieved scores. Finally, building on recent work on inferring goals from behavior, we demonstrate a proof-of-concept for a program-based inverse reinforcement learning algorithm.

Related work

Our treatment of goals as programs follows in a line of work instantiating Fodor's (1979) Language of Thought in code-like representations (Goodman et al., 2008; Piantadosi et al., 2012; Rule et al., 2020; Wong et al., 2023). Our dataset of human-generated goals is inspired by Davidson et al. (2025), which also collected human-generated goals in a rich physics-based environment. Our new dataset was designed to be more broadly useful to the community, with more human-generated goals (~ 400 vs. ~ 100), a simpler environment better suited for comparing humans and machines, and a simple action space that facilitates computing the optimal action for a given goal. This makes our dataset and environment better suited for building agents from autotelic learning principles or for building models of goal inference from the behaviors of others.

Our study of goal inference builds on a body of work related to Bayesian theory of mind—models that seek to infer the goals and/or beliefs that best explain a person's observed behavior (Baker et al., 2009; Ullman et al., 2009; Baker et al., 2011). Our work differs from this previous work in handling more expressive goal representations: previous models have

In D. Barner, N.R. Bramley, A. Ruggeri and C.M. Walker (Eds.), *Proceedings of the 47th Annual Conference of the Cognitive Science Society* ©2025 the author(s). This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY).

focused on much simpler objectives, such as agents reaching one or more target locations (Baker et al., 2011; Velez-Ginorio et al., 2017; Baker et al., 2017) or achieving a specific world state (Zhi-Xuan et al., 2020; Alanqary et al., 2021; Zhi-Xuan et al., 2024). Ultimately, to capture goal inference in our new dataset, we envision a complete model involving Bayesian inference over candidate reward-producing programs.

Experimental Design and Data Collection

We designed an experiment to study how humans create new goals. Our design prioritizes two key elements: enabling natural goal creation while maintaining precise labeling of goal objectives. Participants explored five different configurations of the environment, creating and demonstrating novel goals for each one. For each goal, they provided both natural language descriptions and explicit scoring criteria, allowing us to bridge between intuitive human goal descriptions and formal reward-producing program specifications (Davidson et al., 2025).

Game Environment. Our experimental game environment Figure 1 was designed to study goal creation while balancing ecological validity with computational tractability. We developed a novel two-dimensional launching game inspired by the famous *Angry Birds* game and similar physics-based experimental paradigms (e.g., Allen et al., 2019; LeGris et al., 2024). The core interaction is deliberately kept simple: participants launch a blue ball with adjustable power and angle and then observe the resulting physics-driven interactions. The environment included a carefully chosen set of interactive elements: static objects (bins, shelves), dynamic elements (conveyor belts, bouncers), and special items (magnets, portals, vanishers), see Figure 1. This combination enables rich goal creation while keeping the action space tractable — each play attempt involves just one launching decision (similar to pinball), making it feasible to simulate all possible trajectories in any given configuration of the environment. We implemented the environment using the `Phaser` game engine (Phaser Developers, n.d.), the `Matter.js` physics engine (Liam Brummitt, 2014), resources from Kenney Assets (Kenney, 2025), and the `Smile` web experiment development platform (Gureckis et al., 2025).

Experiment Protocol. The experiment consisted of three phases: participant training, goal creation across five levels, and goal demonstration. Before beginning, participants completed a tutorial introducing the environment mechanics and object interactions, followed by a guided walkthrough of the goal creation process. A comprehension quiz ensured understanding of key experiment requirements. In the main phase, participants encountered five different level configurations, each existing in two variants: a "full" version with many interactive elements and a "minimal" version with fewer objects. Participants were randomly assigned a level order and encountered two to three full variants, with the remaining levels shown in their minimal form. This variation allowed us to study how environmental complexity influences goal creation.

Participants followed a structured protocol for each level designed to capture both their creative process and formal goal

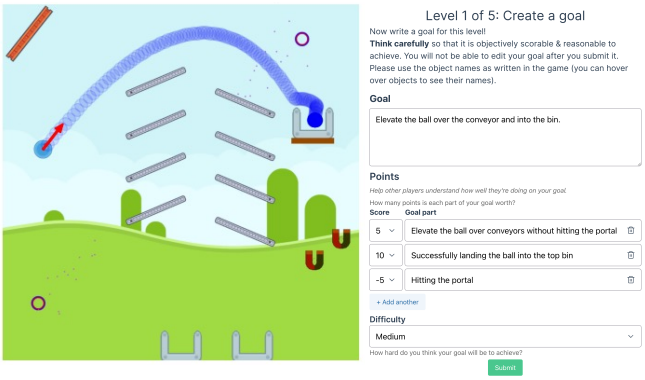


Figure 1: Experiment environment. *Left:* Our game environment, showing the main ball to be shot (colored blue), an example trajectory (blue ball positions) for a shot (red arrow), alongside several other objects. *Right* A form prompting the participant to create, describe, and specify a goal for this level. Our experiment is available online.

