Long short-term prediction guides human metacognitive reinforcement learning

Sang Wan Lee (✉ sangwan@kaist.ac.kr)
KAIST

Dongjae Kim
Dankook University  https://orcid.org/0000-0002-4513-9087

Jee Hang Lee
Sangmyung University

Wooseok Jung
University of Oxford

So Hyun Kim
Sangmyung University

Article

Keywords: Reinforcement learning, Long-short term memory, Meta-reinforcement learning, Prediction error

Posted Date: June 28th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3080402/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License

Additional Declarations: Yes there is potential Competing Interest. The preliminary draft of our work has been reviewed by Sam Gershman at Harvard.
Long short-term prediction guides human metacognitive reinforcement learning

Dongjae Kim\textsuperscript{1,}\textsuperscript{†}, Jee Hang Lee\textsuperscript{2,}\textsuperscript{†}, Wooseok Jung\textsuperscript{3,4}, So Hyun Kim\textsuperscript{2} and Sang Wan Lee\textsuperscript{5,6,7}

\textsuperscript{1}Department of AI-Based Convergence, Dankook University, 152 Daehak-ro, Yongin-si, 31116, Republic of Korea.
\textsuperscript{2}Department of Human-Centered AI, Sangmyung University, 20 Hongjimun 2-gil, Seoul, 03016, Republic of Korea.
\textsuperscript{3}Somerville College, University of Oxford, Woodstock Road, Oxford, OX2 6HD, United Kingdom.
\textsuperscript{4}R&D Center, VUNO Inc., 479 Gangnam-daero, Seoul, 06541, Republic of Korea.
\textsuperscript{5*}Department of Brain and Cognitive Sciences, KAIST, 291 Daehak-ro, Daejeon, 34141, Republic of Korea.
\textsuperscript{6}KAIST Center for Neuroscience-inspired AI, KAIST, 291 Daehak-ro, Daejeon, 34141, Republic of Korea.
\textsuperscript{7}The Kim Jaechul Graduate School of AI, 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; \textsuperscript{†}These authors contributed equally to this work.

Abstract

Long-term prediction ensures stable but slow learning. Short-term prediction is fast but deemed to be risky in context-changing environments. To understand how humans resolve this conflict, we fitted nine biologically plausible reinforcement learning models to 82 human behavior data using a human-model policy matching method. One meta-RL model replicates subjects’ context-dependent behavior, passing underfitting and overfitting tests. The same model survived a generalizability test simulating various Markov decision tasks and an adaptation test.
in which task structure, reward, and uncertainty change unexpectedly. Critically, the model’s ability is attributed to long- and short-term predictions about state transition and reward. Quantitative analyses revealed that conventional instant prediction error and its underlying long-term baseline are sufficient for context embeddings. These findings generate a long short-term prediction hypothesis: humans translate short-term into long-term predictions for context generalization while self-evaluating their long-term predictions to adjust short-term predictions. This creates a virtuous cycle of metacognitive generalization-adaptation.

**Keywords:** Reinforcement learning, Long-short term memory, Meta-reinforcement learning, Prediction error

**One sentence summary**

In human reinforcement learning, translating between short-term and long-term predictions creates a virtuous cycle of context generalization-adaptation.

**Introduction**

Recent reinforcement learning (RL) algorithms solve complex reward-prediction problems [1–5], and they can do so more rapidly by adopting various sample-efficient learning strategies, e.g., hierarchical RL [6, 7], model-based RL [2, 8–11], and memory-based RL [12, 13]. Despite such advances, most RL models still have a hard time making out-of-context predictions; their prediction is highly biased toward the current context. In this regard, we call these types of RL strategies *short-term prediction*.

Alternatively, one can learn information shared between different contexts, such as meta-learning [14, 15]. Although this strategy ensuers stable learning in a less context-dependent manner, it requires longer training time than short-term prediction. Even if one manages to learn across multiple contexts, its in-context prediction is inevitably less accurate. We call this RL strategy *long-term prediction*.

Abundant neural evidence supports animals’ ability to make short-term and long-term predictions. First, the midbrain dopamine neurons of the primates encode reward prediction error (RPE) [16, 17]. The striatal system in the human brain was shown to implement an actor-critic regime [18]. Recent studies have found neural evidence of sample-efficient learning strategies, such as model-based RL [19, 20] and memory-based RL [21, 22], supporting the short-term prediction view.

Recent evidence has shown that the human brain weights these strategies according to context changes ([23, 23–29]), resonating with the long-term prediction view. This evidence raised a new possibility that the brain implements meta-learning [15] or a meta-control of multiple learning strategies [30].
One plausible explanation is that the brain gradually learns to combine different strategies across multiple contexts (long-term prediction) in a way that makes the most accurate predictions in the current context [23–25] (short-term prediction).

This begs the fundamental question about human meta RL: how does the human orchestrate short-term and long-term predictions, and why does so? We hypothesize that human (1) generalizes contexts by translating short-term predictions into long-term predictions (bottom-up) while (2) continuously self-evaluating her long-term predictions to make the most accurate short-term predictions in the current context (top-down). We denote this bidirectional process as a long short-term prediction.

A few candidate RL models, such as model-based RL, meta RL, and successor representations, can implement the long short-term prediction hypothesis. In doing so, we ran two sets of formal tests with various types of RL models: (i) underfitting and overfitting test and (ii) generalizability and adaptability test. First, we present a provably efficient overfitting test to show that a single round of retraining to match the behavioral policy of humans and models is sufficient for replicating context-dependent behavior pattern, called contextual behavior recoverability. Second, nine biologically plausible RL models were fitted to human behavior data using a human-model policy-matching method. Third, we define a fully parameterized task space to quantify the model’s ability to generalize to various task structures and to adapt to context changes. Finally, we examine the characteristics of the best model to gain an insight into how the human’s short-term predictions are translated into long-term and vice versa.

