Human-level reinforcement learning performance of recurrent neural networks is linked to hyperperseveration, not directed exploration

D. Tuzsus[1], A. Brands[1], I. Pappas[2], J. Peters[1]

[1]Department of Psychology, Biological Psychology, University of Cologne, Germany
[2]Department of Neurology, Keck School of Medicine, University of Southern California, USA

Contact: dtuzsus1@uni-koeln.de

Abstract

A key feature of animal and human decision-making is to balance the exploration of unknown options for information gain (directed exploration) versus selecting known options for immediate reward (exploitation), which is often examined using restless bandit tasks. Recurrent neural network models (RNNs) have recently gained traction in both human and systems neuroscience work on reinforcement learning, due to their ability to show meta-learning of task domains. Here we comprehensively compared the performance of a range of RNN architectures as well as human learners on restless four-armed bandit problems. The best-performing architecture (LSTM network with computation noise) exhibited human-level performance. Cognitive modeling first revealed that both human and RNN behavioral data contain signatures of higher-order perseveration, i.e., perseveration beyond the last trial, but this effect was more pronounced in RNNs. In contrast, human learners, but not RNNs, exhibited a positive effect of uncertainty on choice probability (directed exploration). RNN hidden unit dynamics revealed that exploratory choices were associated with a disruption of choice predictive signals during states of low state value, resembling a win-stay-loose-shift strategy, and resonating with previous single unit recording findings in monkey prefrontal cortex. Our results highlight both similarities and differences between exploration behavior as it emerges in meta-learning RNNs, and computational mechanisms identified in cognitive and systems neuroscience work.

Introduction

Reinforcement learning (RL) theory [1] is of central importance in psychology, neuroscience, computational psychiatry and artificial intelligence as it accounts for how artificial and biological agents learn from reward and punishment. According to both the law of effect in psychology [2] and the reward hypothesis in machine learning [1], agents optimize behavior to maximize reward and minimize punishment. In computational psychiatry, RL theory has yielded valuable insights into changes in learning and decision-making associated with different mental disorders [3–5].

To maximize reward, agents have to solve the exploration-exploitation dilemma [1] that can be stated as follows: Should one pursue actions that led to reward in the past (exploitation) or should one explore novel courses of action for information gain (exploration)? In stable environments, where action-outcome contingencies are stable over time, the exploration-exploitation dilemma can be effectively solved by first exploring all available actions to identify the most rewarding one, and subsequently exploiting this action. In contrast, in volatile environments, action-outcome contingencies change over time, such that exploration and exploitation need to be continuously balanced by an agent. A high level of exploitation would make an agent unable to adapt to changes in the environment, whereas an excess of exploration would reduce reward accumulation, as optimal actions would oftentimes not be selected.

A number of computational strategies have been proposed to address the exploration-exploitation tradeoff [1]. In ε-greedy and softmax choice rules, exploration is achieved via choice randomization. While such “random” exploration appears to be one core component of both human and animal exploration [6–10], computational modeling of behavior strongly suggests that humans additionally use “directed” or strategic exploration strategies [8–13]. This is typically modeled via an “exploration bonus” parameter that increases the value of options with greater information value [11–14]. In volatile environments, the uncertainty associated with the outcome of a specific action is often taken as a proxy for information gain [10]. Exploring uncertain courses of action can thus increase information gain, over and above a simpler random exploration strategy.

A further related process pervasive in human and animal behavior is perseveration, the tendency of an agent to repeat previous choices regardless of obtained reward. Perseveration can be categorized into first-order or higher-order perseveration. First-order perseveration refers to the tendency to repeat the choice of the previous trial ($choice_{t-1}$), but in the higher-order case, perseveration can extend to choices n-trials back ($choice_{t-n}$) [15]. First-order perseveration in RL tasks was observed in rats [16], monkeys [15,17] and humans [11,13], and there is evidence for higher-order perseveration in monkeys [15] and humans [18–20]. Recently, Palminteri [19] showed that accounting for higher-order perseveration in RL models of behavior improves the interpretability of other model parameters.

Note that perseveration and exploration are associated with opposite choice patterns. Whereas high perseveration implies “sticky” behavior repeating previous choices, exploration entails a higher proportion of switch trials. Notably, explicitly accounting for perseveration in RL models improves the sensitivity to detect directed exploration effects. High perseveration can attenuate estimates of directed exploration driving the estimates to be lower or even negative if perseveration is not accounted for in the cognitive model [6,11,13,21–23]. Chakroun et al. [11] showed that including a first-order perseveration term in the RL model increases estimates of directed exploration and improves model fit to human data. Thus, taking into account perseveration behavior is crucial when examining exploration.

In humans, exploratory choices are associated with increased activity in the fronto-parietal network [6,11,13,24] and regulated by dopamine and norepinephrine neuromodulatory systems [11,25–28]. Choice predictive signals in prefrontal cortex neural populations are disrupted during exploratory choices, reflecting a potential neural mechanism for random exploration [7].

Such neuroscientific lines of work have increasingly been informed by computational neuroscience approaches [29]. Here, neural network models are applied to clarify the computational principles underlying task performance. In the context of RL problems, recurrent neural network models (RNNs) are particularly powerful tools. They constitute deep artificial neural network models for sequential data [30] such as RL tasks [31]. Agents interact with the environment, and receive environmental feedback (e.g., rewards), which then informs subsequent choices. RNNs can be applied to RL problems due to their recurrent connectivity pattern. Each time step, RNN hidden units receive information regarding the network’s activation state at the previous time step via recurrent connections, thereby endowing the network with memory about what has happened before. Training and analysis of such models offer potential novel insights with implications for neuroscience [31]. For example, the representations that emerge in a network’s hidden unit activation pattern following training (or over the course of training) can be directly analyzed [29,32–34], similar to the analysis of high-dimensional neural data [7,29,35]. This can reveal insights into the computations and representations underlying a network’s performance.

Neural network modeling approaches can also complement computational modeling as typically done in psychology and cognitive neuroscience [36]. Here, computational mechanisms and representations hypothesized to underlie performance of a given task are explicitly and rigidly build into a quantitative model. While this approach is helpful to compare candidate models, the rigid dependency of these models on built-in a priori assumptions preclude the discovery of novel mechanisms and representations that could underlie task performance. In contrast, RNN dynamics and representations might reveal novel potential mechanisms and representations that support similar tasks by virtue of the RNNs independent

data-driven learning capacity [31]. Reward learning [32–34] and decision-making [29,32] are prominent recent examples.