specifications. First, they explored the level for at least 30 seconds, making at least three practice shots to understand the level's dynamics. They then created their goal by providing both a natural language description and by assigning specific point values to different achievement conditions (see form in Figure 1, right). Participants also predicted the difficulty of their goal on a 5-point Likert scale. Finally, they demonstrated their goal through at least six recorded attempts, self-scoring each demonstration and optionally providing explanatory notes. After completing their demonstrations, participants reassessed their goal's difficulty, providing insight into how their understanding evolved through actual attempts.

Experimental Design Constraints and Data Collection. We placed two constraints on the games participants created: that they be objectively measurable, and reasonably achievable. These constraints helped ensure goals could be formally represented in our computational framework while remaining intuitive and engaging for participants. To enable analysis of goal creation and achievement, we collected data at multiple levels of abstraction. For each goal, we recorded natural language descriptions, structured point assignments for different achievement conditions, and difficulty ratings before and after demonstration attempts. The scoring system served as a bridge between participants' intuitive goal conceptualization and formal reward-producing program specifications. We also captured detailed behavioral data: the magnitude and direction of every shot throughout the experiment, self-assigned scores for demonstration attempts, optional explanatory notes, and complete state traces of the physics simulation. This rich dataset enables a high-level analysis of goal creation patterns and detailed validation of our formal goal representations.

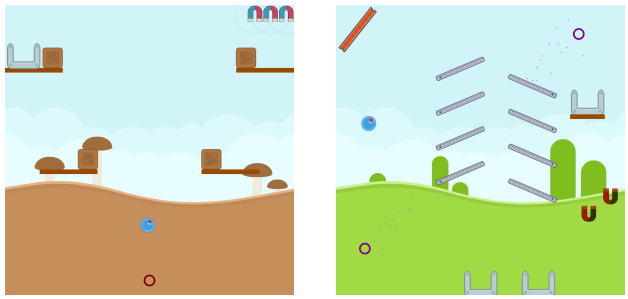
Participants. We contacted 135 participants through Prolific (Palan and Schitter, 2018), of which 107 consented to the study and submitted data on one or more goals. We paid participants a base rate of $8 (roughly $15/hour) with a bonus of $3. We exclude data from 20 participants who made no meaningful attempt to conform to the instructions. Accounting for three participants for whom we only have partial data due to technical difficulties, we arrived at a dataset of 394 goals.

First Example Goal

"land the ball inside grey basket, bonus points for breaking the boxes"

- **10**: land ball inside basket
- **2**: break single box

```
(define (goal p106-bin-and-boxes) (level V-STEPS-MINIMAL)
(preferences (and
    (preference ball_in_bin (exists (?b - bin) (at-end (in ?b main_ball))))
    (preference box_destroyed (exists (?bx - box) (once (vanished ?bx))))
))
(scoring (+
    (* 10 (count-once ball_in_bin) )
    (* 2 (>= (count-once-per-objects box_destroyed) 1 ))
)))
```



Second Example Goal

"Elevate the ball over the conveyor and into the bin"

- **5**: Elevate the ball over conveyors without hitting the portal
- **10**: Successfully landing the ball into the top bin
- **-5**: Hitting the portal

```
(define (goal p97-conveyor-bin) (level PORTALS-FULL)
(preferences (and
    (preference elevate_over_conveyors
      (exists (?c1 ?c2 -
          (either conveyor_1 reverse_conveyor_1))
        (then
          (hold (above ?c1 main_ball))
          (hold (in_motion main_ball))
          (hold (above ?c2 main_ball))
          (at-end (not (portalled main_ball)))
    )))
    (preference ball_bin
      (at-end (in bin_1 main_ball))
    )
    (preference ball_portal (exists (?p - portal)
      (once (touch main_ball ?p))
))))
(scoring (+
    (* 5 (count-once elevate_over_conveyors))
    (* 10 (count-once ball_bin))
    (* -5 (count-once ball_portal))
)))
```

Figure 2: Example goals and program translations. These goals were created by different participants in our experiment. *Left:* The participant created this goal for the level on the left. They specified two conditions, each mapping to a preference, one checking the terminal conditions (`ball_in_bin`, using the **at–end** operator) and one at any point in flight (`box_destroyed`, using the **once** operator). *Right:* The participant created this goal for the level on the right. The constraint about elevating over the conveyors required specifying a set of temporal conditions (under the preference (`elevate_over_conveyors_no_portal`, using the **then** operator).