Fig. 1 Human-model behavioral policy matching framework. (A) Three different approaches to training RL models: goal matching (GM), behavior cloning (BC), and policy matching (PM). PM is a blend of GM and BC, used in [23–25]. (B) Contextual behavior recoverability. After initial training with the original human behavior data (xHuman), the simulation data (xModel) was obtained by having the fitted models (Model) perform the original task. We then quantify the effect of context changes (task parameters) on the human behavior data and the simulation data, respectively (contextual behavior profiling), and test whether the two effects match (Contextual behavior recoverability test).
To build RL models that similarly learn tasks as humans do, we consider three different training strategies: goal matching (GM), behavior cloning (BC), and policy matching (PM) (Figure 1A). Goal matching (GM) trains the RL model with the same goal as human subjects, e.g., reward maximization, without needing to mimic human behavior. Behavior cloning (BC) is focused on imitating human behavior (Michie et al., 1990). Even if a model mimics human behavior [31–33] or matches human-level performance [1–5]), it does not necessarily mean that the model fully embeds essential computations of the human RL. Moreover, model complexity increases the risk of policy bias, doomed to failure of replicating human behavior patterns in similar contexts [34, 35].

Policy matching (PM) alleviates this issue by mimicking the way humans achieve the goal. In each training epoch, the RL model completes one task episode to maximize reward (following the GM regime). Throughout episodes, the model is trained to minimize the loss function, reflecting the difference between the model’s behavior and that of the human (following the GM regime). The PM was previously used for training computational models to account for neural data [20, 23, 24].

After initial training, one needs to ensure that the context-dependent behavior of the trained model matches that of the human subjects (Figure 1B; Contextual behavior recoverability). In an ideal situation where there is no overfitting, retraining evokes little discrepancy of the contextual behavior between subjects and models. On the other hand, overfitting amplifies the loss of information about policy of original human subjects as the retraining repeats, leading to the increase in the magnitude of difference in context-dependent behavior. In the following, we formalize this situation as a sub-Martingale process.

First, the context effect on behavior after the i-th re-trainings is specified by using a general linear model.

\[ g_i(\theta) = \beta_i^T \theta_i, \beta_i \in \mathbb{R}^n, \]

where \( \theta_i \) is a task context vector consisting of, e.g., state-transition uncertainty and goal (See Supp. 7). The coefficient \( \beta_i \) refers to the contextual behavior parameters. To quantify contextual behavior difference between the human subjects’ data (\( g_0 \)) and the model’s simulation data after the n-th retraining (\( g \)), we apply the L1 loss as following:

\[ S_n = \sum_{j=1}^{2} |\beta_{0,j} - \beta_{n,j}|, \]

Because overfitting could increase \( S_i \) in general, the upper bound of \( \mathbb{P}(S_n > \lambda) \) can be a viable overfitting indicator. We approximate its upper bound using a random variable \( D_n \), the sum of the n-step loss.
Lemma 1 \( S_n \leq D_n = \sum_{i=1}^{n} X_i \) where \( X_i = \sum_{j=1}^{2} | \beta_{0,j} - \beta_{n,j} | \) for all \( n \geq 1 \).

Since \( \mathbb{P}(S_n > \lambda) \leq \mathbb{P}(D_n > \lambda) \) holds for all \( \lambda \) by lemma 1, the tail bound of \( D_n \) can serve as an overfitting bound. One can reasonably assume that \( X_i \), which reflects the contextual behavioral loss after the \( i \)-th retraining, is bounded owing to the bounded parameter space associated with an arbitrary training method, and that the average loss will quickly diminish after a few retraining steps if the model behaves consistently:

- \( X_i \) are independent non-negative bounded random variables with mean \( \mu_i \).
- \( \mu_i \) decays exponentially by the rate \( r \): there exists an \( r \in (0, 1) \) such that

\[
\lim_{n \to \infty} C_n < \infty.
\]

The first assumption makes \( D_i \) a submartingale process.

Lemma 2 \( D_n = \sum_{i=1}^{n} X_i \) is a submartingale with respect to a natural \( \sigma \)-algebra \( \sigma(X_1, \ldots, X_{n-1}) : \mathbb{E}[D_n | X_{n-1}, X_{n-2}, \ldots, X_1] \geq D_{n-1} \).

This characteristics incurs the parameter-independent upper bound of the degree of overfitting \( \mathbb{P}(S_n > \lambda) \), which quantifies how quickly the overfitting bound saturates as retraining repeats.

**Theorem 1 (Adjusted from the Azuma-Hoeffding Inequality)** Suppose \( D \) is a submartingale with \( D_0 = 0 \) and \( |X_n| = |D_n - D_{n-1}| \leq c \) almost surely for all \( n \), where \( c \) is a constant such that \( |Y| \leq c \). Then,

\[
\mathbb{P}(D_n \geq \lambda) \leq \exp \left( -\frac{1}{2} \frac{\lambda^2}{c^2 n} \right) \prod_{i=1}^{n} \left( 1 + \frac{\mu_i}{c} \right) = C_n \exp \left( -\frac{1}{2} \frac{\lambda^2}{c^2 n} \right)
\]

Now we focus on \( C_n \) which limits the behavior of the above overfitting bound. Considering the second assumption, the expectations of \( X_1, X_2, \ldots \) are bounded by \( \mu_1, r \mu_1, r^2 \mu_1, \ldots \), respectively, and \( r < 1 \) make this geometric sequence converge to zero. Hence, \( C_n \) could be rewritten as

\[
C_n = \prod_{i=1}^{n} \left( 1 + \frac{r^{n-1} \mu_1}{c} \right)
\]

and \( \lim_{n \to \infty} C_n < \infty. \)

The second assumption implies \( r \geq \frac{\mu_{i+1}}{\mu_i} \) for all \( i = 1, 2, \ldots \). Then, consider an unbiased estimator \( \hat{\mu}_i = X_i \) of \( \mu_i \). Evaluating only the first two step losses \( X_1 \) and \( X_2 \) are sufficient to estimate \( r \): \( \hat{r} \hat{\mu}_2 = \frac{X_2}{X_1} \). As this sequence converges rapidly, \( \hat{r} \) is a reasonable estimate to bound the loss amplification. Incorporating \( \hat{r} \) into \( C_n \) makes it possible to predict the limiting behavior of the overfitting bound without multiple rounds of retraining. This result allows us to run an overfitting test based on contextual behavior recoverability (Figure 1).
A prefrontal RL model replicates human contextual decision making

