How Predictive Minds Explain and Control Dynamical Systems

Roman Tikhonov
Department of Social & Decision Sciences, Carnegie Mellon University
Pittsburgh, PA 15213, USA
rtikhono@andrew.cmu.edu

Sarah E. Marzen
W. M. Keck Science Department, Pitzer, Scripps, and Claremont McKenna College
Claremont, CA 91711, USA
smarzen@kecksci.claremont.edu

Simon DeDeo
Department of Social & Decision Sciences, Carnegie Mellon University
Pittsburgh, PA 15213, USA
Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501, USA
sdedeo@andrew.cmu.edu

Abstract

We study the relationship between prediction, explanation, and control in artificial "predictive minds"—modeled as Long Short-Term Memory (LSTM) units—that interact with simple dynamical systems. We show how to operationalize key philosophical concepts, and model a key cognitive bias, "alternative neglect". Our results reveal, in turn, an unexpectedly complex relationship between prediction, explanation, and control. In many cases, "predictive minds" can be better at explanation and control than they are at prediction itself, a result that holds in the presence of heuristics expected under computational resource constraints.

We interact with dynamical systems (DSs) as a basic part of everyday life, and spend a great deal of effort trying to predict, explain, and control the ones we encounter. We often try predict the future of a social or physical DS (e.g., "How would that person react to my words?" "Will the door unlock if I turn the key?"), explain what happened (e.g., "What made them so upset?" "Why did the door unlock"), or attempt to control or guide the system's future (e.g., "How can I make them happy?" "How can I unlock the door?").

These are distinct tasks. Explaining why something went wrong doesn't mean knowing how to fix it. Predicting what will happen doesn't mean being able to explain it. One can control a system without being able to explain how. Unfortunately, the relationship between explanation, prediction, and control remains poorly understood, since the existing methodologies are incapable of measuring them in a comparable manner.

The goal of this paper is twofold. First, we present a novel experimental design that simulates human performance in prediction, explanation, and control, and allows for direct comparison between cognitive theories and the results of behavioral studies. For brevity, we focus on a particularly influential, information-theoretic, account of cognition, the "predictive processing" framework [1, 2, 3], that places prediction—rather than explanation [4, 5, 6], or control [7, 8]—in a privileged position.

Preprint. Under review.

Then, we show how differential performance on these tasks can be used as a probe into the nature of cognition itself. What are generic features of performance on the three tasks? What aspects of a system make it easier, or harder, for a predictive mind to learn? How do these answers change with the underlying cognitive model and in the presence of heuristics expected under resource constraints? By varying the systems that the agent interacts with, we show how to identify universal patterns in the covariance of performance on prediction, explanation, and control.

1 Methods

We model a behavioral experiment in which participants observe a probabilistic finite state automaton (see, e.g., Table 4), and are then tested on their ability to predict, explain, and control it. The participants in this work are simulated "predictive minds", equipped with an LSTM module [9] provided by the TensorFlow [10] package, which attempts to maximize log-likelihood (see Appendix D). In this preliminary work, we consider a range of different four-state machines with two potential inputs; see Table 4 for two examples. The participant trains by watching the DS evolve on receiving input symbols, and is tested on a variety of tasks that involve processing two steps in time.

For each task, we consider a "visible" version (where the intermediate state is known) and a "hidden" version (where it is not); these are matched so that the surface level complexities of the questions are identical across the tasks. The six tasks are shown schematically in Table 1. To match standard experimental designs in psychology, test items are two-alternative forced choice questions. This helps equalize the different tasks: even though, for example, the general control problem requires selecting from four possible options, the participant is asked only to make a choice between two of them.

Task	Basic Form	Visible	Hidden
Prediction	"What State will the system end up in?"	$1_a 2_b \underline{?}$	$1_a \mathrm{X}_b \underline{?}$
Explanation	"Why did the system end up in State 3?"	$1_{\underline{a}} 2_{\underline{b}} 3$	$1_{\underline{a}} \mathrm{X}_{\underline{b}} 3$
Control	"How would you get the system into State 3?"	$1_{\underline{?}} 2_{\underline{?}} 3$	$1_{\underline{?}} \mathrm{X}_{\underline{?}} 3$

Table 1: Test tasks, and (shorthand) examples of actual questions that might be asked of a simulated reasoner. States encoded as digits, responses as subscript lowercase letters, unknowns as X, and queries as "?" (see text).

