Supplementary Information for Long short-term prediction guides human meta-reinforcement learning

Dongjae Kim†, Jee Hang Lee†, Wooseok Jung, So Hyun Kim and Sang Wan Lee

[1]Department of AI-Based Convergence, Dankook University, 152 Daehak-ro, Yongin-si, 31116, Republic of Korea.
[2]Department of Human-Centered AI, Sangmyung University, 20 Hongjimun 2-gil, Seoul, 03016, Republic of Korea.
[3]Somerville College, University of Oxford, Woodstock Road, Oxford, OX2 6HD, United Kingdom.
[4]R&D Center, VUNO Inc., 479 Gangnam-daero, Seoul, 06541, Republic of Korea.
[5*]Department of Bio and Brain engineering , KAIST, 291 Daehak-ro, Daejeon, 34141, Republic of Korea.
[6]KAIST Center for Neuroscience-inspired AI, KAIST, 291 Daehak-ro, Daejeon, 34141, Republic of Korea.
[7]KI for Health Science and Technology, KAIST, 291 Daehak-ro, Daejeon, 34141, Republic of Korea.

Contributing authors: dongjaekim@dankook.ac.kr; jeehang@smu.ac.kr; wooseok.jung@vuno.co; rlathgus9872@gmail.com; sangwan@kaist.ac.kr;
†These authors contributed equally to this work.

Contents

List of Figures

Building RL models that learn the latent policy of humans

(related to Section 2 of the main text)

We used the following three types of reinforcement learning (RL) models for the simulations: DDQN [2], meta RL [3, 4], and prefrontal RL [1, 5]. The first two models were trained using two different training methods, goal matching (GM) and policy matching (PM). The prefrontal RL models were trained using the PM method only. In total, six RL models were trained to conduct the large-scale simulations for both the recoverability test and the generalizability test (also see Figure 2 in the main text). For the initial training, we used the two-stage Markov-decision task (MDT) [1] as an environment.

The environment

The two-stage MDT was used as an environment [1] for each RL model to train. It was originally designed for human subject experiments with regard to choice behavior. The structure of the task was designed to optimally favor behavioral control by either the model-based or model-free learning strategy. On each trial, it requires sequential binary choices through a two-layered Markov-decision problem (MDP) in order to obtain tokens that are redeemable for a monetary reward (Supplementary Figure 1). Each trial in the task requires two sequential actions.

The task has mainly two types of trials – specific and flexible goal. In specific goal trials, an agent can obtain a monetary reward when the agent chooses only one color of tokens. The color of the tokens allowing the monetary reward is changed on a trial-by-trial basis. For example, if a blue color was given with the specific goal condition, an agent should collect the blue-colored token to obtain a monetary reward. In contrast, in flexible goal trials, an agent can obtain a monetary reward when the agent collects any color of tokens. In the specific goal condition, a model-based strategy is favored since the reward prediction error (PE) will be high on average due to constant changes in goal state-values, whereas a model-free strategy is more encouraged in flexible goal trials.

In addition to this manipulation of goal conditions, the task also manipulated state-action-state transition probabilities within the MDP. Therefore, uncertainty in state-action-state transitions is high (0.5 vs. 0.5) on some occasions and becomes low (0.9 vs. 0.1) on other occasions. This manipulation of state-action-state transition probabilities across the trials intends to elicit either high or low state- PEs on average that should favor either a model-free or a model-based strategy, respectively.