Goal Programs

Program Semantics. We follow Davidson et al. (2022, 2025) in representing participant-generated goals as *reward-producing programs*. While our set of primitives may not perfectly match the cognitive primitives people use, we are able to approximately model the generative capacity our participants demonstrate. The goal programs capture the semantics of the goals participants created, mapping from observed outcomes after launching the ball to a (scalar) signal of success toward the specified goal. To execute these programs, we construct an interpreter that parses a program in our domain-specific language (DSL), evaluates it in the context of a shot trajectory, and returns a number indicating success on this goal. We operationalize this success signal as the point-based conditions participants specify, which we then use (with their goal demonstrations) to verify that we correctly grounded their descriptions in the correct semantics (see "goal to program translation"). We use a version of Davidson et al.'s (2025) PDDL-inspired (Ghallab et al., 1998) DSL, streamlined and adapted to our environment. We now describe key aspects of the grammar (and see Figure 2 for goal program examples).

Preferences. Preferences capture the *what* and *how* of the goals — elements that need to be tracked to assess whether or not a goal has been achieved and, if so, how successfully. Each preference (e.g., `ball_in_bin` in Figure 2, left) has a name by which it is referred to in scoring expressions and optionally quantifies one or more variables. Preference bodies take one of the following forms, each corresponding to a different category of temporal condition. The **once** operator (e.g., `ball_portal` in Figure 2, right) defines a preference that checks for conditions that are true at any point in time once the ball is in flight. The **at–end** operator (e.g., `ball_bin` in Figure 2, right) captures preferences whose conditions must be met at the end of the shot when all objects have settled. Finally, the **then** operator (e.g., `elevate_over_conveyors_no_portal` in Figure 2, right) captures a sequence of conditions that must be true following the ball being launched (which are homologous to linear temporal logic modals; Manna and Pnueli, 1992). Within an operator (or a modal under a **then** operator), conditions are defined as first-order logic expressions over predicates.

Scoring. The scoring section maps preferences to the signal of success toward goal achievement. Scoring expressions center around enumerating preference satisfactions using the following counting operators: The **count–once** operator captures whether or not a preference was satisfied (returning zero if it was not and one if it was). The **count–once–per–objects** operator counts each unique set of objects that satisfy a preference, and the **count** operator measures the total number of times a preference was satisfied. These counts are combined using arithmetic expressions (to assign different values to different preferences), logical expressions (to condition on the number of satisfactions), and a maximization operator.

Goal to Program Translation. Applying Language of Thought models to study participant-created artifacts requires mapping from the form participants provide (often natural language) to expressions in the domain-specific language of choice. This arduous step can introduce a non-negligible degree of subjectivity. To alleviate this burden, we created a pipeline to automate a first pass at mapping natural language to program semantics, using a fron-

tier LLM (`gemini-2.0-flash-thinking-exp`), a grammar parser, and the participant-provided demonstrations. Our pipeline is an example of the 'word model to world model' framework (Wong et al., 2023). The authors checked and (where required) corrected the pipeline-generated translations. We intend to open-source this pipeline with our dataset.

Goal Dataset Analyses

We offer preliminary analyses of our dataset. We simulate goal reward landscapes using our goal program interpreter, extract metrics from the geometry of reward distributions, and analyze the relationship between them and variables of interest.

Feature Description. We devised several metrics to analyze participant games, capturing program structure and reward landscape characteristics. We measured program complexity through the number of defined preferences and the number of nodes in its syntax tree. To characterize reward distributions, we computed the coverage of positive and maximum scores across the action space (shot power $\in 1\% - 100\%$, shot angle $\in 0^\circ - 359^\circ$, both in integer steps). We also devised a risk score that captures sensitivity to small action perturbations. We calculate this risk score by examining the relative drop in performance between the peak action reward and the average reward of its 5*x*5 neighborhood in action space (varying power by $\pm 2\%$ and angle by $\pm 2^\circ$). We capture participant behavior through their mean and max scores across their demonstrations and the average distance between demonstrated actions. We also analyze participant self-reported difficulty scores.

