Human Causal Transfer: Challenges for Deep Reinforcement Learning

Mark Edmonds[1*] markedmonds@ucla.edu
James Kubricht[2*] kubricht@ucla.edu
Colin Summers[3,4] colinxs@uw.edu
Yixin Zhu[5] yixin.zhu@ucla.edu
Brandon Rothrock[3] Brandon.Rothrock@jpl.nasa.gov
Song-Chun Zhu[1,5] sczhu@stat.ucla.edu
Hongjing Lu[2,5] hongjing@ucla.edu
[*] Equal Contributors [1] Department of Computer Science, UCLA [2] Department of Psychology, UCLA
[3] Jet Propulsion Laboratory, Caltech [4] Department of Computer Science, UW [5] Department of Statistics, UCLA

Abstract

Discovery and application of causal knowledge in novel problem contexts is a prime example of human intelligence. As new information is obtained from the environment during interactions, people develop and refine causal schemas to establish a parsimonious explanation of underlying problem constraints. The aim of the current study is to systematically examine human ability to discover causal schemas by exploring the environment and transferring knowledge to new situations with greater or different structural complexity. We developed a novel OpenLock task, in which participants explored a virtual "escape room" environment by moving levers that served as "locks" to open a door. In each situation, the sequential movements of the levers that opened the door formed a branching causal sequence that began with either a common-cause (CC) or a common-effect (CE) structure. Participants in a baseline condition completed five trials with high structural complexity (*i.e.*, four active levers). Those in the transfer conditions completed six training trials with low structural complexity (*i.e.*, three active levers) before completing a high-complexity transfer trial. The causal schema acquired in the transfer condition was either congruent or incongruent with that in the transfer condition. Baseline performance under the CC schema was superior to performance under the CE schema, and schema congruency facilitated transfer performance when the congruent schema was the less difficult CC schema. We compared between-subjects human performance to a deep reinforcement learning model and found that a standard deep reinforcement learning model (DDQN) is unable to capture the causal abstraction presented between trials with the same causal schema and trials with a transfer of causal schema.

Keywords: Active causal learning; schema transfer; deep reinforcement learning

Introduction

Causality has been dubbed the "cement of the universe" (Mackie, 1974). The key research question in the field of causal learning is how various intelligent systems, ranging from rats to humans and machines, can acquire knowledge about cause-effect relations in novel situations. Decades ago, a number of researchers (*e.g.*, Shanks & Dickinson, 1988; Shanks, 1991) suggested that causal knowledge can be acquired by a basic learning mechanism, associative learning, that non-human animals commonly employ in classical conditioning paradigms to learn the relationship between stimuli and responses. A major theoretical account of associative learning is the Rescorla-Wagner model, guided by prediction error in updated associative weights on cue-effect links (Rescorla & Wagner, 1972).

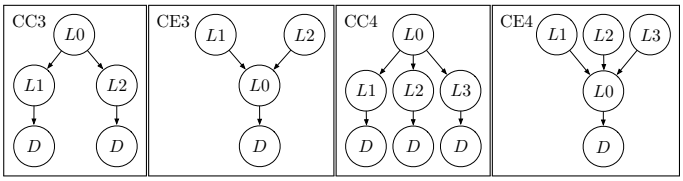
However, subsequent research has produced extensive evidence that human causal learning depends on more sophisticated processes than associative learning of cue-effect links (Holyoak & Cheng, 2011). Human learning and reasoning involves the acquisition of abstract causal structure (Waldmann & Holyoak, 1992) and strength values for cause-effect relations (Cheng, 1997). Causal graphical models (Pearl, 2000) have been integrated with Bayesian statistical inference (Griffiths & Tenenbaum, 2005, 2009; Lu, Yuille, Liljeholm, Cheng, & Holyoak, 2008) to provide a general representational framework for human causal learning (Holyoak & Cheng, 2011).

Nevertheless, most models of human causal learning assume that the hypothesis space of causal variables and causal structures is given and that inference focuses on selecting the best causal structure to explain the observed contingency information relating causal cues to effects. It is unclear how an agent could *actively* explore a completely novel situation in an online fashion and narrow down the set of potential causal structures to enable efficient inference.

In situations in which outcomes depend on the learner's actions rather than simply observations, reinforcement learning (RL) is a widely-used modeling tool. It is useful for designing autonomous, dynamic agents capable of exploration in complex environments. RL focuses on learning what to do by mapping situations to actions, so as to maximize a reward signal (Sutton & Barto, 1998). RL has historically been closely linked with associative learning theory and conceives of learning as essentially a process of trial and error. The connection between classical conditioning and temporal-difference learning, a central element of RL, is widely acknowledged (Sutton & Barto, 1990). Hence, RL could be considered as a modern version of associative learning, where learning is not only guided by prediction error but also by other learning mechanisms, notably the estimation of the reward function. Recent advances in RL, especially deep RL, have demonstrated impressive success in applications involving the design of autonomous, dynamic agents for exploration, including playing Atari and Go (Mnih et al., 2015; Van Hasselt, Guez, & Silver, 2016; Silver et al., 2016) and learning complex robot control policies (Levine, Finn, Darrell, & Abbeel, 2016).

