Modelling human exploration with light-weight meta reinforcement learning algorithms

Thomas D. Ferguson, Alona Fyshe, Adam White

Keywords: exploration, multi-arm bandit, IDBD, meta-learning, non-stationary

Summary

Learning in non-stationary environments can be difficult. Although many algorithmic approaches have been developed, often methods struggle with different forms of non-stationarity such as gradually changing versus suddenly changing contexts. Luckily, humans can learn effectively under a variety of conditions and using human learning could be revealing. In the present work, we investigated if a stateless variant of the IDBD algorithm (Mahmood et al., 2012; Sutton, 1992), which has previously shown success in bandit-like tasks (Linke et al., 2020), can model human exploration. We compared stateless IDBD to two algorithms that are frequently used to model human exploration (a standard Q-learning algorithm and a Kalman filter algorithm). We examined the ability of these three algorithms to fit human choices and to replicate human learning within three different bandits: (1) non-stationary volatile which changed suddenly, (2) non-stationary drifting which changed gradually, and (3) stationary. In these three bandits, we found that stateless IDBD provided the best fit of the human data and was best able to replicate different aspects of human learning. We also found that when fit to the human data, differences in the hyperparameters of stateless IDBD across the three bandits may explain how humans learn effectively across contexts. Our results demonstrate that stateless IDBD can account for different types of non-stationarity and model human exploration effectively. Our findings highlight that taking inspiration from algorithms used with artificial agents may provide further insights into both human learning and inspire the development of algorithms for use in artificial agents.

Contribution(s)

1. Our work is the first to investigate a light-weight, meta-learning algorithm from reinforcement learning (IDBD) as a potential computational model of human exploration. Recovery of IDBD parameters and simulation results from our human data provides suggestive evidence that people modulate their learning rates in a similar manner to IDBD.
   Context: Our work may be limited to the bandit setting, as human data in multi-stage decision making tasks is typically modelled using hybrid model-free/model-based (e.g., successor representation) approaches (Momennejad et al., 2017).
2. Although prior work has shown IDBD-based agents can automatically and continually adapt step-sizes to improve performance in simulation, we are the first to show IDBD can do the same with human exploration data (i.e., a sequence of actions and rewards generated by people performing bandit tasks).
   Context: IDBD-inspired agents have been used in supervised learning tasks (Sutton, 1992; Mahmood et al., 2012), bandit tasks (Linke et al., 2020), MDPs (Mcleod et al., 2021; Kearney et al., 2018; Javed et al., 2024; Jacobsen et al., 2019), and even to help predict data from real robots (Mahmood et al., 2012; Kearney et al., 2018; Jacobsen et al., 2019)
3. Our analysis indicates that IDBD matches human data better when compared to a Q-learning and a Kalman filter algorithm which were used as baselines. (Daw et al., 2006; Hassall et al., 2019).
   Context: Our results are limited to three tasks and a moderate number of human participants. It is always possible that different tasks or a larger number of participants could produce different conclusions. We did not exhaustively study all computational models proposed in the literature, but instead focused on two: a Q-learning algorithm (Hassall et al., 2019) and a Kalman filter algorithm (Daw et al., 2006)

Modelling human exploration with light-weight meta reinforcement learning algorithms

Thomas D. Ferguson[1,2], **Alona Fyshe**[1,2,3,4], **Adam White**[1,2,3]
{tfergus2,fyshe,amw8}@ualberta.ca

[1]**Department of Computing Science, University of Alberta, Canada**
[2]**Alberta Machine Intelligence Institute (Amii)**
[3]**CIFAR AI Chair**
[4]**Department of Psychology, University of Alberta, Canada**

Abstract

Learning in non-stationary environments can be difficult. Although many algorithmic approaches have been developed, methods often struggle with different forms of non-stationarity such as gradually changing versus suddenly changing contexts. Luckily, humans can learn effectively under a variety of conditions so using human learning could be revealing. In the present work, we investigated if a stateless variant of the IDBD algorithm (Mahmood et al., 2012; Sutton, 1992), which has previously shown success in bandit-like tasks (Linke et al., 2020), can model human exploration. We compared stateless IDBD to two algorithms that are frequently used to model human exploration (a standard Q-learning algorithm and a Kalman filter algorithm). We examined the ability of these three algorithms to fit human choices and to replicate human learning within three different bandits: (1) non-stationary volatile which changed suddenly, (2) non-stationary drifting which changed gradually, and (3) stationary. In these three bandits, we found that stateless IDBD provided the best fit of the human data and was best able to replicate different aspects of human learning. We also found that when fit to the human data, differences in the hyperparameters of stateless IDBD across the three bandits may explain how humans learn effectively across contexts. Our results demonstrate that stateless IDBD can account for different types of non-stationarity and model human exploration effectively. Our findings highlight that taking inspiration from algorithms used with artificial agents may provide further insights into both human learning and inspire the development of algorithms for use in artificial agents.

1 Introduction

Often algorithms struggle to deal with non-stationary contexts. While many approaches have been developed to deal with non-stationary contexts (Gupta et al., 2011; Garivier & Moulines, 2011; Mcleod et al., 2021; Linke et al., 2020; Padakandla et al., 2020; Padakandla, 2021; Jain et al., 2024; Khetarpal et al., 2022; Chandak, 2022), one problem is that they may be unable to handle different types of non-stationary contexts. For example, consider a context which is slowly changing – which may require consistent but slow rates of exploration – in comparison to an context which changes suddenly – which would require exploration and a sudden increase in learning rate when the context changes. However, humans are excellent at learning in non-stationary contexts like these (e.g., (Soltani & Izquierdo, 2019; Lee et al., 2023; Payzan-Lenestour & Bossaerts, 2011). Taking inspiration from human learning capabilities could be useful for both deciding which algorithmic directions are promising and focus future algorithm refinement. One ability which allows us to engage in meta-learning is solving the explore-exploit dilemma (Cohen et al., 2007). Specifically, one example of meta-learning in humans is the tuning of hyperparameters in algorithms of learning

and exploration depending on context (Griffiths et al., 2019). While the development of models of human learning under uncertainty (that is, non-stationarity) has been successful (e.g., (Behrens et al., 2007; Daw et al., 2006), much of the focus has been on contexts which only show one form of non-stationarity (e.g., a suddenly changing world or a gradually changing world). If researchers are to take inspiration from human learning, then validating algorithms which can handle different types of non-stationarity should be of importance. As such, the use of light-weight, meta-learning models – which may not require the tuning of numerous hyperparameters – provides one step forward.

