A Causal Model Approach to Dynamic Control

Zachary J. Davis, Neil R. Bramley, Bob Rehder, Todd Gureckis
[zach.davis, neil.bramley, bob.rehder, todd.gureckis] @nyu.edu
Department of Psychology, New York University
6 Washington Place, New York, NY, USA 10003 USA

Abstract

Acting effectively in the world requires learning and controlling dynamic systems, that is, systems involving feedback relations among continuous variables that vary in real time. We introduce a novel class of dynamic control environments using the continuous Ornstein-Uhlenbeck process united with causal Markov graphs that allow us to systematically test people's ability to learn and control various dynamic systems as they change in real time. We find that people's control is robust to changing goals, and exhibits heterogeneity of performance in different environments that matches closely with complexity defined by our optimal model. These results suggest people are capable learners of dynamic systems, able to leverage a rich representation of their environment to accomplish their goals.

Keywords: dynamic control, causal learning, dynamic decision making, reinforcement learning

Introduction

Humans are masters of navigating and manipulating complex dynamic systems. For instance, anyone who has been involved in a particularly engaging conversation, or one that escalates into a fight based on a shift in tone of voice, knows that these systems are highly sensitive and involve complex feedback loops and dynamics (Berry & Broadbent, 1984). Our aim in this paper is to study how people accomplish continuous goal-directed control in complex dynamic natural environments.

Previous approaches to studying human control have been been limited to systems with little formal analysis of the underlying structure (e.g., Berry & Broadbent, 1984; Gureckis & Love, 2009a; Hagmayer et al., 2010). We build on this literature by introducing a class of dynamic systems composed of multiple components related by an underlying causal Markov structure. The causal Markov structure enable us to systematically vary the environment that subjects interact with allows us to understand the formal properties of control problems, including what determines their difficulty. Our approach thus fills a gap in the literature on complex dynamic control by explicitly incorporating notions of causal structure (cf, Osman, 2010).

We begin by briefly reviewing the literature on dynamic control tasks and highlight where we depart from previous studies. We then provide a formal description of the "language" we use to describe various system dynamics. We next report a novel experiment which investigates the extent to which people learn and exploit knowledge of the causal structure of a system to maximize reward. To foreshadow, we find that people generally perform like an an optimal agent in that they achieve a high reward rate and robustly adapt their control decisions in light of changing goals. Importantly, they also struggle on those minority of control problems that our efficient, model based agent performs poorly on.

Past research

Complex dynamic control (CDC) tasks have come under a variety of names. We follow Osman (2010) in grouping, under the umbrella of CDC, tasks described as complex problem-solving tasks, dynamic decision-making tasks, and process control tasks.

Static control. A classic control study is Berry and Broadbent's (1984) sugar factory task, which has spurred a wide variety of computational models (for discussion, see Busemeyer, 2002). Participants' task was to maintain a level of production of a sugar factory by controlling the number of workers employed. Sugar production at time t (P_t) was a function of not only the number of workers (W_t) but also production at the previous time step: $P_t = 2 \times W_t - P_{t-1}$. Berry and Broadbent found that control performance and explicit task knowledge were independent. Participants could learn to control the system but not verbalize how it works, and telling participants how the system works did not improve control performance. Note that Berry and Broadbent were unable to determine whether participants' implicit knowledge of the system involved memorized state-action pairs or a non-verbalizable representation of the system's update function.

Hagmayer et al. (2010) assessed whether people learn causal structure while controlling a dynamic system. The key manipulation was testing whether people were sensitive to an intervention on a node that mediated the relationship between the control and target variables. They found that participants correctly adjusted their behavior to account for the disabled node, suggesting that they had learned the causal model and were anticipating the downstream effects of this intervention.

We expand on these previous studies in several ways. First, we present variables that change value, and can be controlled, in (perceptually) continuous rather than discrete time. Rewards also accumulate in real time depending on the values of the variables. Second, we consider feedback relations involving not only the outcome variable and itself, but between any set of variables in the system (leading to networks that exhibit complex behavior, including oscillations). Crucially, our paradigm allows us to use control behavior to infer the underlying system knowledge, namely to what extent control behavior reflects a deep causal model or shallow memorized state-action pairs.