With these significant developments in RL, is it possible for modern learning models to acquire human-like causal knowledge? To address this question, we designed a novel task to examine learning of action sequences governed by different causal structures, allowing us to determine in what situations humans can transfer their learned causal knowledge. Our design involves two types of basic causal structures (common cause (CC) and common effect (CE); see Figure 1). When multiple causal chains are consolidated into a single structure, they can form either CC or CE schemas. Previous studies using an observational paradigm have found an asymmetry in human learning for common-cause and common-effect structures (Waldmann & Holyoak, 1992).

To design a novel environment for humans, we developed a virtual "escape room". Imagine that you find yourself trapped in an empty room where the only means of escape is through a door that will not open. Although there is no visible keyhole on the door–nor do you see any keys lying around–there are some conspicuous levers sticking out of the walls. Your first instinct might be to pull the levers at random to see what happens, and given the outcome, you might revise your theory about how lever interactions relate to the opening of the door. We refer to this underlying theory as a causal schema: *i.e.*, a conceptual organization of events identified as cause and effect (Heider, 1958). These schemas are discovered with experience and can potentially be transferred to novel target problems to infer their characteristics (Kubricht, Lu, & Holyoak, 2017).



(a) (b) (c) (d)

Figure 1: Common cause (CC) and common effect (CE) structures used in the present study. *D* indicates the effect of opening the door. (a) CC3 condition, three lock cues; (b) CE3 condition, three lock cues. (c) CC4 condition, four lock cues; (d) CE4 condition, four lock cues.[1]

In the escape room example, one method of unlocking the door is to induce the causal schema connecting lever interactions to the door's locking mechanism. However, it remains unclear whether people are equally proficient in uncovering CC and CE schemas in novel situations. In the current study, we first assessed whether human causal learning can be impacted by the underlying structure, comparing learning of a CC structure with learning of a CE structure. We then examined whether learning one type of causal structure can facilitate subsequent learning of a more complex version of the same schema involving a greater number of causal variables. We compared human performance in a range of learning situations with that of a deep RL model to determine whether behavioral trends can be captured by an algorithm which learns solely by reward optimization, with no prior knowledge about causal structure.

In the remainder of the paper, we first describe the RL algorithms used for the present OpenLock task. We then describe the design of an experiment and report human results. Next, we describe our RL model and model results. Finally, we discuss the implications of our findings for causal learning.

Reinforcement Learning

RL focuses on learning a mapping between states and actions to maximize some reward function (Sutton & Barto, 1998). *Q*-learning, a representative model-free RL technique, seeks to learn an action-value function using expected discounted rewards (Watkins & Dayan, 1992). The optimal Q function is defined as:

$$Q^*(s,a) = \max_{\pi} \mathbb{E}\left[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots | s_t = s, a_t = a, \pi\right], \quad (1)$$

where s_t is the state at time t, a_t the action, $\pi = P(a|s)$ the agent's policy, $\gamma \in [0,1]$ a discount factor, and r_t the reward.

A milestone of RL is the introduction of DQN (Mnih et al., 2015), which overcomes limitations presented by previous neural-network-based RL methods. Specifically, DQN uses *experience replay* (ONeill, Pleydell-Bouverie, Dupret, & Csicsvari, 2010) to mitigate networks from over-fitting to recent correlations in the observation sequence. DQN also introduced a *target network* that is only updated every τ steps to further mitigate over-fitting. This method showed a remarkable ability to play Atari games above human ability.

DDQN (Van Hasselt et al., 2016) expands on DQN by reducing over-estimations of the Q function. While DQN uses a single value estimator to both select and evaluate a particular action, DDQN decouples selection and evaluation by learning two value estimators: one for selection and another for evaluation. DDQN shows superior performance and stability over DQN in the vast majority of Atari games and has become one of the state-of-the-art RL methods. In this paper, we choose DDQN as the computational model due to its straightforward implementation and remarkable performance on various tasks.

