Behavioral optimization in a robotic serial reaching task using predictive information

Sen, D.; Kleijn, R. de; Kachergis, G.

Behavioral Optimization in a Robotic Serial Reaching Task using Predictive Information

Deniz Sen[1], Roy de Kleijn[2] and George Kachergis[3]

[1]Mathematical Institute, Leiden University
[2]Leiden Institute for Brain and Cognition, Leiden University
[3]Department of Psychology, Stanford University

***Abstract*—Prediction is a powerful approach to minimize errors and control problems in familiar environments and tasks. In human motor execution of sequential action, context effects can be observed, such as anticipation of or predictive movement towards target objects, where later subactions are affected by the execution of earlier subactions. In this paper, we present a simulation framework for a serial reaching task using a 4 DoF robotic arm to examine the learning of context effects in simulated robotic reinforcement learning agents. As we demonstrate, giving robotic agents access to predictive information about a future target object's identity results in motion optimization, where the identity of the next target modulates earlier subactions. Specifically, agents learn to anticipate and predict the location of the next target object, and move towards it before it appears, thus achieving higher rewards than agents that were not given predictive information.**

***Index Terms*—Prediction, sequential action learning, serial response time task, learning-based control, motion optimization**

I. Introduction

Human infants begin life unable to plan, speak, move around, or to pick up or hold toys. Yet within three to five months, infants learn to control their bodies well enough to at least reach out and grasp small toys with some success. This feat is impressive both because of the large number of degrees of freedom (DoF) in their body (e.g., one arm has 7 DoF), and because their visual system—including predictive knowledge of the world—is still developing. As infants mature, so does their ability to reach, grasp and interact with their environment. Their ability to perform complicated sequential actions becomes an integral part of their lives long before they have matured into adults. But exactly how do infants learn to control their bodies and manipulate objects, which require execution of order-dependent (sub)actions? While there is ample data on human sequence learning, testing computational models of learning to reach a sequence of objects in a simulated 3-dimensional environment allows for close comparison between how robots and humans learn, and may thus yield insight into human learning.

A. Sequence Learning

Sequence learning is a learning paradigm most widely studied within the field of cognitive psychology, in which the human ability to learn and perform context-dependent sequential actions is examined [1], [2], [3]. The ability to perform sequential action is an inherent part of the human ability to execute daily tasks. That is, most of our daily activities, from making coffee or riding a bicycle, are viewed as complex sequential actions, subject to a structured hierarchy [4], able to be adapted under changing circumstances (e.g., making a coffee with milk instead of regular coffee or taking a detour when riding a bicycle). Usually, dependencies between sub-actions arise. For example, the action of using a computer can be divided into the sub-actions (1) turning the computer on; (2) placing a hand on the mouse; (3) placing the other hand near the keyboard. While the order of (2) and (3) does not matter, it is clear that sub-action (1) should be performed before all other sub-actions. As such, sub-action (1) initializes the sequential action routine. The underlying cognitive mechanisms involved in sequential learning have been widely studied using the serial response time (SRT) task [5], [6], but how people make inferences about the dependencies during planning and execution of actions is not very well understood.

B. Mechanisms of Sequence Learning

Within the SRT literature, three main hypotheses try to explain the cognitive mechanism involved in sequence learning: (1) a stimulus-based hypothesis, (2) a response-based hypothesis, and (3) a stimulus–response (S–R) rule hypothesis, each mapping onto a different stage of cognitive processing. That is, a stimulus presentation and encoding stage, a response execution stage, and a response selection stage of information processing.

The stimulus-based hypothesis emphasizes that sequences are learned by forming stimulus–stimulus associations. Learning sequences is viewed as non-motoric, effector independent [7], [8] and purely perceptual [9], emphasizing the stimulus encoding stage of information processing [6]. The response-based hypothesis on the other hand proposes that sequence learning is not purely perceptual and highlights that motor components play a vital role. Both the response and the response location are important [10], [11], meaning that sequences are learned by forming response-response associations, thus implicating the response execution stage.

Lastly, the stimulus–response rule hypothesis states that stimulus–response associations in response selection need to

[1] The corresponding code repository can be found at: https://github.com/sendeniz/context-effect-robotic-sequence-reacher

be formed for successful sequence learning [12] emphasizing the link between perceptual and motor components. The stimulus–response rule can therefore be viewed as a combination of the stimulus-based and response-based learning hypotheses. Experiments with human participants have been interpreted as showing support for each of the three hypotheses in different scenarios, making it difficult to infer the underlying representations and mechanisms that people bring to bear on sequence learning tasks. However, we believe that research using computational models both offers a way to formally specify and to test such theories, and thus may provide insight into how structural dependencies in sequential learning develop in people.