Luckily, work in artificial agents provides one such family of light-weight meta-learning algorithms. The Incremental Delta-Bar-Delta (IDBD) algorithm employs a simple update rule and was able to learn within a non-stationary context through adjusting individual step-size parameters for every input (Sutton, 1992). IDBD has been extended to have no sensitive hyperparameters – known as Autostep – where it was able to solve different non-stationary problems and did not require extensive tuning (Mahmood et al., 2012). Autostep works by increasing the step-size parameter of an input (in the case of a bandit task, each bandit arm) when learning is progressing (i.e., the prediction error of an input is in a consistent direction) while decreasing the step-size parameter when learning is not progressing (i.e., the prediction error is not consistent) through a memory trace of prior prediction errors. Autostep/IDBD has been successfully applied to bandit-like tasks which required the individual tracking of multiple arms each with their own step-size parameter (Linke et al., 2020). The application of a light-weight meta-learning algorithms like Autostep/IDBD (hereinafter: stateless IDBD) to human learning data across contexts may be useful.

We examined both human and artificial agent performance across different learning contexts. Specifically, we examined performance within: (1) a non-stationary bandit where the best arm (i.e., the arm that produced the highest reward) would change suddenly across time-steps, (2) a non-stationary bandit where the best arm would change gradually across time-steps, and (3) a stationary bandit where the best arm remained consistent. The different levels of non-stationarity provide an appropriate paradigm for testing learning across contexts. We collected a large sample of human participants (n = 204) who each completed two of the bandit contexts. The primary algorithm we were interested in was the stateless IDBD algorithm. Stateless IDBD has not been formally validated for use with humans in a multi-arm bandit so we conducted two steps to validate the algorithm for use in humans: "parameter recovery" and "model recovery" (Wilson & Collins, 2019). We ran parameter recovery to determine whether the hyperparameters had distinct effects on task performance. We ran model recovery to determine whether the algorithm made distinct behavioural predictions in our tasks compared to two baseline algorithms. Specifically, to compare stateless IDBD to other algorithms, we examined two popular algorithms often used to model exploration in humans: a simple Q-learning algorithm which relies on a static step-size for all arms (Hassall et al., 2019; Ferguson et al., 2023), and a Kalman filter model where the step-size can change (Daw et al., 2006; Speekenbrink & Konstantinidis, 2015). To compare our three algorithms, we examined how well each of them fit human choices in the bandits and how well they could simulate human learning.

In the present work, we provide five key findings. First, stateless IDBD showed strong parameter recovery and model recovery, suggesting that it is a good candidate to be applied to human learning. Second, we found that stateless IDBD provided the best fit of the majority of the human participants across all three bandit contexts. Third, we found stateless IDBD best replicated human learning compared to our two baseline algorithms. Fourth, we found that stateless IDBD provided the best evidence of transfer learning compared to our two baseline algorithms. Fifth, we found that stateless IDBD's best-fitting hyperparameters based on the human choices differed depending on the bandit context, which may be tied to human meta-learning. In sum, we found that stateless IDBD algorithm was more successful at modelling human exploration than our two baselines. Our findings have implications both for research on human learning and for algorithm development in artificial agents.

2 Problem Setting: Human Bandit Tasks

In the present work, we had agents complete two of three multi-arm bandits (Figure 1). Each of the bandits required completing 300 total time-steps. For the two non-stationary bandits, agents completed three blocks of 100 time-steps. For the stationary bandit, agents completed six blocks of 50 time-steps. Agents completed all blocks of one bandit followed by all blocks of the second bandit.

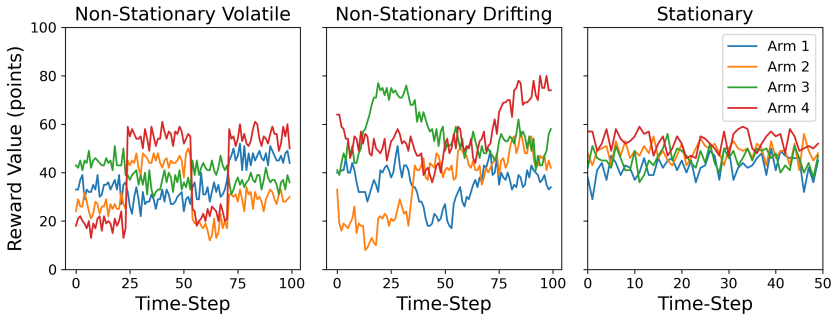

Figure 1: The three multi-arm bandits used in the present work. The reward distributions show the reward values of the four arms for an example block of each bandit that agents encountered.

We believe that when humans completed our learning task, they were completing a multi-arm bandit and not a Markov decision problem. First, there was no additional contextual (state) information to help them solve the task and the task was designed to focus their attention on the immediate block. When completing the task, they saw the four bandit arms represented as different coloured squares, were required to select a bandit within 2000 ms on each time-step, and all participants completed the tasks within 50 minutes. Second, the task was not a sequential decision-making task and choices did not involve future discounted reward. That is, all the rewards of the four arms were independent of each other and of the next time-step (or block/task).

For the non-stationary drifting bandit, we used a task where the reward values of the four arms changed across time-steps randomly and independently (Daw et al., 2006). On each block, the rewards were randomized for each of the four arms, and followed a random walk that drifted towards 50 points on each time step. Specifically, the rewards for each time-step were drawn from a Gaussian distribution with a mean ($\mu_{j,k}$) and a standard deviation which was equal to three. To calculate the mean of the Gaussian distribution for each arm (k) and time-step (j), the point values of the arms were updated using a Gaussian random walk: $\mu_{j+1,k} = \lambda\mu_{j,k} + (1-\lambda)\theta + \nu$, where λ is a decay parameter equal to 0.9836, θ is the decay center (equal to 50), and ν is a diffusion noise parameter. On each time-step, the diffusion noise parameter is sampled from a Gaussian distribution with a mean of zero and a standard deviation of 2.8.