Using three training methods (BC, GM, PM), three categories of RL models (model-free, model-free + model-based, model-free x model-based) were fitted to 82 human subjects’ data, collected with a two-stage Markov decision task [23]. (Figure 2A; Full details in Methods - Computational models of RL). It is an exhaustive simulation, including more than 4,000 individual model fitting processes (9 models × 82 subjects × 3 training types × 2 retraining).
The best fitting models identified by the underfitting test (Figure 2B) were previously implicated in striatal and cortical value-based decision making (MF, MB, and pfcRL). To examine whether the model’s explainability is ascribed to contextual behavior, we measured how much information transferred from the past episodes of events to the agent’s current action pertains to the optimal policy that fully incorporates the current context (Figure 2C; for full details, see Methods - Episodic encoding efficacy test). The categories of models surviving in the underfitting test also showed the highest level of episodic encoding efficacy (For more details, see Supplementary Information).

Next, we ran an overfitting test to identify models that reliably replicate human context-dependent behavior patterns (Figure 2D). As discussed in the previous section, the overfitting test quantifies how much the effect of context (goal and state transition uncertainty) on behavior changes after training. Following the convention of the parameter recovery analysis [35], the models were fitted to individual subjects’ data $x_{\text{Human}}$, and then the simulated data $x_{\text{Model}}$ were generated by running simulations with the fitted model on the original task. We then compared the context effect for $x_{\text{Human}}$ and $x_{\text{Model}}$ ($L_1$ distance between the respective $\beta_i$).

One version of prefrontal RL (pfcRL2), a meta RL algorithm that combines model-based and model-free RL based on the self-evaluation of the long-term prediction error signals[25], demonstrated significantly better contextual behavior recoverability than others (Figure 2D). When computing goodness-of-fit statistics considering both the steepness and the significance of the correlation, the effect becomes more pronounced (Figure 2E). The effect size of the pfcRL2 is more than three times larger than most other models. In summary, pfcRL2 is the only model that survived underfitting and overfitting tests.

Task generalizability and adaptability of human prefrontal meta RL

To examine the models’ ability to generalize from what they learned from the original task to other tasks, we parameterized the two-step Markov decision task (Figure 3A and 3B). What makes generalization and adaptation tests practically challenging is the fact that humans cannot afford to perform as many trials and tasks as algorithms do. To circumvent this issue, the best fitting model, pfcRL2, was used as a proxy for human subjects. We compared this model with other four categories of models that showed reasonably good performance in our underfitting and overfitting tests: meta RL, model-free RL (SARSA and DDQN), model-based RL (FORWARD and SR-DYNA), and distributional RL (IQN). All model parameters were chosen from the initial training phase (Figure 2) to ensure that they never experienced new tasks.

We chose ten tasks from the task space to maximally span task contexts. Each task is associated with different contexts: task structure (ladder and tree) and task uncertainty (state-transition uncertainty: fixed, drift, switch, and
Fig. 3 Generalizability and adaptability test. (A) A parameterized task space spanning various contexts: task structures and uncertainty (state-transition and reward probabilities). The red dot indicates a sample task. (B) We generated ten different types of tasks by systematically manipulating both task structure and task uncertainty, each corresponding to the red dot in (A). (C) Task generalization. Shown are the performance of the top seven models. The rewards were normalized and then averaged to preclude that some task associated with a large reward overrides overall performance. (D) Task adaptation (task switching scenario). Task change took place every 393 trials (number of trials of the two-step MDP in Lee et al. (2014); see Methods), randomly sampled from (B). Shown are the cumulative reward (right) and reward (right). (E) Task adaptation (context changing scenario). The task type is fixed and its task parameter changes every five trials. We used the choice optimality as a performance measure to offset the effect of the task-specific reward scale on overall performance.

The generalizability test involved 5,740 simulations (= 82 subjects \times 7 RL models \times 10 tasks). The pfcRL2 achieved the highest generalization performance (Figure 3C). Most other models performed successfully, except for SR-DYNA and IQN. The adaptability test simulates a highly volatile environment, for which we developed two scenarios: the one with sudden task changes randomly sampled from the set of 10 tasks (Figure 3D - task switching scenario) and the other with continuous, subtle task parameter changes in a random walk fashion within a single task (Figure 3E - context changing scenario). A choice optimality measure was used in the latter scenario because...
a simple performance metric, such as average or cumulative reward, is not
sensitive enough to evaluate adaptive behavior to subtle context changes.\textsuperscript{25} Overall, the pfcRL2 showed the best adaptation performance in both scenarios
(Figure 3D and 3E). Taken together, the results demonstrated the propensity
of human meta RL (pfcRL2) to not only generalize what they learned from
the original task but also adapt to survive highly volatile environments.

We afterwards conducted the generalizability test with 5,740 simulations (= 82 subjects $\times$ 7 RL models $\times$ 10 tasks). The pfcRL2 achieved the highest gener-
alization performance (Figure 3C). Most other models performed successfully,
except for SR-DYNA and IQN. The adaptability test simulates a highly volatile
environment, for which we developed two scenarios: the one with sudden task
changes randomly sampled from the set of 10 tasks (Figure 3D - task switch-
ing scenario) and the other with continuous, subtle task parameter changes
in a random walk fashion within a single task (Figure 3E - context changing
scenario). A choice optimality measure was used in the latter scenario because
a simple performance metric, such as average or cumulative reward, is not
sensitive enough to evaluate adaptive behavior to subtle context changes.\textsuperscript{25} Overall, the pfcRL2 showed the best adaptation performance in both scenarios
(Figure 3D and 3E). Taken together, the results demonstrated the propensity
of human meta RL (pfcRL2) to not only generalize what they learned from
the original task but also adapt to survive highly volatile environments.
Long short-term prediction guides human metacognitive reinforcement learning

(A) Prefrontal reinforcement learning

(B) Long short-term prediction error

Bottom-up context generalization (via short-term/instant prediction error)

Bottom-up context generalization (via long-term prediction error)

Top-down context adaptation

(Caption on next page.)
Long short-term prediction guides human metacognitive reinforcement learning