C. Predictive Information

In any environment that is not purely stochastic, statistical regularities can provide information about the likelihood of possible future observations, and therefore future actions of positive expected value. Human learners are known to be sensitive to predictive information in a variety of domains, and often leverage it to produce smooth, efficient motor action. Consider the everyday context of making coffee or tea: if you want sugar and cream in your beverage, you may begin stirring the sugar in even as you reach for and pour the cream with your other hand. In other domains, learners also optimize their action execution by feedforward planning if predictive information is available, as shown by context effects such as anticipatory lip rounding (i.e., pronouncing /t/ in the world tulip) [13], anticipatory finger flexing in reaching tasks [14], or end-state comfort effects (i.e., when flipping a cup upside down, the is hand flipped prior grasping so to leave the hand at the end of the action in a comfortable resting state) [15], see [16] for a more complete overview of these context effects. Given the pervasiveness of context effects in human sequence learning, it seems reasonable to expect that any good model of biological learning would also show such sensitivity to predictive information.

D. Reinforcement Learning

One candidate family of algorithms comes from reinforcement learning (RL), a machine learning paradigm in which an agent iteratively optimizes their control policy for a given task through repeated interaction with a dynamic environment, which either rewards or punishes the agent's behavior. RL is a paradigm deeply rooted in psychological [17], [18] and neuroscientific studies [19] of animal behavior, where it is more commonly known as operant conditioning and is a type of associative learning process through which the strength or occurrence of an action is modified by reinforcement or punishment [18].

In both RL and operant conditioning, initially random actions are shaped into goal-oriented, purposeful behavior with the help of supervision signals (more commonly referred to as reward signals), which evaluate an agent's behavior during training. Learning occurs through interaction between the agent and the environment, where the supervision signal is made available indirectly in the form of rewards or punishments by the environment [20]. Informally, the agent's goal is to maximize the total amount of reward it receives. In order to achieve this goal, the agent should then not necessarily maximize the immediate reward, but maximize the cumulative long-term reward [21].

II. Related Work

The current simulation is fundamentally based on Nissen & Bullemer's serial response time (SRT) task [5]. In this task, participants are seated in front of a computer screen where stimuli will appear sequentially in one of four locations as shown in Figure 1, separated by an interstimulus interval (ISI). Whenever a stimulus appears, participants should press

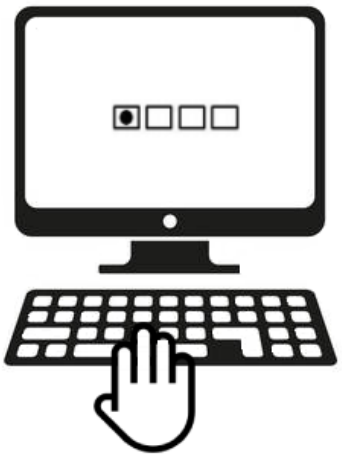

Fig. 1: Setup of a serial response task using human agents.

a corresponding button located directly below the stimulus. In their original task, the authors gave some participants a deterministic sequence of length 10, and others received a random sequence. The results showed that the use of a deterministic sequence leads to a significant decrease in response times in comparison to response times in the random sequence condition, suggesting that participants are predicting (whether implicitly or explicitly) the location of the next stimulus.

Kachergis et al. [22] modified this task into a trajectory task, having participants move their cursor to stimuli in an array instead of making discrete button presses, in order to better investigate their moment-to-moment movements in the task. In this paradigm, they found evidence for predictive curving towards the proximal response location before the next target location is revealed, and even as the cursor approached the location of the current target. De Kleijn et al. [23] replicated these findings and described these predictive movements in more detail, finding that participants who are given a deterministic sequence will optimize their behavior using predictive movements, while those given a random sequence tend to center their mouse cursor during the ISI—a strategy that is optimal given the stochastic environment, since it minimizes participants' expected distance (and thus response time) to the next target location. This predictive and centering behavior

was replicated in a 2D simulation [24] using a robotic arm. However, this simulation did not incorporate physics (friction, gravity), and used an evolutionary algorithm (EA) to train the neural controller.

A. Contributions

To address some short comings of related work, the current study (1) uses a more realistic 3D stimulation of an SRT task utilizing a torque based 4 DoF robotic arm. In addition to this, (2) we also propose a more plausible procedure with respect to human data on sequence learning compared to previous work. That is, while EAs mimic the process of biological evolution to produce organisms with skilled behavior by learning across multiple individual lifetimes [21], [25], human sequence learning occurs within an individual's lifetime, very possibly using reinforcement learning mechanisms. Lastly, due to the close relation of our study to previously gathered data on human participants, (3) we provide a comparison of optimization strategies that have emerged in human participants and our artificial agents under varying conditions and (4) examine which mechanisms are likely to be involved in the emergence of predictive motion optimization of sequential actions.

III. Method

We used a 3D framework for a fully articulated robotic arm performing a sequential reaching task akin to a serial response time task, allowing for close comparison of different control-based learning algorithms with human behavior using the *ML-Agents* toolkit for the Unity 3D engine [26]. In particular, we trained a two-jointed articulated robotic agent with 4-DoF on a four-target object reaching task using proximal policy optimization (PPO).