Real-time control. There have been a few investigations into how people manage CDC tasks in real-time, however few

of these approaches investigate how people *simultaneously* learn and control a dynamic system. For example, there have been studies into how people compete in a bidding scenario (Brandouy, 2001) or collaborate in real time (Sarter, Mumaw, & Wickens, 2007), but these also presuppose that the participants already possess knowledge of the underlying dynamics. Our investigation is novel in that we investigate the extent to which people can learn, in real time, the dynamics of a system while controlling it. This puts our task closer, in some respects, to studies on reinforcement learning in dynamic environments where people make discrete actions to maximize reward in an environment that changes in response to the history of past actions (e.g., Gureckis & Love, 2009a, b). We build off of these studies by having an environment with explicit causal structure, allowing us to draw insights about cognition from the extensive causal learning literature.

Forward models. As in many real world environments, actions taken in the Ornstein-Uhlenbeck environment we define below have uncertain, time-delayed impact on the desired outcome. Learning the mapping from actions to desired outcomes is known as a "distal supervised learning task" (Jordan & Rumelhart, 1992). A crucial component of controlling proximal actions to gain distal rewards is to develop a model of one's environment, also known as a forward model. The Causal Model Based Controller outlined below is a forward model for the dynamic system presented to participants, and it will be shown that having this mapping from actions to future states in the environment allows for generalization to new goals.

Ornstein-Uhlenbeck Process

An Ornstein-Uhlenbeck (OU) process is a stationary Gauss-Markov process in continuous time that asymptotically converges to a stable mean (Uhlenbeck & Ornstein, 1930). It has been used to model phenomena in physics (Lacko, 2012) and finance (Barndorff-Nielsen & Shepard, 2001), and has also been studied in perception (Vul et al., 2009). Because these processes are able to capture dynamic natural phenomena across a wide variety of domains, we believe that the OU process is a reasonable formalism for modeling causal relationships between continuous variables in continuous time.

Generative model. The instantaneous change of some variable x is defined as follows:

$$dx_t = \theta\Big(\sum_{i=1}^{n} \beta_{iX} y_{i,t} - x_t\Big)dt + \sigma dW_t \qquad (1)$$

where x_t is the value of variable x at time t, θ is a parameter greater than 0 that determines how sharply the process reverts to the mean, σ is the variance, and W_t is a Wiener process.

The mean that x reverts to is determined by its parents in the causal structure. The combination of causal strengths (β_{iX}) and values of parent nodes at the current time point ($y_{i,t}$) determines the value that the x trends to. In asymptote the process simply reverts to a sum, for each parent y_i, of the causal strength between variable y_i and x multiplied by the value of the parent.

Experiment: Control Task

Method

Participants. 36 participants (20 female, mean age=33) were recruited from Amazon Mechanical Turk using the psiTurk framework (Gureckis et al., 2016), which has been shown to produce comparable results to lab experiments in cognitive science (Crump, McDonnell, & Gureckis, 2013). They were paid $3.50 for approximately 25 minutes, with additional bonus based on performance (M=$0.52). Of the 36 participants we gathered, 6 were excluded because they did not use the arrow keys on more than 25% of the phases.

Materials and procedure. See Figure 1 for illustration of a trial. Each of the three variables was represented by a vertical slider constrained to be between -100 and 100. The handles of each slider presented a rounded integer value. One slider, the "control" slider, could be intervened on with three keys 'o', 'k', and 'm'. As is intuitive on a QWERTY keyboard, the 'o' key increased the control slider (by 10), the 'k' key held the value steady, and the 'm' key decreased the control slider by 10. If the participant did not press a key, the control slider would move according to the dynamics of the OU system.

The other two sliders could not be directly controlled by participants. One of these sliders, the "target" slider, had 20% of its area colored green to indicate the reward region. For each time step (100ms) that the target slider was in the green region, $0.01 was added to their score (displayed at the bottom of the screen). The top of the screen presented a timer counting down from 20 seconds, at which point the phase finished.