[1]Example solution executions for CE3 and CC3 can be viewed at https://vimeo.com/265596602

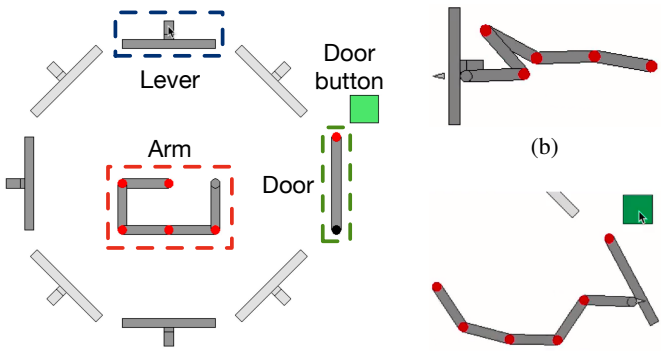


Figure 2: (a) Starting configuration of a 3-lever trial. All levers begin pulled towards the robot arm, whose base is anchored to the center of the display. The arm interacts with levers by either *pushing* outward or *pulling* inward. This is achieved by clicking either the outer or inner regions of the levers' radial tracks, respectively. Only push actions are needed to unlock the door in each lock situation. Light gray levers are always locked, which is unknown to both human subjects and RL at the beginning of training. Once the door is unlocked, the green button can be clicked to command the arm to push the door open. The black circle located opposite the door's red hinge represents the door lock indicator: present if locked, absent if unlocked. (b) Push to open a lever. (c) Open the door by clicking the green button.

Experiment: OpenLock Task

Participants

A total of 240 undergraduate students (170 female; mean age = 21.2) were recruited from the University of California, Los Angeles (UCLA) Department of Psychology subject pool and were compensated with course credit for their participation.

Materials and Procedure

In the OpenLock task, participants were asked to "escape" from a virtual room by opening a locked door that was controlled by a lever mechanism (see Figure 2). The task was to figure out what lever mechanisms can open the door. Each lock situation consisted of seven levers surrounding a robot arm and a door which began in a locked state. The levers pertinent to the locking mechanism (*i.e.*, active levers) were colored grey, and levers irrelevant to the locking mechanism (*i.e.*, inactive levers) were colored white. Participants were not explicitly told which levers were active or inactive but were instead required to learn the distinction through trial and error. This was not generally difficult, however, as the inactive levers could never be moved. The order in which the active levers needed to be moved followed either a common cause (CC) or common effect (CE) schema (see Figure 1), and participants were given 30 attempts to discover *every* solution in each situation. Participants were instructed to consider solutions as "combinations" to each lock, and discovery of every solution/combination was required to ensure that participants understood the underlying causal schema in each situation. Participants also operated under a movement-limit constraint whereby only three movements could be used to both (1) interact with the levers (two movements) and (2) push open the door (one movement). If a participant tried to move an active lever in an incorrect order, the lever would remain stationary and a movement would be expended. Each trial reverted to its

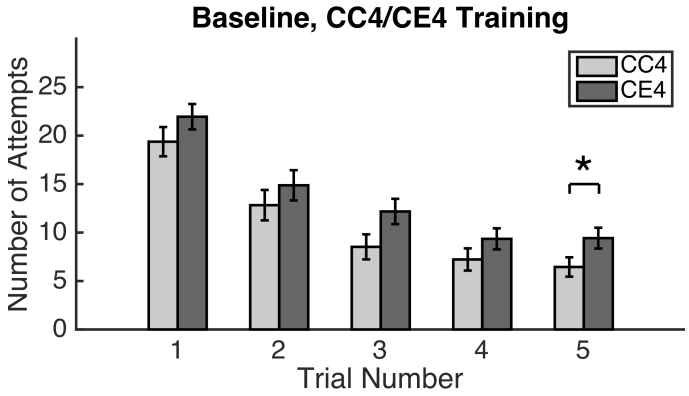


Figure 3: Average number of attempts needed to find all unique solutions in the 4-lever common cause (CC4) and common effect (CE4) baseline conditions. Error bars indicate standard error of the mean.

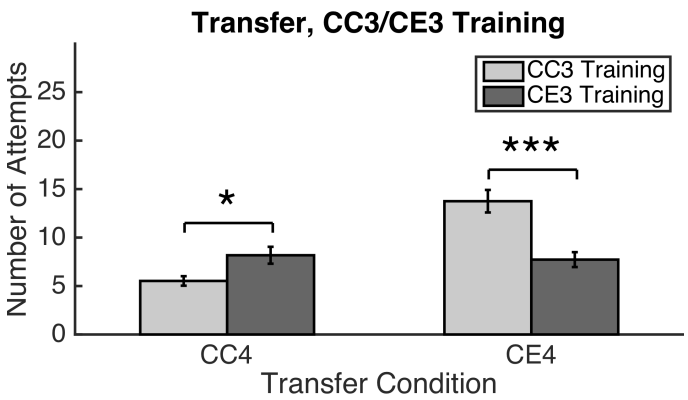


Figure 4: Transfer trial results. Average number of attempts needed to find all unique solutions in the 4-lever common cause (CC4; left) and common effect (CE4; right) conditions. Light and dark grey bars indicate CC3 and CE3 training, respectively. Error bars indicate standard error of the mean.

initial state once the three movements were expended, and the experiment automatically proceeded to the next trial after 30 attempts. The number of remaining solutions and attempts were provided in a console window located on the same screen as the OpenLock application.

