Yoking-Based Identification of Learning Behavior in Artificial and Biological Agents

Manuel Baum[1,2][0000−0002−3083−0887], Lukas Schattenhofer[1][0000−0003−1514−902X], Theresa Roessler[3][0000−0002−2403−8375], Antonio Osuna-Mascaró[3][0000−0002−6954−6453], Alice Auersperg[1,3][0000−0001−7405−9791], Alex Kacelnik[1,4][0000−0002−3188−8255], and Oliver Brock[1,2][0000−0002−3719−7754]

[1] Science of Intelligence, Research Cluster of Excellence, Marchstr. 23, 10587 Berlin*
[2] Robotics and Biology Laboratory, Technische Universität Berlin
{baum,oliver.brock}@tu-berlin.de
[3] Comparative Cognition Group, University of Veterinary Medicine Vienna
{theresa.roessler,alice.auersperg}@vetmeduni.ac.at
[4] Behavioural Ecology Group, University of Oxford, England
alex.kacelnik@zoo.ox.ac.uk

Abstract. We want to understand how animals can learn to solve complex tasks. To achieve this, it makes sense to first hypothesize learning models and then compare these models to real biological learning data. But how to perform such a comparison is still unclear. We propose that yoking is an important component to such an analysis. In yoking, two agents are made to experience the same inputs, rewards or perform the same actions – possibly in combination. We use yoking as an analytical tool to identify the algorithm that drives learning in a target agent. We evaluate this approach in a synthetic task, where we know the ground truth learning algorithm. Then we apply it to biological data from a physical puzzle task, to identify the learning algorithm behind physical problem solving in Goffin's cockatoos. Our results show that yoking works, and can be used to identify the target algorithm more reliably, with less variance and assumptions, than a more unconstrained approach to identify learning algorithms.

Keywords: Learning · Yoking · Off-policy · Reinforcement learning

1 Introduction

Behavioral biology aims to understand how animals learn to solve novel tasks. AI wants to use such knowledge to build general artificial agents. To gain insights into the mechanisms underlying biological learning, we can compare observed behavior to that of artificial agents for which we know the learning method.

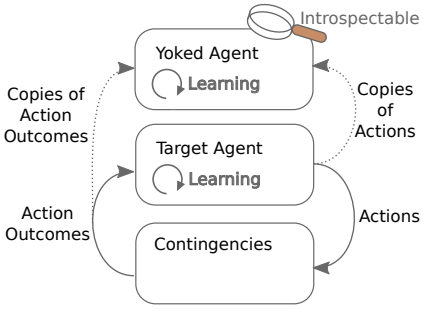
* Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2002/1 "Science of Intelligence" – project number 390523135.

Agreement between learned behaviors serves as evidence for agreement in the mechanisms that drive behavioral adaptation. But to identify behavioral agreement requires meaningful comparisons of the behavioral trajectories of several agents. We will discuss why this is a challenging problem and show how *yoking*—an experimental technique from behavioral biology—enables such comparisons.

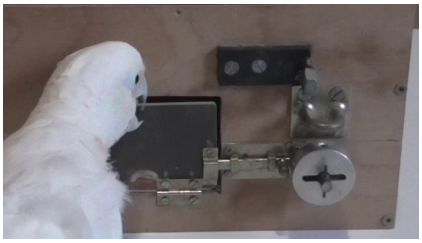
Behavioral trajectories can exhibit substantial variability. Reasons for this include inter-individual differences in embodiment, knowledge, and skills as well as variability in the environment. As a result, the space of action sequences and learning trajectories is extremely large, already for moderately complex tasks and environments. Two learning trajectories may explore entirely different regions in this space, even when they are generated by the same learning algorithm. To gain insights into biological behavior, we therefore must find ways to perform meaningful comparisons in spite of this variability.

We propose a method for identifying learning behaviors based on a set of hypothesized model algorithms, even when behavioral trajectories are sampled from very large spaces. We assume that the agent whose learning behavior we seek to understand (usually a biological agent we call the *target agent*) follows some learning method we would like to identify. The target agent produces behavioral trajectories that sample only a small, but highly relevant subset of all possible behaviors. To identify the mechanisms implemented in the target agent, we compare its behavior to that of a set of alternative agents, each implementing a different learning method. To effectively compare these agents, we *yoke* the other agents to the target agent. This leads to meaningful comparisons, identifying learning models that resemble the target agent's learning behavior.