A. Proximal Policy Optimization

Proximal policy optimization (PPO) is a model-free on-policy reinforcement learning algorithm for continuous or discrete state–action spaces, which introduces a clipped surrogate objective to improve training stability by limiting the change between each policy update [27], [28]. Compared to "vanilla" policy gradients objective

$$L^{PG}(\theta) = \hat{\mathbb{E}}_t \left[\log \pi_\theta \left(a_t \mid s_t \right) \hat{A}_t \right] \tag{1}$$

, which utilizes the log probability of the decision-making policy $\log \pi_\theta \left(a_t \mid s_t \right)$ to trace the impact of an action a_t in a state s_t, PPO is similar to Trust Region Policy Optimization (TRPO), both replacing the log probability in equation 1 with the probability ratio $r_t(\theta)$ of the action under the current policy $\pi_\theta \left(a_t \mid s_t \right)$ and the old policy $\pi_{\theta_{\text{old}}} \left(a_t \mid s_t \right)$ in equation 2

$$r_t(\theta) = \frac{\pi_\theta \left(a_t \mid s_t \right)}{\pi_{\theta old} \left(a_t \mid s_t \right)} \tag{2}$$

The probability ratio in Eq. 2 is defined so that, if an action under the current policy is more likely than it is under the old policy, the probability ratio $r_t(\theta)$ will be greater than 1. Similarly, if an action under the old policy is more likely than under the current policy, the probability ratio $r_t \theta$ will be between 0 and 1. As a consequence, given a large probability difference between an action under the current policy and the old policy (i.e., let the action under the current policy be 100 times more likely than the action under the old policy), the probability ratio $r_t(\theta)$ can explode, leading to large gradient steps, which may cause the policy to go down a path of nonsensical unrecoverable action. Both PPO and TRPO account for this instability by adding constraints to their objective function.

The TRPO [29] objective

$$L^{TRPO}(\theta) = \hat{\mathbb{E}}_t \left[\frac{\pi_\theta \left(a_t \mid s_t \right)}{\pi_{\theta old} \left(a_t \mid s_t \right)} \hat{A}_t \right] = \hat{\mathbb{E}}_t \left[r_t(\theta) \hat{A}_t \right] \tag{3}$$

is subject to the trust region constraint, which ensures that the distance between old and current policy (measured by the KL-divergence) is within some value δ and helps to guarantee that the function is monotonically increasing. As such, TRPO mitigates instability by maximizing the objective function via the KL-Divergence, while PPO incorporates a similar constraint into the objective function via gradient clipping. PPO thus gains the same performance benefits as TRPO by optimizing a simpler, clipped surrogate objective

$$\underbrace{L^{CLIP}(\overset{\text{policy parameter}}{\overset{\downarrow}{\theta}})}_{\text{loss}} = \underbrace{\hat{\mathbb{E}}_t[\min(\underbrace{r_t(\theta)}_{\substack{\text{probability ratio of} \\ \text{current and old policy}}} \hat{A}_t, \overset{\text{gradient clipping}}{\overset{\downarrow}{\text{clip}}}(r_t(\theta), 1-\epsilon, 1+\epsilon)\overset{\text{advantage estimate}}{\overset{\downarrow}{\hat{A}_t}})]}_{\text{expectation over time}} \tag{4}$$

, where the the probability ratio $r_t(\theta)$ is clipped between $(1-\epsilon, 1+\epsilon)$.

B. PPO hyperparameters

When applying PPO on the network architecture with shared parameters for both policy (actor) and value (critic), the objective function is augmented with an error term on the value estimation and an entropy term to encourage sufficient exploration [30].

$$L(\theta) = \mathbb{E}\left[L(\theta^{CLIP}) - \underbrace{c_1 \left(V_\theta(s) - V_{\text{target}} \right)^2}_{\text{error term}} + \underbrace{c_2 H\left(s, \pi_\theta(.)\right)}_{\text{entropy term}} \right] \tag{5}$$

Therefore, through its objective, PPO introduces a number of hyperparameters to stabilize the training process. In particular it introduces control of bias–variance, control of exploration–exploitation, and constrains the divergence between old and new policy. The value coefficient $c_1 \geq 0$ controls the influence of the value estimate and the entropy regularization coefficient $c_2 \geq 0$, controls the stochasticity or the exploration–exploitation balance of an agent's actions. Larger values of the entropy curiosity parameter reflect more stochastic behavior, increasing exploration by discouraging premature convergence to suboptimal deterministic policies [30], [31].

The regularization parameter $0 \leq \lambda \leq 1$, used when computing the generalized advantage estimate (GAE) [32], controls the bias–variance trade-off. The GAE measures the degree to which an action is a good or bad decision given a certain state. As such λ trades variance by decreasing the weights of distant advantage estimates, in favor of bias towards earlier advantage estimates. Low values correspond to relying more on the current value estimate (which can be high bias), and high values correspond to relying more on the actual rewards received in the environment, which can be high variance

Lastly, the parameter ϵ, also referred to as the acceptable divergence threshold, constrains the possible divergence between old and new policy update to ensure that the agent does not jump into a policy that generates nonsensical actions. Small values will stabilize the training procedure, but will slow the training process.