In the environment, users commanded the movement of a simulated robot arm by clicking on desired elements in a 2D display. Levers could either be pushed or pulled by clicking on their inner or outer tracks, although pulling on a lever was never required to unlock the door. There were either 3 or 4 active levers in each lock situation. We refer to the 3- and 4-lever common cause situations as CC3 and CC4 (Figure 1a, 1c), respectively, and the 3- and 4-lever common effect situations as CE3 and CE4 (Figure 1b, 1d), respectively. Note that these numbers correspond with the number of *active* levers. The status of the door (*i.e.*, either locked or unlocked) was indicated by the presence or absence of a black circle located opposite the door's hinge. Once the door was unlocked and the black circle disappeared, participants could command the robot arm to push the door open by clicking on a green *push* button. The robot arm consisted of five segments that were free to rotate such that all elements in the display were easily reached by the arm's free end; the arm position control was implemented using inverse kinematics. Box2D (Catto, 2011) was used to handle collision, and the underlying simulation environment uses OpenAI Gym (Brockman et al., 2016) as the virtual playground to train agents and enforce causal schemas through a finite state machine.

Participants were randomly assigned to one of six conditions in a between-subjects experimental design (40 participants per condition) and began the experiment by viewing a set of instructions outlining important components and details in the lock environment[2]. Fifteen additional participants were recruited but subsequently removed from the analysis due to their inability to complete any trial in the allotted number of attempts. The first two experimental conditions were baselines that contained five different lock situations comprised of either CC4 or CE4 trials, exclusively. These baseline conditions for the two control groups, denoted as CC4 and CE4, were included to assess whether human causal learning can be impacted by the underlying structure, comparing learning of a common-cause structure with learning of a common-effect structure. For the remaining four conditions, we examined whether learning one type of causal structure can facilitate subsequent learning of a more complex version of the same schema involving a greater number of causal variables (*i.e.*, active levers). The four conditions contained six training trials with 3-lever situations, followed by one transfer trial with a 4-lever situation. The schema underlying the 3- and 4-lever situations was either congruent (CC3-CC4, CE3-CE4) or incongruent (CC3-CE4, CE3-CC4) and always remained the same throughout the 3-lever training trials. Participants required approximately 17.4 min to complete the baseline trials and 17.3 min to complete the training and transfer trials.

[2]The instructional video can be viewed at `https://vimeo.com/265302423`

Human Results

We first compared performance across the two baseline conditions where participants only completed the CC4 and CE4 trials. The average number of attempts to solve a 4-lever task in each of the baseline trials is shown in Figure 3. Participants showed a clear learning effect as fewer attempts were needed for later trials, $F(4,75) = 40.16, p < .001$. The main effect of causal structure was trending towards significance, $F(1,78) = 3.63, p = .06$, and results from a two-sample t-test at the final trial (*i.e.*, Trial 5) indicate that the task with the CE structure took significantly more attempts to solve than the CC structure, $t(78) = 2.00$, $p < .05$. This result suggests that when a situation involved relatively high structural complexity, the CE structure was more difficult to discern than the CC structure.

Next, we examined the training performance in the four groups who completed both the training trials with 3-levers and the transfer trial with 4-levers. A clear learning improvement was found, indicated by a significant main effect of training trials, $F(5,152) = 56.02, p < .001$. There was no difference in training performance between the CC3 and CE3 groups, $F(1,158) = 0.11$. Compared with the two control groups in the four-lever situations, participants showed similar performance in the three-lever situations, suggesting that structural complexity impacts the comparative difficulty between CC and CE trials. For simple structures with fewer causal variables, people appear to learn different types of causal structures equally well. However, as complexity increases, some causal structures appear easier to learn than others. To further investigate whether the four training groups achieved the same level of learning, we compared the performance at the final training trial in the three-lever task. There were no differences in performance between the CC3-CC4 and CC3-CE4 groups, $t(78) = 0.87$, or the CE3-CC4 and CE3-CE4 groups, $t(78) = 0.48$. This suggests that participants in each training group had approximately the same level of understanding of the underlying causal schema before moving to their respective transfer trials.