In this line of work, RNNs are trained to solve reinforcement learning and decision-making tasks from the human and animal neuroscience literature, and the mechanisms underlying their performance are examined [29,33,34,37,38]. Note that this is distinct from using RNNs to model human behavioral data, where standard computational models are replaced by RNNs [39]. RNNs trained on such tasks achieved a form of “meta-learning”: when weights were held fixed following training, the models had acquired the ability to solve novel instantiations of tasks from the same task family [33,34,37,40]. Reinforcement learning over a large number of training episodes via slow adjustments of network weights, gave rise to a much faster reinforcement learning algorithm embedded in the network dynamics, and not involving further weight changes [31,34,37].

Finally, RNNs with noisy computations might be more resilient to adverse conditions (e.g. contingency reversals, volatility) than their counterparts with deterministic computations [32]. This resonates with findings from the machine learning literature suggesting improved performance of neural networks with noisy computations under some conditions [41–43]. Likewise, mental representations [44] and neural representations [37,45,46] might benefit from some degree of representational imprecision (e.g., representations infused with task-independent noise).

RNNs trained on cognitive tasks from the neuroscience literature can reveal how artificial neural networks diverge from (or converge with) human and animal behavior. Also, RNNs might show human-like behavior by mere statistical learning without the use of human-like abstract rule learning [47]. While deep RL agents show superhuman ability in games like Go, Shogi, Chess and Atari games [48–50] they fail to perform better than chance level on a standard T-maze task from animal learning [51]. One of the most prominent differences between human learners and neural networks is the number of interactions with the environment required to learn a task [52–55]. This is in part related to exploration inefficiency (“sampling inefficiency”) during training [56]. Even though there is ongoing work on endowing deep RL agents with improved exploration strategies [55–57], there is so far limited evidence with respect to exploration strategies emerging in meta-learning RNNs. Binz & Schulz [58] showed that the large language model (LLM) GPT-3 shows no evidence of directed exploration in the “Horizon Task” from human cognitive neuroscience [9]. However, LLMs are trained passively on vast amounts of text data, whereas humans learn actively by interacting with dynamic environments. Whether meta-learning RNNs trained on dynamic environments employ directed exploration is an open question.

Bandit tasks constitute a classical testing bed for RL agents [1], and are regularly applied to study human and animal exploration [6,7,11,13,24,45,45,59,60]. In non-stationary

(*restless*) bandit tasks, agents select among a number of options ("bandits") with dynamically changing reinforcement rates or magnitudes (Fig. 1 b & c). In contrast, in stationary bandit problems reinforcement rates are fixed. RNNs achieve state-of-the-art performance on stationary bandit tasks [34] and in reversal schedules [61] adapt their learning rates to environmental volatility [34]. Furthermore, RNNs with computation noise can solve restless bandit tasks when trained on stationary bandits [32], in contrast to their counterparts with deterministic computations. Human exploration behavior in restless bandit tasks is typically better accounted for by models with dynamic uncertainty-dependent learning rates such as the Kalman Filter [6,62]. Furthermore, humans regularly apply a directed exploration strategy on restless bandit tasks. This is modeled using an additional "exploration bonus" parameter that typically takes on positive values, reflecting directed exploration of uncertain options [10–14,24].

Initial work on RNN mechanisms supporting bandit task performance [32,34,38], have predominantly focused on stationary bandits [34]. However, stationary bandits preclude a comprehensive analysis of exploration mechanisms, because exploration is restricted to the first few trials. Furthermore, previous work often focused on two-armed bandit problems [32,38,45]. However, these tasks are limited in that only one alternative can be explored at any given point in time. Although previous work has begun to use classical computational modeling to better understand RNN behavior [34,63], a comprehensive comparison of human and RNN behavior and computational mechanisms when solving the exact same RL problems is still lacking. In addition, similar to so-called researcher's degrees of freedom in experimental work [64], the study of RNNs is associated with a large number of design choices, e.g. with respect to the specifics of the architecture, details of the training schemes as well as hyperparameter settings. Yet, a comprehensive comparison of different network architectures and design choices in the context of RL tasks from the human cognitive neuroscience literature is still lacking.

Here, we addressed these issues in the following ways. First, we comprehensively compared a large set of RNN architectures in terms of their ability to exhibit human-level performance on restless four-armed bandit problems. Note that RNNs were trained as stand-alone RL agents, and were not used to replace standard computational models to account for human behavior. Second, to compare computational strategies between RNNs and human subjects, we used comprehensive cognitive modeling of human and RNN behavior during performance of the exact same RL problems. Finally, we expanded upon previous approaches to the analysis of RNN hidden unit activity patterns [29,32,34] by leveraging analysis approaches from systems neuroscience studies of exploration [7].

Methods

RNN unit types

The present study systematically compared network architectures consisting of four different types of RNN units, standard recurrent units ("vanilla" units) [65], standard LSTM units [66], as well as both unit types endowed with computation noise.

Vanilla RNN

This is a simple Elman recurrent neural network [65] ("Vanilla" RNN). Let X_t denote the input to the network at time *t*, H_t the recurrent state of the network and Y_t the output of the network. Then the network is governed by the following equations:

$$\widehat{H}_{\mathrm{t}} = W_{H_{t-1}} \cdot H_{t-1} + W_X \cdot X_t + B \tag{1}$$

$$H_t = \sigma_{tanh}(\widehat{H}_t) \tag{2}$$

$$Y_t = \sigma_{sm}(W_H \cdot H_t) \tag{3}$$

The input X_t is a vector of length 5 with the first element corresponding to the reward observed on the previous trial (r_{t-1}) and the remaining elements containing a one-hot encoding of the previously chosen action (a_{t-1}). The latter is a vector of length 4 (No. of actions) with all elements being "0" except the element which corresponds to the previous action being set to "1". The parameters of the network are weight matrices $W_{H_{t-1}}$, W_X and W_H and the bias vector B, which are optimized during the training process. Non-linear activation functions σ_{tanh} and σ_{sm} denote the hyperbolic tangent and the softmax function, respectively.

The forward pass of the model starts by passing information from the input layer to the hidden layer by calculating the updated state $\widehat{H_t}$ as a linear combination of the input X_t and the previous recurrent activity H_{t-1} weighed by corresponding weight matrices W_X and $W_{H_{t-1}}$and an additive bias term B (Eq. 1). Within the hidden layer, $\widehat{H}_t$ is non-linearly transformed by the hyperbolic tangent function which results in the current recurrent activity H_t (Eq. 2). Recurrent activity H_t in the hidden layer is then transformed to action probabilities Y_t by applying the softmax function to H_t weighed by matrix W_H (Eq. 3).