Key Findings. Our analysis revealed several interesting relationships between game design, player performance, and perceived difficulty: *Participants are sensitive to goals requiring precision.* Participants' post-gameplay difficulty assessments (but not pre-gameplay) showed positive correlation ($r = 0.22, p = 10^{-3}$) with reward sensitivity but negative correlation with demonstration mean and max performance ($r = -0.26, p < 10^{-5}$ and $r = -0.28, p < 10^{-6}$). *Program complexity predicts reward variability:* Program complexity, measured by the number of preferences and syntax tree nodes, was associated with a broader range of achievable reward values ($r = 0.50, p < 10^{-10}$ and $r = 0.39, p < 10^{-10}$) and a higher proportion of positively rewarding actions ($r = 0.22, p < 10^{-4}$ and $r = 0.18, p < 10^{-3}$).This suggests additional preferences are often used to introduce possible bonuses and minor objectives. *Risk scores capture demonstration difficulty.* Higher risk scores, corresponding an increased presence of brittle solutions, show negative correlations with performance (demonstration mean: $r = -0.52, p < 10^{-10}$; max: $r = -0.42, p < 10^{-10}$). *Positive reward clusters and coverage trade off.* Created goals tend to either have few large reward regions or many small ones, as seen by a strong negative correlation between the number of positive reward clusters and the overall positive space coverage ($r = -0.56, p < 10^{-10}$).

Goal Inference

To demonstrate the usefulness of these goal program representations, we offer a proof-of-concept goal inference method through reward-producing programs, visualized in Figure 3. We presently eschew the complexity of the full, open-ended goal-inference problem, and instead formulate the task as a N-alternative forced choice problem: given a set of goals and the demonstrations provided by a participant for one of them, can we predict which goal yielded these demonstrations?

Data. We excluded goals with fewer than six demonstrations (the minimum we asked participants to record) and those with zero or one non-zero scores. Otherwise, we do not filter for how successfully the participants demonstrated various or optimal outcomes in their goal. This yields a dataset of 354 of the original 394 goals, and their corresponding demonstrations, where each demonstration includes the power (1%-100%) and the angle ($0^\circ - 359^\circ$) of the shot, both discretized to integers.

Setup. We follow a Bayesian approach to this problem. We are given a set of N goals $\{g_1, g_2, \cdots, g_N\}, g_i \in \mathcal{G}$ and a set of M ball shot demonstrations consisting of a power and angle, $D = \{d_1, d_2, \cdots, d_M\}$. We wish to infer the most likely goal given these demonstrations, $\max_{g_i} P(g_i \mid D)$. By Bayes' theorem, $P(g_i \mid D) = \frac{P(D|g_i)p(g_i)}{P(D)} \propto P(D \mid g_i)$ (we currently do not set a prior over goals, and leave that for future work to consider). If we assume the demonstrations are conditionally independent given a goal, we can factor $P(D \mid g_i) = \prod_{j=1}^{M} P(d_j \mid g_i)$, that is, the probabilities of each action under the goal.

Policy. We compute a ground-truth goal-conditioned policy by enumerating over all shot strengths and angles. We evaluate the score under the goal (using our interpreter and simulation) for each possible shot to compute each goal's full action-value table (in reinforcement learning parlance, Q-values). Next, we normalize the Q-value table (computing action advantages). We then smooth the table by convolving it with a Gaussian kernel with hyperparameters k (the kernel size) and σ (the standard deviation). We do so to capture an intuition of having a sense that a shot was close, allowing credit sharing between an accurate shot and its nearby neighbors. Finally, we convert these smoothened values to a policy by applying a softmax with temperature T, our third and final hyperparameter.

Hyperparameter Optimization. We fit hyperparameters by optimizing the negative log-likelihoods $\log P(d_j \mid g_i)$ for each goal with its corresponding demonstrations. Our overall objective is the grand mean of negative log-likelihoods over demonstrations and goals, $\mathcal{L} = -\frac{1}{G}\sum_{i=1}^{G} \frac{1}{|D_i|} \sum_{j=1}^{M_i} \log P(d_j \mid g_i)$, where G is the total number of goals and D_i is the demonstration set for goal g_i. We perform Bayesian hyperparameter optimization using `optuna` (Akiba et al., 2019). We hold out 15% (53/354) goals to evaluate our hyperparameter optimization procedure, holding out from each level proportionally.

Inference. We seek $\max_{g_i} P(g_i \mid D) = \max_{g_i} \log P(g_i \mid D) = \sum_{j=1}^{M} \log P(d_j \mid g_i)$. We compute the right-hand quantity, the sum of action log probabilities, using the distributions induced with the fitter hyperparameters, and predict the goal with the largest sum of log-likelihoods over the demonstrations.