Finally, we examined participants' transfer performance. The average number of attempts needed to solve the transfer trials are depicted in Figure 4. A two-way ANOVA revealed a significant interaction effect between the training structure and the testing structure, $F(1,156) = 24.94, p < .001$, indicating superior transfer when the same type of causal structures were used in the training and transfer trials. The resulting plot shows that participants trained under a CC3 structure performed better in the CC4 condition than those trained under a CE3 causal structure, $t(78) = 2.62, p = .01$. Similarly, participants trained under a CE3 structure performed better in the CE4 test trials than did those who trained under a CC3 structure, $t(78) = 4.27, p < .001$. Consistent with the baseline groups, there was also a significant main effect of causal structure in the transfer test, as the CE4 condition required more attempts than the CC4 condition, $F(1,158) = 17.14, p < .001$.

Model Details

The state space of the lock environment consists of 16 binary dimensions: 7 for the state of each lever (*pushed* or *pulled*), 7 dimensions for the color of each lock (grey or white), 1 dimension for the state of the lock (*locked* or *unlocked*), and 1 dimension for the state of the door (*closed* or *open*). The action space consists of 15 dimensions: each of the 7 levers has 2 actions (*push* and *pull*), and the door has one action (*push*).

DDQN is used as the underlying Q-learning algorithm. The neural network is set to consist of 4 hidden layers (Figure 5): a 16-dimensional state space input vector, densely connected to 4 layers with 128 nodes, each of which using a ReLU activation function, leading to an output layer with 15 dimensions and a linear activation. During policy evaluation, the action with the highest output is chosen as the next action to take, $a_t^* = \underset{a}{\text{argmax}}\, P(a|s)$.

Reward Functions are perhaps the most critical part of RL. The purpose of a reward function is to signal the agent when an action helps or hurts achieving the goal (Sutton & Barto, 1998). The agent's goal is to maximize the accumulated reward over its experience in the environment. We design a multitude of reward functions to encode information about the environment:

- *Basic* A reward of 10 is given if the door is unlocked, 50 is given if the door is opened, and otherwise 0.
- *State change* Builds on the basic reward function but adds a reward of 0.5 if the agent's action changes the observation vector in any way.
- *Unique solutions* Builds on the basic reward function, but only gives rewards if the successful action sequence has not previously been executed.
- *Negative immovable* Builds on the basic reward function, but also gives a reward of -0.5 if the agent interacts with a lever that is immovable.
- *Negative repeat* Builds on the basic reward function but adds a penalty of -0.25 for repeated actions to minimize the chance that the agent repeats the same action.
- *Partial action sequence* Builds on the basic reward function and state change. Awards a reward of 1 if the first action taken is part of a solution. This allows for a smoother reward function and is equivalent to awarding a state change reward for the first move only.
- *Solution multiplier* Builds on the base reward function but adds a reward multiplier for each successive solution found. For example, if the multiplier is set to 1.5x, the first solution found has a reward of 1 for opening the door, then the second solution has a reward of 1.5, and the third 2.25. The order in which solutions are found does not matter. Intuitively, this is an alternative mechanism to encode the importance of finding multiple solutions rather than a single solution.

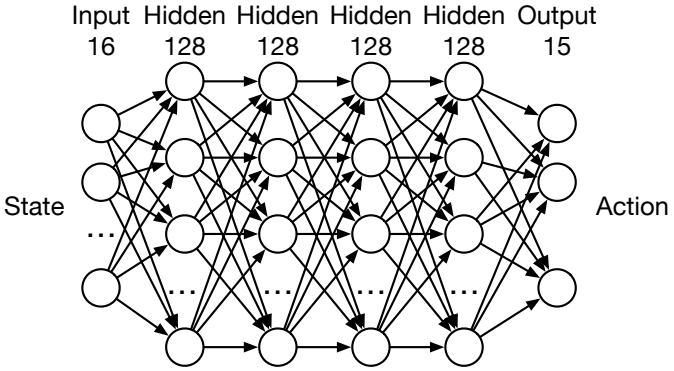


Figure 5: Neural network architecture of DDQN. Input consists of a 16-dimensional state vector. All hidden layers are 128-dimensional and densely connected with ReLU activation. The output layer is 15-dimensional with linear activation.

Model Results

We tested the RL's ability to solve the OpenLock task by starting with reward conditions and parameter values as close to human participants as possible. The unique solutions reward function only gives rewards for successful action sequences, which is equivalent to the information human participants received from the console window. However, this reward function results in an agent incapable of meaningfully interacting with the environment. During DDQN's experience replay, the same state-action pair can yield different reward values if the agent executes the same successful action sequence more than once (reward is only given on the first execution). The agent may experience the same state, take the same action, and receive a different reward. Even worse, the agent will receive the reward only once per solution per trial, making the probability of correctly updating networks weights during experience replay low.