Noisy Vanilla RNN

Computation noise might aid in adverse conditions during decision-making [32]. Following Findling et al. [32], we therefore modified the standard RNN unit by adding update-dependent computation noise to the recurrent activity H_t.

$$\widehat{H}_{t,k}^{noisy} \sim \mathcal{N}\left(\widehat{H}_{t,k}, \zeta \cdot \left|H_{t-1,k}^{noisy} - H_{t,k}\right|\right) \tag{4}$$

To transform the exact recurrent activity $H_{t,k}$ in unit k in trial t to noisy recurrent activity $H_{t,k}^{noisy}$ we added noise according to a Gaussian distribution with mean equal to the exact updated state $\widehat{H}_{t,k}$ (see Eq. 1) and a standard deviation of the absolute magnitude of the difference between the previous noisy recurrent activity $H_{t-1,k}^{noisy}$ and the current exact recurrent activity $H_{t,k}$ (see Eq. 2). Because of this difference term the spread of the noise added to a unit scales with the amount of reconfiguration in recurrent activity between subsequent trials similar to a prediction error. The standard deviation is further scaled by the hyperparameter $\zeta > 0$ denoted the *Weber fraction* [32]. Finally, after having sampled the noisy updated state $\widehat{H}_{t,k}^{noisy}$ the hyperbolic tangent activation function is applied (see Eq. 2) to calculate noisy recurrent activity $H_{t,k}^{noisy}$. Importantly, similar to Findling et al. [32] we treated computation noise as an endogenous constraint to the network where the source of the noise is not modifiable, thereby ignoring the gradients resulting from it in the optimization procedure during gradient descent.

LSTM

One issue with standard vanilla RNN units is that during gradient descent, they can suffer from exploding and vanishing gradient problems, resulting in the network not learning the task [67]. Long short-term memory networks (LSTMs) can mitigate this problem by using gated units that control the information flow, which allows the network to learn more long term dependencies within the data [66].

LSTM behavior is governed by the following standard equations:

$$I_{\mathrm{t}} = \sigma_{sm}(W_{XI} \cdot \mathrm{X}_t + W_{HI} \cdot \mathrm{H}_{t-1} + W_{CI} \cdot C_{t-1} + B_I) \tag{5}$$

$$F_{\mathrm{t}} = \sigma_{sm}(\mathrm{W}_{XF} \cdot \mathrm{X}_{\mathrm{t}} + \mathrm{W}_{\mathrm{HF}} \cdot H_{\mathrm{t}-1} + \mathrm{W}_{\mathrm{CF}} \cdot C_{\mathrm{t}-1} + B_F) \tag{6}$$

$$C_t = F_t \circ C_{\mathrm{t}-1} + I_t \circ \sigma_{tanh}(W_{XC} \cdot \mathrm{X}_{\mathrm{t}} + W_{HC} \cdot H_{t-1} + B_C) \tag{7}$$

$$O_t = \sigma_{sm}(W_{XO} \cdot \mathrm{X}_{\mathrm{t}} + W_{HO} \cdot H_{t-1} + W_{CO} \cdot \mathrm{C} + B_O) \tag{8}$$

$$H_t = O_t \circ \sigma_{tanh}(C_t) \tag{9}$$

Here, X is the same input as in Vanilla RNNs, I is the input gate, F is the forget gate (also called the maintenance gate), C is the cell state, O is the output gate, H is the hidden state, t indexes trials and σ_{sm} and σ_{tanh} denote the softmax or the hyperbolic tangent activation

function, respectively. The trainable parameters of the network are weight matrices W and the bias vectors B, where subscripts indicate the connected gates/states.

Noisy LSTM

Following Findling & Wyart [32], we introduce Weber noise at the level of the hidden state H of an LSTM unit k at trial t:

$$H_{t,k}^{noisy} \sim \mathcal{N}\left(H_{t,k}, \zeta \cdot \left|H_{t-1,k}^{noisy} - H_{t,k}\right|\right) \tag{10}$$

That is, noisy LSTM units are the direct analogue to the noise extension outlined above for vanilla units.

Training and test environments

Networks were trained and tested on four-armed restless bandit tasks [6]. On each trial, agents choose between one of four actions (bandits). Associated rewards slowly drift according to independent gaussian random walks. A single episode during training and testing consisted of 300 trials.

During training, the reward associated with the ith bandit on trial t was the reward on trial t-1, plus noise:

$$\hat{R}_{i,t} = R_{i,t-1} + \epsilon_{i,t} \tag{11}$$

Noise ϵ was drawn from a gaussian distribution with mean 0 and standard deviation 0.1:

$$\epsilon_{\mathrm{i,t}} \sim \mathcal{N}(0,\ 0.1) \tag{12}$$

To ensure a reward range: [0,1], reflecting boundaries were applied such that

$$\mathrm{R_{i,t}} = \begin{cases} R_{i,t-1} + \epsilon_{i,t} & if\ 0 < \hat{R}_{i,t} < 1 \\ R_{i,t-1} - \epsilon_{i,t} & if\ \hat{R}_{i,t} < 0\ or\ \hat{R}_{i,t} > 1 \end{cases} \tag{13}$$

Note that $\mathrm{R_{i,t}}$ is a continuous reward as implemented in past research with similar tasks [6,11,13].

Following training, networks weights were fixed, and performance was examined on three random walk instances previously applied in human work [6,11,13]. Here, the reward associated with the i_{th} bandit on trial t was drawn from a gaussian distribution with standard deviation 4 and mean $\mu_{i,t}$ and rounded to the nearest integer.

$$R_{i,t} \sim \mathcal{N}\left(\mu_{i,t}, 4\right) \tag{14}$$

On each trial, the means diffused according to a decaying gaussian random walk:

$$\mu_{i,t+1} = \lambda\mu_{i,t} + (1-\lambda)\theta + \epsilon_{\mathrm{i,t}} \tag{15}$$

with decay $\lambda = 0.9836$ and decay center $\theta = 50$. The diffusion noise $\epsilon_{\mathrm{i,t}}$ was sampled from a gaussian distribution with mean 0 and SD 2.8:

$$\epsilon_{\mathrm{i,t}} \sim \mathcal{N}(0,\ 2.8) \tag{16}$$

During test, each network was exposed to the same three instantiations of this process used in human work [6,11,13]. Rewards in these three random walks were in the range: [0,100]. As RNNs were trained with rewards in the range: [0,1] we scaled rewards accordingly before testing the RNNs.