For the non-stationary volatile bandit, we developed a bandit where the reward values of the four arms would change suddenly. For each block, one arm's mean was initialized to be between 30 and 90 points, while the other arms had their means shifted by: -8, -16, and -24 points relative to the first arm. On each time-step, the values of each of the arms were sampled from a Gaussian distribution with the specified mean values and standard deviation equal to three. Following the completion of between 20 to 30 time-steps, the mean reward of each arm were shifted by between 10 and 15 points.

For each block in the stationary bandit, the first arm's mean was set to be between 30 and 90 points, while the other three arms had their means shifted -5, -10, and 5 points relative to the first arm. For the stationary bandit, the reward on each time-step was drawn from a Gaussian distribution using the means specified above and a standard deviation of three.

Human Participants We collected data from 204 people. 107 people completed both the drifting and volatile bandit, while 97 people completed both the volatile and the stationary bandit (see 6 for the instructions participants received). Participants were recruited from the local institution, compensated with course credit, and completed a consent form. All experiments were conducted with the approval of the local institution's research ethics board.

3 Model Fitting & Algorithms

To find the best fitting parameters of each algorithm from the human choices, we applied the Bayesian optimization algorithm from the PyBADS package (Version 1.0.5) in Python (Version 3.9). Algorithm parameters were optimized individually for each participant within each of the three bandits (each participant ended up with three sets of parameters; one set per bandit). Using the Softmax choice probabilities of each of the algorithms, we applied a posteriori estimation based on the minimization of the negative log-likelihood across all time-steps (t) and blocks (b) per the input set of hyperparameters (Daw, 2011).

The goal is to select hyperparameters for each algorithm that would most likely recreate the choices in the human data. Assume we pick hyperparameter set w which generates a sequence of corresponding softmax probabilities $P_{i,j,w}$ for block i and time-step j. Then we compute the negative log-likelihood of the observed actions a_k which is the selected arm on each block and time-step.

$$\ell(w) = -\sum_{i=1}^{b}\sum_{j=1}^{t} \ln P_{i,j,w}(a_k). \tag{1}$$

The actual loss given to the Bayesian optimizer is the AIC = $2p + 2\ell(w)$ where p is the number of hyperparameters. We use the AIC, instead of the negative log-likelihood, to penalize model complexity.

Stateless IDBD The stateless IDBD algorithm relies on Autostep (Mahmood et al., 2012) which has been successfully applied to non-stationary bandit-like tasks (Linke et al., 2020). The stateless IDBD algorithm involves the calculation of an individual step-size parameter for each of the arms, which changes on each time-step per the sign of the prediction error (see Algorithm 1 in 7).

The meta-learning rate parameter (κ) determines how quickly the individual step-size parameters change. Because of the low number of time-steps, the meta-learning rate parameter did not have any appreciable effect on performance, and we chose to keep it constant (.15) across each of the bandits. While we found that this meta-learning rate parameter maximized performance across 10000 simulations, there was little effect on performance overall.

For stateless IDBD, we updated value estimates using model-free reinforcement learning (Sutton & Barto, 2018). Specifically, we had the value estimates updated for chosen arms on each time-step by multiplying the prediction error of the chosen arm by a step-size parameter (α). On each time-step, the value estimates for the selected arm (k) were updated by: $q_{i,j+1,k} = q_{i,j,k} + \alpha_k \times \delta_{i,j}$, where α_k is the step-size for an arm and $\delta_{i,j}$ is the prediction error: $\delta_{i,j} = r_{i,j} - q_{j,k}$, and $r_{i,j}$ is the reward obtained from the selected arm. For the stateless IDBD algorithm, we fit three hyperparameters per person and per bandit: the inverse temperature parameter of the softmax policy, the initial Q value for each block, and the initial step-size parameters for each block.

Q-Learning Baseline For the Q-learning baseline, the algorithm relied on the Q-update as in Stateless IDBD. However, we instead fit a single step-size (α) for all arms. We initialized each of the arm values (q) optimistically on each block as 100. We note that while we attempted to fit the initial Q values for this algorithm, we were unable to successfully recover the initial Q value estimates – a point we will return to in the discussion. For the Q-learning baseline algorithm, we fit two hyperparameters per person per bandit: the inverse temperature parameter from the softmax policy, and the single step-size parameter for all arms.

Kalman Filter Baseline For a second baseline, we examined a Kalman filter algorithm (Kalman, 1960). The Kalman Filter algorithm used approximate Bayesian updating for changing the agent's value estimates across time-steps. In comparison to Q-learning, the Kalman filter incorporates a variable step-size (the Kalman Gain) which changes across time. On each time-step the value estimate (q; initialized to 100 on each block) and the variance estimate (v; initialized to 100 on each block) of for each arm are updated by:

$$q_{i,j+1,k} = q_{i,j,k} + \text{KG}_{i,j,k}(r_{i,j} - q_{i,j,k}) \tag{2}$$

$$v_{i,j+1,k} = (1 - \text{KG}_{i,j,k})v_{i,j,k} + \sigma_\xi^2. \tag{3}$$

The Kalman Gain was updated by

$$\text{KG}_{i,j,k} = \begin{cases} \frac{v_{i,j,k}+\sigma_\xi^2}{v_{i,j,k}+\sigma_\xi^2+\sigma_\epsilon^2} & \text{if k = selection} \\ 0 & \text{otherwise} \end{cases}, \tag{4}$$

where σ_ξ^2 is the innovation variance parameter and σ_ϵ^2 is the error variance parameter. These two variance parameters determine how the Kalman Gain updates on each time-step. Only one of these two noise parameters can typically be recovered successfully from human choices, because it is the ratio of the two parameters that determines how arms are selected and how values are updated (Piray & Daw, 2024). Thus, the only free parameter was the error variance parameter while we set the innovation variance parameter to a constant value of five (similar to previous work - (Daw et al., 2006). For the Kalman Filter baseline algorithm, we fit two hyperparameters: the inverse temperature parameter from the softmax policy, and the error variance parameter.

