Computational discovery of human reinforcement learning dynamics from choice behavior

Daniel Weinhardt
Osnabrück University
dweinhardt@uos.de

Maria Eckstein
Google Deepmind
mariaeckstein@deepmind.com

Sebastian Musslick
Osnabrück University,
Brown University
smusslick@uos.de

1 Introduction

One of the central goals of behavioral science is to understand the cognitive processes that occur within individuals. A significant challenge lies in the fact that that these processes are not directly observable and must be inferred from noisy behavioral data. Human reinforcement learning (RL) is a prime example, positing that individuals adjust the values they assign to actions based on the rewards they receive. However, uncovering the learning dynamics underlying these action values is difficult because the action values are latent and cannot be directly measured. Instead, they must be inferred from observable behaviors, such as action choices. In this work, we introduce a novel machine learning approach for inferring latent human RL dynamics from human behavior leveraging recurrent neural networks (RNN) and sparse identification of non-linear dynamics (SINDy). This approach automates the discovery of interpretable models that explain human RL.

Our approach recovers human RL dynamics using a two-step process. First, drawing inspiration from previous work [1, 2, 3, 4], we train an RNN to the behavior of humans performing a reinforcement learning task. Critically, we disentangle the RNN into multiple components, each representing a distinct cognitive mechanism. Each RNN component is constrained to a low-dimensional memory state representing the latent dynamical system variables for that cognitive mechanism. RNNs are a popular approach to fit human behavior due to their flexibility in capturing cognitive computations [5, 6, 7]. Once fitted, we simulate the RNN to generate time series data for each component's memory state. Next, we apply SINDy [8], a data-driven algorithm developed for identifying dynamical systems from time series data by representing it as a linear combination of predefined functions, often incorporating non-linear expressions like polynomials of the input variables. Here, we apply SINDy to the time series generated by each RNN component's memory state. This enables us to characterize the cognitive mechanism implemented by each component by identifying the underlying dynamical system that governs how it updates memorized action values based on previous rewards and choices. Together, these extracted dynamical systems yield an interpretable model of human RL.

In this work, we evaluate the algorithm's ability to recover various ground truth human RL models from simulated behavioral data in the context of a two-armed bandit task. These ground truth models incorporate a range of cognitive mechanisms, including Q-learning [9], value forgetting over time [10], choice perseverance [11], asymmetrical learning rates for positive and negative outcomes [12], and confirmation bias [13]. Critically, we attempted to infer these mechanisms from the choice behavior of the ground truth alone and under decision noise. Our results indicate that our approach is capable of recovering underlying cognitive mechanisms robustly and with a high accuracy, enabling the automated discovery of expressive yet interpretable human RL models.

NeurIPS 2024 Workshop on Behavioral Machine Learning.

2 Methods

2.1 Experimental paradigm

The two-armed bandit task is a paradigm for studying human RL where a participant repeatedly chooses between two options (or "arms"), each offering a potential reward $r = 1$ with some probability $P(r_\mathrm{i})$. Participants are tasked to maximize the rewards across a series of trials by selecting one the two arms a_i, $i \in 1, 2$, on each trial. In this study, $P(r_\mathrm{i})$ is unknown to the participant and fluctuates over the trials t with a drift rate of σ according to

$$P_{\mathrm{t+1}}(r_\mathrm{i}) = P_\mathrm{t}(r_\mathrm{i}) + \mathcal{N}(0, \sigma). \tag{1}$$

Maximizing rewards in this task requires participants to balance exploration (trying new options) and exploitation (sticking with known good ones) to adapt to the changing environment. Critically, participants must learn, based on experience, which of the two options is currently more rewarding.

2.2 Ground truth models

Task representation and action selection. Each ground truth is a simulated reinforcement learning agent that chooses between two actions $\vec{A} = (a_1, a_2)$, and learns from the respective outcomes—the reward r—similarly to a human participant. Learning is represented by applying a Q-learning algorithm to update its estimate about the value of each action $\vec{Q} = (q_1, q_2)$. Action selection is a stochastic procedure with the action probabilities $P(\vec{A})$ given by the Softmax function

$$P(\vec{A}) = \mathrm{Softmax}(\vec{Q}). \tag{2}$$

The single action values are computed by

$$q_\mathrm{i} = \beta(v_\mathrm{i} + c_\mathrm{i}), \tag{3}$$

where β is the inverse noise temperature, $i \in 1, 2$ indicates the action, v_i is the reward-based component of the action value, and c_i is the choice-based component, reflecting the influences of prior rewards and choices, respectively.