Results. Our hyperparameter optimization procedure yielded a Gaussian kernel of size $k = 23$ with $\sigma = 0.951$ and a softmax temperature $T = 1.243$. The mean negative log-

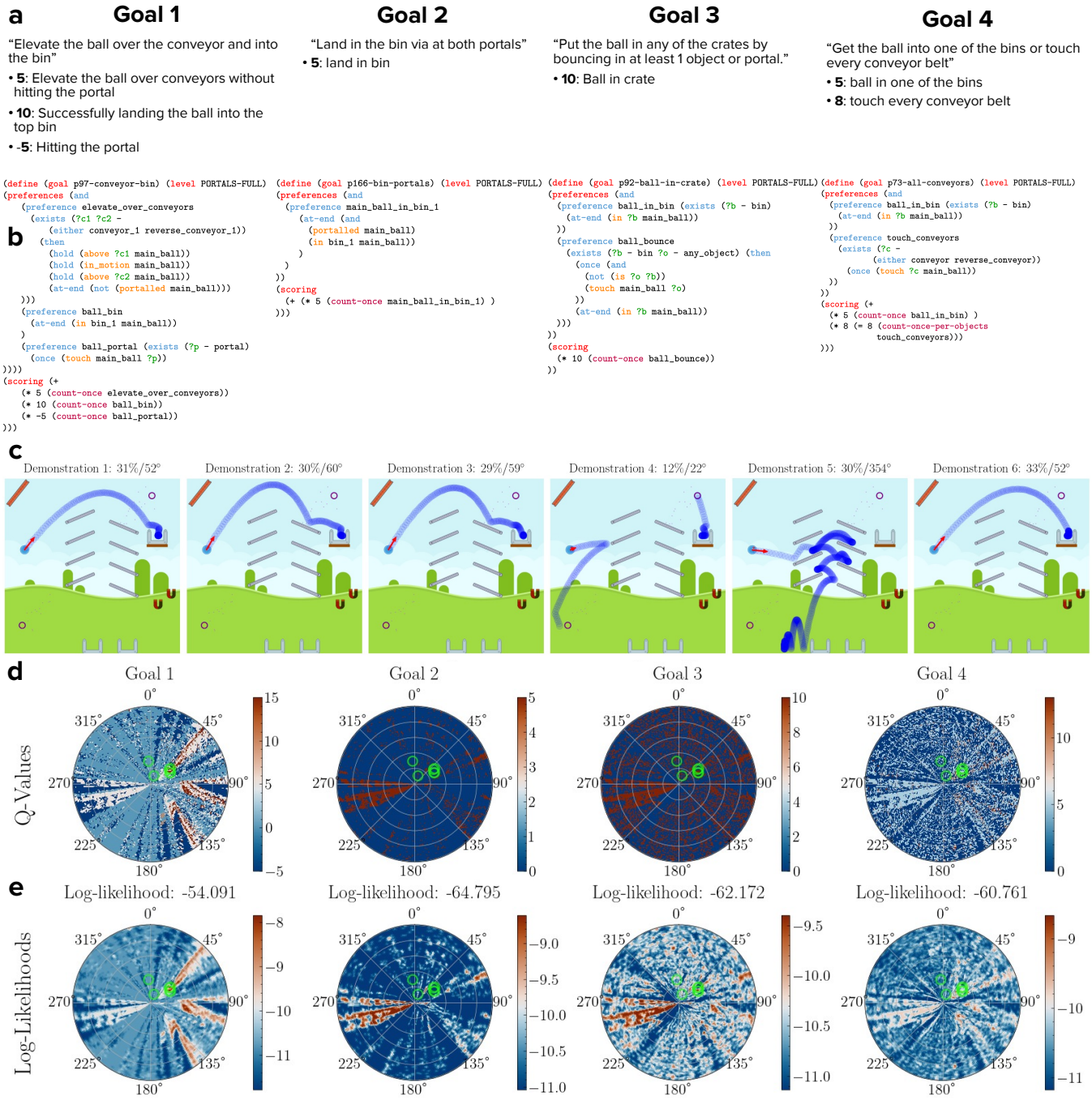


Figure 3: Goals and goal inference visualization. We set up our goal inference problem as N-alternative forced choice: given a set of goals (visualized for $N = 4$) and demonstrations provided by the participant who created one of them, which goal yielded these demonstrations? We visualize the goals and scoring elements as described by the participants in panel **a**, and the reward-producing programs they map into in panel **b**. In panel **c**, we visualize the trajectories of the six demonstrations provided by the participant (the blue circles represent ball positions, and the red arrow the direction and strength of each shot, also in the figure titles). In panel **d**, we plot, in polar coordinates, the Q-values (point outcomes) for each power and angle of shot. We enumerate all possible ball launches in increments of 1% of the maximal force and 1°. The green circles mark the locations of demonstrations visualized in panel **c**. Finally, in panel **e**, we plot the log probabilities of the distributions induced by our method for each goal after normalizing the Q-values, spatially smoothing, and applying a (log-) softmax. Goal 1, whose demonstrations we use, has the highest likelihood and, therefore, is (accurately) predicted as the goal in this example.