We empirically evaluated combinations of the other reward functions. We found that the best DDQN performance can be achieved by using the negative immovable, partial action sequence, and solution multiplier reward functions. This combination has a number of properties that make it conducive to learning because it: (1) penalizes the agent for performing meaningless actions, (2) encodes a reward for finding multiple solutions, and (3) creates a smoother reward function by giving rewards for performing the correct first action.

Specifically, the optimal agent uses the following DDQN parameters: discount factor, $\gamma = 0.8$; starting epsilon, $\varepsilon = 1$; minimum epsilon, $\varepsilon_{\text{min}} = 0.01$; learning rate, $\alpha = 0.001$; epsilon decay, $\varepsilon_{\text{decay}} = 0.995$ (refer to Van Hasselt et al., 2016 for more information on these parameters). At the end of each trial, ε was set to 0.5 to encourage more exploration and to prevent the agent from adopting a policy specific to the previous trial.

For baseline cases with only CC4 and CE4 conditions, the RL agent was given 300 attempts per trial and looped over all 4-lever trials 10 times. For transfer conditions, the agent was given 300 attempts per trial (in contrast to 30 attempts per trial for human subjects) and looped over the training trials 100 times (in contrast to only once for human subjects) during training. In the testing, the agent was given 300 attempts per trial and looped over the testing trials 10 times. This is different from the human experiments, however, when the agent trained with one iteration over the trials (with a higher number of attempts per trial), the agent performed extremely poorly after the first trial. Although

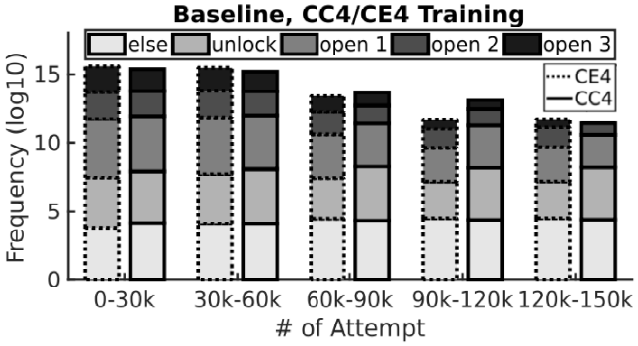


Figure 6: Baseline trial results of RL agent. The frequency of each reward category is plotted in log-scale; the number of attempts is the same in each group. The decreasing height of the bars indicates that one reward category is dominating; specifically the *else* category. The agent's performance decreases as the number of attempts increases, meaning that the agent is getting worse at the task during training.

these differences in the experimental setup make a quantitative comparison to human results difficult, general qualitative assessments can be made to judge the overall performance of DDQN in this task.

First, we examine the performance of the baseline over time (Figure 6). The categories represent how close the agent was to a solution when the attempt ended (*i.e.*, when the agent had executed three actions). The categories correspond to various values of the reward accumulated over an attempt: (1) a category for finding each possible solution (60 for the first, 90 for the second, and 135 for the third using a solution multiplier of 1.5x), (2) a category for unlocking the door, and (3) a category for everything else (a reward lower than the other categories). We aggregate the counts of each category to examine how the agent learns over time.

For baseline conditions with only CC4 and CE4 trials, the RL model shows that it is able to find all 3 solutions, evidenced by the proportion of attempts in the *open 3* category. However, the proportion of attempts in the *open 3* category is lower than in *open 2* and *open 1* (the same is true for *open 2* and *open 1*). This indicates that the agent has a difficult time finding the second and third solution after finding the first, despite the higher reward of the second and third solution. Even worse, the agent finds fewer solutions as the agent trains more. Figure 6 shows the *else* category increasing as the attempt number increases, meaning that the agent is performing attempts that result in little to no reward *more* often in later training than in earlier training.

Figure 7 shows the results of the transfer trial from training on the 3-levers and transferring to the 4-levers. Models for both CE3-CC4 and CC3-CC4 executed solutions approximately 30% of all attempts. We note that the CE3-CE4 transfer case is slightly easier than the other cases; a solution was executed 42% of all attempts. CC3-CE4 transfer is harder than the other cases; a solution was executed only 16% of all attempts. In contrast, CE3-CE4 was the second hardest transfer case for humans; however, CC3-CE4 was also the most difficult case for humans. Overall, it appears the asymmetry between transfer cases is less pronounced in the RL's model compared to human performance.

These results suggest that while the RL model is able to uncover some knowledge about the mechanics behind the OpenLock task, the agent fails to form a useful abstraction between trials, both when the agent is transferring between congruent causal schema and incongruent causal schema. If the RL model was learning an abstract causal schema and applying it successfully, in the baseline results, we would expect to see the relative proportion of the *else* category to decrease while the relative proportion of the *unlock*, *open 1*, *open 2*, and *open 3* increased.