A Markov-decision process subject to the training method

With the aforementioned two-stage MDT, we formalize the choice behavior of agents as an MDP, which is a 5-tuple (S, A, T, R, γ), where S is the state space,

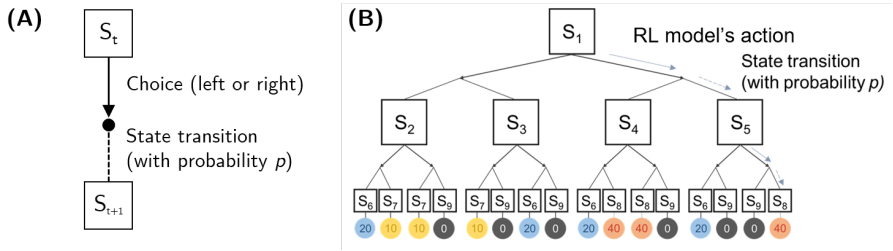

Fig. 1 Two-stage Markov-decision task (MDT) [1].
(A) Sequential two-choice Markov decision task ([4], redrawn). An agent moves from one state (St) to the other state (St+1) with a certain state-transition probability p following a binary choice, either left or right. (B) Illustration of the task.

A is the action space, T is the transition function, R is the reward function, and $\gamma \in [0, 1)$ is the discount factor. Since the GM and the PM have different objectives (and associated goal conditions), the specific definition of the MDP varies depending on the training method.

Goal matching (GM). In this training method, the RL model interacts with the two-stage MDT to simply maximize the expected amount of reward. We applied this GM method to two RL models, DDQN [2] and meta RL [3, 4], which we called GM-DDQN and GM-metaRL hereinafter, respectively. The MDP is specifically as follows:

- State: The state S is defined as the combination of the current state in the task and the current goal condition. The current state in the task is defined as S1, S2 ... S9, as shown in Supplementary Figure 1. The goal condition includes flexible and specific goal types.
- Action: The action space $a_t \in A$ of state $s_t \in S$ is defined as two actions, moving either to *Left* or to *Right* at state s_t. This action space allows the agent to move to the next state to get a token redeemable for the monetary reward subject to the goal condition.
- Transition: $T : S \times A \times S \to [0, 1]$ is a transition function driven by the environment E. Given the state s_t and an action a_t, the transition to the next state s_{t+1} is determined by the state-transition probability in each trial, either (0.9 vs. 0.1) for the low-uncertainty condition or (0.5 vs. 0.5) for the high-uncertainty condition.
- Reward: $R : S \times A \times R \to R$ is the reward function, where R is a discrete set of possible rewards in a range of {0, 10, 20, 40}. The reward that the RL agent actually receives varies depending on the goal condition given on a trial-by-trial basis. As described, an agent receives a (monetary) reward when the agent satisfies the goal condition in each trial. For example, on specific goal trials, the agent receives the reward if the agent successfully collected the token whose color is the same as the color that the specific goal condition indicated. On flexible goal trials, the agent receives the reward regardless of the color of tokens.

Policy matching (PM). In this training method, an RL agent was trained in such a way that it mimics the way the human performs reward maximization while performing a two-stage MDT. Thus, this method enables the achievement of both goal matching and behavior matching through combining the GM and behavior cloning (also see Figure 1 in the main text). In each training epoch, the RL model completes one episode consisting of 200–400 games of the two-stage MDT to maximize the reward (GM), and then the discrepancy between the model's behavior and human subject's behavior is translated into the loss function (behavior cloning). Here, the actual training of the RL model is guided by the loss function constructed by the difference between the agent's and the human subject's behavior, not by the reward acquired while performing each game of the two-stage MDT. Therefore, we present the new reward of the MDP in PM, a fundamental difference between GM's and PM's MDP.

- Reward: $R : S \times A \times R \to R$ is the reward function, where R is a discrete set of possible rewards in a range of $\{k-n, k, k+n\}$, $k > 0, n \geq 0$. The terminal reward R_Ω is defined as

$$R_\Omega = \begin{cases} k-n, & \text{if } a^1_{ag} \neq a^1_H \text{ and } a^2_{ag} \neq a^2_H \\ k+n, & \text{if } a^1_{ag} = a^1_H \text{ and } a^2_{ag} = a^2_H \\ k, & \text{otherwise} \end{cases}, \tag{1}$$

  where k, n is a pre-defined constant value, a^i_{ag} is an action that the agent performed at stage i in the two-stage MDT, and a^i_H is an action that a human subject originally performed at the same stage i.