Softmax Policy To ensure our algorithms could be easily compared, we used a softmax policy. The softmax policy involves probabilistic random exploration, where agents usually select the highest estimated value arm, while occasionally exploring the other arms in decreasing probability depending on their estimated value. The inverse temperature parameter (τ) determines how often exploration occurs. The softmax policy relies on the formula: $P_{i,j}(a_k) = \frac{\exp(\tau \times q_{i,j,k})}{\sum \exp(\tau \times q_{i,j})}$.

4 Experimental Results

Below we provide the main results. First, stateless IDBD was validated for use in humans as the algorithm demonstrated good parameter and model recovery. Second, stateless IDBD was the best-fitting algorithm across all bandits. Third, stateless IDBD was best able to simulate human learning curves. Fourth, we demonstrate stateless IDBD was also best able to simulate transfer performance across contexts. Fifth, the best fitting hyperparameters of stateless IDBD differed across bandits.

Model Validation of Stateless IDBD To validate stateless IDBD for use in humans, we ensured that the three hyperparameters could be recovered effectively. This was true for the volatile bandit (all r > .88), the drifting bandit (all r > .83) and the stationary bandit (all r > .83). We also found strong model recovery across all three of our bandits. Specifically, stateless IDBD showed good model recovery in the volatile bandit (90%), the drifting bandit (90%), and the stationary bandit (96%). Please see the supplemental materials for additional details and figures for parameter recovery (8) and model recovery (9).

Best Fitting Algorithms Across Bandits To determine which algorithm provided the best fit of the human data, we computed each of the algorithm's AIC values on a participant-by-participant basis for the three bandits. We found that the stateless IDBD bandit provided the best fit of 64% (130/204) of participants in the volatile bandit, 59% (63/107) of participants in the drifting bandit, and 88% (85/97) of participants in the stationary bandit.

Learning Curve Simulation To adjudicate between algorithms, we investigated whether three algorithms could replicate the learning behaviour of humans (Figure 2)[1]. To determine how well the algorithms could replicate human performance, we selected the best fitting parameters for each participant within each bandit. We then used those best fitting parameters to simulate performance by having each algorithm make choices and obtain rewards in each of the three bandits. To determine how well each algorithms learned compared to the humans, we examined four measures: (1) optimal arm choice (i.e., how often the agent selected the highest value arm), (2) switching (how often the agent switched), (3) win-stay behaviour (how often the agent stayed following a win), and (4) lose-switch behaviour (how often the agent switched following a loss).

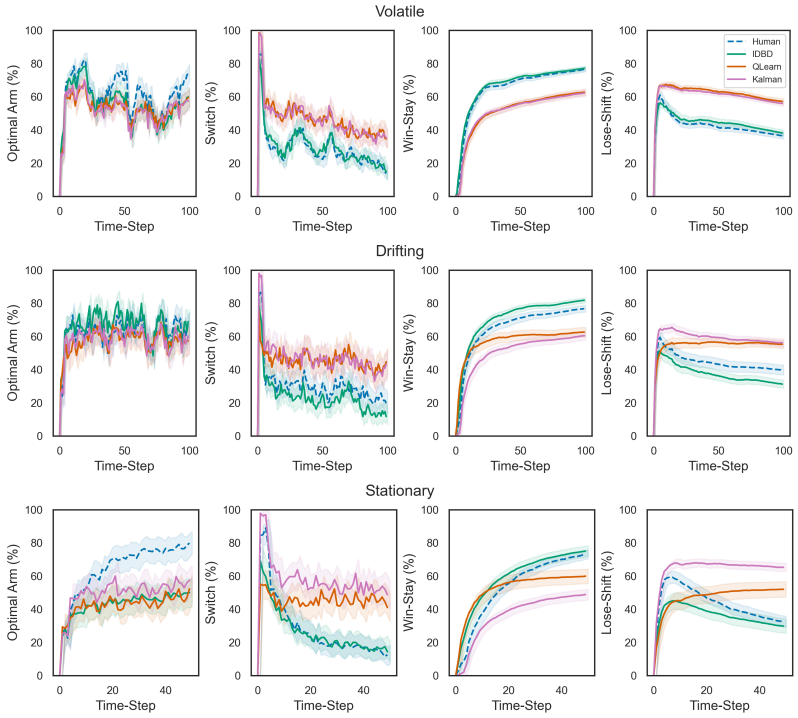

Figure 2: Human (blue dotted line) and algorithm (solid lines) learning curves. Shaded regions indicate 95% confidence intervals.

Overall, stateless IDBD was best able to simulate performance across three of our four measures. Generally, the algorithms attained similar levels of performance in terms of optimal arm selection in both of the non-stationary bandits. However, in the stationary bandit, the Kalman filter algorithm was most similar to the human optimal arm curve. For switching, the stateless IDBD algorithm attained performance most closely aligned with human switching. For win-stay and lose-shift behaviour, the stateless IDBD algorithm simulated curves most like the humans although it tended to stay following wins at a higher rate and shift following losses at a lower rate. In contrast, the two baselines tended to stay following wins at a lower rate and switch following loses at a higher rate.

Transfer Performance In addition, we compared how well the algorithms could transfer performance across bandits. To do this, we took the best fitting hyperparameters of the human learning data from one bandit and used those parameters to simulate participants in the second bandit that a participant completed. That is, we examined how well the algorithms fit to one bandit were able to replicate human learning when made to complete a second bandit. To assess transfer performance, we compared the algorithms to the human performance using the mean square deviation (Ahn et al.,

[1]The confidence intervals on Figure 2 are standard normal 95% confidence intervals. This is true for all confidence intervals in our work.