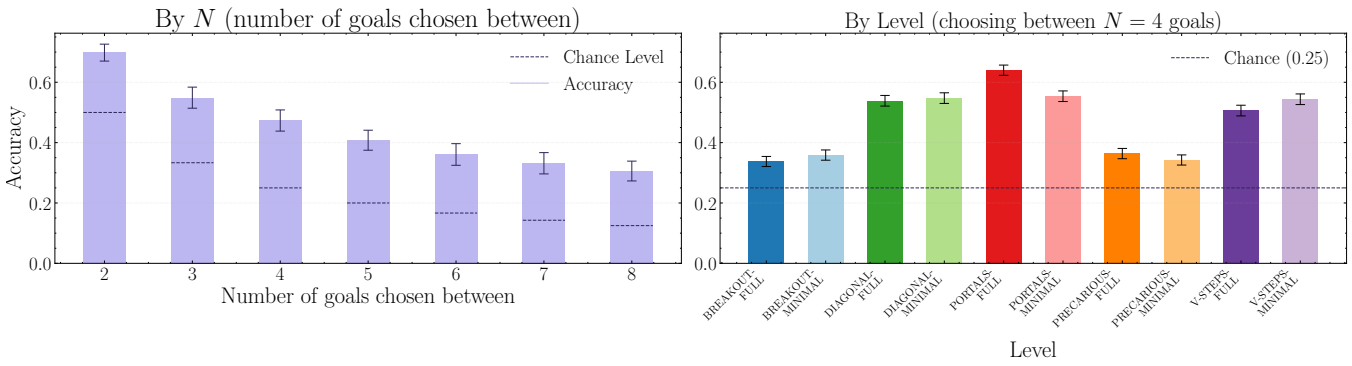


Figure 4: Goal inference through reward-producing programs succeeds substantially above chance. *Left:* We compare the accuracy of our goal inference procedure as we increase the number of goals to choose between. In the range evaluated, our approach remains substantially above chance accuracy. *Right:* Comparing between $N = 4$ goals, we compare the accuracy for each level and variation. While we observe meaningful deviations, all results are above chance accuracy. Error bars indicate the standard errors of the mean.

likelihood on our held out set ($\mathcal{L}_{hold} = 9.609$) is lower than on the hyperparameter-optimization set ($\mathcal{L}_{hyper} = 9.759$). We take this as evidence we avoided overfitting and proceed to analyze the entire dataset jointly. For each of the ten level variations, and each value of $N = \{2, 3, \cdots, 8\}$, we randomly sampled 200 sets of N goals created for that level. For each set, we evaluate the goal inference through reward-producing programs method with each goal's demonstrations to see if it predicts the correct goal, making for $200N$ total trials. Figure 4 summarizes our results. For every value of N tested, our method predicts the correct goal substantially above chance accuracy (Figure 4, left). We see meaningful variation in accuracy between the different levels, albeit with all levels remaining above chance accuracy. We hypothesize this arises from the variation in goals created in each given level, an analysis we aim to pursue in future work. Visualizing the method's steps in Figure 3, we observe that the raw Q-values (top row) are rather noisy; adjacent values often differ drastically. This motivated the intermediate smoothing step before applying the softmax to arrive at the policies depicted in the bottom row.

Discussion

We demonstrate a proof of concept for a novel goal inference algorithm leveraging structured program representations to perform inverse reinforcement learning. Our method implements a simple approach to using the rich information provided by the reward-producing programs, and even this succeeds above chance level in every evaluation we devised; on the other hand, it is far from perfect and admits opportunities to improve it in future work. To study goal inference, we collected a rich dataset of almost 400 user-generated goals and demonstrations in a novel physics environment. Our dataset highlights insights into goal creation: participants generated contextually appropriate goals that leveraged available interactions, with varying complexity and reward structures, demonstrating systematic relationships between goal properties and perceived difficulty.

A key advantage of our experimental design is the ability to enumerate the entire action space for each goal. This enables us to compute detailed reward landscape features that are intractable in more complex environments. Future work could build on this by integrating our goal analysis with cognitive models of intuitive physics (Allen et al., 2020), allowing "simulation-in-the-loop" approaches to goal generation and understanding. This could help bridge the gap between symbolic goal representations and grounded physical understanding.