The practical meaning of R_Ω is that an RL model is able to receive the maximum reward (e.g., $k+n$) when the RL model can duplicate the choice behavior of the human subject in one game. If the RL model completely fails to the duplication of the human subject's behavior in one game, the RL model receives the minimum reward (e.g., $k-n$). The RL model receives a neutral reward (e.g., k) otherwise. Here, the amount of n determines the impact of the terminal reward R_Ω on the reinforcement of the RL model's policy.

We note that to determine the terminal condition, we compared only two actions of the RL model and the human subject's actions since the two-stage MDT environment requires sequential two choices. For the entire training using PM, we ran 1,000 epochs and 8,000 epochs for the training of PM-DDQN and PM-metaRL, respectively.

Training

Given the two training methods, we trained the following six RL models: DDQN and meta RL using GM (GM-DDQN, GM-metaRL), DDQN and meta RL using PM (PM-DDQN, PM-metaRL), and prefrontal RL using PM (PM-pfcRL1, PM-pfcRL2).

DDQN is a deep reinforcement learning algorithm designed to effectively maintain a thematic consistency between value and policy.

$$Y_t^{DDQN} \equiv R_{t+1} + \gamma Q(s_{t+1}, \text{argmax } Q(s_{t+1},\ a;\ \theta_k), \theta_k^-). \tag{2}$$

We implemented the DDQN using the following network architecture: one input layer; two hidden layers, which were composed of 64 nodes each; and one output layer. All layers are fully connected. A set of hyper-parameters is as follows: the discount factor γ is 0.99, the learning rate α is 0.001, the target update frequency to the primary network τ is 0.001, and the batch size is 32.

For meta RL, we used the A3C-LSTM model implemented by A. Juliani (https://github.com/awjuliani/Meta-RL). The size of the LSTM is 256. The state, previous reward, and previous actions were encoded in the form of one-hot vectors. The LSTM receives an input as the concatenated form of these vectors. The policy and value networks are fully connected feedforward networks without hidden layers. These two networks receive LSTM outputs.

For prefrontal RL, we used the computational model adaptively combining model-based and model-free control, inspired from the findings on the neural activity of the lateral prefrontal cortex and ventral striatum [1, 5]. Two versions were selected: a baseline model for PM-pfcRL1 [1] and an adaptive model for PM-pfcRL2 [5]. Both models successfully implement arbitration control based on the Bayesian reliability estimation over model-based RL (FORWARD learning [6]) and model-free RL (SARSA learning [7]). PM-pfcRL2 [5] is slightly different from the original one (PM-pfcRL1). PM-pfcRL2 computes Bayesian reliability based on the clusters of PE distribution estimated by the Dirichlet process Gaussian mixture model, while PM-pfcRL1 simply divides PEs with the pre-defined threshold. By considering the PE distribution to be a Gaussian mixture model, PM-pfcRL2 is able to incorporate dynamic changes of task variables in the environment.

In addition, we used a random agent as an agent for the control group. The random agent chose the action randomly while performing the two-stage MDP. Supplementary Figure 2 shows the training results of all the RL models.

In the GM training, we used a fixed number of episodes for all RL models, GM-DDQN and GM-metaRL to train. In the PM training, we used the early-stopping condition for PM-DDQN, and maximum number of episodes for both PM-DDQN and PM-metaRL. For the computational efficiency, we stopped the training of PM-DDQN earlier when its likelihood summation in a single episode was greater than the likelihood summation of PM-pfcRL. Here, the likelihood stands for the probability showing the extent to which the choice of the agent at state s_t is similar to the actual choice of the human subject at the same state s_t. We followed equation 3 to compute the likelihood,

$$\text{Likelihood } L = p(a\mathsf{s\_i}), \tag{3}$$

where a is the human subject's choice at the i^{th} trial in state s.