The term *yoking* originates in behavioral biology. In the classical yoking setting, two animals are aligned so that both experience the outcomes of the



(a) A yoked, introspectable agent receives copies of a target agent's actions and action outcomes. Learning in the yoked agent is evaluated as model for learning in the target agent.



(b) A Goffin's cockatoo opens a baited mechanical puzzle, called lockbox. The lockbox consists of a cashew reward behind an acrylic door, a metal bar blocking that door, and a metal disk blocking the metal bar.

Fig. 1: We evaluate and apply the yoking experimental paradigm with to identify the learning mechanisms that enable animals to solve complex problems.

behavior of just one of them (the target agent). This serves to isolate the role of contingent behavior from the effect of receiving rewards. Whereas it is difficult to yoke perception and actions of biological agents, we have full control over synthetic models. We can enforce their experience to align with respect to rewards, percepts, and actions. Figure 1a shows a schematic of this approach. Yoking causes the artificial agent to mimic the behavior of the target agent, effectively confining learning to plausible behavioral trajectories, rendering the comparison meaningful in spite of large unexplored regions of the behavioral space.

We develop and evaluate a methodology for yoking-based identification of learning algorithms. We apply the yoking paradigm to learning a physical puzzle task, the lockbox depicted in Figure 1b. To solve this task, an agent must open several mechanical locks in sequence to obtain a reward. We evaluate the proposed method in two sets of experiments. First, we evaluate the yoking approach by successfully identifying the learning algorithm of an artificial target agent for which we know the true learning algorithm. The yoking-based comparison between the target agent and other artificial agents successfully identifies the learning method of the target agent from a set of candidate models. Our experiments show that yoking does this with fewer assumptions and less variance than a more unconstrained approach. We apply yoking to data obtained with real-life cockatoos performing the same task. Our results show that yoking is an important tool in identifying learning behavior in artificial and biological agents.

2 Related Work

We discuss applications of yoking in biological learning experiments and then describe recent applications based on machine learning.

In the biological learning literature, the dominant application of yoking is as a "yoked control." Here, two animals are placed inside identical skinner boxes, where both animals can act and perceive their environments independently, however, they are yoked with respect to the rewards they receive. The release of rewards in both boxes is contingent only on the actions of the *target* animal. This setup reveals if changes in behavior are due to operant conditioning or purely due to a changing frequency of reward or punishment [10]. Yoked control has been criticized because it can bias results towards conclusions in favor of the operant conditioning hypothesis [5]. It seems the main source for bias in yoked controls are individual differences [3]. In this paper we use simulated experiments to evaluate the effect that inter-individual differences have on the yoking procedure. In contrast to the classical application of yoking as a control, we use it with the goal to directly identify a target learning algorithm.

Yoking an animal to another animal's rewards is simple, but yoking perceptual inputs is more difficult. The classical kitten carousel experiment [6] is an example of yoking two animals together so that they get the same sensory inputs. Besides yoking through mechanical linking, powered mobility devices could also be used to yoke two subjects perceptually [1].

To yoke the perception of synthetic agents to real animals, we must know the percepts of the animal. An interesting approach to this problem is a controlled rearing approach, in which newborn chicks are motion-tracked and raised in a mostly virtual environment [13]. This setting was used to train artificial neural networks on the same visual input that the birds received [7].

It is practically impossible to yoke two animals such that they perform the same *actions*. But this can be easily done if the yoked agent is a computational model. Recently, a so-called tandem learning setting was used to yoke deep Q-Learning agents to one another [9]. This was done to analyze the difficulty of off-policy learning problem. Off-policy learning means that an agent learns using data that was not generated by the behavior it executes to solve the task, but by another behavior. Results showed that difficulties in off-policy learning mainly stem from the use of non-linear function approximators and the inherent misfit of target agents' data distributions to yoked agents policies. As it turns out, this insight is not only relevant to machine learning, but also to biological research. The yoked experimental design necessarily involves off-policy learning. And because we likely need complex, non-linear learning models to explain learning in complex animals, biological analysis needs to be aware of the challenges involved in the off-policy learning problem. In this paper we circumvent this issue by using tabular models that are capable of off-policy learning.