To model human cognition, we must both specify the normative content of the task—what it would mean to get it right—and describe the way in which a human might actually go about producing an answer. Normative specifications are philosophical: they are accounts of what behavior ought to aim for. Descriptive specifications are scientific: they are theories of how agents answer the question ways that might, or might not, approximate the normative standard.

1.1 Normative Operationalization of Prediction, Explanation, and Control

The proper operationalization of the tasks has a number of subtleties; see Appendix B for a full discussion. In each of the three tasks, the subject is given a partial (two time-step) history of the DS states and input symbols. In the visible **prediction** case, the subject sees (for example) that the DS was in State 1, received Input a, transitioned to State 2, and received Input b; they are then asked "which state will DS end up in?" The correct answer is $\mathrm{argmax}_i P(i|1_a 2_b) P(2|1_a)$.

In the visible **explanation** case, the subject sees (for example) that the DS was in State 1, received Input a, transitioned to State 2, received Input b, and transitioned to State 3. They are then asked "why did the DS end up in State 3?" Let us consider two possible explanations. Option 1: "Because the DS received Input a after State 1, rather than b". Option 2: "Because the DS received Input b after State 2, rather than a". The correct answer can be evaluated counterfactually; Option 1 is the correct answer if $P(3|1_b 2_b) P(2|1_b)$ is less than $P(3|1_a 2_a) P(2|1_a)$; Option 2 is the correct answer in the opposite case.

This operationalizes a natural account of causal reasoning in terms of counterfactuals with either the first input or the second being altered, and everything else fixed as before. The cause of the final state is the action that, were it varied, would make the outcome less likely compared to alternatives. This matches long-standing accounts of explanation as counterfactual causation [11]; importantly, it

contrasts with description-length accounts of explanation [12] that rely on a purely Bayesian analysis in terms of material conditionals.

In the visible **control** case, the subject sees (for example) that the DS is in State 1; they are asked to choose a combination of inputs (e.g., ab or ba) that will most likely get the DS into State 2, and then State 3. The correct answer is $\text{argmax}_{\{i,j\}} P(3|1_i 2_j) P(2|1_i)$.

Task	Visible Form	Hidden Form
Prediction	$\text{argmax}_i P(i\|1_a 2_b)$	$\text{argmax}_i \left(\sum_{k=1}^{N} P(i\|1_a k_b) P(k\|1_a) \right)$
Explanation	$P(3\|1_b 2_b) P(2\|1_b) < P(3\|1_a 2_a) P(2\|1_a)$	$\sum_{k=1}^{N} P(3\|1_b k_b) P(k\|1_b)$ $< \sum_{k=1}^{N} P(3\|1_a k_a) P(k\|1_a)$
Control	$\text{argmax}_{\{i,j\}} P(3\|1_i 2_j) P(2\|1_i)$	$\text{argmax}_{\{i,j\}} \left(\sum_{k=1}^{N} P(3\|1_i k_j) P(k\|1_i) \right)$

Table 2: Normative answers to the tasks, based on Table 1; here N is the number of intermediate states. In the explanation task, we show the case where the correct answer is "because the first input was a"; this example assumes that the DS in question has only two possible input symbols, a and b.

The hidden case is identical, but the subject is not told the intermediate state, forcing them to consider different possibilities for the effect of the first action. This is a natural extension of the visible one, where we marginalize over the intermediate state, as shown in Table 2; see Appendix B. Hidden problems are computationally harder, but can, because the marginalization amounts to averaging over potentially noisy estimates, be more robust to error.

1.2 Descriptive Operationalization of Prediction, Explanation, and Control

Reasoners are equipped with a "simulation module" (SM), here taken to be an LSTM, that enables them to predict the future state of a DS based on the prior history of both states and user inputs. We assume that the SM gives answers of the form $\hat{P}(i|1_a 2_b)$; *i.e.*, for an arbitrary two-step sequence of both states and inputs, we have an estimate, $\hat{P}$ of the probability that any particular state i will be the result.