Supplementary Figure 2 presents the training results of all the RL models. In Supplementary Figure 2A, a family of prefrontal RL models (PM-pfcRL1,

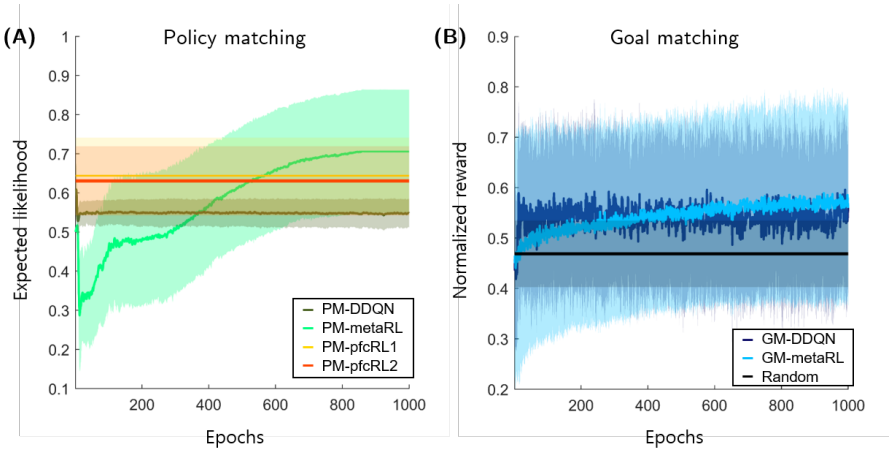

Fig. 2 Training results of all RL models.
(A) Training result of RL models trained using PM. The X-axis represents the number of epochs, and the Y-axis represents the expected likelihood, meaning the extent to which the RL model accurately predicts the action at the state compared to that of the human subject at the same state. (B) Training result of RL models trained using GM. The X-axis represents the number of epochs, and the Y-axis represents the normalized reward that each RL model trained using the GM method received.

PM-pfcRL2) shows the best performance in the aspect of rapid learning. It shows the consistent performance from the beginning stage of training to the end at a relatively higher level. The deep RL model, PM-DDQN, shows the worst performance. Rapidly decreased at the beginning, it shows a consistently low-level performance. Perhaps this gives us an intuition that a purely model-free RL agent may not be able to learn the optimal policy via the PM method. In the case of meta RL, PM-metaRL presents an interesting pattern – it learns slowly but can achieve the best performance in the end out of four RL models. Supplementary Figure 2B shows the training results of the RL models in which GM was used: GM-DDQN and GM-metaRL.

1 Latent behavior profile recovery test (related to Section 3 of the main text)

General linear model (GLM) for the latent behavior profile recovery test

We used the GLM to conduct the parameter recovery analysis on choice behavior. Here, the choice optimality, which is the behavioral measurement of model-based control [5], was chosen for a dependent variable, and uncertainty, complexity, previous reward, previous action, and max goal value were chosen for independent variables. Thus, the model was,

$$y = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4, \tag{4}$$

where y is a choice optimality, and $x = \{x_1 : \text{Uncertainty},\ x_2 : \text{Goal},\ x_3 : \text{Action (Prev.)},\ x_4 : \text{State (Prev.)}\}$.

2 Testing empirical generalizability of the RL models (related to Section 4 of the main text)

In order to test the capacity of the three types of RL models (see also Figure 2 in the main text) to generalize what they learned from a single task, we situated those RL models already trained in the two-stage MDT in the new volatile environments that the RL models never experienced. We created ten different MDTs for the environments, where the following two task parameters can be manipulated: task structure (ladder and tree) and task uncertainty (fixed, drift, switch, and drift + switch).