IV. Experiment

In the current study, we studied context effects by measuring the Euclidean distance between the location of the robot hand and the active target as it becomes visible (at the end of the ISI), a measure used to determine predictive movement in SRT experiments with human participants [33]. We obtain a single measure by averaging results over 5 runs, one run consisting of 6×10^7 episodes and a single episode equal to $20,000$ timesteps. Our simulated robotic agent, task environment and sensors were implemented in Unity. We sampled an activation sequence for each of the four target objects from a uniform distribution and created three conditions: (1) a control condition in which agents have no access to predictive information, (2) a predictive information condition in which the agent's observation vector is expanded by the next target's one-hot coded identity, and (3) a sparse reward condition in which the agent is not encouraged for quick reaching, but still has access to predictive information in the same fashion the predictive condition does.

A. Setup

1) Robotic Arm: The artificial agent consisted of an arm with two actuators with a hand sensor, represented as a blue sphere attached to the agent, which could touch targets as shown in Figure 2. The robotic system has a total number of 6 DoF. The first actuator positioned at the shoulder, which is represented as a black marble consists of a ball joint with 3 DoF and the second actuator, positioned at the elbow, is also modeled as a ball joint and consists of 3 DoF. However, as the agent operates on a continuous action space of size 4, corresponding to torque applicable to the two joints, axial motion in each joint is ignored, resulting in 4 DoF.

2) Task Design: A representation of our reaching task is shown in Figure 3. The agent's shoulder joint is located above the center C with the center being equidistant to all of the four targets t_i with $i = 1 \ldots 4$. Target stimuli appear in a random order at each of the four target locations with an ISI of 500 timesteps between each successful touch, meaning that the next target will appear after 500 timesteps after successful touch of an active target. The target activation sequence is generated by sampling from a discrete uniform distribution $U(1, 4)$, with replacement, constrained so that an active target cannot be active twice in succession.

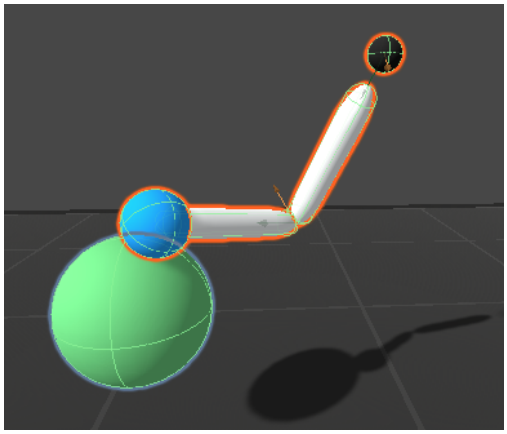

Fig. 2: The robot arm touching a target (green) with its hand (blue).

A successful touch of the active target is associated with a reward $r = 1.0$ at timestep $t = 0$, which decays by a factor of $\gamma = 0.992^t$ for every timestep t, resulting in a reward per timestep of $r_t = 1.0 \times 0.992^t$, resetting after the appearance of a new target. Curriculum learning is used to divide the training procedure in two lessons, which can improve speed of convergence and the quality of local minima obtained in non-convex optimization landscapes [34], [35]. In the first lesson, the agent is given a decaying reward $r_t = 1.0 \times 0.992^t$ as described above, with no movement penalty to avoid constraining the agents' exploration of the action space. After $4,000,000$ timesteps, the second lesson is initialized in which an additional penalty of $-0.001 \times$ the absolute distance of hand displacement is incurred for every time step to penalize the agent for making inefficient movements to encourage optimized target reaching. To avoid penalty accumulation, which can paralyze the agent during training, the final reward is clipped to $max(0.7, 1.0)$ across both lessons. This ensures that reward scores are on the same scale and both lessons are comparable. Furthermore, to further facilitate learning and encourage quick movement the agent received an additional reward equal to the difference between a chosen scalar (6) and the reaction time (RT) it took for a successfully reach. The scalar value was chosen based on the RT from preliminary experiments using the sparse reward condition (see Figure 7.), in which predictive movement did not occur as seen in Figure 4. To ensure that the additional reward is positive, we take the $\max(6 - RT, 0)$.