There are also other approaches in machine learning where a target agent, potentially human, is copied by another learning agent. Behavioral cloning [12], inverse reinforcement learning [8] and generally the learning-from-demonstration setting [2] are related areas of research. However, the common goal in those cases is to copy the target agent and to achieve high reward, not to identify the target agent's learning algorithm.

3 Comparing Synthetic Learning Models to Target Data

We want to understand how animals learn to solve novel problems. Our approach is to compare their learning to the adaptation of artificial learning models in similar settings. Because the artificial models are introspectable, we learn which mechanisms in artificial models are most likely to explain the observed biological learning. We use two different ways to perform the required comparisons: one in which the behavior of the agent is yoked and one in which it is unconstrained.

3.1 Problem formalization

We assume that the underlying learning problem can be formalized as reinforcement learning (RL) problem on a discrete Markov Decision Process (MDP). A discrete MDP is a four-tuple (S, A, T, R) that consists of a set of states S, a set of actions A, a probabilistic state-transition function $T = p(s'|s, a)$ and a reward function $R(s, a)$. The probabilistic state-transition function T captures the probability that an action a will cause the system to transition from state s to state s'. The goal of the reinforcement learning problem then is to learn

a policy $\Pi(s) = p(a|s)$ that maximizes the expected reward in the MDP. The problem this paper tackles is the question: Given sequences of actions that a reinforcement learning agent performed as it learned to solve the task, can we identify the ground truth learning algorithm that adapted the behavior?

To answer this question we assume an episodic RL setting, where the target agent acts in the MDP during a sequence of m sessions $D_1, ..., D_m$ (episodes). In each session the agent performs n_m actions $\boldsymbol{a_m} = a_m^1 ... a_m^n$ using its policy Π_m^* which is adapted after each session using the ground truth learning algorithm. As we generally do not have access to the ground truth policy P^* or its output distribution over actions $p_m^*(a_m)$, we can only infer the learning algorithm from the actions $\boldsymbol{a}$ that were performed. Our approach is to compare the actions performed by the target policy Π^* to the output distributions p_m^k of k different policies Π^k that are each adapted by a different candidate learning algorithm. For this comparison we use the Sørensen–Dice similarity [4] between each candidate policy's posterior over actions p_m^k and a categorical distribution fit to the actions performed by the ground truth policy. Given a set of candidate learning algorithm models, this lets us compute a per-session similarity measure between the policy adapted by the target learning algorithm and several candidate policies – each adapted by a different candidate learning algorithm.

3.2 *Unconstrained* approach to identify learning algorithms

A straightforward approach to this problem is to simulate the MDP using the state-transition function $T = p(s'|s, a)$ and to choose actions based on the candidate policy Π^k. This yields sequences of actions and rewards that can be used to adapt the policy using the candidate learning algorithm in between sessions.

But this approach has drawbacks. The first drawback is that modelling errors accumulate, as the actions performed by the agent depend on the changes made by the learning algorithm, and those changes, in return, depend on the performed actions. If either the simulation (state-transition function T) or the candidate model for the learning algorithm are not exactly correct, errors accumulate significantly over time. The simulation and the learning algorithms are models, so in any case both will be at least slightly wrong. The accumulated errors will lead to high variance not only in the outcomes of the simulations, but also in the metric used to compare those data to the actions performed by the target agent. The second drawback is that we need a full model of the domain in which the behavior is observed. Especially a state-transition function T can be challenging to obtain in contact-rich, real world scenarios.

3.3 *Yoked* approach to identify learning algorithms

High variance and the need for a state-transition model are challenging problems that can be avoided with a yoking-based approach. The idea of yoking is simple, yet powerful. Instead of using a full simulation to generate actions, we can instead directly use the actions $\boldsymbol{a^*}$ that were performed by the target policy Π^* and align the state to follow the same trajectory as encountered by the ground-truth agent.