Fig. 4 Long short-term prediction for context generalization and adaptation. (A) Schematic diagram of the prefrontal RL (pfcRL2). The red and blue dotted arrow refers to state (SPE) and reward prediction error (RPE), respectively. The dark and light arrow corresponds to the instant (PE) and long-term baseline prediction error ($PE_0$), respectively. The former signal is used for value learning while the latter guides the level of engagement of the environmental model in value learning strategies (indicated by a cylindrical arrow), acting as a meta RL. (B) Long short-term prediction error of the prefrontal RL. The long-term baseline prediction error ($PE_0$) is defined as a baseline estimate of the absolute amount of the instant PE, which serves as a long-term performance estimator of the current value learning strategy. (C) Context sensitivity of the instant PE. The colored dots indicate trials in which a linear regression model can recover a binarized instant PE from the current task context (“PE-based context embedding”). Instant PE concerns state-transition (state prediction error; SPE) and reward (reward prediction error; RPE). (D) Instant PE signals become more consistent (indicated by lower variability) when it gets sensitive to the current context (“context embedding trials”). For example, the variability of SPE is lower in the SPE-based context embedding trials (corresponding to the red dots of (B)) compared to the RPE-based context embedding trials (corresponding to the blue dots of (B)). (E) Proportion of trials in which the pfcRL2 chooses the model-based learning strategy over model-free in context embedding trials. (F-H) The same analyses as (C-E), but with the long-term PE ($PE_0$). (I) The preference for model-based learning strategy ($P_{MB}$) changes when the context changes. Measured is the entropy of the probability distribution of choosing the model-based learning strategy. (J) Effect of context information on instant PE, long-term PE ($PE_0$) and $P_{MB}$ (ROC analysis with SVM; input: context vector, output: binarized $PE/PE_0/P_{MB}$).

Long short-term prediction hypothesis

By examining the latent process of the pfcRL2, we attempt to understand computational principles underlying long short-term prediction. The agent makes long short-term predictions in terms of reward prediction error (RPE) and state-prediction error (SPE), each being used for learning the value and the model of the environment, respectively (Figure 4A).

First, the context-sensitivity analyses confirmed that the long short-term prediction guides bottom-up context generalization (Figure 4C-H). When evaluating how much context information is gleaned from short-term (Figure 4C) and long-term prediction errors (Figure 4F) (See Methods - Context sensitivity analysis), we found that the short- and long-term prediction error signals are sufficient for decoding a wide range of contexts, and the decodability is attributable to both SPE and RPE. These results evince bottom-up context generalization via the two channels (state transition and reward) in two different time scales (short-term and long-term). Moreover, the more sensitive a PE signal is to the current context (indicated by context embedding trials in Figure 4D and 4G), the more consistent the corresponding PE becomes (indicated by the entropy of the PE distribution in Figure 4D and 4G). This indicates that the PEs also signal the confidence level of the pfcRL2’s own context encoding, enabling metacognitive context adaptation.

Accordingly, we examined the effect of top-down context adaptation. The context information is effectively propagated to the control of learning strategies (Figure 4E and 4H). For example, when SPE signals context information, the agent favors the model-based learning strategy that primarily
Long short-term prediction guides human metacognitive reinforcement learning

uses SPE for value learning. Furthermore, the changes in the preference for model-based learning strategies coincide with context changes (Figure 4I); on the other hand, the preference changes occur less within each context. The formal quantitative analysis, in which an optimal linear encoder (support vector machine; SVM) was trained to measure the amount of mutual information between the context information and the PE signals, confirmed that the bottom-up context generalization (guided by the instant and long-term PE) and top-down adaptation (indicated by the model-based and model-free learning strategy) reflects context changes (Figure 4J; PE sensitivity AUC = 0.84, $PE_0$ sensitivity AUC = 0.92, $P_{MB}$ sensitivity AUC = 0.90).

**Fig. 5** Predictive meta-learner for controlling optimal learning strategies. (A) A schematic diagram of the predictive meta-learner. It learns to predict the optimal learning strategy (model-based or model-free RL) using the past PE signals and context information. The labels are determined in the same way as in the context sensitivity analysis (Figure 4). (B) Ablation study. Ten simulations were conducted in the context changing scenario using predictive meta-learners. The model trained with PEs (PEs+CTX) showed a significantly higher AUC than the one without PEs (CTX) (paired t-test; $p < 0.001$). (C) (Left) Predictive context adaptation. A context-changing scenario in which the reward probability, uncertainty, and task structure change over time (upper) and the corresponding label information (lower). (Right) The illustration of the predictive meta-learner’s prediction about the optimal learning strategy during the trials indicated by the grey box in the left figure. The blue and red triangles refer to correct and wrong predictions, respectively.
Long short-term prediction guides human metacognitive reinforcement learning

Predictive metacognitive RL with a long short-term memory networks

The above observations not only afford insight into how human long short-term prediction can guide context-dependent behavior, but also they could benefit the design of predictive meta-learning algorithms in machine learning. As a proof of concept, we implemented a predictive meta-learner with a long short-term memory network (LSTM), which is the minimal and most straightforward architecture that can accommodate our hypothesis (see Figure 5A). The learner takes as an input the past context information (e.g., task structure, state-space uncertainty, reward probability) and the history of its own short- and long-term prediction error. This design is intended to learn the long-term trends of context changes, thereby choosing the most suitable learning strategy for upcoming events.

The predictive meta-learner was trained to predict the optimal learning strategy using our previous simulations’ data. We confirmed that the model uses the prediction error signals to perform context-predictive learning (Figure 5B). It uses the model-based RL strategy when SPE signals context information statistically significantly (‘SPE sensitive’ trials; see Methods - context sensitivity analysis), whereas it uses model-free RL when RPE strongly signals context information (accuracy = 0.93, AUC = 0.79). Importantly, this design enables the learner to engage in top-down modulation of learning strategies only when it is confident about context change patterns, which is the key characteristics of metacognitive RL. It also allows the learner to fall back into an alternative, rule-based learning strategy when less confident, e.g., in volatile and highly uncertain environment[26].