3) Neural Controller & Hyperparameters: The agents' neural controller was implemented using PyTorch and consists of a feedforward network with two fully connected hidden layers, each with 128 units. The input vector for both the control condition and the predictive information condition consisted of 37 elements with the following observations:

- position of both arm segments (2×3 elements);
- velocity of both arm segments (2×3 elements);

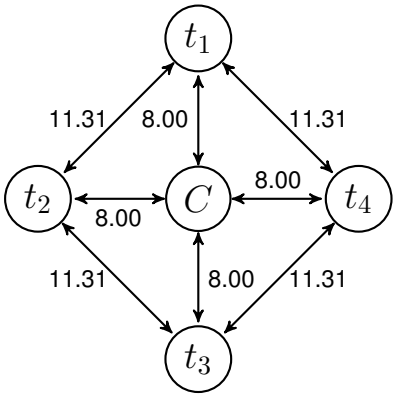

Fig. 3: Bird's-eye graphical representation of our reaching task in 2D Euclidean space, where C represents the anchor point of the shoulder joint, which is centered between the four target locations.

- angular velocity of both arm segments (2×3 elements);
- rotation of both actuators, represented as a quaternion (2×4 elements);
- position of the hand (3 elements);
- position of the goal (3 elements);
- target touched (1 element)
- control condition: zero padding (4 elements)
- predictive information condition: next target's hot-coded identity (4 elements)
- sparse reward condition: next target's hot-coded identity (4 elements)

To ensure that the observation vector of the control condition has the same length as the predictive information condition, the observation vector was zero-padded. The output layer of the network determines the torques to be applied to the actuators, leading to corresponding changes in the simulated environment. A swish activation function [36] (with $\beta_{swish} = 1.0$) is used, which has been shown to outperform ReLu under more complex optimization problems, varying batch size and different datasets.

Due to the inherently noisy distribution of the policy function, we use the Adam optimizer [37] with a constant learning rate schedule of 0.0001. To stabilize training, we break the temporal dependence of consecutive actions by defining a replay buffer [38] of size $102,400$ and randomly sample a mini-batch of size 512 observations from this replay buffer. To avoid the computational and time expenditure associated with training and fine-tuning network parameters of simulated robotic agents, we rely on values that have been shown to work well. Parameter settings for the networks are detailed in Table I.

TABLE I: Proximal policy optimization (PPO) hyperparameters used for training.

PPO Parameters	
learning rate α	0.0001
c_1	0.001
c_2	0.5
lambda λ	.992
epsilon ϵ	0.2
batch size	512
buffer size	102400

B. Results

Across multiple simulations, we consider 1) agents' mean distance to the active target at target onset (just after the ISI), 2) the agents' mean distance to the center (just after the ISI), 3) the mean cumulative reward achieved, and 4) in the predictive information condition, compared to the control and sparse reward condition. Figure 4 displays the mean distance of the agents' hands from the active target immediately after the end of the ISI across training across all three conditions.

An independent samples *t*-test showed that the Euclidean distance between the last $50,000$ epochs is smaller for the predictive information condition ($M = 7.23$) than for the control condition ($M = 9.87$), $t(18) = -11.05$, $p < 0.0001$. The difference between the control group ($M = 9.87$) and the sparse reward group ($M = 10.31$) is in comparison much smaller and not significantly different, $t(18) = -1.89$, $p = 0.075$. This shows that agents in the predictive information condition can learn to leverage that information about the environment, and begin to optimize their behavior by moving the arm toward the target they predict will become active.

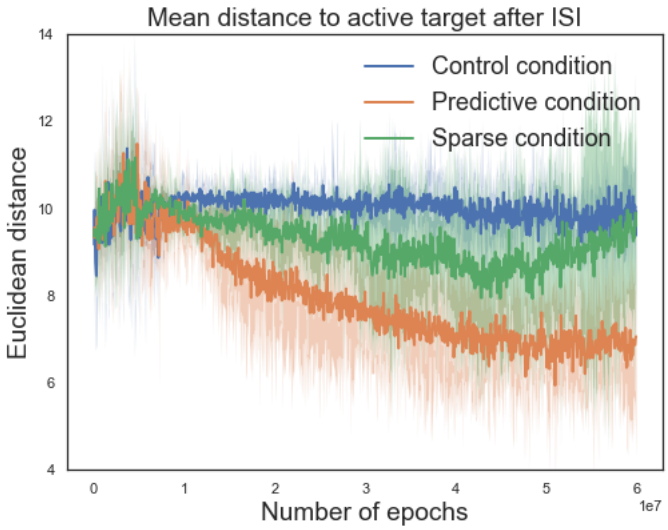

Fig. 4: Mean Euclidean distance to the active target at target onset, with shaded areas representing $2\times$ the standard deviation.

The mean Euclidean distance of the agents' hand to the center of all four target locations is illustrated in Figure 5. We compared the agents' distance to the center in the last 50,000 epochs of training, and found that there was no significant difference between the means of the control condition ($M = 5.94$), the predictive information condition ($M = 5.86$), or the sparse reward condition ($M = 5.74$), as shown by independent

t-tests (control vs. predictive condition: $t(18) = 0.89$, $p = 0.39$, control vs. sparse: $t(18) = 1.81$, $p = 0.09$, and predictive vs. sparse: $t(18) = 1.01$, $p = 0.33$.)