If the reasoners are given a perfect SM and unbounded computational abilities, then they will be capable of 100% accuracy, since they can simply substitute $\hat{P}$ for P in Table 2; the mathematical form remains identical. Errors can result in one of two ways.

Firstly, $\hat{P}$ might differ from P. The reasoner may have insufficient data, or the SM may be internally inconsistent or miscalibrated. If the DS is a finite state machine, for example, but the SM allows for long-range correlations, the SM may be influenced by spurious coincidences.

Secondly, the reasoner might make systematic errors in using the SM. These are at a higher, Marr-algorithmic, level, and include many of the heuristics and biases in the psychological literature. We consider a common bias, "alternative neglect" (AN), which occurs when the reasoner considers a greedy simulation of the path through states rather than all possible options. This leads to deviations from the normative form, as shown in Table 3; see Appendix C for further discussion.

2 Results

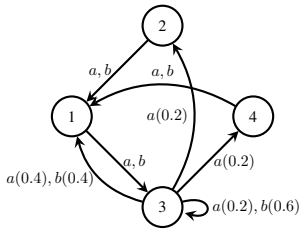
Table 4 shows the results for two test cases, "DS One" and "DS Two", with a short 32-step training set. To show how these results generalize, Fig. 1 of the Appendix overlays the results for DS One and DS Two with 100 randomly generated systems. Our goal here is not to exhaust the full range of possible phenomena, but rather to (1) demonstrate some key, counterintuitive results, and (2) provide proof-of-concept for effect sizes in performance differences.

Counterintuitively, explanation or control can be easier than prediction; this happens 76% and 81% of the time respectively. Also counterintuitively, the visible problem can sometimes be harder than the hidden problem; DS One, in particular, is constructed to make visible explanation particularly hard, and in the random sample, this happens about 38% of the time for prediction, 27% for explanation,

Task	Hidden Form with Alternative Neglect
Prediction	$\hat{P}(i\|1_a\epsilon(1_a)_b)$
Explanation	$\max_k \hat{P}(3\|1_b\epsilon(1_b)_b) < \max_k \hat{P}(3\|1_a\epsilon(1_a)_a)$
Control	$\text{argmax}_{\{i,j\}} \left(\hat{P}(3\|1_i\epsilon(1_i)_j)\right)$

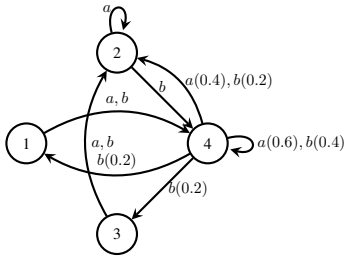
Table 3: Alternative Neglect (AN), a heuristic that deviates from the normative standard of Table 2. Here, $\epsilon(1_a)$ is equal to $\text{argmax}_k\ P(k|1_a)$, the most likely next state if one starts in 1 and takes action a; use of an ϵ term like this is the key simplification of the AN heuristic, that constructs the relevant path in a greedy "fashion". $\hat{P}$ refers to the output of the simulation module (SM) after training on the prediction task. Errors in the reasoner can come from two paths: a failure to simulate the DS well ($\hat{P}$ deviating from P), or a failure to combine $\hat{P}$ in the normative fashion (as in alternative neglect).

(a) DS One



Task	Visible	Hidden	Hidden+AN
Prediction	89.1%	89.7%	85.9%
Explanation	57.0%	80.8%	59.1%
Control	63.6%	73.2%	51.4%

(b) DS Two



Task	Visible	Hidden	Hidden+AN
Prediction	93.4%	92.7%	89.8%
Explanation	91.2%	89.3%	81.9%
Control	93.8%	91.7%	86.9%

Table 4: Average LSTM accuracy rates for prediction, explanation, and control tasks in two simple machines. The training phase has $N_l = 32$ state-symbol pairs. Standard errors are $< 1\%$.

and 9% for control. This happens because the visible case requires more precise calibration of particular transitions. Meanwhile, as expected, the alternative neglect heuristic leads to significant deficits in performance; over our random sample, prediction and control decline by 6 points, and explanation by 8 points, on average, with a broad distribution; Fig. 1(a) shows the explanation case, and Fig. 2, prediction and control.