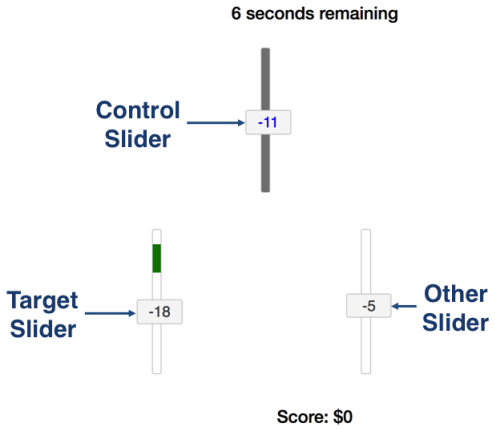

Figure 1: Example of the interface used by participants. Participants moved the control slider while observing the effects on the target/other slider. The goal was to exert forces on the control slider to keep the target slider within the green region.

In the instructions phase, participants were shown four videos of an agent interacting with the structures to familiar-

ize them with the interface (participants were told the structure that the agent was interacting with). They were shown examples of (1) a network with no connections, (2) one with a single, direct positive relationship between controller and target, (3) a single negative causal relationship, and (4) an indirect connection through the auxiliary slider (with both links being positive). Participants were then presented with a four question comprehension check to ensure that they understood the task. Questions established that participants understood that the connections, but not reward regions, stayed the same between Phase 1 and Phase 2, that the 'o' key moved the control slider up and the 'm' key moved it down, and that the reward would be a randomly selected phase of the experiment. Participants could not continue without answering all questions correctly. The parameters used during training and the learning task were $\theta = .1$, $\sigma = 5$, and βs were either -1, 0, or 1.

In the learning task, participants initiated the trial by pressing the "Start" button and the sliders started jittering according to an OU process, with unknown β weights driving the movement (there were no causes outside the network). The values of the sliders updated every 100ms, and each phase lasted for 20 seconds. At any time, participants were free to manipulate the control slider. Scores were presented in real time during the trial, counting up by \$0.01 per 100ms that the target slider was in the reward region. After the first 20 second phase for a structure, participants received a pop-up inviting them to begin the second phase, and were reminded that the connections will not change, but the reward region will. After the second phase, the participants moved onto the next device, repeating this process for each of the 12 structures[1]. After seeing all structures, participants completed a brief post-test questionnaire.

Results

A paired-samples t-test revealed that participants were rewarded at a slightly higher rate during phase 2 of the study than phase 1 (M=4.25, SD=40.52); t(359)=-1.99, p=.047. Figure 2 shows results for each tested structure, revealing heterogeneity in the effect of experience on expected reward. This heterogeneity is at least in part due to the inherent complexity of different structures (see Modeling section below).

Models

Causal Model Based Controller The goal of the Causal Model Based Controller (CMBC) agent is to use its best estimate of the causal structure of the environment to act flexibly to maximize reward[2]. The CMBC agent, then, must estimate the probability of there being causal connections between sliders. For the current environment, causal strengths are defined as β weights between sliders (see the generative model section). Given some movement of slider x, and the values of other sliders y_i, the log-likelihood of causal strength β is:

$$ln(p(\beta_{iX}|dx_t, y_{i,t})) \propto -(dx_t - \theta\Big(\sum_{i=1}^{n} \beta_{iX} y_{i,t} - x_t\Big)dt)^2 \quad (2)$$

For each observation, we jointly estimate the full space of beta values for possible edges. For example, for three variables there are six possible edges, $\boldsymbol{\beta} = \{\beta_{XY}, \beta_{XZ}, \beta_{YX}, \beta_{YZ}, \beta_{ZX}, \beta_{ZY}\}$. Multiplying by the (initially uniform) prior probability of each hypothesis and normalizing yields the posterior over hypotheses.

The CMBC uses its online estimate of the causal structure of the environment to act. In particular, it imagines taking each of the four possible choices available to it[3]. A given choice at time t will impact the controlled variable's state at time $t+1$. The CMBC then projects forward the effects that this choice would have over time. For this study, we project forward the impact of a choice for three time steps from the time of the decision. Because the process is stochastic, the impact of a choice will yield a probability distribution over possible states of the target variable. For each possible causal structure, it takes the integral of the expected distribution within the target range

$$EV(choice|structure) = \int_{range\ min}^{range\ max} N(\mu_T, \sigma)dx \quad (3)$$

where μ_T is the mean of the target variable's distribution. The CMBC then weights the expected value of some choice given a structure by the probability of that structure:

$$EV(choice, struct) = EV(choice|struct) * p(struct) \quad (4)$$

Marginalizing over structures gives the $EV(choice)$. The CMBC agent chooses the action that maximizes expected value.