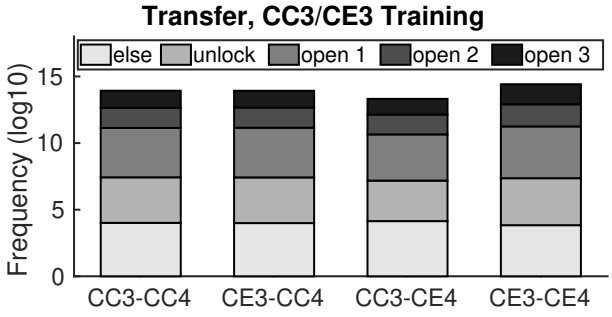


Figure 7: Transfer trial results of RL. The frequency of each reward category is plotted in log-scale; the number of attempts are the same in each transfer case. Note that CC3-CC4 and CE3-CC4 have nearly the same proportions while CC3-CE4 appears more difficult and CE3-CE4 is easier.

In the human results, we see a monotonically decreasing number of attempts (thereby monotonically increasing performance on the OpenLock task). In contrast, we see the RL model monotonically increasing the number of unsuccessful of attempts during baseline training (thereby monotonically decreasing performance). This result suggests that our DDQN agent is incapable of forming the abstract causal structure humans are implicitly or explicitly encoding. If our RL model was learning an encoding of the common causal structure between trials, we would expect the performance to increase over time.

These results suggest our DDQN agent is not forming the causal abstractions humans form. The different configurations of the levers only switch the location of each lever in the casual structure; once the position of each lever in the causal structure has been identified, an optimal agent can solve the task in two attempts in the 3-lever case or three attempts in the 4-lever case. In our experiments, DDQN is incapable of forming a policy that encodes this casual structure.

Discussion

Why is CE more difficult than CC? Human results show that the CE condition required a greater number of attempts in all cases. One potential explanation of this phenomenon relies on the ambiguity from environmental feedback after executing the first action. In the CC situation, the environment only changes if the first action is correct (*i.e.*, the agent pushes on the $L0$ lever). After pushing on $L0$ in the first action, the agent can then push either of the remaining active levers to unlock the door. Once the agent receives positive environmental feedback, it is less likely to make a mistake.

In contrast, if the first action is correct in the CE situation (pushing on $L1$ or $L2$), pushing on one of the remaining active levers is not guaranteed to unlock the door (*e.g.* if $L2$ is pushed after $L1$, the door remains locked). This introduces additional ambiguity to the agent after executing the correct first action. However, CE has two correct first actions in contrast to CC's single correct first action. While this makes the first action easier, we speculate that CC contains less ambiguity from the agent's planning perspective. Even though it is more difficult to select the correct first action, the environmental feedback from the CC's first action (*i.e.*, $L1$ and $L2$ not moving) provides more guidance than the environmental feedback from CE's second action. Additional experiments are needed to verify this hypothesis and will be conducted as future work.

Why is this task difficult for DDQN? The OpenLock environment presented here presents many challenges to traditional RL. First, the variation of the lever configurations of trials requires learning abstractions between configurations; each trial can be thought of as a different "game" with the same causal schema. DDQN was designed to learn singular games at a time rather than transfer knowledge *between* different games (Van Hasselt et al., 2016).

Second, the environment's state and action spaces are low dimensional and discrete. This results in a discrete and sparse reward function, which makes gradient descent difficult for DDQN. In contrast to most Atari games where random actions typically move the player (or perform another typically inconsequential action), exploratory mistakes in OpenLock are very common and almost always result in failing to open the door.

Third, state changes modify the underlying mechanics of the environment; *e.g.*, for CC trials, pushing on $L0$ unlocks $L1$ and $L2$. This is unlike traditional Atari games where the visual dynamics of the environment directly influence the reward function. While this maintains the Markov property assumed in Q-learning, it requires reasoning about the latent state space of the causal schema, which is not present in most Atari games.

Fourth, humans using an optimal policy must remember their previous solutions; *i.e.*, an optimal policy is non-Markovian. If humans were using a Markovian policy, their attempts to find the second and/or third solutions should be evenly distributed with the first solution found. However, many participants find all solutions within 2-3 attempts (finding two solutions in two attempts requires a lucky guess on the first attempt).

RL assumes the problem is Markovian and is therefore unable to remember the solutions already found. We relaxed this constraint by allowing the state space to be semi-Markovian; the number of solutions found was appended to the state space as a binary vector. However, empirically, this made no difference in performance to the fully-Markovian RL results. In fact, using any combination of the *unique solutions* reward function resulted in essentially no learning; after the agent finds a solution and takes the exact same action sequence again, they are given no reward. This means the agent only has one positive example per trial per solution, making it difficult to learn a meaningful policy during experience replay and gradient descent. However, future work should include an exploration into RL agents explicitly equipped with memory, such as a recurrent neural network (RNN). These agents may be better equipped to handle the long-term temporal constraints of finding all solutions.