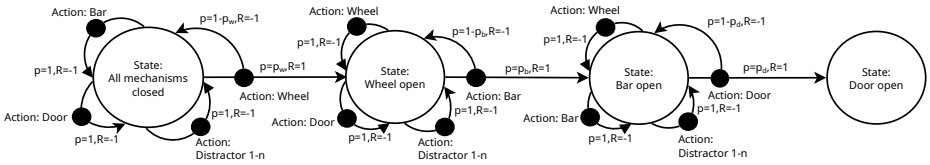


Fig. 2: The lockbox task, modelled as a finite Markov Decision Process with four discrete states. Actions with their probabilistic outcomes and rewards are depicted as branching arrows.

Using this approach, we cannot avoid *all* modeling aspects that come with an MDP model. We still need a set of states S, a set of actions A and a reward function R. However we can avoid modelling the state-transition function T. This approach significantly reduces variance as it avoids randomness introduced by the simulation. Instead, the yoked model is closely aligned to the target agent.

4 Evaluation in a Synthetic Lockbox Task

To assess the performance of both approaches we need an experiment where the ground-truth learning algorithm is known. Thus, we implemented a simulated experiment, similar to the biological one we will use for evaluation in Section 5. In this environment we simulate several learning agents and assess how well each of the two approaches can identify the ground truth learning algorithm.

The task we consider is the physical puzzle depicted in Figure 1b, called a lockbox. This lockbox represents a sequential manipulation problem where the agent has to first remove a metal disk, so that it can then push a bar to the side, which makes it possible to finally open a door and retrieve a reward. This lockbox task can be modelled as a dicrete MDP with four states and three actions shown in Figure 2. The transition probabilities depend on the agent's mechanical skills p_w, p_b, p_d to open the wheel, bar and door respectively. The reward function yields a reward of 1 for every action that could successfully open a lock, and -1 otherwise. Importantly, opened locks can not be closed again.

We consider five different candidate learners. Two of these learners use tabular Q-Learning [11], but with different learning rates. The model *QLearn (slow)* uses a learning rate $\alpha = 0.1$ and the model *QLearn (fast)* uses a learning rate $\alpha = 0.9$. Two other models follow a custom learning algorithm, we call *Myopic RL*. For each state, this model has a parameter vector ω with as many elements as the number of available actions. In each state, actions are sampled from the categorical distribution $\sigma(\omega)$, where $\sigma(.)$ is the soft-max function. Whenever this agent successfully performs an action i to change the state of the lockbox, then the respective entry $\omega_\mathbf{i}$ is increased by a constant amount β. The two *Myopic RL* learners differ in this learning rate β, where *Myopic RL (slow)* uses $\beta = 1$ and *Myopic RL (fast)* uses $\beta = 100$. Finally, we also use a baseline algorithm which implements no learning whatsoever, called *No Learning*.

4.1 Identifying known ground truth learning algorithms

We cannot use biological data to compare the performance of the *yoked* and *unconstrained* approach, as we don't know what the ground-truth learning algorithm is behind real animals' learning. Thus we will simulate the aforementioned algorithms as the ground-truth learning algorithms and evaluate which of the approaches identifies the target algorithm more reliably.

To generate the ground-truth learning data, we simulated each of the algorithms 32 times on a synthetic lockbox experiment. This lockbox experiment had three states and three relevant actions, as described above, however we added seven additional distractor actions that do not have an effect on the state. The initial action selection probabilities $p(a|s)$ for each policy were sampled from a Dirichlet distribution with $\alpha_1^{dir} = \ldots = \alpha_{10}^{dir} = 1.0$, and blackbox optimization was used to find corresponding parameters for *QLearn* and *Myopic RL* that map to such an initial distribution. We constrained learning to 12 sessions of maximally 200 actions each. As can be seen in Figure 3, both *Myopic RL* conditions can learn to solve the task, *QLearn (fast)* also shows adequate performance while *QLearn (slow)* manages to only slightly outperform *No Learning* with the very restricted number of actions and sessions. The expected number of required actions for a perfect agent would be 30, given the mechanical skill setting of 0.1.