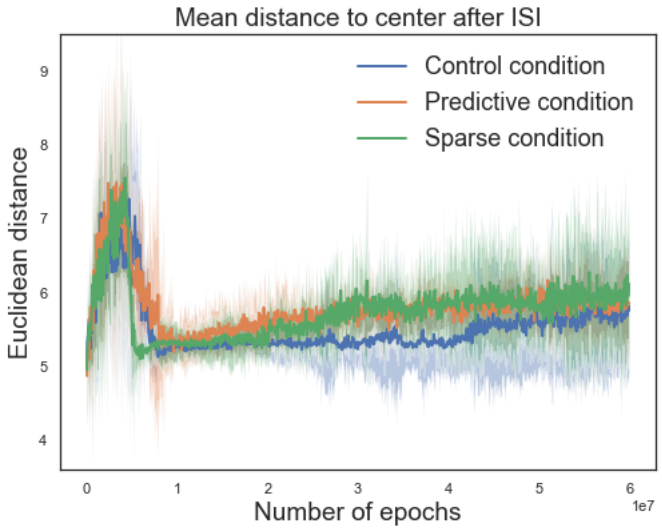

Fig. 5: Mean Euclidean distance to the center at target onset across all conditions, with shaded areas representing $2\times$ the standard deviation.

Figure 6 shows the mean cumulative reward achieved in the predictive information condition vs. the control condition across training. Although the agents in the control condition initially outperformed the agents given predictive information, which had to learn to effectively leverage the predictive information, eventually the agents given predictive information consistently outperformed those in the control condition. An independent samples *t*-test showed that the mean cumulative reward over the last 50,000 epochs is significantly higher for the predictive information condition ($M = 1.99$) than for the control condition ($M = 1.79$), $t(18) = -2.80$, $p = 0.015$).

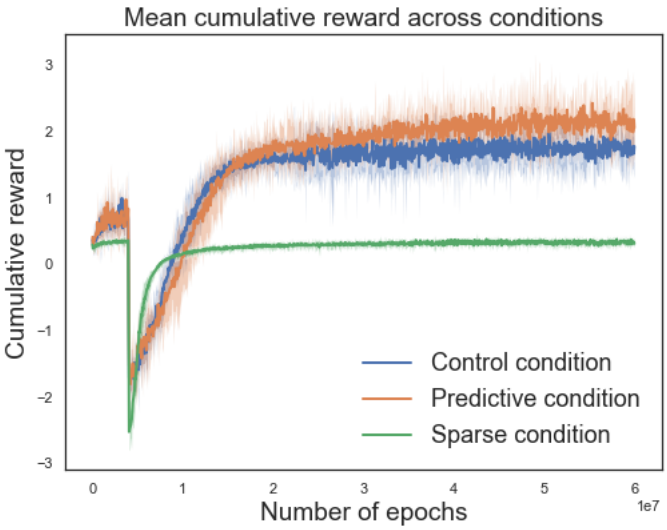

Fig. 6: Comparison of the mean cumulative reward across conditions, with shaded regions representing $2\times$ the standard deviation.

The reaction time across all three conditions is shown in Figure 7. Mean reaction time for the predictive condition ($M = 3.37$) and the control condition ($M = 2.87$) are similar, with the sparse reward condition having the highest mean reaction time ($M = 3.93$) averaged over the last $50,000$ epochs.

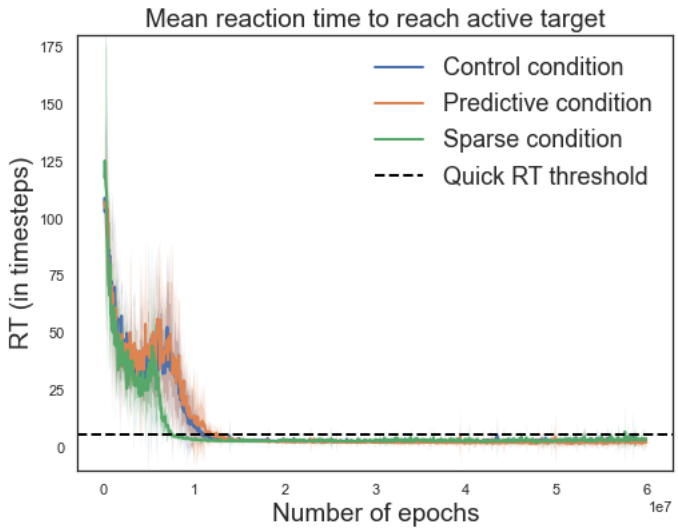

Fig. 7: Mean reaction time (RT) across all conditions, with shaded areas representing $2\times$ the standard deviation. The dotted black line represents the quick movement scalar 6.

To summaries, the predictive information conditions' shorter distances to the next target at the end of the ISI, this result suggests that agents in the predictive information condition are making faster responses. Thus, even-though all conditions are nearly equally as far from the center, we conclude that agents given predictive information are able to begin optimizing their behavior in a sequential reaching task.