Cognitive mechanisms. The model updates the chosen action value q_ch and the non-chosen action value q_nch separately, with each update driven by multiple cognitive mechanisms.. The reward-based value v_ch of the chosen action is updated by using the reward prediction error (RPE), calculated as $e_\mathrm{rp} = (r - v_\mathrm{ch})$, and scaled by an adaptive learning rate α [9]. This learning rate is further influenced by two cognitive mechanisms: (a) asymmetric learning rates for positive versus negative outcomes [12] and (b) a confirmation bias that enhances learning for outcomes that confirm existing estimates (e.g. high v_ch and positive outcome) while reducing learning for outcomes that contradict those estimates (e.g. high v_ch and negative outcome) [13]. The choice-based value c_ch of the chosen action is updated by incorporating a choice persistence bias which makes the model favor previously chosen actions [11]. Meanwhile, the the non-chosen action is updated by gradual value forgetting over time of the reward-based value v_nch [10], while the choice-based value is set to $c_\mathrm{nch} = 0$. In summary, these mechanisms result in the following update equations,

$$v_{\mathrm{ch,t+1}} = v_{\mathrm{ch,t}} + (\alpha_\mathrm{r} r + \alpha_\mathrm{p} \overline{r} + b_\mathrm{cb}(v_{\mathrm{ch,t}} - v_0)(r - v_0))e_\mathrm{rp}, \tag{4}$$

$$v_{\mathrm{nch,t+1}} = v_{\mathrm{nch,t}} + b_\mathrm{f}(v_{\mathrm{nch,t}} - v_0), \tag{5}$$

$$c_{\mathrm{ch,t+1}} = b_\mathrm{cp} a_\mathrm{repeat}, \tag{6}$$

$$c_{\mathrm{nch,t+1}} = 0, \tag{7}$$

where t is the current time step, α_r the learning rate for positive outcomes, α_p the learning rate for negative outcomes, $\overline{r} = (1 - r)$ signifying a non-rewarded trial, b_cb the confirmation bias weight, v_0 the initial reward estimate, b_f the forget rate, b_cp the choice persistence bias, and a_repeat a binary signal identifying whether a choice is repeated. We will generate synthetic choice data using various parameterizations of this ground truth and assess the extent to which we can recover its underlying cognitive mechanisms.

2.3 Framework for the automated discovery of cognitive models

The automated discovery method involves fitting an RNN to choice data and then using SINDy to extract interpretable update equations from the RNN.

Recurrent neural network. The RNN is trained to emulate the behavioral choice data by learing the actions $\vec{A}$ a human participant (or simulated agent) selected based on their experienced reward outcomes and prior actions. To achieve this, the RNN must internalize cognitive processes that can reproduce the original choice data.

To facilitate the discovery of distinct cognitive mechanisms, we disentangle the RNN into several smaller sub-networks and apply information bottlenecks for each of them (for similar approaches, see [2, 14]). These information bottlenecks control the inputs and outputs each sub-network is processing. The sub-networks represent distinct action value updates. Here, the sub-networks $\mathrm{f}_{\mathrm{q,ch}}$ and $\mathrm{f}_{\mathrm{c,ch}}$ update the reward-based and choice-based components of the chosen action value q_{ch}, respectively. The reward-based and choice-based components of the non-chosen action value q_{nch} are updated by the sub-networks $\mathrm{f}_{\mathrm{q,nch}}$ and $\mathrm{f}_{\mathrm{c,nch}}$, respectively. These updates have the structure

$$v_{\mathrm{ch,t+1}} = v_{\mathrm{ch,t}} + \mathrm{Sigmoid}(\mathrm{f}_{\mathrm{q,ch}}(v_{\mathrm{ch,t}}, r))e_{\mathrm{rp}}, \tag{8}$$