The success of our proof-of-concept goal inference system suggests promising directions for future development. Though our participants are far from perfect, the model currently assumes demonstrations are meant to be optimal; people might offer imprecise or pedagogical demonstrations, intentionally showing a range of outcomes. Future work could address these by incorporating models of bounded rationality and pedagogical reasoning. We currently tackle a limited, N-alternative forced-choice goal inference problem. Future work should explore open-ended goal program synthesis from demonstrations, using learned priors from our dataset and flexible generative models (Davidson et al., 2025; Todd et al., 2024).

Our formal representations of goals as reward-producing programs have potential applications beyond cognitive modeling. They could inform the development of autotelic AI systems capable of open-ended learning through self-generated goals (Colas et al., 2022). From a developmental perspective, our work raises intriguing questions about how humans acquire and refine their "language of goals." How do children learn to compose basic goal primitives into more complex objectives? How does this capability interact with developing physical understanding and causal reasoning? Our experimental paradigm and formal framework could provide tools for investigating these questions empirically. Furthermore, our choice to focus on objectively scoreable goals raises is experimentally useful but may limit the applicability of these approaches. Naturally occurring human goals often involve subjective or fuzzy success criteria. Future work could extend these formal goal representations to capture more open-ended objectives while striving to maintain analytical tractability.

These directions highlight the broader potential of studying goal creation as a window into human cognition and artificial intelligence. By formalizing and analyzing this fundamental capability, we can better understand both human learning and potential paths toward more capable AI systems.

Acknowledgments

G.D. thanks members of the Human and Machine Learning Lab and the Computation and Cognition Lab at NYU for their feedback at various stages of this project. The authors thank Gabe Grand for helpful discussions at early iterations of this work. G.D.is supported by a Meta AI Mentorship fellowship. G.D. and B.M.L. are supported by the National Science Foundation (NSF) under NSF Award 1922658 NRT-HDR: FUTURE Foundations, Translation, and Responsibility for Data Science. G.T.'s work on this project is supported by the National Science Foundation Graduate Research Fellowship under Grant DGE-2234660. T.M.G.'s work on this project is supported by NSF BCS 2121102.

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). "Optuna: A Next-generation Hyperparameter Optimization Framework". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (p. 4).

Alanqary, A., Lin, G. Z., Le, J., Zhi-Xuan, T., Mansinghka, V. K., and Tenenbaum, J. B. (2021). "Modeling the mistakes of boundedly rational agents within a Bayesian Theory of mind". *arXiv [cs.AI]* (p. 2).

Allen, K. R., Smith, K. A., and Tenenbaum, J. B. (2020). "Rapid trial-and-error learning with simulation supports flexible tool use and physical reasoning". *Proceedings of the National Academy of Sciences* 117.47, pp. 29302–29310 (p. 6).

– (2019). "The Tools Challenge: Rapid Trial-and-Error Learning in Physical Problem Solving" (p. 2).

Austin, J. T. and Vancouver, J. B. (1996). "Goal constructs in psychology: Structure, process, and content." *Psychological Bulletin* 120 (3), pp. 338–375 (p. 1).

Baker, C., Saxe, R., and Tenenbaum, J. (2011). "Bayesian theory of mind: Modeling joint belief-desire attribution". In: *Proceedings of the annual meeting of the cognitive science society*. Vol. 33 (pp. 1, 2).

Baker, C. L., Jara-Ettinger, J., Saxe, R., and Tenenbaum, J. B. (2017). "Rational quantitative attribution of beliefs, desires and percepts in human mentalizing". *Nature Human Behaviour* 1 (4), p. 64 (p. 2).

Baker, C. L., Saxe, R., and Tenenbaum, J. B. (2009). "Action understanding as inverse planning". *Cognition* 113.3. Reinforcement learning and higher cognition, pp. 329–349 (p. 1).

Chu, J. and Schulz, L. (2020). "Exploratory play, rational action, and efficient search" (p. 1).

Chu, J., Tenenbaum, J. B., and Schulz, L. E. (2024). "In praise of folly: flexible goals and human cognition". *Trends Cogn. Sci.* (p. 1).

Colas, C., Karch, T., Sigaud, O., and Oudeyer, P.-Y. (2022). "Autotelic Agents with Intrinsically Motivated Goal-Conditioned Reinforcement Learning: a Short Survey". *Journal of Artificial Intelligence Research* 74, pp. 1159–1199 (p. 6).

Czikszentmihalyi, M. (1990). *Flow: The psychology of optimal experience*. New York: Harper & Row (p. 1).

Davidson, G., Gureckis, T. M., and Lake, B. M. (2022). "Creativity, Compositionality, and Common Sense in Human Goal Generation". In: *Proceedings of the 44th Annual Meeting of the Cognitive Science Society, CogSci 2022* (p. 3).