2008). Briefly, the mean square deviation was calculated by: MSD $= \frac{1}{c}\sum_{1}^{c}(P_{human} - P_{sim})^2$. Here, c is the three bandits, P_{human} is the human participants' performance averaged across all blocks and time-steps for a bandit, and P_{sim} is the average simulated performance from the algorithm from that same bandit. The lower the mean squared deviation, the better the algorithm's ability to simulate transfer performance. We compared the ability of algorithms to transfer performance using one-way analysis of variances (ANOVAs), and followed up with independent samples t-tests (Benjamini-Hochberg corrected; (Benjamini & Hochberg, 1995)).

In terms of the ability of the algorithms to transfer performance (Figure 3) we found that stateless IDBD performed best for three of our four measures. We found an effect of algorithm on optimal arm choice (F(1, 1170)= 9.14, p= 0.0001, η_p^2 = 0.015). Stateless IDBD performed worse at replicating the human optimal arm choices compared to both Q-learning (t(780) = 2.24, p = 0.03, d = 0.16) and the Kalman filter (t(780) = 4.04, p < 0.0001, d = 0.29). In addition, the Kalman filter outperformed Q-learning (t(780) = 3.71, p < 0.03, d = 0.15). In terms of replicating human switching behaviour, we again found an effect of algorithm type (F(1, 1170) = 19.43, p = 5e-9, η_p^2 = 0.032). The follow-up t-tests revealed that stateless IDBD was better at replicating human switching compared to both Q-learning (t(780) = 3.52, p = 0.0007, d = 0.25), and the Kalman filter (t(780) = 6.67, p = 1e-10, d = 0.48). Interestingly, Q-learning replicated human switching behaviour better than the Kalman filter (t(780) = 2.65, p = .008, d = 0.19).

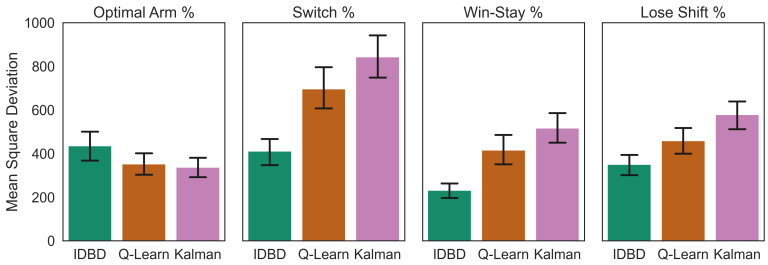

Figure 3: Mean square deviation across our four measures of performance - lower scores indicate the algorithm was better able to simulate human performance. Error bars are 95% confidence intervals.

For replicating win-stay behaviour, we found an effect of algorithm type (F(1, 1170)= 14.07, p = 9e-7, η_p^2 = 0.023). This effect was primarily driven by the fact that stateless IDBD was better able to replicate win-stay behaviour than both Q-learning (t(780) = 4.07,p= 7e-5, d = 0.29) and the Kalman Filter (t(780) = 5.68, p = 6e-8, d = 0.41). There was no difference between Q-learning and the Kalman Filter (t(780) = 1.16, p = 0.24, d = 0.08). For lose-shift behaviour, again the algorithms differed (F(1, 1170) = 12.64, p = 3e-6, η_p^2 = 0.021). Stateless IDBD outperformed both Q-learning (t(780) = 2.51, p = 0.02, d = 0.25) and the Kalman filter (t(780) = 5.26, p = 5e-7, d = 0.38). The Q-learning algorithm outperformed the Kalman filter algorithm in terms of lose-shift behaviour (t(780) = 2.42, p = 0.02, d = 0.17).

Hyperparameter Comparison For our final analysis, we compared the best-fit hyperparameters of the stateless IDBD algorithm across bandits (Figure 4). For the comparison between the volatile bandit to the stationary bandit, we found that all three hyper parameters differed. Specifically, participants in the non-stationary volatile bandit had lower inverse temperature parameters (t(94) = 4.44, p = 2e-5, d = 0.64), higher initial Q values (t(94) = 8.88, p = 4e-14, d = 1.15) and higher initial step-size parameters (t(94) = 7.20, p = 1e-10, d = 1.03). When instead comparing the two non-stationary bandits, we found that humans had a higher inverse temperature parameter (t(103) = 2.24, p = 0.02, d = 0.27) and higher initial step-size values (t(103) = 2.72, p = 0.008, d = 0.36) in the volatile bandit compared to the drifting bandit. There was no difference in terms of initial Q values (t(103) = 0.24, p = 0.80, d = 0.03).

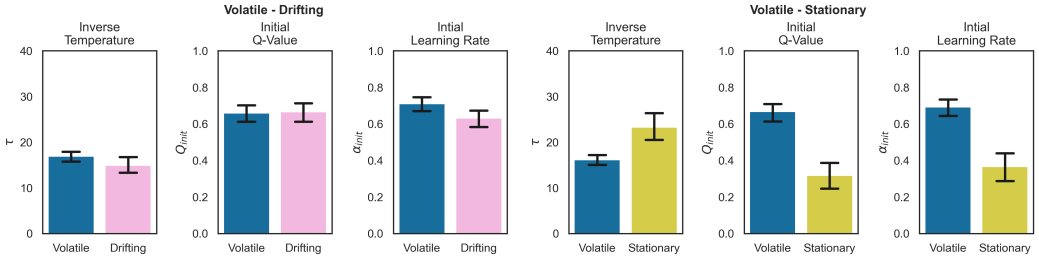

Figure 4: Best fit hyperparameters from the human data. Error bars are 95% confidence intervals.