Deep Reinforcement Learning To compare to the CMBC, we considered a model-free reinforcement learning agent based on a deep-Q learning network (DQN). This model represents the state of the art for sequential decision making in complex environments similar to those studied here. Recently DQNs have been used to push the limits of what reinforcement learning algorithms can accomplish (e.g., learning to play Atari at near human levels, Mnih et al., 2015).

This model is interesting to compare here for a number of reasons. First, a DQN is explicitly non-causal in that it has no direct represention of the environment and is unable to counter-factually plan future states and actions. At the same time, such models are powerful tools for dynamic control because they can learn forward-looking policies by approximating the solution to Bellman's equation. Related approaches

[1] see Appendix for all presented causal structures

[2] It is important to note that the learning is all passive, reducing uncertainty is not factored into the choice the CMBC agent makes

[3] (1) increase controlled variable's state by 10 (2) hold it constant (3) decrease its state by 10 (4) do nothing and allow it to vary according to the OU process

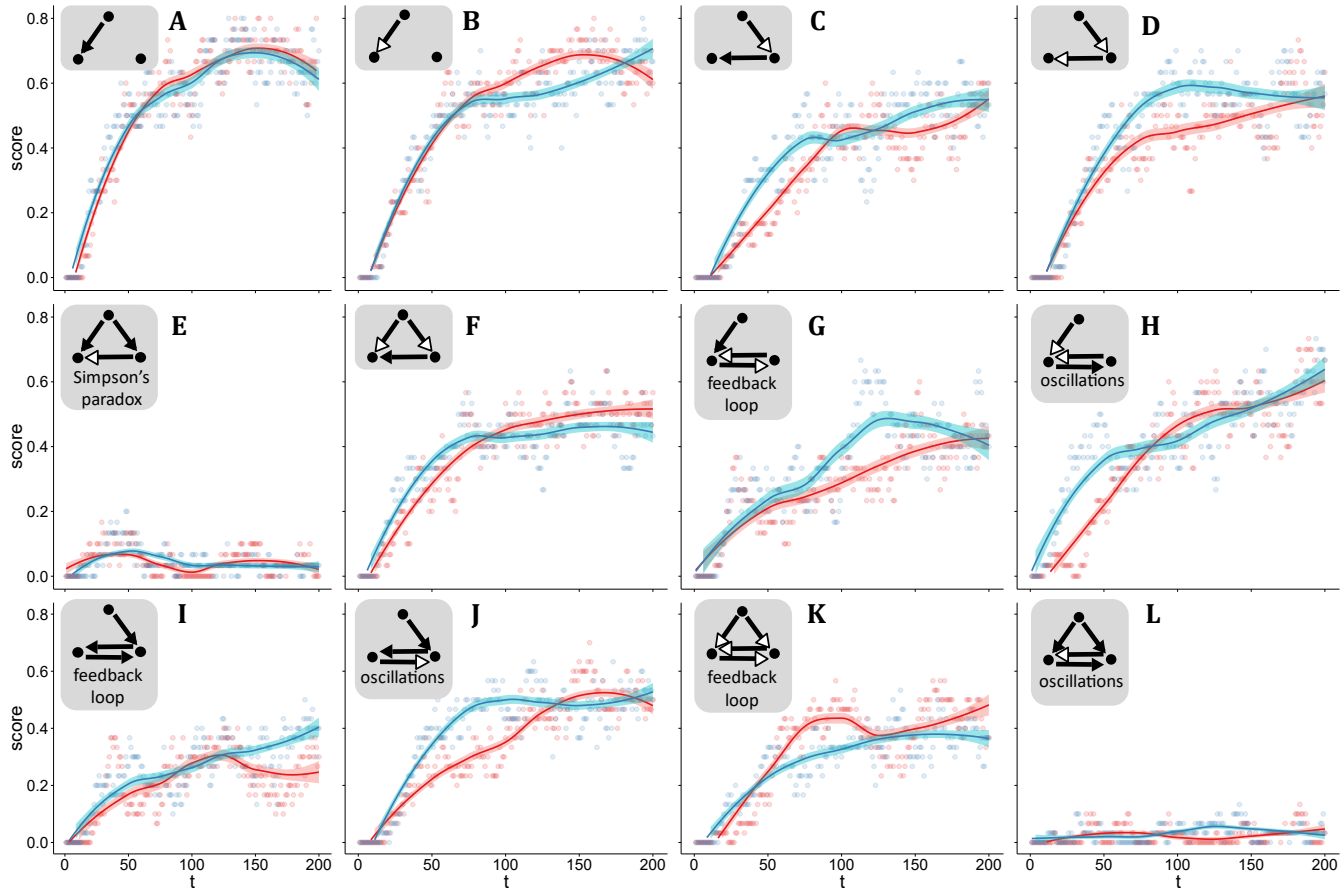

Figure 2: Proportion of possible reward received per-time step over the course of a trial. Red lines indicate phase 1, blue lines phase 2. The graphs in the corner of each plot label the causal structure that determined the dynamics of the environment with the node at the top of the triangle mapped to the control slider and the node on the left mapped to the target slider. Solid arrowheads denote regular connections ($\beta = 1$), white arrowheads denote inverse connections ($\beta = -1$).

have been used to successfully model human performance in discrete-time control and learning problems (e.g., Gureckis & Love, 2009). Furthermore, the flexible mapping between states and actions possible in DQNs may admit certain types of generalization to new reward regimes. Together these factors make such models interesting comparison cases against the CMBC approach.