Training procedure

The networks were trained to optimize their weights and biases by completing 50.000 task episodes. For each episode, a new instantiation of the environment was created according to the equations above. We compared two training schemes, the standard REINFORCE algorithm [68] and advantage actor-critic (A2C, [69]). The objective of the network was to maximize the expected sum of discounted rewards according to following equation:

$$\mathrm{L}(\pi) = \mathrm{E}^{\pi}\left[\sum_{\mathrm{t}\geqq 1}\sum_{\mathrm{k}\geqq 0}\gamma^{k}\cdot r_{t+k}\right] \tag{17}$$

Here, t is the trial number, γ the discount factor, t_{t+k} the observed reward at trial $t+k$ (k is a positive integer) and π is the policy followed by the network.

Following each episode, one of the following algorithms was used to update the network parameters to improve the policy. REINFORCE [68] relies on a direct differentiation of the objective function:

$$\nabla \mathrm{L}_{\pi} = E^{\pi}\left[\sum_{t\geqq 1} \nabla l\, og\pi(a_t) \cdot \left(\sum_{k\geqq 0} \gamma^k \cdot r_{t+k}\right)\right] \tag{18}$$

Here the gradient of the policy loss ($\nabla \mathrm{L}_{\pi}$) is calculated by summing the derivatives of the log probabilities of chosen actions ($log\pi(a_t)$) weighted by the discounted sum of expected rewards from the current trial until the end of the episode. Note that this ensures that action probabilities will increase or decrease according to the expected rewards following these actions. If the expected returns $\left(\sum_{k\geqq 0} \gamma^k \cdot r_{t+k}\right)$ are positive, the gradient will be positive and therefore gradient descent will increase the log probabilities of chosen actions. Conversely, if expected returns are negative, the gradient will be negative and therefore gradient descent will decrease the log probabilities of chosen actions. Thus, action probabilities for actions that led to rewards will be increased, and action probabilities for actions that led to punishments will be decreased.

Advantage Actor-Critic (A2C, [69]) uses a weighted sum of the policy gradient (∇L_{π}), the gradient with respect to the state-value function loss (∇L_{v}), and an optional entropy regularization term (∇L_{ent}), defined as follows:

$$\begin{aligned} \nabla L &= \nabla L_{\pi} + \nabla L_{v} + \nabla L_{ent} \\ &= \frac{\partial \log\pi(a_t|\theta)}{\partial\theta}\delta_t(\theta) + \beta_v \delta_t(\theta)\frac{\partial \mathrm{V}}{\partial \theta_v} + \beta_{\mathrm{e}}\left[\frac{\partial \mathrm{H}\big(\pi(a_t|\theta)\big)}{\partial\theta}\right] \end{aligned} \tag{19}$$

$$\delta_{\mathrm{t}}(\theta_v) = [\mathrm{G_t} - \mathrm{V}(\theta_{\mathrm{v}})] \tag{20}$$

$$G_t = \sum_{i=0}^{k}\left[\gamma^i r_{t+1} + \gamma^k V(\theta_v)\right] \tag{21}$$

Here, on a given trial t, the chosen action is denoted by a_t, the discounted return is G_t with k being the number of steps until the end of the episode, the actor component with the action policy π is parameterized by RNN parameters θ, the critic component with the value function V estimating the expected return is parameterized by θ_v, the entropy of policy π is denoted by $H(\pi)$, the advantage function that estimates the temporal-difference error is denoted by $\delta_t(\theta_v)$. A2C in this formulation contains two hyperparameters β_v and β_{e} scaling the relative influence of the state-value function loss and the entropy regularization term, respectively.

Note that in equations 19 – 21, RNN parameters corresponding to the policy θ and to the value function θ_v are separated, but in practice, as in Wang et al. [34], they share all layers except the output layer where the policy corresponds to a softmax output and the value function to a single linear output. Weights and biases for REINFORCE and A2C were updated using the RMSProp algorithm as implemented in TensorFlow 1.15.0. We initialized all weights with the standard Xavier initializer [70] as implemented in TensorFlow. Bias parameters were initialized with 0 vectors.

A common problem of policy gradient methods such as REINFORCE is high variance in the gradients used during the stochastic gradient descent (SDG) optimization procedure [1]. This is the case because the magnitude of the gradients depends on the empirical returns (sum of collected rewards in a given episode). We therefore mean-centered rewards to reduce the variance in the gradients, which improved the training process and performance.

A subset of hyperparameters were systematically varied, as outlined in Table 1. The entropy cost (β_e) was either set to 0.05 (fixed entropy), linearly annealed over the course of training from 1 to 0, or omitted (*none*). In networks with computation noise, the Weber fraction ζ was set to 0.5 [32]. Additional hyperparameters (learning rate, discount factor, no. of training episodes etc.), were selected based on previous work [34] and held constant across all architectures (see Table 2).

Table 1. Overview of factors that are systematically explored in RNN training. Total number of RNN models: 2 (Unit type) x 2 (Computation noise) x 3 (Entropy cost) x 2 (Learning algorithm) = 24 RNN models.

Factor	
Unit type	Vanilla / LSTM
Computation noise	None / Update-dependent
Entropy cost β_e	None / Fixed / Annealed
Learning algorithm	REINFORCE / A2C

Table 2. Hyperparameter values used during RNN training.

Hyperparameters	
Number of hidden units	48
Learning rate	0.0001
State-value estimate cost β_v	0.5
Weber fraction ζ	0.5
Discount factor	0.5
Training Episodes	50.000

Trials per Episode	300

Human data

For comparison with RNN behavior, we re-analyzed human data from a previous study (placebo condition of [11], n=31 male participants). Participants performed 300 trials of the four-armed restless bandit task as described in the environment section.

Cognitive modeling

Our model space for RNN and human behavior consisted of a total of 21 models (see Table 3). Each model consisted of two components, a *learning rule* (Delta rule or Bayesian learner) describing value updating, and a *choice rule* mapping learned values onto choice probabilities.

Delta rule: Here, agents update the expected value ($\mathrm{v}_{\mathrm{c_t}}$) of the bandit chosen on trial t ($\mathrm{c_t}$) based on the prediction error (δ) experienced on trial t:

$$v_{c_t,t+1} = v_{c_t,t} + \alpha\delta_t \tag{22}$$

$$\delta_t = R_t - v_{c_t,t} \tag{23}$$

The learning rate $0 \leq \alpha \leq 1$ controls the fraction of the prediction error used for updating, and R_t corresponds to the reward obtained on trial *t*. Unchosen bandit values are not updated between trials and thus remain unchanged until a bandit is chosen again. Bandit values were initialized at $v_1 = 50$.