5 Discussion

Implications for Human Learning The present work has several implications for understanding human learning. First, we found that stateless IDBD is a well-behaved algorithm (per parameter and model recovery) and can successfully model human learning. Thus, we no longer need to use a single static step-size (the Q-learning algorithm), and can extend the benefit of the varying step-size (the Kalman filter) to multiple actions. While it is unclear whether humans would maintain separate step-size parameters for different options, combining brain imaging techniques like EEG (which can detect neural signals tied to step-size changes; (Jepma et al., 2016) with the stateless IDBD algorithm could provide an answer. Second, because we combined stateless IDBD with Q learning, our work should be easily extended to study different aspects of exploration such as directed exploration (Auer, 2002), or the positivity bias (Palminteri, 2022) providing directions for future work. Third, typically initial Q values are not recovered from human data, although there are some exceptions when using Hierarchical Bayesian approaches (Dubois et al., 2021). Being able to recover the initial Q values means we can model how much humans value a context, providing a link to foraging algorithms which model how good a forager thinks a context is (Avgar & Berger-Tal, 2022). Fourth, the hyperparameters of stateless IDBD varied across contexts which may suggest that these parameters are related to cognitive processes involved in meta-learning (Wang, 2021). That is, humans modulate their rate of probabilistic exploration, modify their assessments of context quality, and increase or decrease their initial rate of learning, to learn across contexts.

Implications for Artificial Agents Our results also suggest meta-learning algorithms similar to IDBD could be useful for developing continual reinforcement learning algorithms. Continual reinforcement learning algorithm development is still in its infancy. There is still little consensus on problem formulations (Abel et al., 2023), empirical benchmarks to evaluate progress (Khetarpal et al., 2022), or how hyperparameters should be dealt with (Mesbahi et al., 2024). However, our results suggest two foci for future algorithm development. First, hyperparameter free algorithms (or at least algorithms less sensitive to hyper choices) model human data well. Recent work has shown that hyperparameter tuning in continual reinforcement learning is fundamentally different compared with conventional reinforcement learning and that tuning in continual tasks can obfuscate good directions for algorithmic progress (e.g., (Mesbahi et al., 2024). Second, the vast majority of deep reinforcement learning algorithms (including continual ones) make use of the Adam optimizer (Kingma & Ba, 2015). The belief is that Adam eliminates step-size tuning and provides a vector of step-sizes—one for each weight in the network. However, recent studies have shown counter-examples where Adam degenerates into a single global step-size parameter (Degris et al., 2024) and performs poorly (Elsayed & Mahmood, 2024). Our results provide yet another piece of evidence suggesting that effective continual learning systems (in this case people) modulate a collection of step-sizes based on meta learning and that approaches similar to, or inspired by, IDBD should be developed and examined in continual reinforcement learning settings.

Broader Impact Statement

As our work studies human exploration, it could be used by malicious actors to manipulate people to explore (or not) as a benefit to the actor. We caution readers to not use this work for that purpose.

References

David Abel, André Barreto, Benjamin Van Roy, Doina Precup, Hado Van Hasselt, and Satinder Singh. A definition of continual reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Woo Young Ahn, Jerome R. Busemeyer, Eric Jan Wagenmakers, and Julie C. Stout. Comparison of decision learning models using the generalization criterion method. *Cognitive Science*, 32, 2008.

H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Second International Symposium on Information Theory*, 1973.

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3, 2002.

Tal Avgar and Oded Berger-Tal. Biased learning as a simple adaptive foraging mechanism. *Frontiers in Ecology and Evolution*, 9, 2022.

Timothy E.J. Behrens, Mark W. Woolrich, Mark E. Walton, and Matthew F.S. Rushworth. Learning the value of information in an uncertain world. *Nature Neuroscience*, 10, 2007.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57, 1995.

Yash Chandak. Reinforcement learning for non stationary problems. In *PhD Dissertation, University of Massachusetts Amherst*, 2022.

Jonathan D. Cohen, Samuel M. McClure, and Angela J. Yu. Should i stay or should i go? how the human brain manages the trade-off between exploitation and exploration. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362, 2007.

Nathaniel D. Daw. Trial-by-trial data analysis using computational models. *Decision Making, Affect, and Learning: Attention and Performance XXIII*, 2011.

Nathaniel D. Daw, John P. O'Doherty, Peter Dayan, Ben Seymour, and Raymond J. Dolan. Cortical substrates for exploratory decisions in humans. *Nature*, 441:876–879, 2006.

Thomas Degris, Khurram Javed, Arsalan Sharifnassab, Yuxin Liu, and Richard S. Sutton. Step-size optimization for continual learning. *arXiv:2401.17401 [cs.LG]*, 2024.

Magda Dubois, Johanna Habicht, Jochen Michely, Rani Moran, Ray J. Dolan, and Tobias U. Hauser. Human complex exploration strategies are enriched by noradrenaline-modulated heuristics. *eLife*, 10, 2021.

Mohamed Elsayed and Ashique Rupam Mahmood. Addressing loss of plasticity and catastrophic forgetting in continual learning. In *International Conference on Learning Representations*, 2024.

Thomas D. Ferguson, Alona Fyshe, Adam White, and Olave E. Krigolson. Humans adopt different exploration strategies depending on the environment. *Computational Brain and Behavior*, 6, 2023.

Aurélien Garivier and Eric Moulines. On upper-confidence bound policies for switching bandit problems. In *International conference on algorithmic learning theory*, 2011.

Thomas L. Griffiths, Frederick Callaway, Michael B. Chang, Erin Grant, Paul M. Krueger, and Falk Lieder. Doing more with less: meta-reasoning and meta-learning in humans and machines. *Current Opinion in Behavioral Sciences*, 29, 2019.

Neha Gupta, Ole Christoffer Granmo, and Ashok Agrawala. Thompson sampling for dynamic multi-armed bandits. In *International Conference on Machine Learning and Applications*, 2011.

Cameron D. Hassall, Craig G. McDonald, and Olave E. Krigolson. Ready, set, explore! event-related potentials reveal the time-course of exploratory decisions. *Brain Research*, 1719, 2019.

Andrew Jacobsen, Matthew Schlegel, Cameron Linke, Thomas Degris, Adam White, and Martha White. Meta-descent for online, continual prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3943–3950, 2019.

Arushi Jain, Josiah P. Hanna, and Doina Precup. Adaptive exploration for data-efficient general value function evaluations. In *Advances in Neural Information Processing Systems*, 2024.