Davidson, G., Todd, G., Togelius, J., Gureckis, T. M., and Lake, B. M. (2025). "Goals as Reward-Producing Programs". *Nature Machine Intelligence* (pp. 1–3, 6).

Dweck, C. S. (1992). "Article Commentary: The Study of Goals in Psychology". *Psychological Science* 3.3, pp. 165–167 (p. 1).

Fishbach, A. and Ferguson, M. J. (2007). "The goal construct in social psychology". In: *Social psychology: Handbook of basic principles*. Ed. by A. W. Kruglanski and E. T. Higgins. Vol. 2. New York, NY, US: The Guilford Press, xiii, pp. 490–515 (p. 1).

Fodor, J. A. (1979). *The language of thought*. Harvard University Press, p. 214 (p. 1).

Ghallab, M., Howe, A., Knoblock, C., et al. (1998). *PDDL – The Planning Domain Definition Language* (p. 3).

Goodman, N. D., Tenenbaum, J. B., Feldman, J., and Griffiths, T. L. (2008). "A rational analysis of rule-based concept learning". *Cogn. Sci.* 32.1, pp. 108–154 (p. 1).

Gureckis, T. M., Liquin, E., Davidson, G., Intara, P., and LeGris, S. (2025). *Smile—A happy approach to online research.* Package under active development (p. 2).

Kenney (2025). *Kenney.NL* (p. 2).

LeGris, S., Lake, B., and Gureckis, T. (2024). "Predicting insight during physical problem solving". In: *Proceedings of the 46th Annual Conference of the Cognitive Science Society*. Cognitive Science Society. Austin, TX: Cognitive Science Society (p. 2).

Liam Brummitt (2014). *Matter.js* (p. 2).

Manna, Z. and Pnueli, A. (1992). *The Temporal Logic of Reactive and Concurrent Systems*. Springer New York (p. 3).

Palan, S. and Schitter, C. (2018). "Prolific.ac—A subject pool for online experiments". *Journal of Behavioral and Experimental Finance* 17, pp. 22–27 (p. 2).

Phaser Developers (n.d.). *Phaser* (p. 2).

Piantadosi, S. T., Tenenbaum, J. B., and Goodman, N. D. (2012). "Bootstrapping in a language of thought: a formal model of numerical concept learning". *Cognition* 123.2, pp. 199–217 (p. 1).

Rule, J. S., Tenenbaum, J. B., and Piantadosi, S. T. (2020). "The Child as Hacker". *Trends in Cognitive Sciences* 24.11, pp. 900–915 (p. 1).

Ryan, R. M. and Deci, E. L. (2000). "Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being." *American psychologist* 55.1, p. 68 (p. 1).

Todd, G., Padula, A. G., Stephenson, M., Piette, E., Soemers, D. J., and Togelius, J. (2024). "GAVEL: Generating Games

via Evolution and Language Models". In: *The Thirty-eighth Annual Conference on Neural Information Processing Systems* (p. 6).

Tomasello, M. (2022). *The evolution of agency: Behavioral organization from lizards to humans*. MIT Press (p. 1).

Ullman, T., Baker, C., Macindoe, O., Evans, O., Goodman, N., and Tenenbaum, J. (2009). "Help or Hinder: Bayesian Models of Social Goal Inference". In: *Advances in Neural Information Processing Systems*. Ed. by Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta. Vol. 22. Curran Associates, Inc. (p. 1).

Velez-Ginorio, J., Siegel, M. H., Tenenbaum, J. B., and Jara-Ettinger, J. (2017). "Interpreting actions by attributing compositional desires". In: *Proceedings of the 39th Annual Meeting of the Cognitive Science Society, CogSci 2017, London, UK, 16-29 July 2017*. Ed. by G. Gunzelmann, A. Howes, T. Tenbrink, and E. J. Davelaar. cognitivesciencesociety.org (p. 2).

Wong, L., Grand, G., Lew, A. K., Goodman, N. D., Mansinghka, V. K., Andreas, J., and Tenenbaum, J. B. (2023). "From Word Models to World Models: Translating from Natural Language to the Probabilistic Language of Thought". arXiv (pp. 1, 4).

Zhi-Xuan, T., Kang, G., Mansinghka, V., and Tenenbaum, J. B. (2024). "Infinite Ends from Finite Samples: Open-Ended Goal Inference as Top-Down Bayesian Filtering of Bottom-Up Proposals". *Proceedings of the Annual Meeting of the Cognitive Science Society* 46.46 (p. 2).

Zhi-Xuan, T., Mann, J. L., Silver, T., Tenenbaum, J. B., and Mansinghka, V. K. (2020). "Online Bayesian goal inference for boundedly-rational planning agents". *arXiv [cs.AI]* (p. 2).