To evaluate the ability of these networks to learning in the OU environment, we constructed a neural network in pyTorch (Paszke et al., 2017) with three layers. The input layer was made up of 6 inputs. The first three coded (on a scale of -100 to 100) the current location of each of the three sliders. The next two input units coded the upper and lower bound of the current reward region on the target slider (information which was concurrently visible to human participants). The final input was a continuous input that coded the difference between the current position of the target slider and the mid-point of the target region. A fully connected hidden layer with 256 (rectified linear) units was in turn fully connected to an output layer with 4 linear units representing the estimated Q-values of moving the control slider up, down, hold it steady, or do nothing. The target objective for training was the standard "on policy" Q-learning algorithm that learns how an action might effect future states of the system as well as the value of each action at the current time (Watkins & Dayan, 1989). For the first 1000 trials of learning the the model choose actions based on a linearly decaying epsilon-greedy choice rule, there after using softmax (Sutton & Barto, 1998). For each time step the target slider was maintained in the target reward region the network earned 10 units of reward. Furthermore, to punish extreme deviations from the target the reward was the negative of the absolute value of the distance between the target slider and the center of the reward region anytime the slider moved outside the target window. Learning was accomplished via gradient descent on the *smooth_l1_loss* of the difference between the prediction and actual Q-values for each action. To speed learning, the network maintained a buffer of past state-action-reward-next state transitions and randomly sampled 64 of these each trial for use in a single batch gradient update. The majority of the network features