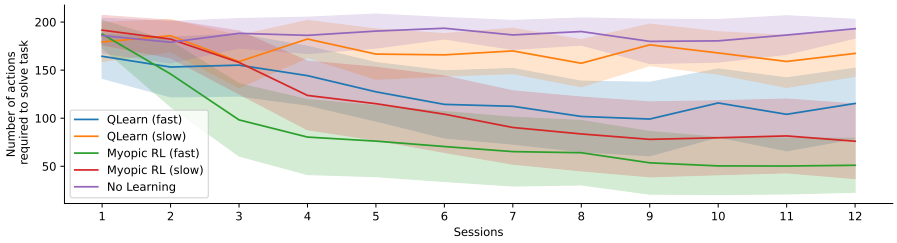


Fig. 3: Mean and $\frac{1}{2}$ standard deviation of the learning curves for different agents in a simulated lockbox task. The Myopic RL agent learns the task the quickest on average, with both sets of parameters. Q-Learning also learns the task with high learning rate, while the learning rate of *QLearn (slow* is too small to learn the task reliably in 12 sessions. Each model was simulated 32 times, sessions limited to a length of 200 actions and the mechanical skill was set as $p_w, p_b, p_d = 0.1$.

Next we perform a comparison of the *yoked* and *unconstrained* approach. We use each of the five previously simulated models individually as the ground truth target algorithm to be identified. In that identification problem, all five models also serve as a candidate model that may be the underlying ground truth algorithm. So, for example, when *QLearn (fast)* is the target, all 5 learning models are used as hypotheses and the goal would be to identify that *QLearn (fast)* is indeed the most probable learner behind the observed changes in behavior.

For each of the 32 runs per target algorithm, we either yoke or simulate each candidate algorithm ten times. These ten models are initialized so that they follow the same distribution of actions as the target algorithm in its first session. We cannot directly use the probabilities $p(a|s)$ of the target algorithm's policy, as in a setting where the target agent is biological we would not have access to this information. Instead we use the relative frequency of actions the target agent performed and, again, use blackbox-optimization to find parameters for the candidate models that yield a $p(a|s)$ according to these frequencies. After this initialization, agents in the *unconstrained* condition are simulated and learn independently, while agents in the *yoked* condition learn from the same actions and receive the same action outcomes as the target model.

Next, we assess which of the agents best explain the target agent's behavior. The following procedure is applied to all pairs of target models and candidate learning models, irrespective of how the candidate models were trained. The similarity measure is computed per session.

For each of the 32 instances of the target agent, there are ten *yoked* and ten *unconstrained* candidate model executions. For each instance i of the target agent, we fit a categorical distribution $\mathbf{p}_i$ its performed actions. Then we compare this distribution to the known action distribution $\mathbf{p}_{ij}(a|s)$ of the $j \in [1, \ldots, 10]$ candidate models using Sørensen–Dice similarity [4]. For each instance of the target agent, this yields ten similarity scores for which we compute mean μ_i and standard deviation σ_i. Finally, to ensure statistical support, we average these statistical moments over the 32 instances such that $\overline{\mu} = \mathbb{E}_i(\mu_i)$ and $\overline{\sigma} = \mathbb{E}_i(\sigma_i)$.

Figure 4 shows this statistic of scores when it is applied session-wise to all pairs of target models and candidate models. The data shows that the *yoking* approach is superior to the *unconstrained* approach. *Unconstrained* suffers from higher variance and reveals the ground-truth algorithm less clearly than *yoking*. In the *yoked* condition, the ground truth algorithm is almost at all points the most likely (topmost) hypothesis. The results are most distinctive after the first few sessions when the learning algorithms could actually take effect, and before the last few sessions where those algorithms that learn the solution already converged to that very same solution.