V. Discussion

We find that giving agents access to predictive information (i.e., information about the next targets' identity) leads to a shorter mean distance to the active target after the ISI, which indicates that agents exhibit predictive motion optimization when reaching for targets. This is an interesting finding, as the agent must solve an associative learning task, in which a link between the stimulus identity (i.e., the next target identity), stimulus location (i.e., the target locations) and reaching responses need to be formed for successful motion optimization. Even-though, our experiments show that sequential reaching can emerge through simple reinforcement learning mechanism that reward responses that lead to successful reach of the presented stimulus in the control condition, predictive optimization seems to be modulated by knowledge about sequence order.

That is, as the agent is rewarded for responses that lead to the successful reach of the target stimulus, the agent can if predictive information is given further optimize for higher rewards by matching target identity to target locations, allowing the agent to move towards the target stimulus location

before its appearance during the ISI. This would mean that information about sequence dependencies modulate predictive behavior in sequential reaching, which is reflected in the predictive conditions' success in minimizing its distance to the active target right after the ISI.

Similar optimization strategies have been observed in human serial reaching tasks. Participants in a repeating (predictable) sequence condition were able to learn about the sequence order, which led to anticipatory hand movement to the target stimulus before its appearance and a minimized distance. In comparison, a random (unpredictable) sequence condition led to slower response times, and to a centering strategy which actually minimizes expected distance to the next target, given the stochasticity. While in the current study, agents in the predictive condition exhibited a faster RT and optimized reaching, agents that did not receive predictive information, in contrast did not learn a distinguishable optimal centering strategy: rather, agents in all three conditions remained similarly close to the center. A variety of reasons for this difference are possible, as the people in the trajectory SRT task are subject to the additional constraints of their elastic muscle system as well as the potential need to re-center the mouse on the mousepad. Nonetheless, the PPO-based RL agents given predictive information show many of the same behavioral characteristics of humans, making it a valuable platform for detailed comparisons of human and RL-based sequential action learning.

To summarize, we find that given 1) that sequential reaching can emerge in the control condition, where responses which approximate successful reach of appearing target stimuli are rewarded, that stimulus-response association may explain the development of sequential reaching and 2) that contextual information about sequence ordering can modulate the expression of these stimulus-response associations, leading to optimized reaching similar to humans. As a result, it would seem that agents stores some form of stimulus-response associations in its weight matrix, suggesting that the stimulus-response associations may be sufficient to explain predictive behavior in sequential reaching, although future work will have to explore whether more sophisticated representations are warranted.

A. Limitations

The present study diverged from human studies in a number of ways, which may be explored in future research. In the current study, predictive information was provided to the agent by an oracle, rather than being learned through repeated exposure to a static sequence of target locations, as humans were given in prior work. Future studies should examine whether RL agents given a static, repeating sequence of locations could still learn to leverage this predictive information—and in a random sequence condition, whether they could perhaps learn to converge on an optimal re-centering strategy. We propose that such computational work, along with detailed comparisons to human learning in similar tasks with sequences of varying length and transition probabilities between locations, will lead to a better understanding of how memory and learning constraints lead to context effects in sequential action, both in people and in robots.

VI. Conclusion

In this study we implemented a sequential reaching task in a simulated 3D environment with a robotic arm controlled by a neural controller, trained using PPO, a state-of-the-art deep reinforcement learning algorithm. We found that providing the agent with predictive information (e.g., information about the next target) led to the agent being closer to the target stimulus and thus achieved higher rewards than agents without predictive information. Thus, we conclude that predictive information can allow RL agents to exhibit predictive optimization of reaching behavior in a sequential reaching task and suspect that stimulus-response associations are involved in its development.

Acknowledgment

We would like to thank Aske Plaat, Wojtek Kowalczyk, Zhao Yang, Paul Peters and Daniel Klassen for their support, time and thoughts.