$$v_{\mathrm{nch,t+1}} = v_{\mathrm{nch,t}} + \mathrm{Tanhshrink}(\mathrm{f}_{\mathrm{q,nch}}(v_{\mathrm{nch,t}})), \tag{9}$$

$$c_{\mathrm{ch,t+1}} = c_{\mathrm{ch,t}} + \mathrm{Tanhshrink}(\mathrm{f}_{\mathrm{c,ch}}(c_{\mathrm{ch,t}}, a_{\mathrm{repeat}})), \tag{10}$$

$$c_{\mathrm{nch,t+1}} = c_{\mathrm{nch,t}} + \mathrm{Tanhshrink}(\mathrm{f}_{\mathrm{c,nch}}(c_{\mathrm{nch,t}})), \tag{11}$$

$$\vec{Q} = \mathrm{ReLU}(\beta)(\vec{A}(v_{\mathrm{ch}} + c_{\mathrm{ch}}) + (1 - \vec{A})(v_{\mathrm{nch}} + c_{\mathrm{nch}})), \tag{12}$$

where Sigmoid is the sigmoid function imposing a value range of $0 < \mathrm{f}(x) < 1$ on the computed learning rate. The function $\mathrm{Tanhshrink}(x) = x - \tanh(x)$ imposes a quasi-zero-plateau on an otherwise quasi-linear function enhancing the RNN's ability to sparsify the value representation and therefore omit unnecessary cognitive mechanisms. ReLU is the rectified linear unit function imposing a value range of $0 \leq x < \inf$ on the inverse noise temperature β.

A significant challenge in recovering cognitive mechanisms is ensuring their identifiability. This issue is not unique to our approach but is a common problem in the inherently ill-posed task of recovering cognitive mechanisms from noisy choice data. For example, the learning rate function $\mathrm{f}_{\mathrm{q,ch}}$ can approximate multiple mechanisms at once (e.g. asymmetric learning rates and confirmation bias), including the RPE e_{rp}, due to their identical inputs. To address this issue, we hard-code the RPE e_{rp}, reflecting the consensus that RPEs are fundamental to most human RL models. This allows $\mathrm{f}_{\mathrm{q,ch}}$ to discover mechanisms distinct from the RPE, despite relying on the same inputs.

The RNN is trained by minimizing the cross entropy loss between the participant's actual next action $\vec{A}_{\mathrm{t+1}}$ chosen by the participant and the predicted probabilities $P(\vec{A}_{\mathrm{t+1}})$, based solely on the observed current action $\vec{A}_{\mathrm{t}}$, the received reward r_{t} and the RNN's latent state space $\vec{Q}_{\mathrm{t}}$.

Sparse identification of non-linear dynamics. SINDy is a data-driven method for the discovery of dynamical systems from time series data. Applying SINDy—and symbolic regression algorithms in general—to uncover cognitive mechanisms presents a challenge, as the time series of relevant variables (e.g., action values) are not directly observable. However, after fitting an RNN to the observable choice data, we can extract these latent variables over the trials t from each sub network within the RNN. We then apply SINDy to derive interpretable equations that govern the cognitive mechanisms implemented by each sub network, yielding a fully interpretable model of human RL. SINDy represents the cognitive mechanisms as sparse linear combinations of non-linear terms according to

$$\Delta v_{\mathrm{ch}} = \Theta([1, v_{\mathrm{ch}}, r, \overline{r}])\vec{W}_{\mathrm{q,ch}}, \tag{13}$$

$$\Delta v_{\mathrm{nch}} = \Theta([1, v_{\mathrm{nch}}])\vec{W}_{\mathrm{q,nch}}, \tag{14}$$

$$\Delta c_{\mathrm{ch}} = \Theta([1, c_{\mathrm{ch}}, a_{\mathrm{repeat}}])\vec{W}_{\mathrm{c,ch}}, \tag{15}$$

$$\Delta c_{\mathrm{nch}} = \Theta([1, c_{\mathrm{nch}}])\vec{W}_{\mathrm{c,nch}}, \tag{16}$$

where Θ is a library of polynomial combinations up to the second degree of the given candidate terms and $\vec{W}$ is the array of corresponding fitted weights to each of the RNN's sub-networks. We used two optimizers (STLSQ [8] and SR3 [15]) in our simulation study due to their different performances in different contexts. The SINDy weight arrays $\vec{W}$ are trained by minimizing the mean-squared error between the observed variables and the predicted ones. Additionally, the weights are regularized (STLSQ: $L0$ and $L2$-norm; SR3: $L1$-norm). This way SINDy fits a predictive but sparse and thus interpretable equation by setting most weights $w_{\mathrm{i}} \in \vec{W}$ to $w_{\mathrm{i}} = 0$. SINDy and the used optimizers are implemented in the python package `pysindy` [16].