3 Discussion

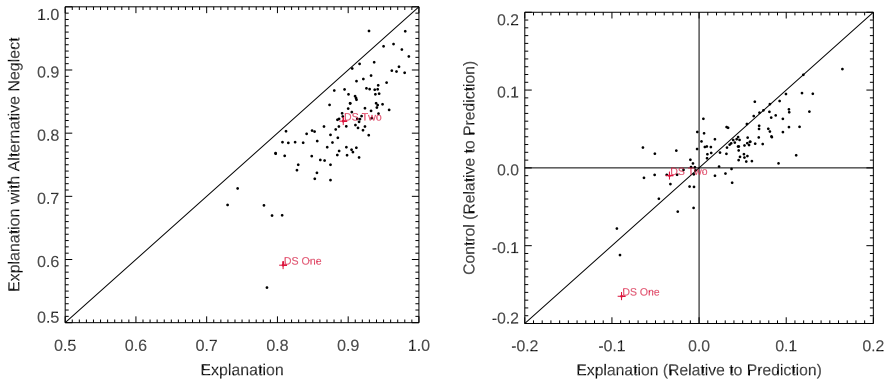
Our work began with the key idea that prediction, explanation, and control ought to be examined simultaneously. A long tradition in dynamic decision-making studies has examined human power to *control* DSs [13, 14, 15]). Prediction and explanation, however, have been largely overlooked. A few works have used DSs to investigate causal inference (e.g., [16]), but did not measure control and prediction. One exception is in implicit learning experiments that included a prediction task in a post-experimental questionnaire, to measure awareness of knowledge obtained about the dynamic system [17, 18]. Consistent with our new results, they found that people can learn to control a dynamic system, but still fail to accurately predict its behaviour.

Our main result is that this holds even for "predictive" minds. Agents can find control and explanation easier than prediction—even when they are trained to minimize loss—on prediction tasks. These effects from DS to DS, providing a guide for laboratory tests to tease apart the underlying mechanisms associated with each task.

A second result is how normative constraints (Table 2) compare with how a psychologically-realistic subject might actually perform (Table 3). "Alternative neglect" is a common bias in the real world [19]; our work shows how this heuristic leads to characteristic patterns of sub-optimal performance that vary both from DS to DS, and between the tasks (prediction, explanation, and control) themselves.

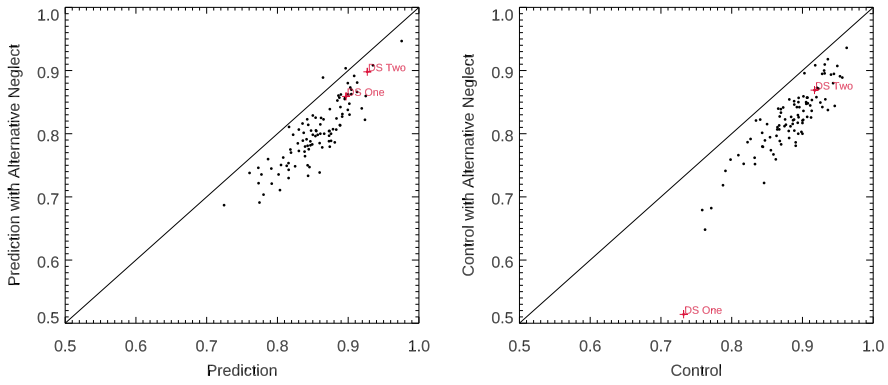
A Random Machine Sampling

To explore the range of behaviors, we compare our test cases, DS One and DS Two, to a set of 100 randomly generated machines. Each random system has deterministic transitions except for three randomly-chosen state/input pairs, where it branches, probabilistically.



(a) (b)

Figure 1: Key results for random machines: (a) alternative neglect induces systematic deficits in all three tasks, as can be seen here in the case of explanation; (b) it is often easier to control or explain a system than it is to predict one. Dots label randomly-sampled machines; "+" symbols label DS One and DS Two.