Testing the generalizability of the RL models trained using PM

We used the following four RL models for this scenario: PM-DDQN, PM-metaRL, PM-pfcRL1, and PM-pftRL2. First, we trained these four RL models in the two-stage MDT using the PM training method, which resulted in a total of 328 RL models (82 human subjects' data $\times$ 4 RL models). We saved the entirety of the parameters and weights of each RL model once all the RL models had been trained. To measure the empirical generalizability of the RL models, we situated the restored RL models in the ten tasks. The order of tasks that the RL agent experiences was pseudo-randomly determined. Once one RL model was loaded, the RL model performed the task assigned under the condition that both the parameters and weights of the model were frozen. The sequence on the manipulation of task variables (e.g., state-transition uncertainty and goal condition) were fixed across the entirety of the RL models involving the simulations.

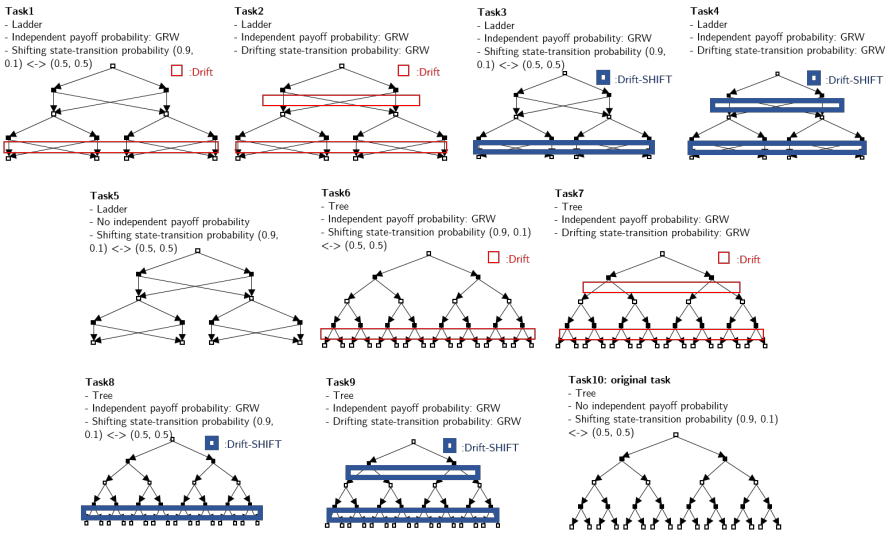

Fig. 3 Ten MDTs used for the empirical generalizability test (see also Figure 4 in the main text).

Testing the generalizability of the RL models trained using GM

We used the following two RL models for this scenario: GM-DDQN and GM-metaRL. Here, the two RL agents were allowed to perform all tasks pseudo-randomly, given without any constraints. Once the two RL models had been successfully trained in all ten tasks, the RL models followed the same steps as in the case of PM. After restoring all parameters and weights of each model, the RL model performed the task assigned under the condition that both the parameters and weights of the model were frozen.