Bayesian learner: Here we used a standard Kalman filter model [6,62], where the basic assumption is that agents utilize an explicit representation of the process underlying the task's reward structure. The payoff in trial t for bandit i follows a decaying Gaussian random walk with mean $\mu_{\mathrm{i,t}}$ and observation variance $\theta_o^2 = 4^2$. Payoff expectations ($\hat{\mu}_{i,t}^{pre}$) and uncertainties (variances $\hat{\sigma}_{i,t}^{2\,pre}$) for all bandits are updated between trials according to

$$\hat{\mu}_{i,t+1}^{pre} = \hat{\lambda}\hat{\mu}_{i,t}^{post} + \left(1 - \hat{\lambda}\right)\hat{\vartheta} \tag{24}$$

and

$$\hat{\sigma}_{i,t+1}^{2\,pre} = \hat{\lambda}^2 \hat{\sigma}_{i,t}^{2\,post} + \hat{\sigma}_d^2 \tag{25}$$

with decay $\lambda = 0.9836$, decay center $\vartheta = 50$ and diffusion variance $\hat{\sigma}_d^2$ = 4.
The chosen bandit's mean is additionally updated according to

$$\hat{\mu}_{c_t,t}^{post} = \hat{\mu}_{c_t,t}^{pre} + \kappa_t \delta_t \tag{26}$$

with

$$\delta_t = r_t - \hat{\mu}_{c_t,t}^{pre}. \tag{27}$$

Here, κ denotes the Kalman gain that is computed for each trial t as:

$$\kappa_t = \hat{\sigma}_{i,t}^{2\,pre} / \left(\hat{\sigma}_{i,t}^{2\,pre} + \hat{\sigma}_o^2\right) \tag{28}$$

κ_t determines the fraction of the prediction error that is used for updating. In contrast to the learning rate in the delta rule model, $\kappa_t$varies from trial to trial, such that the degree of updating scales with a bandit's uncertainty $\hat{\sigma}_{i,t}^{2\,pre}$. The observation variance $\hat{\sigma}_o^2$ indicates how much rewards vary around the mean, reflecting how reliable each observation is for estimating the true mean. Initial values $\mu_1^{pre}$and $\sigma_1^{2\,pre}$ were fixed to 50 and 4 for all bandits, respectively. Estimates of the random walk parameters $\hat{\lambda}$, $\hat{\vartheta}$, $\hat{\sigma}_o^2$ and $\hat{\sigma}_d^2$ were fixed to their true values (see Table 3).

Choice rules: Delta rule models

Choice rule 1 used a standard softmax function (SM):

$$\text{Choice rule 1 (SM)}: P_{i,t} = \frac{exp\left(\beta v_{i,t}\right)}{\sum_j exp\left(\beta v_{j,t}\right)} \tag{29}$$

Here, $P_{i,t}$ denotes the probability of choosing bandit i on trial t and β denotes the inverse temperature parameter controlling the degree of choice stochasticity.

Choice rule 2 extended choice rule 1 with a heuristic directed exploration term:

$$\text{Choice rule 2 (SM + T)}:\ P_{i,t} = \frac{exp\left(\beta\left[v_{i,t} + \varphi(t - T_i)\right]\right)}{\sum_j e\,xp\left(\beta\left[v_{j,t} + \ \varphi\left(t - T_j\right)\right]\right)} \tag{30}$$

This simple "trial heuristic" [12] models a bandit's uncertainty as linearly increasing with the number of trials since it was last selected $(t - T_i)$, where T_i denotes the last trial before the current trial t in which bandit i was chosen. The free parameter φ models the impact of directed exploration on choice probabilities.

Choice rule 3 then replaced the trial-heuristic with a directed exploration term based on a "bandit identity" heuristic:

$$\text{Choice rule 3 } (SM + B): P_{i,t} = \frac{exp\left(\beta\left[v_{i,t} + \varphi x_i\right]\right)}{\sum_j \mathrm{e\,xp}\left(\beta\left[\mathrm{v_{j,t}} + \varphi \mathrm{x_j}\right]\right)} \tag{31}$$

Here, x_i denotes how many unique bandits were sampled since bandit i was last sampled. E.g., $x_i = 0$ if bandit *i* was chosen on the last trial, and $x_i = 1$ if one other unique bandit was selected since *i* was last sampled. x_i therefore ranges between 0 and 3.

Choice rule 4 then corresponds to choice rule 1 with an additional first-order perseveration term:

$$\text{Choice rule 4 (SM + P)}: \mathrm{P_{i,t}} = \frac{\exp\left(\beta\left[v_{i,t} + I_{c_{t-1=i}}\rho\right]\right)}{\sum_j \exp\left(\beta\left[\mathrm{v_{j,t}} + \mathrm{I_{c_{t-1=j}}}\rho\right]\right)} \tag{32}$$

The free parameter ρ models a perseveration bonus for the bandit selected on the preceding trial. I is an indicator function that equals 1 for the bandit chosen on trial $t - 1$ and 0 for the remaining bandits.

Choice rule 5 then corresponds to choice rule 1 with an additional higher-order perseveration term:

$$Choice\ rule\ 5\ (SM + DP): P_{i,t} = \frac{exp\left(\beta\left[\hat{\mu}_{i,t}^{pre} + \rho H_{i,t}\right]\right)}{\sum_j \mathrm{e\,xp}\left(\beta\left[\hat{\mu}_{j,t}^{pre} + \rho H_{j,t}\right]\right)} \tag{33}$$

With

$$H_t = H_{t-1} + \ \alpha_h(I_{t-1} - H_{t-1})$$

Here, the indicator function I is used to calculate the habit strength vector H_t [71]. H_t is a recency-weighted average of past choices, were the step size parameter α_h [0,1] controls how much recent actions influence current habit strength. In case of α_h = 1, SM+DP corresponds to the first-order perseveration model SM+EP, as only the most recent (previous) action is updated in the habit strength vector. Values of 0 < α_h < 1 result in recency weighted average values for each action in H, giving rise to higher-order perseveration behavior. The resulting habit strength values of each action are then weighted by the perseveration parameter ρ, and enter into the computation of choice probabilities.