Khurram Javed, Arsalan Sharifnassab, and Richard S Sutton. Swifttd: A fast and robust algorithm for temporal difference learning. In *Reinforcement Learning Conference*, 2024.

Marieke Jepma, Peter R. Murphy, Matthew R. Nassar, Mauricio Rangel-Gomez, Martijn Meeter, and Sander Nieuwenhuis. Catecholaminergic regulation of learning rate in a dynamic environment. *PLoS Computational Biology*, 12, 2016.

Rudolf E. Kalman. A new approach to linear filtering and prediction theory. *Transactions of the ASME-Journal of Basic Engineering*, 82:35–45, 1960.

Alex Kearney, Vivek Veeriah, Jaden B Travnik, Richard S Sutton, and Patrick M Pilarski. Tidbd: Adapting temporal-difference step-sizes through stochastic meta-descent. *arXiv preprint arXiv:1804.03334*, 2018.

Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. Towards continual reinforcement learning: a review and perspectives. *Journal of Artificial Intelligence Research*, 75, 2022.

Diederik P. Kingma and Jimmy Ba. Adam: a method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Junseok K Lee, Marion Rouault, and Valentin Wyart. Adaptive tuning of human learning and choice variability to unexpected uncertainty. *Science Advances*, 9, 2023.

Cam Linke, Nadia M Ady, Martha White, Thomas Degris, and Adam White. Adapting behavior via intrinsic reward: A survey and empirical study. *Journal of Artificial Intelligence Research*, 69: 1287–1332, 2020.

Ashique Rupam Mahmood, Richard S. Sutton, Thomas Degris, and Patrick M. Pilarski. Tuning-free step size adaptation. In *International Conference on Acoustics, Speech, and Signal Processing*, 2012.

Matthew Mcleod, Chunlok Lo, Matthew Schlegel, Andrew Jacobsen, Raksha Kumaraswamy, Martha White, and Adam White. Continual auxiliary task learning. In *Advances in Neural Information Processing Systems*, 2021.

Golnaz Mesbahi, Parham Mohammad Panahi, Olya Mastikhina, Martha White, and Adam White. K-percent evaluation for lifelong rl. *arXiv:2404.02113 [cs.LG]*, 2024.

I. Momennejad, E. M. Russek, J. H. Cheong, M. M. Botvinick, N. D. Daw, and S. J. Gershman. The successor representation in human reinforcement learning. *Nature Human Behaviour*, 1, 2017.

Sindhu Padakandla. A survey of reinforcement learning algorithms for dynamically varying environments. *ACM Computing Surveys*, 54, 2021.

Sindhu Padakandla, Prabuchandran K. J, and Shalabh Bhatnagar. Reinforcement learning algorithm for non-stationary environments. *Applied Intelligence*, 50, 2020.

Stefano Palminteri. Choice-confirmation bias and gradual perseveration in human reinforcement learning. *Behavioral Neuroscience*, 137, 2022.

Elise Payzan-Lenestour and Peter Bossaerts. Risk, unexpected uncertainty, and estimation uncertainty: bayesian learning in unstable settings. *PLoS Computational Biology*, 7, 2011.

Payam Piray and Nathaniel D. Daw. Computational processes of simultaneous learning of stochasticity and volatility in humans. *Nature Communications*, 15:9073, 2024.

Alireza Soltani and Alicia Izquierdo. Adaptive learning under expected and unexpected uncertainty. *Nature Reviews Neuroscience*, 20, 2019.

Maarten Speekenbrink and Emmanouil Konstantinidis. Uncertainty and exploration in a restless bandit problem. *Topics in Cognitive Science*, 7, 2015.

Richard S. Sutton. Adapting bias by gradient descent: an incremental version of delta-bar-delta. In *AAAI Conference on Artificial Intelligence*, 1992.

Jane X. Wang. Meta-learning in natural and artificial intelligence, 2021.

Robert C Wilson and Anne G E Collins. Ten simple rules for the computational modeling of behavioral data. *eLife*, 8, 2019.

Supplementary Materials

The following content was not necessarily subject to peer review.

6 Task Instructions

Participants were informed that their goal was to obtain as many points as possible by selecting the arm that gave the most points on each time-step. In the two non-stationary bandits they were told that the best option could change both between time-steps and between blocks. In the stationary bandit, participants were told that the best option would stay consistent across a block but may change between blocks. Participants were not compensated on the basis of performance. When changing between bandit types (e.g., stationary to volatile; drifting to volatile), participants were told they were switching environments and the points values of the task may show different patterns. In addition, they were given the instructions mentioned above for the new bandit type.

See https://github.com/tomferg/RLC_2025 for data and experiment code.

7 Stateless IDBD Pseudocode

Below, we provide pseudocode for a stateless version of Autostep (Mahmood et al., 2012) adapted from previous work (Linke et al., 2020). Autostep was designed for non-stationary environments and has a separate step-size parameter for each input. For stateless IDBD, the step-size parameter can thus change depending on the obtained rewards. As such, when learning is progressing the step-size parameter of an arm should increase, but when learning is not progressing the step-size parameter of an arm should decrease. This occurs through the computation of a memory trace (h) of the prediction errors (δ) across the task. If the predictions errors of an arm are all of the same sign then the step-size parameter of that arm should increase but if the prediction errors are changing signs repeatedly then the step-size parameter should decrease. The update of the step-size parameters depends on a meta-learning rate parameter (β). However, because we had agents only complete a small number of time-steps (100 time-steps for each block of each bandit), we found that the meta-learning rate had little effect on performance when simulating agents within the bandits we used here.