After the training, we tested the model in a new task environment where task structure, state-space uncertainty, and reward probability continuously change (Figure 5C). The model seemingly predicted the correct optimal strategy (Figure 5C right; blue triangles) though some transient error occurs when the context changes rapidly (Figure 5C right; red triangles). The result implies that long short-term prediction errors are a simple and effective means for predictive context adaptation.

Discussion

Is it possible for algorithms to learn generalizable policies from humans? This study attempts to formally examine this question. To do so, we implemented a set of formal tests as prerequisites: (i) overfitting and underfitting test, and (ii) empirical generalizability and adaptability test, followed by a context sensitivity analysis. We compare humans’ latent context-dependent behavior with the four categories of RL models by fitting 82 human subjects’ data with two-stage MDTs, in which the goal, state-transition uncertainty, and state-space complexity were experimentally manipulated. The overfitting and underfitting
test showed that the one adaptively combining MB and MF RL, called prefrontal RL, is the only model replicating human subjects’ context-dependent learning patterns. The subsequent information-theoretic analysis showed that the prefrontal RL has better episodic encoding efficacy, an important feature in learning compact context representations. Next, we ran a large-scale simulation with ten new MDTs to the RL models to test the empirical generalizability of the RL models, an ability to generalize original contexts, and transfer its knowledge to deal with a variety of new contexts. The prefrontal RL showed the highest normalized reward on average. We also tested the RL models’ adaptability with the context-switching scenario. The prefrontal RL showed the highest performance in terms of context-wise reward and cumulative reward.

Such noticeable capacity of the prefrontal RL motivates us to interrogate the key variables underlying human RL: short-term and long-term prediction errors (PEs) about state-transition and reward. These signals are then used to adjust the balance between two complementary learning strategies, a stable model-free and sample-efficient model-based RL, in a way that best responds to new environmental challenges. This view resonates with humans’ propensity to cope with the trade-off between performance, sample efficiency, and computational cost [37].

The in-depth investigation of the prefrontal RL offers an interesting insight into how humans use short and long-term predictions during RL. First, by translating short-term predictions into long-term predictions, humans gradually learn to generalize contexts in a bottom-up manner. Second, based on the metacognitive process to self-evaluate her long-term predictions, humans learn to combine model-free and model-based RL in a top-down manner, leading to short-term predictions suitable for context changes. Moreover, the bottom-up and top-down processes are not separable; they co-occur as a single dynamical system, creating a virtuous cycle of generalization-adaptation (Figure 4A). We name this the long short-term prediction hypothesis.

To avoid any misinterpretation, we should note that our results do not imply that deep RL or other variants cannot generalize their policy. It is mainly because all RL models in our study were trained to learn the policy of humans as opposed to performing the task by themselves. In this regard, this suggests an interesting implication that the way deep RL algorithms solve tasks is distinctly different from that of humans. Our framework allows RL algorithms to glean valuable insights from human problem-solving processes.

Although machine learning has focused on task transfer, generalization, or adaptation, these issues have been examined separately. Our work offers a unified view: (1) translating short-term into long-term prediction about rewards and states (dubbed as long short-term prediction) enables bottom-up context generalization, (2) orchestrating short-term prediction based on long-term prediction serves as top-down context adaptation, (3) a functional cycle of generalization-adaptation within a single network implements predictive meta RL. Our theoretical insight helps AI veer in a new direction for predictive meta RL: metacognitive long-term and short-term evaluation of external environments and internal valuation.
Understanding computational principles underlying context generalizability and adaptability in humans would lend valuable insight into how machine learning algorithms can adapt to the specific context by transferring the generalized knowledge ([14, 38–41]). There has been a wide range of algorithmic resolutions focusing on a shared structure of state space an agent can exploit across the environments ([14, 15, 41–47]). Further neuroscientific investigations in this regard would improve biologically-plausible algorithm design, such as successor representation ([48–50]), or meta learning ([15]).

Methods

Two-stage Markov decision task

The two-stage MDT was used as an environment [23] 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 goals. 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.

Empirical likelihood estimation

RL algorithms have several policies to select an action from values, such as greedy policy, $\epsilon$-greedy policy, and softmax. To compare various RL algorithms on the same basis, we empirically estimated likelihood by calculating mean hit rate, the ratio of matched models’ simulated choices to human choices.
Computational models of RL

We used four different categories of RL models to conduct the tests (Figure 2A). The first category is Model-free RL which was implemented with SARSA [51], Double DQN [52] also known as DDQN and distributional RL [53]. It is one of the typical deep RL models approximating model-free RL. The second category is Model-based RL, implemented with Forward, Successor representation (SR; [48–50]) and meta RL [14, 15]. This category accommodates both model-free and model-based RL. In particular, meta RL is known to adaptively respond to contextual changes in the environments. We used the goal matching and policy matching methods to train this model (GM-metaRL and PM-metaRL, respectively). The third category of RL model was implemented with the computational model to account for the neural activity of the lateral prefrontal cortex and ventral striatum (prefrontal RL) [23–25]. There are two versions of this model: the baseline model [23] and the adaptive model [25]. These models learn a task by dynamically arbitrating between model-free and model-based RL. Specifically, they adjust on a trial-by-trial basis the degree of control allocated to the model-free and model-based RL strategies, and this top-down control signal is computed based on the prediction reliability of each RL strategy. We used the policy matching method to train these two models (PM-pfcRL1 and PM-pfcRL2, respectively). We did not use goal matching in this case because previous studies have found that this method is not effective in fitting these models to data [23]. We additionally considered a standard inverse RL implemented with GAIL [54].

Here, we laid out each category of RL algorithms in the order of its methods for value updates, and identified which behavioral profile of the RL algorithm belongs to.

Model-free RL

SARSA We implemented SARSA RL [51], to be consistent with the MF RL that is used in pfcRL, which is believed to be the model that is closest to the human brain.