Choice rules 6 and 7 likewise extend choice rules 3 and 4 with first-order perseveration terms:

$$Choice\ rule\ 6\ (SM+TP):\ P_{i,t} = \frac{exp\left(\beta\left[v_{i,t} + \varphi(t-T_i) + I_{c_{t-1=i}}\rho\right]\right)}{\sum_j e\,xp\left(\beta\left[v_{j,t} + \varphi\left(t-T_j\right) + I_{c_{t-1=j}}\rho\right]\right)} \tag{34}$$

$$Choice\ rule\ 7\ (SM+BP): P_{i,t} = \frac{exp\left(\beta\left[v_{i,t} + \varphi x_i + I_{c_{t-1=i}}\rho\right]\right)}{\sum_j e\,xp\left(\beta\left[v_{j,t} + \varphi x_j + I_{c_{t-1=i}}\rho\right]\right)} \tag{35}$$

Choice rules 8 and 9 extend choice rules 3 and 4 with higher-order perseveration terms:

$$Choice\ rule\ 8\ (SM+TDP):\ P_{i,t} = \frac{exp\left(\beta\left[v_{i,t} + \varphi(t-T_i) + \rho H_{i,t}\right]\right)}{\sum_j e\,xp\left(\beta\left[v_{j,t} + \varphi\left(t-T_j\right) + \rho H_{j,t}\right]\right)} \tag{36}$$

$$Choice\ rule\ 9\ (SM+BDP): P_{i,t} = \frac{exp\left(\beta\left[v_{i,t} + \varphi x_i + \rho H_{i,t}\right]\right)}{\sum_j e\,xp\left(\beta\left[v_{j,t} + \varphi x_j + \rho H_{j,t}\right]\right)} \tag{37}$$

Choice rules: Bayesian learner models

with $\hat{\mu}_{i,t}^{pre}$ instead of $v_{i,t}$ in Equations 29 - 37 yields choice rules 1-9 for the Kalman filter models (equations omitted for brevity). Given that the Bayesian Learner models include an explicit representation of uncertainty, we included two additional models:

$$Choice\ rule\ 10\ (SM+E): P_{i,t} = \frac{exp\left(\beta\left[\hat{\mu}_{i,t}^{pre} + \varphi\hat{\sigma}_{i,t}^{pre}\right]\right)}{\sum_j \mathrm{e}\,\mathrm{xp}\left(\beta\left[\hat{\mu}_{j,t}^{pre} + \varphi\,\hat{\sigma}_{j,t}^{pre}\right]\right)} \tag{38}$$

Here, φ denotes the exploration bonus parameter reflecting the degree to which choice probabilities are influenced by the uncertainty associated with each bandit, based on the model-based uncertainty $\hat{\sigma}_{i,t}^{pre}$. Again, including first order perseveration yields choice rule 10:

$$Choice\ rule\ 11\ (SM + EP): P_{i,t} = \frac{exp\left(\beta\left[\hat{\mu}_{i,t}^{pre} + \varphi\hat{\sigma}_{i,t}^{pre} + I_{c_{t-1=i}}\rho\right]\right)}{\sum_j \mathrm{e}\,\mathrm{xp}\left(\beta\left[\hat{\mu}_{j,t}^{pre} + \varphi\,\hat{\sigma}_{j,t}^{pre} + I_{c_{t-1=j}}\rho\right]\right)} \quad (39)$$

Including higher-order perseveration yields choice rule 12:

$$Choice\ rule\ 12\ (SM + EDP): P_{i,t} = \frac{exp\left(\beta\left[\hat{\mu}_{i,t}^{pre} + \phi\hat{\sigma}_{i,t}^{pre} + \rho H_{i,t}\right]\right)}{\sum_j \mathrm{e}\,\mathrm{xp}\left(\beta\left[\hat{\mu}_{j,t}^{pre} + \phi\,\hat{\sigma}_{j,t}^{pre} + \rho H_{i,t}\right]\right)} \quad (40)$$

Table 3. Free and fixed parameters of all computational models.

Note: Choice rules for the Delta rule: Choice rule 1: softmax; Choice rule 2: softmax with directed exploration (trial heuristic); Choice rule 3: softmax with directed exploration (bandit heuristic); Choice rule 4: softmax with first-order perseveration; Choice rule 5: softmax with second-order perseveration; Choice rule 6 and 7 are choice rules 2 and 3 with first-order perseveration; Choice rule 8 and 9 are choice rules 2 and 3 with higher-order perseveration. Choice rules for the Bayes learner rule: Choice rules 1-9: Are analogous to respective Delta rule choice rules. Choice rule 10: softmax with directed exploration (Kalman); Choice rule 11 and 12: Choice rule 10 with first-order and higher-order perseveration, respectively. α: learning rate; α_h: habit step size; β: inverse temperature; φ: exploration bonus; ρ: perseveration bonus; v_1: initial expected reward values for all bandits; λ: decay parameter; $\hat{\vartheta}$: decay center; σ_0^2: observation variance; σ_d^2: diffusion variance; μ_1^{pre}: initial mean of prior expected rewards for all bandits; σ_1^{pre}:initial standard deviation of prior expected rewards for all bandits.

	Delta rule		**Bayesian learner**	
Choice rule 1 (SM)	α, β	Fixed: v_1	β	Fixed: $\hat{\lambda}$, $\hat{\vartheta}$, $\hat{\sigma}_o^2$, $\hat{\sigma}_d^2$, $\hat{\mu}_1^{pre}$, $\hat{\sigma}_1^{pre}$
Choice rule 2 (SM + T)	α, β, φ	Fixed: v_1	β, φ	Fixed: $\hat{\lambda}$, $\hat{\vartheta}$, $\hat{\sigma}_o^2$, $\hat{\sigma}_d^2$, $\hat{\mu}_1^{pre}$, $\hat{\sigma}_1^{pre}$
Choice rule 3 (SM + B)	α, β, φ	Fixed: v_1	β, φ	Fixed: $\hat{\lambda}$, $\hat{\vartheta}$, $\hat{\sigma}_o^2$, $\hat{\sigma}_d^2$, $\hat{\mu}_1^{pre}$, $\hat{\sigma}_1^{pre}$