Algorithm 1: Stateless IDBD Algorithm

Only the chosen arm (k) is updated
β is the meta-learning rate parameter
n and h are scalar memory variables initialized to 1 and 0
δ_j is the prediction error at time-step j and α_k the step-size parameter of predictor k
Procedure Stateless IDBD
1. $n_k = \max(|\delta_j h_k|,\ n_k + \frac{1}{100}\alpha_k(|\delta_j h_k| - n_k))$
2. $\alpha_k = min(\alpha_k \exp(\beta \frac{\delta_j h_k}{n_k}), 1)$
3. $h_k = h_k(1 - \alpha_k) + \alpha_k \delta_j$

8 Parameter Recovery

To ensure that the stateless IDBD algorithm was stable and that its parameters had distinct effects on behaviour, we conducted parameter recovery (Figure 5). Parameter recovery involves the generation of a set number of simulated datasets (in our case, 50 datasets per bandit). To generate each dataset, we first randomly selected a set of true (i.e., simulated) parameter values for the stateless IDBD model and then had the model generate a dataset (i.e., make action choices and obtain rewards) using those parameters within a bandit. Following this, we ran our parameter fitting procedure on each of the simulated datasets. We then correlated the simulated parameters (which we input and

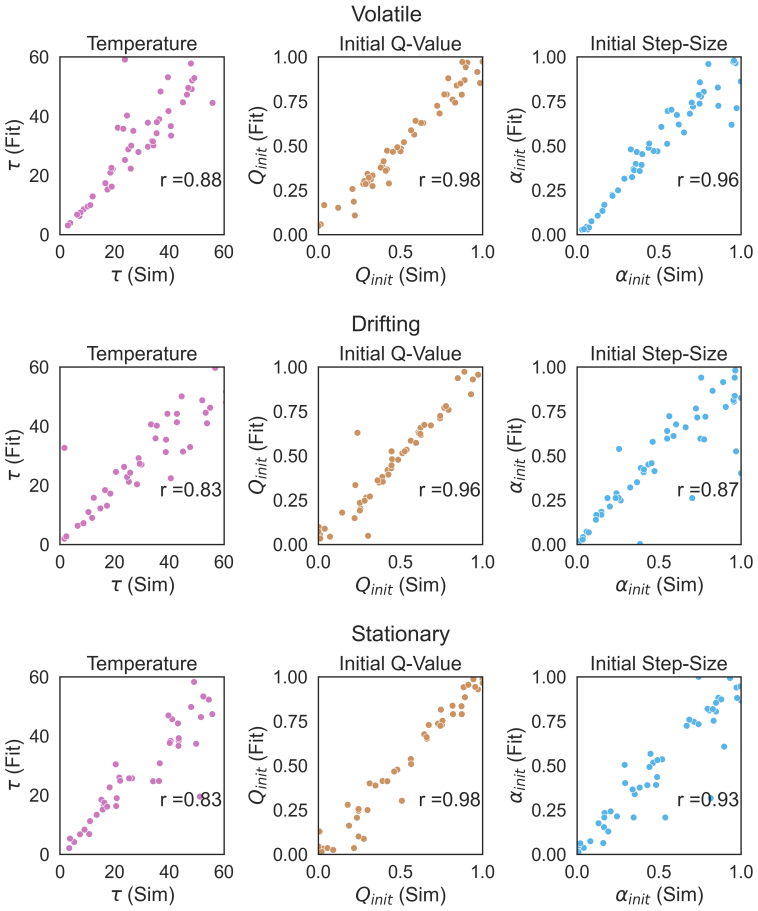

Figure 5: Parameter recovery of the three hyperparameters for the stateless IDBD model across each of the bandits

know) with the fitted parameters recovered from minimizing the negative log likelihood. This was repeated for each bandit individually.

9 Model Recovery

We conducted model recovery (Figure 6) to determine whether our three chosen algorithms made quantitatively distinct behavioural patterns in our bandits. Akin to our parameter recovery, we first generated a set of 50 datasets from each algorithm using different random parameter values (n = 50 for each of the three algorithms; 150 total simulations). We next passed each of these simulated datasets through our fitting procedure (i.e., minimizing the negative log-likelihood) for each of the three algorithms individually. Following this, we transformed our negative log likelihood values into Akaike Information Criterion values (AIC, (Akaike, 1973) to determine which algorithm provided the best fit of each dataset.

We then generated a 3 x 3 confusion matrix for the three algorithms. To do this, we recorded each time an algorithm provided the best fit of a simulated dataset. For the confusion matrix, the 3 rows are the algorithms used to simulate the dataset and the 3 columns are the algorithms used to fit those datasets. To be clear, each row saw the same 50 datasets (generated by a specific algorithm) which were passed through the three algorithms fitting procedures, and we recorded which algorithm provided the best fit within the cells of the 3 x 3 matrix. After this, we transformed those best fit

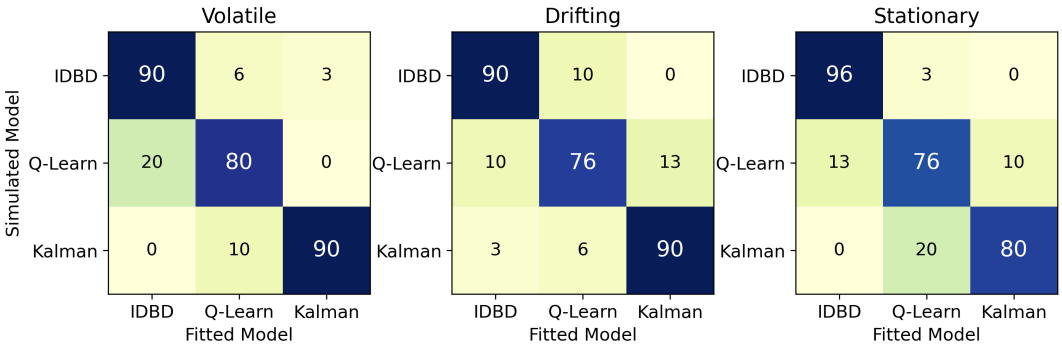

Figure 6: Model recovery confusion matrices across each of the bandits. The numbers indicate the percentage of the simulated data best fit by each of the models.

numbers into percentages by dividing each cell by the number of datasets from that row (50). Within the confusion matrix we expect to observe that the diagonal of the matrix would be where the largest values are present as that is where the algorithm used to simulate the dataset matches up with the model used to fit the dataset.