Double deep Q-network (DDQN) DQN is widely used as a standard MF deep RL algorithm, after its huge success on video game (REF: Minh, 2015). A DDQN, on the other hand, is a variant of DQN that has two copies of value networks to resolve some issues in DQN such as high correlation of histories, etc. DDQN is a deep reinforcement learning algorithm designed to effectively maintain a thematic consistency between value and policy.

\[ Y_t^{DDQN} \equiv R_{t+1} + \gamma Q(s_{t+1}, \text{argmax } Q(s_t, a; \theta_k), \theta_k^-) \]

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 \( \gamma \) is 0.99, the learning rate \( \alpha \) is 0.001, the target update frequency to the primary network \( \tau \) is 0.001, and the batch size is 32.
Implicit Quantile Network (IQN) An IQN is a distributional RL algorithm, which is inherently a MF RL with distributional value representation [53]. There are other and even simpler algorithms for distributional algorithms as well (e.g. C51, QR-DQN), but IQN uses uniformly sampled quantiles so that it can learn any value distribution no matter how skewed it is.

Model-based RL

FORWARD We implemented the FORWARD MB model which updates its own state-transition probability matrix based on Temporal difference learning [20]. This MB RL is consistent with the MB RL of pfcRLs.

Meta-RL 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. Wang et al. [15] showed that meta-RL can represent human-like behaviors in two-stage MDP, which is widely agreed as behavioral evidence of mixed MB and MF of human mind.

SR-DYNA Successor representation is a reinforcement learning algorithm that stores future state-transition occupancies. By doing so it can endow MB-like behavior (Dayan 1993). Ida Mommenejad employed Sutton’s Dyna framework, which is an offline update of successor representation based on simulation and MB planning. This algorithm is particularly useful in tasks requiring goal-directed behavior, such as our generalizability.

Inverse RL

Generative adversarial imitation learning (GAIL) Inverse reinforcement learning algorithms are generally used to infer value function from the behaviors, which directly relates to the model’s ability to generalize what can be learned in a task. To fully address all varieties of RL algorithms with various training methods (especially including PM and BC), we need to compare them with IRL algorithms. As a representative candidate of IRL, we used GAIL [54] that marked expect-level performance, especially by utilizing Generative Adversarial Networks (GANs) that is broadly used in machine learning because of its sample-efficient generalization.

Prefrontal RL

For prefrontal RL, we used the computational model adaptively combining MB and MF control, inspired from the findings on the neural activity of the lateral prefrontal cortex and ventral striatum [23, 24]. Two versions were selected: a baseline model for pfcRL1 [23] and an adaptive model for pfcRL2 [25]. Both models successfully implement arbitration control based on the Bayesian reliability estimation over MB RL (FORWARD learning [20]) and MF RL (SARSA...
Long short-term prediction guides human metacognitive reinforcement learning

Learning [51]), pfcRL2 [25] is slightly different from the original one (pfcRL1). pfcRL2 computes Bayesian reliability based on the clusters of PE distribution estimated by the Dirichlet process Gaussian mixture model, while pfcRL1 simply divides PEs with the predefined threshold. By considering the PE distribution to be a Gaussian mixture model, pfcRL2 is able to incorporate dynamic changes of task variables in the environment.

Random

We also used a random agent as an agent for the control group. The random agent chose the action randomly while performing any two-stage MDP.

A Markov-decision process subject to the training methods (PM, GM, BC)

With the aforementioned two-stage MDT, we formalize the choice behavior of agents as an MDP, which is a 5-tuple \((S, A, T, R, \gamma)\), where \((S)\) is the state space, \((A)\) 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 [52] and meta RL [14, 15], 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 \(S_1, S_2 \ldots S_9\), 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:** \((T : S \times A \times S \rightarrow [0, 1])\) 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:** \((R : S \times A \times R \rightarrow R)\) 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 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:** \((R : S \times A \times R \rightarrow R)\) 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 \((R_\Omega)\) is defined as

\[
R_\Omega = \begin{cases} 
  k - n, & \text{if } a_{ag}^1 \neq a_H^1 \text{ and } a_{ag}^2 \neq a_H^2 \\
  k + n, & \text{if } a_{ag}^1 = a_H^1 \text{ and } a_{ag}^2 = a_H^2 \\
  k, & \text{otherwise} 
\end{cases}
\]

(1)

where \((k, n)\) is a predefined constant value, \((a_{ag}^i)\) is an action that the agent performed at stage \((i)\) in the two-stage MDT, and \((a_H^i)\) is an action that a human subject originally performed at the same stage \((i)\).

The practical meaning of \((R_\Omega)\) 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 \((R_\Omega)\) 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.
Behavior cloning (BC)

In this training method, an RL agent was trained in a way that it mimics the way the human performs reward maximization while performing a two-stage MDT. The agent gets rewarded (+1) when the action the agent chose matches with the action that the human subject made, while losing reward (−1) otherwise. The main assumption of the BC training is that the

• Reward: \( R : S \times A \times R \rightarrow R \) 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.

Fitting RL models to human data (training)

In the GM and BC training, we used a fixed number of episodes for all deep RL models to train (number). However in the PM training, we stopped the PM training of deep RL models (i.e. DDQN, meta-RL, IQN) earlier when its likelihood summation in a single episode was greater than the likelihood summation of PM-pfcRL1 and PM-pfcRL2 for computational efficiency. 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 \( ?? \) to compute the likelihood,

\[
\text{Likelihood } L = p(as_i),
\]

\( (a) \) is the human subject’s choice at the \( (i^{th}) \) trial in state \( (s) \).

Behavioral measurements

We used normalized reward to measure an agent’s performance that is normalized by dividing reward with maximum reward in a given trial. We used it to quantify an agent’s generalizability in multiple tasks because it simplifies the reward to be ranged from 0 to 1 and make comparison of agents fair.

However, where the various task parameters continually and continuously change over time, normalized reward is inapplicable because it becomes a noisy measure when rewards are sparse. For example, although the theoretical maximum reward in a given trial is high, it is often unattainable. In this case, an agent should learn what is the optimal action that can maximize both frequency and magnitude of rewards.

Thus, we employed choice optimality as another behavioral measure to measure an agent’s performance, especially to quantify adaptation performance. Choice optimality is a binary value (i.e. 1 or 0) that is the proximity of
an agent’s choices to the optimal agent’s choices, and is often used to assess the
degree of engagement of MB in the context of the arbitration control between
MB and MF [25, 55]. It also can be defined in terms of mutual information
(See Episodic encoding efficacy test via information theoretic analysis).