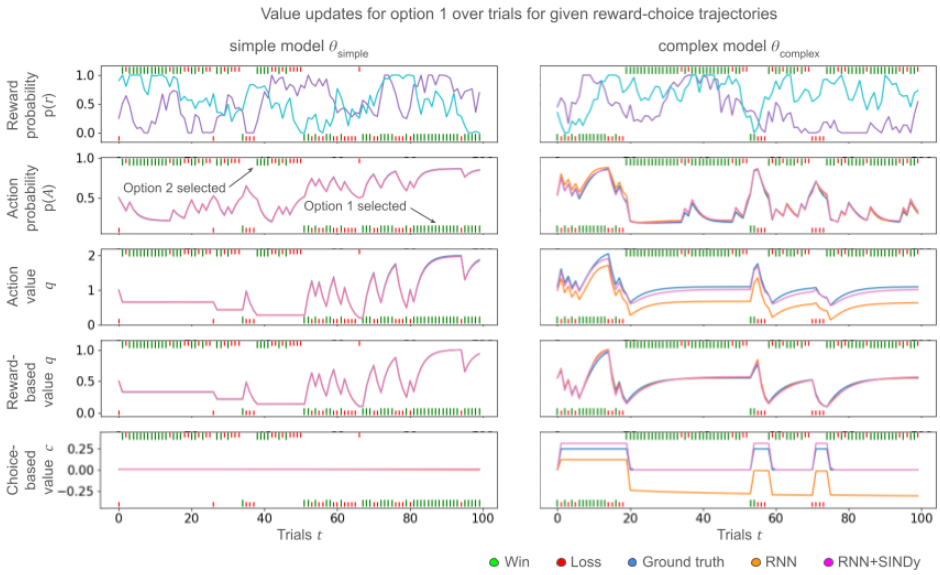

Figure 1: Reward probabilities $p(r)$, action selection probabilities $p(A)$, and value updates for option 1 over a given reward-choice trajectory for the simple model (left) and the complex model (right) computed by the respective ground truth (blue), the fitted RNN (orange) and the SINDy model (pink). The selected action a_1 or a_2 at each trial is marked by the lower and upper ticks, respectively. The outcome is marked by the color of the ticks (green: rewarded; red: non-rewarded).

2.4 Simulation experiments

We tested our approach in a simulation study with two ground truth models. We initialized one ground truth θ_{complex} with all described cognitive mechanisms (i.e. $\beta = 3.0$, $\alpha_{\text{r}} = 0.25$, $\alpha_{\text{p}} = 0.5$, $b_{\text{cb}} = 0.5$, $v_0 = 0.5$, $b_{\text{cp}} = 0.25$, $b_{\text{f}} = 0.2$) and a second ground truth model θ_{simple} with Q-learning only (i.e. $\beta = 2.0$, $\alpha_{\text{r}} = \alpha_{\text{p}} = 0.35$, $v_0 = 0.5$, and the remaining parameters set to 0).

Each ground truth model performed the two-armed bandit task for $n_{\text{sessions}} = 5,120$ sessions, each encompassing $n_{\text{trials}} = 64$ trials. We trained the discovery method on 80% of the sessions and evaluated the recovered model on the remaining 20% of the sessions. The two-armed bandit task had drifting reward probabilities with a drift rate of $\sigma = 0.1$ and yielded binary rewards $r \in \{0, 1\}$.