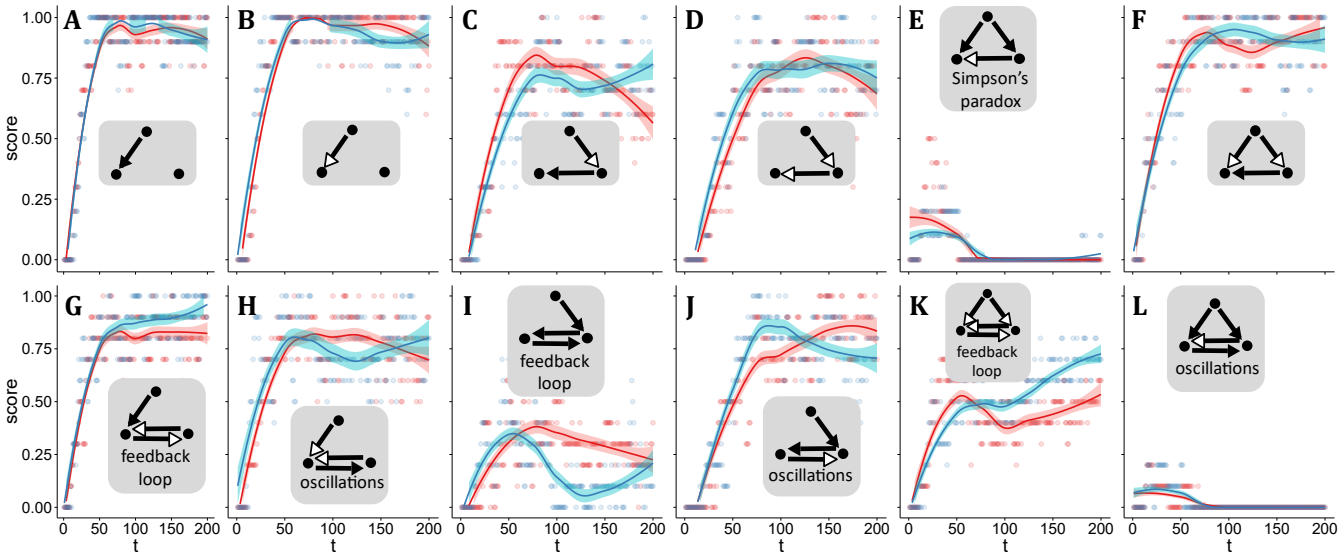

Figure 3: Proportion of possible reward received over time for the CMBC agent. Red lines indicate phase 1, blue lines phase 2. Solid arrowheads denote regular connections ($\beta = 1$), white arrowheads denote inverse connections ($\beta = -1$).

were determined by the Deep Q Learning tutorial provided in the pyTorch documentation. Specific network parameters were set as follows: discount rate ($\gamma = 0.98$), learning rate ($\eta = 0.001$), and softmax temperature ($\tau = 8$).

Even with these powerful learning features, the DQN is at a disadvantage in learning the task because it comes with randomly intialized weights and can only learn the objective of the task by experiencing certain state-action transitions paired with reward. As a result it cannot learn to perform the task in real-time (e.g., using the same number of time steps as human participants). However, it can still provide insight into the relative difficulty of learning each environment. To evaluate this, we ran the DQN 50 times on each structure. The model was allowed to repeatedly interact with the phase 1 environment 20 times and we recorded the total number of time steps that the target slider was in the target window on each episode. After this training, the network was trained on the altered reward window (phase 2) for 20 games, again recording the number of time steps per episode earning reward.

Model Results.

To compare participant judgments to model predictions, we had the models perform the task. As would be expected given its ability to react and replan immediately on each frame, the CMBC achieved much sharper learning curves than the DQN agent, reaching and maintaining the target variable within the reward region in the first phase of the first trial for most environments. The DQN required more experience, generally reaching its maximum performance only after 10 playthroughs of an environment. To test the extent to which the models and participants agreed on the relative difficulty of environments, we measured the average reward for the last 20% of time-points (for participants and CMBC) or playthroughs (DQN). This gives a measure of the asymptotic performance for each agent. Participant judgments correlated with (parameter-free) predictions from both the CMBC (r=.91, p<.001) and the DQN agent (r=.87, p<.001).

Figure 3 shows average reward curves for the CMBC in each of the environments given to participants. As can be seen, they largely agree on the relative difficulty of problems. For example, reward curves in Cells A and B (direct links) have higher plateaus than Cells C and D (indirect links). This is because noise propagates through causal links. In an environment with a direct control-target relationship, a control action has a direct influence on the probability distribution of the target. For an indirect control-target relationship, a control action influences the probability distribution of an intermediate variable, which then further spreads the distribution of the target.

Of course, the most dramatically different environments are cells E and L, termed "Simpson's paradox" environments. In these environments, holding the control variable at any point trends the target toward 0, because the control variable exhibits a direct influence on the target, but also an indirect (and hence time-lagged) influence of the opposite sign. The mean that the control variable trends to, then, is the function $Control_t - Control_{t-1}$. Learning this function, and planning far enough ahead to exploit a strategy that maximizes reward, would be an interesting problem in hierarchical planning that we do not investigate here.

Discussion

In this paper, we presented a new class of environments that can be systematically varied in order to test people's ability to learn and control dynamic systems. We found that people are well matched by an optimal model, not only in correla-

tion between plateaus in reward, but also in their robustness to changing goals and environment-dependent reward curves. We expect that this new class of environments will be useful to the field as experimental environment and test bed, as well as drawing links between the formal analyses in the causal literature and the sophisticated but black-box style learning of contemporary control tasks.

A key limitation of the current work is the relatively simple model comparison. The sample inefficiency of the DQN relative to both participants and the CMBC complicated using more sophisticated analyses. For example, a more sensitive analysis would involve feeding participants' past state and reward observations into the models and softmaxing over each model's preference for control actions to get a prediction for participant performance. This analysis is easily done for the CMBC, but the DQN is so much more sample-inefficient than people that we opted to analyze plateaus instead.