4.2 Comparison over size of action space

We compare the analytical performance of the *yoked* and *unconstrained* approaches when we vary the size of the action space. In the analysis in the previous subsection, we increased the number or possible actions to ten, by introducing seven actions without effect. In Figure 5, we assess the performance of the approaches when we vary the number of additional, effect-less actions to between 0 and 17. We measure analytical performance of either approach as follows. First, we compute the probability to identify the correct algorithm on a per-session basis. We do this by computing the probability $p_c^i = p(m_t \geq m_0 \wedge \ldots \wedge m_t \geq m_i)$ that the correct model is the most probable explanation for each session i, using monte-carlo inference based on the distributions $\mathcal{N}(\overline{\mu}, \overline{\sigma})$ described above. Then we average these session-wise scores into an overall score. Figure 5 shows that

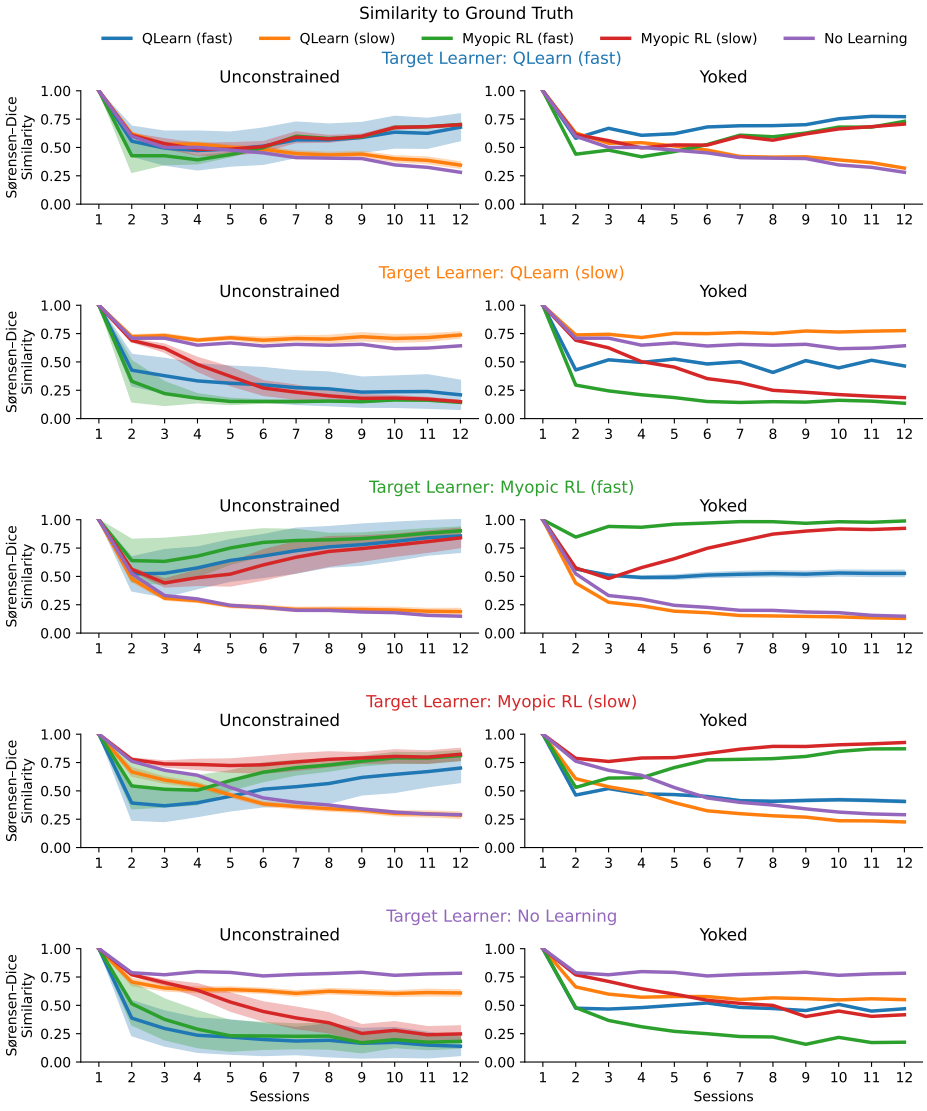


Fig. 4: A comparison of the *unconstrained* approach (left column) to the *yoked* approach (right column). For all five target algorithms, the *yoked* approach identifies the target algorithm as the most likely one (it is the topmost line). It does so with less variance and a larger margin than the *unconstrained* approach. Yoking can even differentiate variants of *QLearn* and Myopic RL with different learning rates. In contrast, the *unconstrained* condition suffers from higher variance which makes conclusions more difficult.