3 Results

We verified our method's ability to recover different ground truth models both qualitatively and quantitatively. First, for qualitative validation, we visually compared the trajectories of single value updates for the reward-based value q, the choice-based value c and the action value v, and the action selection probability $p(\vec{A})$ of the ground truth models against the corresponding recovered models over the trials t for given choice and reward trajectories. As shown in Figure 1, the value updates and the action selection probability closely matched those of the ground truth models. In the case of the complex model, the RNN established an offset for the choice value. However, this offset did not alter the action selection probabilities $p(\vec{A})$ because this offset is applied to both options equally, and therefore did not influence the difference between q_1 and q_2. For the RNN+SINDy model, this minor issue was resolved by simply removing the offset in the generated training data. As already mentioned, we couldn't identify one best-suited optimizer from the `pysindy` package. The optimizer selection is akin to hyperparameter optimization in traditional machine learning and can be performed by e.g. minimizing the loss between the predicted and the generated values computed by RNN+SINDy and the RNN, respectively. We ended up using STLSQ and SR3 to recover θ_{complex} and θ_{simple}, respectively.

Second, for qualitative validation, we trained in total 16 RNNs on 8 datasets from the simple ground truth model and on 8 datasets from the complex ground truth model to verify our approach's stability. The mean test losses and standard deviations were $\mathcal{L}_{\text{simple}} = 0.5970 \pm 0.0011$ and $\mathcal{L}_{\text{complex}} = 0.5265 \pm 0.0024$ for the simple and complex ground truth models, respectively. These results verify our approach's stability. The lower loss $\mathcal{L}_{\text{complex}}$ compared to $\mathcal{L}_{\text{simple}}$ is explained by the higher β-parameter leading to less noise and also to probabilities closer to 0 and 1, thereby reducing the overall distance to the actual binary targets in $\vec{A}_{\text{t+1}}$.

We compared the mean weights $\vec{W}$ and standard deviations of the fitted RNN+SINDy model with those of the ground truth models. As shown in Table 1, the weights $\vec{W}$ of the fitted models ϕ_{complex} and ϕ_{simple} matched the ground truth with high precision, identifying present terms and omitting absent ones. The simple ground truth model was better recovered, despite its higher noise. For the complex model, our approach struggled with ambiguity, especially for simpler choice functions (columns $C_{\text{c,ch}}$ to c^2_{nch}), where standard deviations exceeded mean weights. In the light of the stable test loss $\mathcal{L}_{\text{complex}}$, this suggests that our approach identified multiple solutions for the given observations.

Table 1: Weights of each candidate term for the complex and simple ground truths, θ_{complex} and θ_{simple}, along with the mean weights and standard deviations (σ) of the corresponding fitted models, ϕ_{complex} and ϕ_{simple}. Constants are denoted by $C_{\text{q,ch}}$, $C_{\text{q,nch}}$, $C_{\text{c,ch}}$, and $C_{\text{c,nch}}$. Model weights span multiple rows, with mean weights and standard deviations computed over 8 runs.

Model	β	$C_{\text{q,ch}}$	α	v_{ch}	r	$\overline{r}$	α^2	αv_{ch}	αr	$\alpha\overline{r}$
θ_{complex}	3.00	0.13	0.00	-0.25	0.00	0.50	0.00	0.00	0.00	0.00
ϕ_{complex}	2.58	0.17	0.00	0.08	0.00	0.22	0.00	0.00	0.00	0.00
(σ)	0.07	0.01	0.00	0.05	0.00	0.01	0.00	0.00	0.00	0.00
θ_{simple}	2.00	0.35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ϕ_{simple}	2.00	0.35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(σ)	(0.02)	(0.01)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Model	v^2_{ch}	$v_{\text{ch}}r$	$v_{\text{ch}}\overline{r}$	r^2	$r\overline{r}$	$\overline{r}^2$	$C_{\text{q,nch}}$	v_{nch}	v^2_{nch}	$C_{\text{c,ch}}$
θ_{complex}	0.00	0.5	0.00	0.00	0.00	0.00	0.10	0.80	0.00	0.00
ϕ_{complex}	-0.08	0.21	-0.11	0.00	0.00	0.22	0.08	0.81	0.00	0.32
(σ)	0.05	0.02	0.02	0.00	0.00	0.01	0.03	0.04	0.00	0.02
θ_{simple}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
ϕ_{simple}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
(σ)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Model	c_{ch}	a_{repeat}	c^2_{ch}	$c_{\text{ch}}a_{\text{repeat}}$	a^2_{repeat}	$C_{\text{c,nch}}$	c_{nch}	c^2_{nch}		
θ_{complex}	0.00	0.25	0.00	0.00	0.00	0.00	0.00	0.00		
ϕ_{complex}	0.14	-0.06	0.04	0.19	-0.06	0.00	0.49	-0.97		
(σ)	0.22	0.08	0.39	0.30	0.08	0.00	0.67	1.48		
θ_{simple}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		
ϕ_{simple}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		
(σ)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)		