Choice rule 4 (SM + P)	α, β, ρ	Fixed: v_1	β, ρ	Fixed: $\hat{\lambda}$, $\hat{\vartheta}$, $\hat{\sigma}_o^2$, $\hat{\sigma}_d^2$, $\hat{\mu}_1^{pre}$, $\hat{\sigma}_1^{pre}$
Choice rule 5 (SM + DP)	$\alpha, \alpha_h, \beta, \rho$	Fixed: v_1	α_h, β, ρ	Fixed: $\hat{\lambda}$, $\hat{\vartheta}$, $\hat{\sigma}_o^2$, $\hat{\sigma}_d^2$, $\hat{\mu}_1^{pre}$, $\hat{\sigma}_1^{pre}$
Choice rule 6 (SM + TP)	$\alpha, \beta, \rho, \varphi$	Fixed: v_1	β, ρ, φ	Fixed: $\hat{\lambda}$, $\hat{\vartheta}$, $\hat{\sigma}_o^2$, $\hat{\sigma}_d^2$, $\hat{\mu}_1^{pre}$, $\hat{\sigma}_1^{pre}$
Choice rule 7 (SM + BP)	$\alpha, \beta, \rho, \varphi$	Fixed: v_1	β, ρ, φ	Fixed: $\hat{\lambda}$, $\hat{\vartheta}$, $\hat{\sigma}_o^2$, $\hat{\sigma}_d^2$, $\hat{\mu}_1^{pre}$, $\hat{\sigma}_1^{pre}$
Choice rule 8 (SM + TDP)	$\alpha, \alpha_h, \beta, \rho, \varphi$	Fixed: v_1	$\alpha_h, \beta, \rho, \varphi$	Fixed: $\hat{\lambda}$, $\hat{\vartheta}$, $\hat{\sigma}_o^2$, $\hat{\sigma}_d^2$, $\hat{\mu}_1^{pre}$, $\hat{\sigma}_1^{pre}$
Choice rule 9 (SM + BDP)	$\alpha, \alpha_h, \beta, \rho, \varphi$	Fixed: v_1	$\alpha_h, \beta, \rho, \varphi$	Fixed: $\hat{\lambda}$, $\hat{\vartheta}$, $\hat{\sigma}_o^2$, $\hat{\sigma}_d^2$, $\hat{\mu}_1^{pre}$, $\hat{\sigma}_1^{pre}$
Choice rule 10 (SM + E)			β, φ	Fixed: $\hat{\lambda}$, $\hat{\vartheta}$, $\hat{\sigma}_o^2$, $\hat{\sigma}_d^2$, $\hat{\mu}_1^{pre}$, $\hat{\sigma}_1^{pre}$
Choice rule 11 (SM + EP)			β, ρ, φ	Fixed: $\hat{\lambda}$, $\hat{\vartheta}$, $\hat{\sigma}_o^2$, $\hat{\sigma}_d^2$, $\hat{\mu}_1^{pre}$, $\hat{\sigma}_1^{pre}$
Choice rule 12 (SM + EDP)			$\alpha_h, \beta, \rho, \varphi$	Fixed: $\hat{\lambda}$, $\hat{\vartheta}$, $\hat{\sigma}_o^2$, $\hat{\sigma}_d^2$, $\hat{\mu}_1^{pre}$, $\hat{\sigma}_1^{pre}$

Model estimation and comparison

Models were fit using Stan and the rSTAN package [72] in R (Version 4.1.1, R Core Team, 2022). To fit single subject models to human and RNN data, we ran 2 chains with 1000 warm-up samples. Chain convergence was assessed via the Gelman-Rubin convergence diagnostic $\hat{R}$ [74] and sampling continued until $1 \leq \hat{R} \leq 1.02$ for all parameters. 1000 additional samples were then retained for further analysis.

Model comparison was performed using the loo-package in R [75] and the Widely-Applicable Information Criterion (WAIC), where lower values reflect a superior model fit [76]. WAICs were computed for each model and human subject/ RNN instance. RNN model comparison focused on the model architecture with the lowest cumulative regret (see Eq. 41). For visualization purposes, we calculated delta WAIC scores for each model by first summing WAIC values for each model over all participants/RNN instances and then subtracting the summed WAIC value of the winning model (Model with the lowest WAIC value if summed over all participants/RNN instances).

Cumulative regret

Task performance was quantified using *cumulative regret*, i.e. the cumulative loss due to the selection of suboptimal options, a canonical metric to compare RL algorithms in machine learning [34,77,78]. Formally, this corresponds to the difference between the reward of the optimal action ($\mathrm{R}_{\mathrm{a}^*,\mathrm{t}}$) and the obtained reward ($\mathrm{R}_{\mathrm{a}_\mathrm{t},\mathrm{t}}$), summed across trials:

$$Cumulative\ Regret = \sum_{t} \mathrm{R}_{\mathrm{a}^*,\mathrm{t}} - \mathrm{R}_{\mathrm{a}_\mathrm{t},\mathrm{t}} \tag{41}$$

Lower cumulative regret corresponds to better performance. Note that human-level cumulative regret implies that an agent solved the exploration-exploitation tradeoff with human-level performance. Because other measures such as switch rates do not directly reflect performance, we refrained from using such measures as metrics for RNN architecture selection.

Hidden unit analysis

Deep learning algorithms like RNNs are often described as “black boxes” due to difficulties in understanding the mechanisms underlying their behavior [79]. To address this issue, hidden unit activity was analyzed in relation to behavior via dimensionality reduction techniques, in particular principal component analysis (PCA) and targeted dimensionality reduction (TDR, [7,29]). PCA was used to obtain a first intuition about internal dynamics, and TDR was used to analyze interpretable dimensions of the hidden unit data.

For PCA, hidden unit values were first centered and standardized. Then, the time course of the first three principal components (PCs) was plotted for individual RNN instances, and color-coded according to chosen option, state-value estimate, and stay versus switch decisions.

In contrast to PCA, TDR is a dimensionality reduction technique where the resulting high-dimensional neural activation data is projected onto axes with a specific interpretation. This is achieved by first using PCA to obtain an unbiased estimate of the most salient patterns of activations in the neural data and then regressing the resulting principal components against variables of interest. The resulting predicted values form the interpretable axes [7,29]. Following previous work in primate neurophysiology [7,29], for discrete variables (e.g. choice, stay/switch behavior) we used logistic regression:

$$p(choice = i|X) = \left(\frac{1}{1 + e^{-(X\beta_i)}}\right) \tag{33}$$

For continuous state-value estimates, we used linear regression:

$$\hat{Y} = \mathrm{B}_0 + X\mathrm{B}_1 \tag{34}$$

Here, X are the principal components based on the standardized hidden unit predictor matrix of size N (no. of trials) x M (no. of hidden units) and $\mathrm{B}_0\ and\ \mathrm{B}_1$ are vectors of size M (no. of hidden units). The resulting axis ($\hat{Y}$), a vector of size no. of trials, now has a specific meaning - "Given the principal components on a given trial, what is the predicted value of the state-value-estimate?". For discrete outcomes, the PCA-based hidden unit data were projected onto a choice predictive (or stay/switch-predictive) axis by inverting the logistic link function, i.e., for the case of the choice axis:

$$choice\ axis_i = log\left(\frac{p(choice = i|X)}{1 - p(choice = i|X)}\right) = X\beta_i \tag{35}$$

The $choice\ axis_i$, a vector of size no. of trials, again has a concrete interpretation: "Given the de-noised hidden units on a trial, what are the log-odds of observing $choice_i$"? If the log odds are positive, it is more likely, if it's negative it is less likely to observe $choice_i$. If predicted occurrence and non-occurrence of $choice_i$ is equiprobable the log odds are 0.