Parameter recovery analysis - Behavioral profile
correlation test
To explicitly test the RL algorithm’s recoverability of human behavior, we con-
ducted behavioral profile analysis [24]. Behavioral profile analysis is based on
a generalized linear model (GLM), using task contexts in the environment as
independent variables and a choice optimality, which is the behavioral mea-
surement of MB control [24], was chosen for a dependent variable. Thus, the
model was.

\[ y = \beta_1 x_1 + \beta_2 x_2, \]

Note that these two task contexts are goal condition \((x_1)\) and state-
transition uncertainty \((x_2)\) in Lee et al. (2014), and the only variable that
possibly modulates human behavior during the task. As described in [24],
choice optimality was the key behavioral measure for explaining human RL’s
adaptive behavior using arbitration control between model-based and model-
free RL [23, 24], and two sets of coefficients can be extracted from the result of
GLM using human’s actual behavior and models’ simulated behaviors. Then,
models’ recoverability of human RL’s flexible behaviors can be quantifiably cal-
culated using correlation coefficient between regression coefficients extracted
from human behavior \(\beta_{human}\) and regression coefficients extracted from mod-
els’ simulated behavior \(\beta_{model}\). Positive and significant results of both task
contexts indicate that the model is truly recovering human RL’s behavioral
adjustment to task contexts in the environment.

Episodic encoding efficacy test
Following a recent work [56], we computed (1) the mutual information from
the episodic events and the agent’s action [56] (“episodic encoding efficiency”)
and (2) the mutual information of the agent’s action and the optimal action
(“choice optimality”). First, the episodic encoding efficiency is defined as
\(I(F_{t-1}; a_t)\), where \(F_{t-1}\) and \(a_t\) are the episode variable at trial \(t - 1\) and the
action at trial \(t\), respectively. The episode \(F_t\), is defined as
\[ F_t = \{a_{t-1}^2, S_{t-1}^3, a_{t-1}^{2,*}, R_{t-1}\}, \]

where \((S_{t-1}^3)\) is a state at stage 2 in the previous trial, \((a_{t-1}^2)\) is an action
that was taken by an agent at stage 2 in the previous trial, \((S_{t-1}^3)\) is a rewarding
state at stage 3 in the previous trial, \((a_{t-1}^{2,*})\) is an optimal action at stage 2
in the previous trial, and \((R_{t-1})\) is a reward in the previous trial. We then
computed \( I(F_t, a_t) \) to measure the amount of mutual information between the current action and the history of the previous trial.

Second, the choice optimality is defined as \( I(a_t; a^*_t) \), where \( a_t \) and \( a^*_t \) are the actions of the RL agent and ideal agent with which the current context information is fully provided. It is a 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 \([56]\).

Finally, we computed the correlation between episodic encoding efficiency and choice optimality (“episodic encoding efficacy”). It can be a potential information-theoretic measure to predict generalizability of RL models.

**Modeling distribution of two-step MDT that embraces real-world challenges**

We created a task distribution to embrace the volatile context changes of the real-world. The task distribution is defined with the following 5-tuple [1) task structure, 2-4) payoff probabilities of three goals, 5) state-transition probability].

For the task structure we have a binary value that indicates whether it is “ladder” or “tree”. For the “ladder” state transition, each of which between left and right choice has the exact opposite state-transition probability of the next state. The “ladder” task structure is shown in the pioneering work by Daw et al. (2005). For the “tree” task structure, each choice results in a different branch. Unlike the “ladder” task structure, an agent needs to reevaluate at every stage, which increases the needs for goal-directed (or MB) control and arbitration control as well.

For the pay-off probabilities, which are defined for three rewards in the environment (40, 20, 10) are ranged from 0 to 1. Daw et al. (2005) noted that the perturbation of pay-off probabilities is required for continual necessity of learning, not satisfying with and converging on habitual learning (or MF). This is the same for the state-transition probability, which is ranged from 0 to 1. We updated each in three different ways: “switch”, “drift”, “noisy shift”. First, “switch” refers to the changes of either payoff probabilities and state-transition probability between (0.5, 0.5) and (0.9, 0.1) every 5 trials \([23]\).

“Drift” refers to the Gaussian random walk as in Daw et al. (2005), with the same step size they reported in the paper: \( p \leftarrow p + \mathcal{N}(0, 0.025) \).

“Noisy-shift” refers to the intermediate way of “switch” and “drift”, which switches and then drifts with Gaussian random walk. With same step size ( \( \sigma = 0.025 \)). Lastly, we have “fixed”, which does not change but have fixed probabilities of (0.7, 0.3).
Task switching scenario

Given task distribution defined with 5-tuples, we sampled 10 tasks to test RL models general performances (i.e. generalizability; see Fig. 3B?). We specifically included two tasks that used to highlight that humans use a mixture of MB and MF, and even has an additional hierarchical process called arbitration control between MB and MF [23, 36]. Task switches every 393 trials (a number of trials of a participant who played approximately equal number of trials for each condition in Lee et al. (2014), while performing comparably great in the task), and totally 10 tasks * 393 trials = 3930 trials are played by each RL model.

Context changing scenario

To further test adaptability of agents we manipulated the five-tuple parameters every 5 trials, which reportedly is enough to recognize the changes and get adapted to it [23]. We applied a Gaussian random walk process with step size 0.0025 to probability parameters, and the task structure was retained the same at least 1,000 trials while the number of trials for both task structures were matched. Totally 10,000 trials were simulated for one batch and repeated 100 batches of trials for an agent.

Context sensitivity analysis

Linear regression model was fitted with binarized PEs and context vectors as a dependent and independent variable, respectively. PEs and context vectors were taken from the adaptability test. After fitting, we ran 100 simulations using the linear regression model and analyzed a discrepancy (e.g., residual error) between the model’s estimation on PE signals and ground truth of them in every trial per simulation. We labeled each trial as PE sensitive when the discrepancy between one type of PE signals was statistically smaller than that between the other. The PE sensitive trials can be viewed as the context embedding trials. For example, we labeled one trial as SPE sensitive when the residual error between estimated and ground truth SPE was statistically smaller than that between the estimated RPE and the ground truth RPE, and RPE sensitive otherwise. The same labeling criterion was applied to all other PE signals.