(a) (b)

Figure 2: Parallel figures to 1(a), for prediction and control.

B Normative Definitions

In this section, we describe and justify our normative definitions in greater detail, discuss some alternative choices that could be made, and build further intuition for how the three tasks engage the underlying predictive simulation model (SM) in different ways.

In each case, tasks are completed on the basis of an underlying simulation; we assume that agents are first trained to optimize simple prediction tasks of the form $P(i|X)$, where i is the next machine state, and X is some explicitly, fully-specified past history.

B.1 Prediction

In the simplest, "visible", prediction case, we have problems of the form "$1_a 2_b i$", where the observer is given a sequence of two machine state/user input pairs, and asked what is most likely to happen next (the value of i). In the finite state machine case, the Markovian condition means that knowledge

of the second machine state and input is sufficient to make the optimal prediction, but this is not true in general where, for example, different first inputs could lead to second-state results that are observationaly-indistinguishable but have different downstream consequences—as can happen, for example, in a hidden Markov model.

The normative distribution to work from, in this case, is

$$P(i) = P(i|1_a 2_b) \tag{1}$$

or, in words, given that the user has seen $1_a 2_b$, what is most likely to happen next. In cases like the finite state machine, where the observable machine state is a sufficient statistic for future dynamics, then this reduces to $P(i|2_b)$.

In the hidden form, things are more complicated. Here, the problem is of the form "$1_a k_b i$", the observer is asked to predict i, but they are unaware of the particular value of the intermediate state k.

A natural, but incorrect, approach is to compute

$$\sum_{k=1}^{N} P(i|1_a k_b), \tag{2}$$

but this is, in general, incorrect, because it gives equal weight to values of k that might, in fact, be very unlikely given the prior history of (in this case) "1_a". The correct solution is to condition only on the start state and intermediate actions, which means that we should marginalize over the intermediate state in the following fashion,

$$P(i) = \sum_{k=1}^{N} P(i|1_a k_b) P(k|1_a), \tag{3}$$

i.e., to remember to correct for the fact that some values of k are very unlikely. In the finite state machine case, this reduces to

$$\sum_{k=1}^{N} P(i|k_b) P(k|1_a). \tag{4}$$

B.2 Explanation

As described in the main text, we take an "off the shelf" account of explanation from the philosophical literature where (1) explanation-like "why" questions are to be interpreted causally—*i.e.*, to ask "why did X happen" is to ask "what caused X to happen"; and (2) causal questions are to be understood counterfactually and on the basis of a model, where an intervention is made at one location, with everything else kept constant, and the relative effects of different interventions are compared. Pearl's "do" operator is an example of such an interpretation. This is, certainly, not the only way to explain explanation, but it is an influential and popular account.

In the visible case, the problem is of the form "$1_a 2_b 3$", and the question is "which input caused the system to end up in State 3, given that it passed through State 2".

To compute this, we compare two probabilities,

$$P(3|1_b 2_b) P(2|1_b) \text{ and } P(3|1_a 2_a) P(2|1_a) \tag{5}$$

These concern two alternate universes, or counterfactual conditions; in one, instead of doing a and then b, we do b and then b; in the other, we do a and then a. For these two counterfactual universes, we ask, "what is the likelihood that, given these inputs, we see the system pass through State 2 and then State 3?"

In many (though not all cases), these two probabilities will differ. The cause is then assigned to the input symbol which, when varied, produces the greatest *decrease* in probability. In words, the cause of the outcome is the thing that, if someone else was done instead, would reduce the likelihood of the outcome the most. If, for example, $P(3|1_b 2_b) P(2|1_b)$ is less than $P(3|1_a 2_a) P(2|1_a)$, then "the fact that we started by doing a" is the cause, rather than "the fact that we finished by doing b".

There is one subtlety here, which is that it is might be the case that $P(3|1_a 2_b) P(2|1_a)$, *i.e.*, the probability under the observed behavior, is lower than either of the counterfactuals, and even (say) to

the counterfactual where both inputs are varied. The normative prescription will still assign a cause to one of the counterfactual, but there is a certain counterintuitive feel here—one wants to say "there was no cause, it was an accident, and the alternatives would have made it worse". For simplicity, we eliminate these situations by hand.