The incorporation of causality in our dynamical system allows for a diverse range of future experiments to further test people's flexibility in control. Analogously to changing the reward region of a single variable, future experiments could test people's sensitivity to switching reward variables, counterfactuals, multiple target or control variables, etc. These studies would allow for a deeper investigation into whether people are actually learning the causal structure of the environment, or are doing something more model-free (akin to the DQN agent). In a slightly different vein, the system described in this paper could be used to study a type of real-time 'systems programming', where dynamical systems are learned individually and then linked up into a larger structure.

Problem-solving, here operationalized as the ability to manipulate one's environment in service of some goal, is fundamental to higher-level cognition (Newell & Simon, 1972). Problems that we have to solve in our everyday lives do not often come as pre-packaged word problems or with clearly delimited trials. Rather, we often get noisy feedback from systems with unknown structure that change as we attempt to control them. Impressively, people are able learn how these systems work, and can leverage this knowledge to be robust and flexible in controlling complex systems.

References

Barndorff-Nielsen, O. E., & Shephard, N. (2001). Non-Gaussian Ornstein-Uhlenbeck-based models and some of their uses in financial economics. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2), 167-241.

Berry, D. C., & Broadbent, D. E. (1984). On the relationship between task performance and associated verbalizable knowledge. *The Quarterly Journal of Experimental Psychology*, *36*(2), 209-231.

Brandouy, O. (2001). Laboratory incentive structure and control-test design in an experimental asset market. *Journal of Economic Psychology*, *1*, 126.

Crump, M. J., McDonnell, J. V., & Gureckis, T. M. (2013). Evaluating Amazon's Mechanical Turk as a tool for experimental behavioral research. *PloS one*, 8(3), e57410.

Gureckis, T.M. & Love, B.C. (2009a) Learning in Noise: Dynamic Decision-Making in a Variable Environment. *Journal of Mathematical Psychology*, 53, 180-193.

Gureckis, T.M. & Love, B.C. (2009b) Short Term Gains, Long Term Pains: How Cues About State Aid Learning in Dynamic Environments. *Cognition*, 113, 293-313.

Hagmayer, Y., Meder, B., Osman, M., Mangold, S., & Lagnado, D. (2010). Spontaneous causal learning while controlling a dynamic system. *The Open Psychology Journal*, *3*, 145-162.

Jordan, M. I., & Rumelhart, D. E. (1992). Forward models: Supervised learning with a distal teacher. *Cognitive science*, *16*(3), 307-354.

Lacko, V. (2012). Planning of experiments for a nonautonomous Ornstein-Uhlenbeck process. *Tatra Mountains Mathematical Publications*, 51(1), 101-113.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., & Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40.

Levinson, S. C. (2016). Turn-taking in human communicationorigins and implications for language processing. *Trends in cognitive sciences*, 20(1), 6-14.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Petersen, S. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529.

Newell, A., & Simon, H. A. (1972). *Human problem solving* (Vol. 104, No. 9). Englewood Cliffs, NJ: Prentice-Hall.

Osman, M. (2010). Controlling uncertainty: a review of human behavior in complex dynamic environments. *Psychological bulletin*, 136(1), 65.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., ... & Lerer, A. (2017). Automatic differentiation in PyTorch.

Pearl, J. (2017). Theoretical Impediments to Machine Learning.

Sarter, N. B., Mumaw, R. J., & Wickens, C. D. (2007). Pilots' monitoring strategies and performance on automated flight decks: An empirical study combining behavioral and eye-tracking data. *Human Factors*, *49*, 347357.

Sterman, J. D. (1989). Misperceptions of feedback in dynamic decision making. *Organizational Behavior and Human Decision Processes*, *43*, 301335.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction* (Vol. 1, No. 1). Cambridge: MIT press.

Uhlenbeck, G. E., & Ornstein, L. S. (1930). On the theory of the Brownian motion. *Physical review*, *36*(5), 823.

Vul, E., Alvarez, G., Tenenbaum, J. B., & Black, M. J. (2009). Explaining human multiple object tracking as resource-constrained approximate inference in a dynamic probabilistic model. In *Advances in neural information processing systems* (pp. 1955-1963).