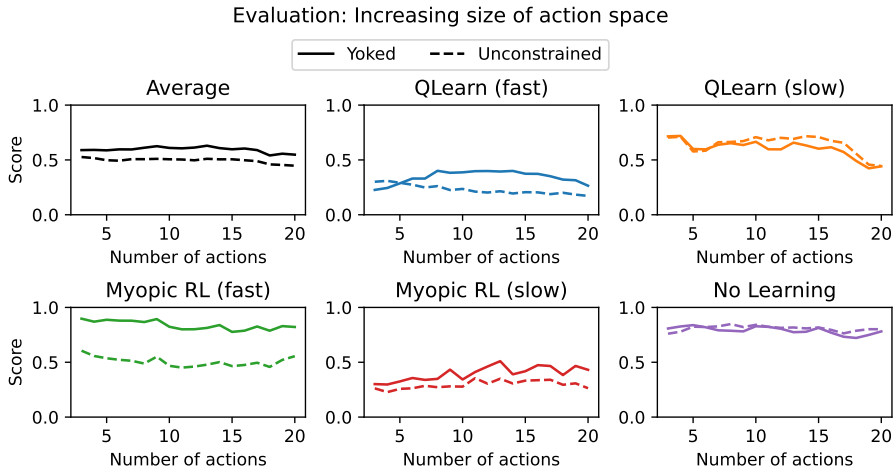


Fig. 5: The average probability of identifying the correct algorithm, plotted over increasing size of action space. All actions beyond the first three are distractors that do not have an influence on the state. The *yoked* condition is again generally superior to *unconstrained*, just slightly worse for *QLearn (slow)* and *No Learning*. However, contrarily to our initial assumption, increasing size of the action space does not disproportionately impact the *unconstrained* condition.

yoking is generally superior to the unconstrained condition, but the performance does not vary with increasing number of actions. We believe this is because the task only has a single, working solution strategy. However even in this condition the *yoking*-based approach is more reliable and suffers less from variance. With more complex tasks, where agents can learn a more diverse set of approaches, we expect that the gap between the *yoked* and *unconstrained* approach will widen even more, in favor of the *yoked* approach.

5 Evaluation in a Real Cockatoo Lockbox Learning Task

We will now apply the yoking approach to real biological data of Goffin's cockatoos opening a lockbox. In this experiment, three Goffins's cockatoos learned to open the lockbox depicted in Figure 1b and described above in Section 4. This lockbox is baited with a cashew reward behind the final plexi-glass door. To obtain the cashew, the birds need to unlock the individual mechanism in the described sequence. For each bird, the data consists of 12 sessions with a maximum of duration of 15 minutes per session. The birds were habituated and pre-trained to open the last two-stages of this lockbox, the door and the bar, so the main learning problem considered here is learning to open the lockbox with the additional metal disk that needs to be unlocked first.

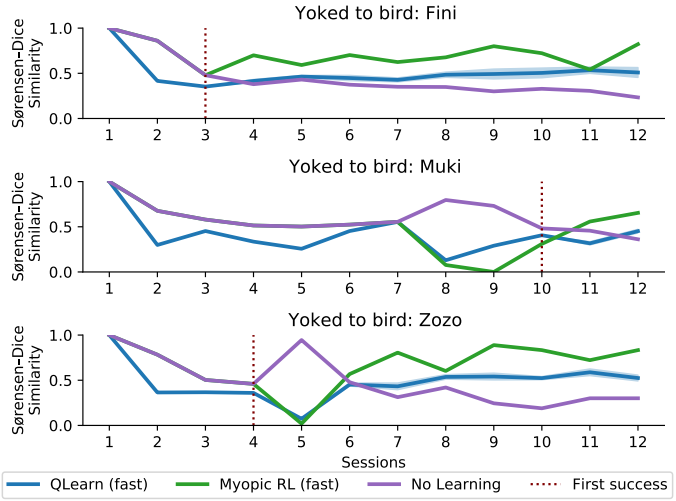


Fig. 6: Similarity of three candidate learning algorithm hypotheses to real bird data. Over the course of 12 sessions, all three birds are first best explained by the *No Learning* hypothesis, until the *Myopic RL* model becomes the best explanation. These switching points also align with the sessions where the real birds made learning progress in the experiments. The *First success* line indicates the session when the birds solved the lockbox for the first time.