4 Conclusion

We presented an approach that combines RNNs with SINDy to discover complex yet interpretable cognitive mechanisms underlying human RL from noisy behavioral data. Simulation results indicate that this data-driven method can successfully recover both simple and complex models, providing a promising alternative to traditional theory-driven models of human RL which consider only a small space of cognitive mechanisms [17]. While further refinement is needed, particularly around identifiability constraints, optimizer selection, and a more efficient experimental design, the framework holds promise to uncover novel mechanisms of human RL from human behavioral data.

References

[1] Maria K. Eckstein et al. *Predictive and Interpretable: Combining Artificial Neural Networks and Classic Cognitive Models to Understand Human Learning and Decision Making*. Pages: 2023.05.17.541226 Section: New Results. May 17, 2023.

[2] Kevin J. Miller et al. *Cognitive Model Discovery via Disentangled RNNs*. Pages: 2023.06.23.546250 Section: New Results. June 26, 2023.

[3] Li Ji-An, Marcus K Benna, and Marcelo G Mattar. "Automatic discovery of cognitive strategies with tiny recurrent neural networks". In: *bioRxiv* (2023), pp. 2023–04.

[4] Paul I Jaffe et al. "Modelling human behaviour in cognitive tasks with latent dynamical systems". In: *Nature Human Behaviour* 7.6 (2023), pp. 986–1000.

[5] Matthew Botvinick and David C Plaut. "Doing without schema hierarchies: a recurrent connectionist approach to normal and impaired routine sequential action." In: *Psychological review* 111.2 (2004), p. 395.

[6] Amir Dezfouli et al. "Integrated accounts of behavioral and neuroimaging data using flexible recurrent neural network models". In: *Advances in neural information processing systems* 31 (2018).

[7] Baihan Lin, Djallel Bouneffouf, and Guillermo Cecchi. "Predicting human decision making in psychological tasks with recurrent neural networks". In: *PloS one* 17.5 (2022), e0267907.

[8] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. "Discovering governing equations from data by sparse identification of nonlinear dynamical systems". In: *Proceedings of the National Academy of Sciences* 113.15 (Apr. 12, 2016). Publisher: Proceedings of the National Academy of Sciences, pp. 3932–3937.

[9] Roy A Wise. "Dopamine, learning and motivation". In: *Nature reviews neuroscience* 5.6 (2004), pp. 483–494.

[10] Asako Toyama, Kentaro Katahira, and Hideki Ohira. "Reinforcement learning with parsimonious computation and a forgetting process". In: *Frontiers in human neuroscience* 13 (2019), p. 153.

[11] Samuel J Gershman. "Origin of perseveration in the trade-off between reward and complexity". In: *Cognition* 204 (2020), p. 104394.

[12] Christopher Mark Hill. *Reward and punishment: the neural correlates of reinforcement feedback during motor learning*. 2019.

[13] Max Rollwage et al. "Confidence drives a neural confirmation bias". In: *Nature communications* 11.1 (2020), p. 2634.

[14] Amir Dezfouli et al. "Disentangled behavioural representations". In: *Advances in neural information processing systems* 32 (2019).

[15] Peng Zheng et al. "A unified framework for sparse relaxed regularized regression: SR3". In: *IEEE Access* 7 (2018), pp. 1404–1423.

[16] Brian de Silva et al. "PySINDy: A Python package for the sparse identification of nonlinear dynamical systems from data". In: *Journal of Open Source Software* 5.49 (2020), p. 2104. DOI: `10.21105/joss.02104`. URL: `https://doi.org/10.21105/joss.02104`.

[17] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 2018.