What DDQN parameters can be learned from human participants? We fit an exponential decay model to human performance during the 6 training trials; this exponential decay is used to control the exploration vs. exploitation of DQN/DDQN agents. This regression shows humans are learning with a decay rate of 0.548 and 0.743 for the median and mean, respectively.

Epsilon decay parameters of 0.548 and 0.743 are extremely low; for the higher of the two settings, 0.743, the RL agent's exploration rate is less than one percent within 16 steps of the simulation. Typical RL epsilon decay settings are above 0.99 to allow exploration for thousands of simulation steps. These human-extracted parameter settings result in no meaningful learning for the RL agent. Instead, the RL agent adopts an uninformed, meaningless policy quickly and does not effectively explore the environment.

Future work should include a more thorough exploration of fitting a RL model to human performance data. Fitting such a model might yield additional insights into the differences between RL and human causal learning. Additional work should also include directly integrating causal models into RL. DDQN uniformly samples over the action space during exploration, regardless of prior experience. A Bayesian network could be learned simultaneously to the RL model and used to select more optimal explorations (*i.e.*, explorations that aid the most in identifying or refuting causal links in the Bayesian network). This could drastically improve the exploration process of DDQN.

Acknowledgment The authors thank Feng Gao, Chi Zhang, Prof. Keith Holyoak, and Prof. Ying Nian Wu at UCLA for their constructive discussions on causal inference and RL. The work reported herein was supported by DARPA XAI grant N66001-17-2-4029, ONR MURI grant N00014-16-1-2007, NSF grant BSC-1655300, and an NSF Graduate Research Fellowship.

References

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016). *Openai gym.*

Catto, E. (2011). *Box2d: A 2d physics engine for games.*

Cheng, P. W. (1997). From covariation to causation: a causal power theory. *Psychological Review*, *104*(2), 367-405.

Griffiths, T. L., & Tenenbaum, J. B. (2005). Structure and strength in causal induction. *Cognitive Psychology*, *51*(4), 334-384.

Griffiths, T. L., & Tenenbaum, J. B. (2009). Theory-based causal induction. *Psychological Review*, *116*(4), 661-716.

Heider, F. (1958). *The psychology of interpersonal relations*. Psychology Press.

Holyoak, K. J., & Cheng, P. W. (2011). Causal learning and inference as a rational process: The new synthesis. *Annual Review of Psychology*, *62*, 135-163.

Kubricht, J. R., Lu, H., & Holyoak, K. J. (2017). Individual differences in spontaneous analogical transfer. *Memory and Cognition*, *45*(4), 576-588.

Levine, S., Finn, C., Darrell, T., & Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, *17*(1), 1334–1373.

Lu, H., Yuille, A. L., Liljeholm, M., Cheng, P. W., & Holyoak, K. J. (2008). Bayesian generic priors for causal learning. *Psychological Review*, *115*(4), 955-984.

Mackie, J. L. (1974). *The cement of the universe: A study of causation*. Oxford.

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

ONeill, J., Pleydell-Bouverie, B., Dupret, D., & Csicsvari, J. (2010). Play it again: reactivation of waking experience and memory. *Trends in neurosciences*, *33*(5), 220–229.

Pearl, J. (2000). *Causality: Models, reasoning and inference*. Cambridge University Press.

Rescorla, R. A., & Wagner, A. R. (1972). A theory of pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. *Classical conditioning II: Current research and theory*, *2*, 64–99.

Shanks, D. R. (1991). Categorization by a connectionist network. *Journal of Experimental Psychology*, *17*(3), 433-443.

Shanks, D. R., & Dickinson, A. (1988). Associative accounts of causality judgment. In *Psychology of learning and motivation* (Vol. 21, pp. 229–261). Elsevier.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... others (2016). Mastering the game of go with deep neural networks and tree search. *nature*, *529*(7587), 484–489.

Sutton, R. S., & Barto, A. G. (1990). Time-derivative models of pavlovian reinforcement. In *Learning and computational neuroscience: Foundations of adaptive networks.*

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

Van Hasselt, H., Guez, A., & Silver, D. (2016). Deep reinforcement learning with double q-learning. In *Aaai* (Vol. 16, pp. 2094–2100).

Waldmann, M. R., & Holyoak, K. J. (1992). Predictive and diagnostic learning within causal models: asymmetries in cue competition. *Journal of Experimental Psychology: General*, *121*(2), 222-236.

Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine learning*, *8*(3-4), 279–292.