While we acknowledge that none of our candidate algorithms is realistically implemented in the birds, it is still worthwhile and informative to see how the models' adaptation compares to learning in the birds. Due to space constraints we cannot explain the bird experiments in detail, but the most important feature of these experiments is that the birds rapidly improve in the task after they discovered the solution to the lockbox for the first time. Figure 6 shows the results of a *yoked* analysis, and indeed their (not) learning in the first sessions seems to be best explained by the *No Learning* model until the *Myopic RL (fast)* model takes over as the best explanation. This inflection point is largely synchronous with those sessions where the birds learn to open the lockbox for the first time. *Myopic RL (fast)* is the fastest candidate learning model in our analysis, which suits the rapid learning we observe in the birds.

6 Conclusion

If we aim to understand how animals learn to solve complex tasks, we need tools to compare biological learning data to synthetic learning models. In this paper, we show that yoking should be an important ingredient to such a comparison. In this context, yoking means that synthetic learning algorithms directly makes use of the actions, percepts, and rewards that the target agent experienced. Our

analysis demonstrates that yoking is an appropriate tool to for identifying the learning algorithm underlying an observed learning behavior. But this approach also faces a challenge, namely the off-policy learning problem. Not all reinforcement learning algorithms are equally capable of learning from data generated by another agent [9]. Thus, analyses like ours need to take great care to not bias results towards off-policy capable algorithms. We believe this is a highly significant insight, also for other analytical approaches in behavioral biology.

Our results achieved with yoking are an example for the deep connections that exists between the worlds of machine learning and biological learning. We believe there is much to gain when we transfer concepts between these domains. The method of yoking and the problem of off-policy learning, both, are relevant in either domain. Learning in both domains is confronted with similar problems and each domain can benefit from leveraging insights of the other.

References

1. Anderson, D.I., Campos, J.J., Anderson, D.E., Thomas, T.D., Witherington, D.C., Uchiyama, I., Barbu-Roth, M.A.: The flip side of perception–action coupling: Locomotor experience and the ontogeny of visual–postural coupling. Human Movement Science **20**(4-5), 461–487 (2001)
2. Argall, B.D., Chernova, S., Veloso, M., Browning, B.: A survey of robot learning from demonstration. Robotics and Autonomous Systems **57**(5), 469–483 (2009)
3. Church, R.M.: Systematic effect of random error in the yoked control design. Psychological Bulletin **62**(2), 122 (1964)
4. Dice, L.R.: Measures of the amount of ecologic association between species. Ecology **26**(3), 297–302 (1945)
5. Gardner, R.A., Gardner, B.T.: Feedforward versus feedbackward: An ethological alternative to the law of effect. Behavioral and Brain Sciences **11**(3), 429–447 (1988)
6. Held, R., Hein, A.: Movement-produced stimulation in the development of visually guided behavior. Journal of Comparative and Physiological Psychology **56**(5), 872 (1963)
7. Lee, D., Gujarathi, P., Wood, J.N.: Controlled-rearing studies of newborn chicks and deep neural networks. preprint arXiv:2112.06106 (2021)
8. Ng, A.Y., Russell, S.J.: Algorithms for inverse reinforcement learning. In: Proceedings of the Seventeenth International Conference on Machine Learning. p. 663–670. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2000)
9. Ostrovski, G., Castro, P.S., Dabney, W.: The difficulty of passive learning in deep reinforcement learning. Advances in Neural Information Processing Systems **34** (2021)
10. Salkind, N.J.: Encyclopedia of Research Design, vol. 1. sage (2010)
11. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT press (2018)
12. Torabi, F., Warnell, G., Stone, P.: Behavioral cloning from observation. preprint arXiv:1805.01954 (2018)
13. Wood, S.M., Wood, J.N.: Using automation to combat the replication crisis: A case study from controlled-rearing studies of newborn chicks. Infant Behavior and Development **57**, 101329 (2019)