The hidden version of the explanation task goes through in a similar fashion; the questions take the form "$1_a k_b 3$"– the formal question is "given that the system started in State 1, and we did a, and then b, why did it end up in State 3", and we compare

$$\sum_{k=1}^{N} P(3|1_b k_b) P(k|1_b) \text{ and } \sum_{k=1}^{N} P(3|1_a k_a) P(2|k_a) \tag{6}$$

which reduces to

$$\sum_{k=1}^{N} P(3|k_b) P(k|1_b) \text{ and } \sum_{k=1}^{N} P(3|k_a) P(2|k_a) \tag{7}$$

in the finite state machine case.

B.3 Control

In the visible case for control, we are given problems of the form "$1_? 2_? 3$". There are two ways to interpret this problem: "if you start in state 1, what actions can you take to get it to State 2, and then State 3" or, alternatively, "if you start in state 1, what actions can you take to get to State 3, given that we assume that you end up in State 2 along the way."

Under the first interpretation, you find the action pair $\{i, j\}$ that maximizes $P(3|1_i 2_j) P(2|1_i)$; under the second interpretation, you find the action pair $\{i, j\}$ that maximizes $P(3|1_i 2_j)$.

Both interpretations are things we might find in the real world. Interpretation one is natural when (for example) the goal is to reproduce a performance; one should take the action that will get you to the desired intermediate state, and will set you up well to get to the final state. We use this interpretation in the main paper.

Interpretation two is natural, however, when, for example, planning for worst-case scenarios; a fire-marshal, for example, might ask "given that there will be a fire, how should we act to minimize resulting casualties". In this case, we want to take the action that, given the (unfortunate) fact that we will have a fire at the next time step, we will be in a good position to recover. Of course, these kinds of questions, under interpretation two, are most natural when the machine has hidden state; in the finite state case, $P(3|1_i 2_j)$ reduces to $P(3|2_j)$; the action i becomes irrelevant.

The hidden version of the control task does not have this subtlety of interpretation; the questions take the form "$1_? k_? 3$", the verbal form is "given that the system started in State 1, what can we do to get it to State 3', and we maximize

$$\sum_{k=1}^{N} P(3|1_i k_j) P(k|1_i) \tag{8}$$

C Alternative Neglect

In all three cases for the hidden case–Eqs 3, 6 and 8—we have a sum over intermediate conditions. Psychologically, this amounts to considering a potentially large number of alternative possibilities, and keeping them all in mind simultaneously. It is natural to consider a simple heuristic, "alternative neglect", where psychologically-realistic observers consider only a single path for each possibility.

There are two natural ways to subselect a path to consider. One can consider the "most likely path overall" (MLP), or, alternatively, "the path constructed by taking the most likely next step at each point in time" (MLNS). In the case of prediction, for example, this would amount to deviating from the normative prescription, Eq. 3, in one of two ways, computing either, in the MLP case,

$$\max_k \ [P(i|1_a k_b) P(k|1_a)] \,, \tag{9}$$

or, in the MLNS case,

$$P(i|1_a \epsilon (1_a)_b) \tag{10}$$

where $\epsilon(1_a) = \text{argmax}_k\ P(k|1_a)$ is the most likely next state if one starts in 1 and takes action a.

In the case of explanation, it takes the form of comparing, in the MLP case,

$$\max_k P(3|1_b k_b)P(k|1_b) \text{ and } \max_k P(3|1_a k_a)P(k|1_a) \tag{11}$$

or, in the MLNS case,

$$\max_k \hat{P}(3|1_b \epsilon(1_b)_b) \text{ and } \max_k \hat{P}(3|1_a \epsilon(1_a)_a) \tag{12}$$

Finally, in the case of control, it takes the form of either, in the MLP case,

$$\max_k P(3|1_i k_j)P(k|1_i) \tag{13}$$

or, in the MLNS case,

$$\text{argmax}_{\{i,j\}} \left(\hat{P}(3|1_i \epsilon(1_i)_j)\right) \tag{14}$$

MLP is related to the concept in physics of the "minimum action path", while MLNS is a cognitively frugal, greedy optimization algorithm. Both are, in different situations, plausible ways for a human to approximate the normative standard. For example, MLP is likely to operate in cases where the observer has many, many samples of the system, and has the heuristic of "remembering the most common relevant path". MLNS is more likely to be in operation when the person is actively simulating the situation. In this paper, where we imagine subjects encountering only a limited amount of data, and ask questions that are out of sample, we consider the MLNS case.