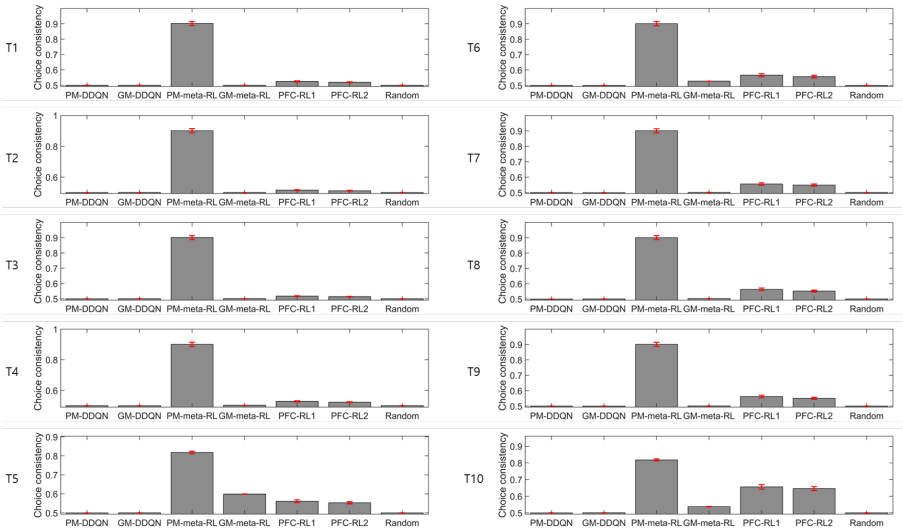

Fig. 4 Behavior profile – Choice consistency.
The X-axis represents the RL model, and the Y-axis represents choice consistency while each RL model performs the task. The label on the left side of the Y-axis represents the index of the task, e.g., T1 refers to Task 1 as presented in Supplementary Figure 3.

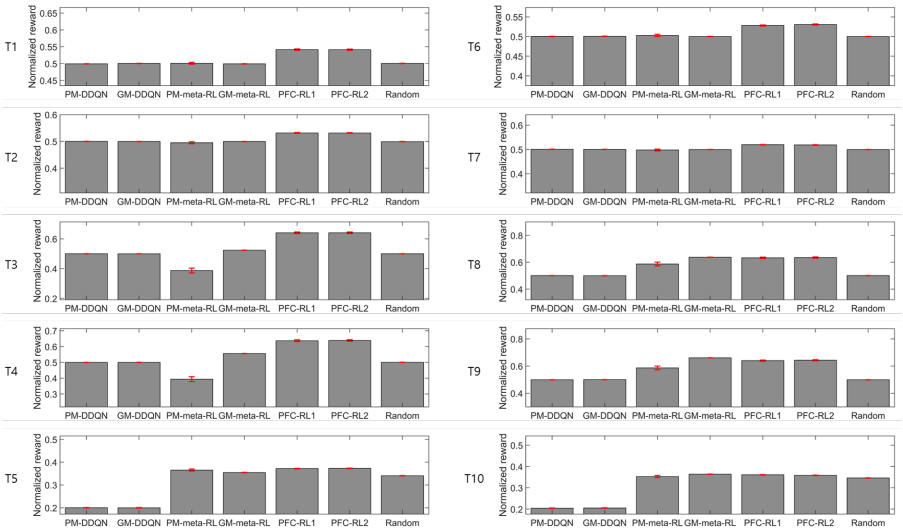

Fig. 5 Behavior profile – Normalized reward.
The normalized reward when model changed an action (i.e., $a_{t-1}! = a_t$). The X-axis represents the RL model, and the Y-axis represents a normalized reward while each RL model performs the task. The label on the left side of the Y-axis represents the index of the task, e.g., T1 refers to Task 1 as presented in Supplementary Figure 3.

3 Episodic encoding efficacy test via information theoretic analysis (related to Section 4 of the main text)

The information theoretic analysis is designed to measure the mutual information between two measures collected while the human subjects participated in the task on choice behavior [8]. Here we used episode F, defined as,

$$F_t = \{a^2_{t-1}, S^3_{t-1}, a^{2,*}_{t-1}, R_{t-1}\}, \tag{5}$$

where a^2_{t-1} is an action that was taken by an agent at stage 2 in the previous trial, S^3_{t-1} is a rewarding state at stage 3 in the previous trial, $a^{2,*}_{t-1}$ is an optimal action at stage 2 in the previous trial, and R_{t-1} is a reward in the previous trial.

In this analysis, we computed $I(F_t, a_t)$, which stands for the level of compression. It measures the amount of mutual information between the current action and the history of the previous trial. In addition, we computed $I(a_t, a^*_t)$, which measures the optimality of the choice behavior [8].

We call $I(F_t, a_t)$ the episodic encoding efficacy. We assume that this quantifies the amount of information required to achieve the optimal choices by the agent. It is direct measure of the agent's capacity to mimic human-like behavior in relation to the governing of the trade-off between performance and efficiency under the limited cognitive resources.