References

[1] R. P. Cooper and T. Shallice, "Hierarchical schemas and goals in the control of sequential behavior." *Psychological Review*, vol. 113, no. 4, p. 887–916, 2006.
[2] M. F. Washburn, *Movement and mental imagery*. Boston, MA: Houghton Mifflin, 1916.
[3] K. S. Lashley, "The problem of serial order in behavior," in *Cerebral mechanisms in behavior*, L. A. Jeffress, Ed. New York: Wiley, 1951, pp. 112–136.
[4] M. Botvinick and D. Plaut, "Doing without schema hierarchies: a recurrent connectionist approach to normal and impaired routine sequential action," *Psychological Review*, vol. 111, 2004.
[5] M. J. Nissen and P. Bullemer, "Attentional requirements of learning: Evidence from performance measures," *Cognitive Psychology*, vol. 19, no. 1, pp. 1–32, 1987.
[6] H. Schwarb and E. Schumacher, "Generalized lessons about sequence learning from the study of the serial reaction time task," *Advances in Cognitive Psychology*, vol. 8, no. 2, p. 165–178, 2012.
[7] W. B. Verwey and B. A. Clegg, "Effector dependent sequence learning in the serial rt task," *Psychological Research*, vol. 69, no. 4, p. 242–251, 2004.
[8] A. Cohen, R. I. Ivry, and S. W. Keele, "Attention and structure in sequence learning," 2004.
[9] J. H. Howard, S. A. Mutter, and D. V. Howard, "Serial pattern learning by event observation." *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 18, no. 5, p. 1029–1039, 1992.
[10] D. B. Willingham, "Implicit motor sequence learning is not purely perceptual," *Memory Cognition*, vol. 27, no. 3, p. 561–572, 1999.
[11] D. B. Willingham, L. A. Wells, J. M. Farrell, and M. E. Stemwedel, "Implicit motor sequence learning is represented in response locations," *Memory Cognition*, vol. 28, no. 3, p. 366–375, 2000.
[12] N. Deroost and E. Soetens, "The role of response selection in sequence learning," *Quarterly Journal of Experimental Psychology*, vol. 59, no. 3, p. 449–456, 2006.
[13] F. Bell-Berti and K. S. Harris, "Anticipatory coarticulation: Some implications from a study of lip rounding," *Journal of the Acoustical Society of America*, vol. 65, pp. 1268–1270, 1979.
[14] M. Jeannerod, M. Arbib, G. Rizzolatti, and H. Sakata, "Grasping objects: the cortical mechanisms of visuomotor transformation," *Trends in Neurosciences*, vol. 18, no. 7, pp. 314–320, 1995.
[15] R. G. Cohen and D. A. Rosenbaum, "Where grasps are made reveals how grasps are planned: Generation and recall of motor plans," *Experimental Brain Research*, vol. 157, pp. 486–495, 2004.
[16] R. De Kleijn, G. Kachergis, and B. Hommel, "Everyday robotic action: lessons from human action control," *Frontiers in Neurorobotics*, vol. 8, p. 13, 2014.
[17] E. L. Thorndike, *Animal intelligence; experimental studies*. New York,The Macmillan Company,, 1911.
[18] B. Skinner, *The Behavior of Organisms: An Experimental Analysis*. B. F. Skinner Foundation, 1938.

[19] W. Schultz, P. Dayan, and R. P. Montague, "A neural substrate of prediction and reward," *Science*, vol. 275, pp. 1593–1599, 1997.
[20] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*. The MIT Press, 2006.
[21] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. The MIT Press, 2018.
[22] G. Kachergis, F. Berends, R. de Kleijn, and B. Hommel, "Human reinforcement learning of sequential action," in *Proceedings of the 38th Annual Conference of the Cognitive Science Society*, 2016, pp. 193–198.
[23] R. de Kleijn, G. Kachergis, and B. Hommel, "Predictive movements and human reinforcement learning of sequential action," *Cognitive Science*, vol. 42, no. S3, pp. 783–808, 2018.
[24] R. de Kleijn, G. Kachergis, and B. Hommel, "Optimized behavior in a robot model of sequential action," in *Proceedings of the 40th Annual Conference of the Cognitive Science Society*, 2018.
[25] D. Fogel, "An introduction to simulated evolutionary optimization," *IEEE Transactions on Neural Networks*, vol. 5, no. 1, pp. 3–14, 1994.
[26] A. Juliani, V. Berges, E. Teng, A. Cohen, J. Harper, C. Elion, C. Goy, Y. Gao, H. Henry, M. Mattar, and D. Lange, "Unity: A general platform for intelligent agents," *arXiv preprint arXiv:1809.02627*, 2020.
[27] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017.
[28] C. C.-Y. Hsu, C. Mendler-Dünner, and M. Hardt, "Revisiting design choices in proximal policy optimization," 2020.
[29] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel, "Trust region policy optimization," 2017.
[30] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," *CoRR*, vol. abs/1602.01783, 2016.
[31] R. J. Williams and J. Peng, "Function optimization using connectionist reinforcement learning algorithms," *Connection Science*, vol. 3, no. 3, pp. 241–268, 1991.
[32] J. Schulman, P. Moritz, S. Levine, M. I. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," in *Proceedings of the 4th International Conference on Learning Representations, ICLR 2016*, Y. Bengio and Y. LeCun, Eds., 2016.
[33] R. Dale, N. D. Duran, and J. Morehead, "Prediction during statistical learning, and implications for the implicit/explicit divide," *Advances in Cognitive Psychology*, vol. 8, pp. 196 – 209, 2012.
[34] R. De Kleijn, D. Sen, and G. Kachergis, "A critical period for robust curriculum-based deep reinforcement learning of sequential action in a robot arm," *Topics in Cognitive Science*, 2022.
[35] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 41–48.
[36] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *CoRR*, vol. abs/1710.05941, 2017.
[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.
[38] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," *CoRR*, vol. abs/1602.01783, 2016.