Proofs of Lemma 1 and Lemma 2

Here we provide proofs of Lemmas.
Proof of Lemma 1

\[ S_n = \sum_{j=0}^{2} | \beta_{0,j} - \beta_{n,j} | \]
\[ = \sum_{j=0}^{2} | \beta_{0,j} - \beta_{1,j} + \beta_{1,j} - \beta_{2,j} + \cdots - \beta_{n,j} | \]
\[ \leq \sum_{i=1}^{n} \sum_{j=0}^{2} | \beta_{i-1,j} - \beta_{i,j} | \text{ by the triangle inequality} \]
\[ = D_n \]

Proof of Lemma 2

\[ \mathbb{E} [D_n | X_{n-1}, \ldots, X_1] \]
\[ = \mathbb{E} \left[ \sum_{i=1}^{n} X_i | X_{n-1}, \ldots, X_1 \right] \]
\[ = \mathbb{E} \left[ \sum_{i=1}^{n-1} X_i | X_{n-1}, \ldots, X_1 \right] + \mathbb{E} [X_n] \text{ by independence of } X_i \]
\[ = D_{n-1} + \mu_n \]
\[ \geq D_{n-1} \text{ since } \mu_i \text{ is positive for all } i. \]

Proof of Theorem (Azuma, 1967)

Lemma 3 Let a random variable \( Y \) has positive mean \( \mu \) and \( Y \leq c \). Then \( \mathbb{E}[e^{\delta Y}] \leq (1 + \frac{\mu}{\delta^2 c^2}). \)

Proof of Lemma 3

Let \( f(y) = e^{\theta y} \). Since \( f \) is convex,
\[ f(y) \leq \frac{c-y}{2c} f(-c) + \frac{c+y}{2c} f(c) \]
for all \( y \in [-c, c] \). Then taking expectations on each side returns
\[ \mathbb{E}[f(Y)] \leq \mathbb{E} \left[ \frac{c-y}{2c} f(-c) + \frac{c+y}{2c} f(c) \right] \]
\[ = \frac{1}{2} (f(c) + f(-c)) + \frac{\mu}{2c} (f(c) + f(-c)) \]
\[ \leq \frac{1}{2} (f(c) + f(-c)) + \frac{\mu}{2c} (f(c) + f(-c)) \text{ since } f(-c) > 0 \forall c \]
\[ = \left( 1 + \frac{\mu}{c} \right) \frac{1}{2} (f(c) + f(-c)). \]
Lastly, Taylor expansion of the RHS can be written as

\[
\frac{e^{-\theta c} + e^{\theta c}}{2} = \sum_{n=0}^{\infty} \frac{(\theta c)^{2n}}{(2n)!} \leq \sum_{n=0}^{\infty} \frac{(\theta c)^{2n}}{2^n n!} = \exp \left( \frac{1}{2} \theta^2 c^2 \right)
\]  

Similar logic is applied to prove the case when \( Y \) is a martingale with some filtration \( G \) i.e., \( \mathbb{E}[Y | G] = \mu \).

**Lemma 4** Suppose \( D \) is a submartingale with \( D_0 = 0 \) and \( |D_n - D_{n-1}| = |X_n| \leq c \) a.s. for all \( n \) and \( \mathbb{E}[X_n] = \mu_n \). Then \( \mathbb{E}[e^{\delta D_n}] \leq C_n \exp \left( \frac{1}{2} \delta^2 c^2 n \right) \), where \( C_n = \prod_{i=1}^{n} \left( 1 + \frac{\mu_i}{c} \right) \).

**Proof of Lemma 4**

Let \( W_n = e^{\theta D_n} \), where \( W_n \geq 0 \) and \( W_n = W_{n-1}e^{D_n-D_{n-1}} = W_{n-1}e^{X_n} \). Applying **Lemma 3** with filtration \( G_{n-1} = (X_1, X_2, \ldots, X_{n-1}) \) returns

\[
\mathbb{E}[W_n | G_{n-1}] = W_{n-1} \mathbb{E}[e^{\theta X_n} | G_{n-1}] 
\leq W_{n-1} \left( 1 + \frac{\mu_n}{c} \right) \exp \left( \frac{1}{2} \theta^2 c^2 \right) \text{ a.s.}
\]

Again, taking expectations on each side returns

\[
\mathbb{E}[W_n] \leq \mathbb{E}\left[ W_{n-1} \left( 1 + \frac{\mu_n}{c} \right) \exp \left( \frac{1}{2} \theta^2 c^2 \right) \right],
\]

and the result follows by the induction.

**Proof of Theorem**

\[
P(D_n \geq \lambda) = \mathbb{P}(e^{\theta D_n} \geq e^{\theta \lambda}) \leq C_n e^{-\theta \lambda} \exp \left( \frac{1}{2} \theta^2 c^2 \right)
\]  

The last line comes from the Chernoff bound and **Lemma 4**.

**Declaration**

**Supplementary information**

Supplementary Sections 1–3, including Figs. 1–6, and References.
Data and Code availability
The data and code to recreate our analyses is available at https://github.com/dongjae-kim/simul-mdps

Authors’ contributions
DK conducted simulations and analyzed data. WJ proved theorems. DK and JHL contributed equally to this manuscript. SWL supervised this project. All authors wrote the manuscript together.

Acknowledgments
We sincerely appreciate Samuel Gershman for providing valuable feedback on the preliminary draft of our work. This work is supported by SW star lab, the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2019M3E5D2A01066267), Samsung Research Funding Center of Samsung Electronics under Project Number SRFC-TC1603-52. WJ received Catherine Hughes Fund from Somerville College, University of Oxford in 2020 during the internship at KAIST.

References
Long short-term prediction guides human metacognitive reinforcement learning


Long short-term prediction guides human metacognitive reinforcement learning


Long short-term prediction guides human metacognitive reinforcement learning


Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- 239350supp178055rgjg0.pdf