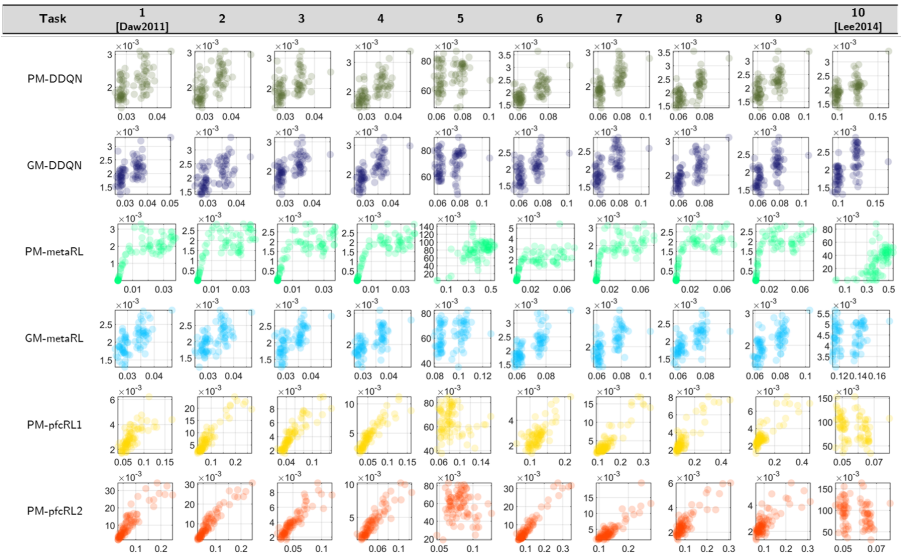

Fig. 6 Episodic encoding efficacy for six RL models across ten tasks.
Each plot shows an episodic encoding efficiency-optimality information plane. It represents the agent's ability to transfer information from past episodes to the agent's action and task performance. The X-axis shows the mutual information of the episodic events and the agent's action ("episodic encoding efficiency"). The Y-axis is the mutual information of the agent's action and the optimal action ("choice optimality"). In the table, the higher the position in the row, the worse capacity to generalize what the agent learned from a single task. For example, PM-DDQN is the worst, and PM-pfcRL2 is the best.

References

[1] Lee, S.W., Shimojo, S., O'Doherty, J.P.: Neural computations underlying arbitration between model-based and model-free learning. Neuron **81**(3), 687–699 (2014)

[2] van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double q-learning (2015) arXiv:1509.06461 [cs.LG]

[3] Wang, J.X., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J.Z., Munos, R., Blundell, C., Kumaran, D., Botvinick, M.: Learning to reinforcement learn (2016) arXiv:1611.05763 [cs.LG]

[4] Wang, J.X., Kurth-Nelson, Z., Kumaran, D., Tirumala, D., Soyer, H., Leibo, J.Z., Hassabis, D., Botvinick, M.: Prefrontal cortex as a meta-reinforcement learning system. Nat. Neurosci. **21**(6), 860–868 (2018)

[5] Kim, D., Jeong, J., Lee, S.W.: Prefrontal solution to the bias-variance tradeoff during reinforcement learning. Cell Rep. **37**(13), 110185 (2021)

[6] Gläscher, J., Daw, N., Dayan, P., O'Doherty, J.P.: States versus rewards: dissociable neural prediction error signals underlying model-based and model-free reinforcement learning. Neuron **66**(4), 585–595 (2010)

[7] Sutton, R.S., Barto, A.G.: Reinforcement Learning, Second Edition: An Introduction. MIT Press, ??? (2018)

[8] Filipowicz, A.L.S., Levine, J., Piasini, E., Tavoni, G., Kable, J.W., Gold, J.I.: The comparable strategic flexibility of model-free and model-based learning (2020)