D LSTM configuration

Our LSTM, deliberately chosen to be resource-challenged, consists of one unit of ten nodes [20] used, with the logits layer being a fully connected layer with linear activations on top of the LSTM. The number of epochs was 1000 with a learning rate of 10^{-1}, trained with an Adam [21] optimizer on categorical cross-entropy loss. The input were two-time-step time series of the past state and action; the output was the next state. All states and actions were encoded using one-hot vectors. Across multiple trials, the loss was approximately 0.1; in other words, the LSTM was able to model the state-state transitions well, as would be expected for a DFA; errors are partly driven by small-number statistics on the input data.

References

[1] Andy Clark. Whatever next? predictive brains, situated agents, and the future of cognitive science. 36(3):181–204, 2013. Publisher: Cambridge University Press.

[2] Karl Friston. The free-energy principle: a unified brain theory? 11(2):127–138, 2010.

[3] Jakob Hohwy. *The predictive mind*. Oxford University Press, 2013.

[4] Zachary Horne, Melis Muradoglu, and Andrei Cimpian. Explanation as a cognitive process. 23(3):187–199, 2019.

[5] Tania Lombrozo. The structure and function of explanations. 10(10):464–470, 2006.

[6] Ruth M.J. Byrne. Counterfactual thought. *Annual Review of Psychology*, 67(1):135–157.

[7] David Silver, Satinder Singh, Doina Precup, and Richard S. Sutton. Reward is enough. 299:103535, 2021.

[8] Cleotilde Gonzalez, Javier F. Lerch, and Christian Lebiere. Instance-based learning in dynamic decision making. *Cognitive Science*, 27(4):591–635, 2003. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1207/s15516709cog2704_2.

[9] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. 9(8):1735–1780, 1997. Conference Name: Neural Computation.

[10] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous systems, 2015.

[11] Peter Menzies and Helen Beebee. Counterfactual Theories of Causation. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2020 edition, 2020.

[12] Zachary Wojtowicz and Simon DeDeo. From probability to consilience: How explanatory values implement bayesian reasoning. *Trends in Cognitive Sciences*, 24(12):981–993, 2020.

[13] Berndt Brehmer. Dynamic decision making: Human control of complex systems. 81(3):211–241, 1992.

[14] Jared M. Hotaling, Pegah Fakhari, and Jerome R. Busemeyer. Dynamic decision making. In *International Encyclopedia of the Social & Behavioral Sciences*, pages 708–713. Elsevier, 2015.

[15] Cleotilde Gonzalez, Pegah Fakhari, and Jerome Busemeyer. Dynamic Decision Making: Learning Processes and New Research Directions. *Human Factors*, 59(5):713–721, August 2017. Publisher: SAGE Publications Inc.

[16] Mark Steyvers, Joshua B. Tenenbaum, Eric-Jan Wagenmakers, and Ben Blum. Inferring causal networks from observations and interventions. *Cognitive Science*, 27(3):453–489, 2003.

[17] Dianne C. Berry and Donald E. Broadbent. On the Relationship between Task Performance and Associated Verbalizable Knowledge. *The Quarterly Journal of Experimental Psychology Section A*, 36(2):209–231, May 1984.

[18] Dianne C. Berry and Donald E. Broadbent. Interactive tasks and the implicit-explicit distinction. *British Journal of Psychology*, 79(2):251–272, 1988.

[19] Philip M. Fernbach, Adam Darlow, and Steven A. Sloman. Neglect of Alternative Causes in Predictive but Not Diagnostic Reasoning. *Psychological Science*, 21(3):329–336, March 2010. Publisher: SAGE Publications Inc.

[20] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.