To decode decisions from de-noised hidden unit activity, we used the results of the logistic regression (Equation 40) to calculate the probability of each action. The action with the maximum probability in a given trial was taken as the predicted action, and the proportion of correctly predicted choices was taken as the decoding accuracy.

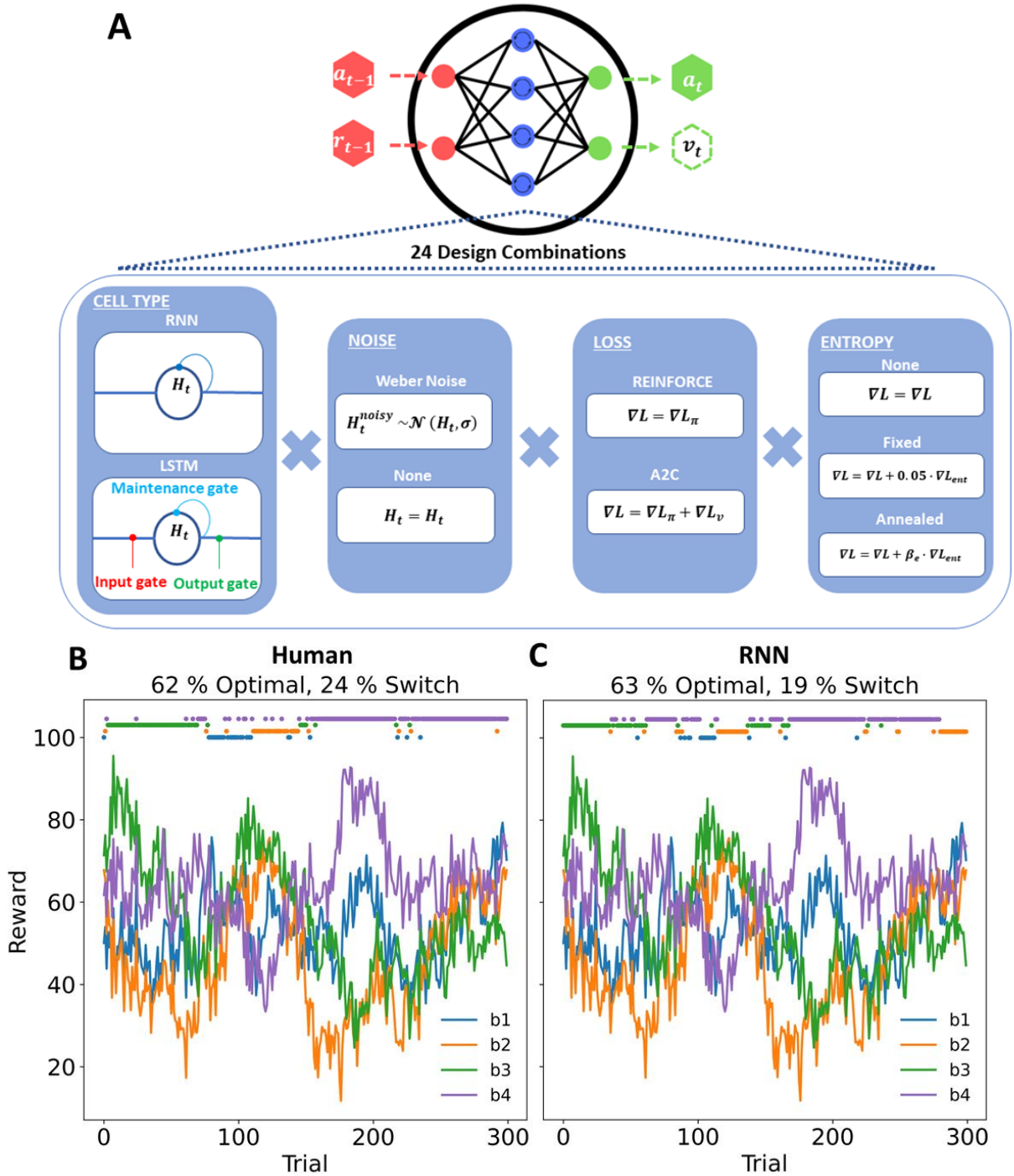


Figure 1. Artificial agent architectures and comparison of task performance to human agent. *(A) The input to the artificial agent is the previous reward (r_{t-1}) and the previous action (a_{t-1}) which is transformed within the hidden layer to output an action in the current trial (a_t) and an optional state-value estimate (v_t) (if the loss function is A2C). We systematically trained different network architectures varying in the factors Cell type (RNN or LSTM), Noise (Weber noise or none), Loss (REINFORCE or A2C) and Entropy (none, fixed or annealed) resulting in 24 design combinations (see methods section for details). (B) Example data from a human learner. (C) Example data from an LSTM network with computation noise solving the same task. In B and C, individual choices (colored dots on top) show selected action, and lines denote drifting rewards for each action. % Optimal: Proportion of choices of the most rewarding action. % Switches: Proportion of switches, i.e $choice_t$ not equal to $choice_{t-1}$.*

Code & Data Availability

Code (Python code for the RNN, Stan code for the cognitive models) and behavioral data of human and RNN (winning architecture) agents will be shared publicly upon publication.

For review purposes use the following view-only link:
https://osf.io/ndpbw/?view_only=a24df99b080f4fe0821a4bff65dc8c47

Results

Model-agnostic behavioral results

Our first aim was to identify the best-performing RNN architecture, as deep learning algorithms can be sensitive to hyperparameter settings [80,81]. The factors considered in the RNN model space are summarized in Table 1. Performance asymptote was reached by each architecture, such that there was no further improvement from 49.500 to 50.000 training episodes (all $p > .2$)). According to cumulative regret (Supplemental Figure 1), the best-performing architecture used LSTM units in combination with computation noise [32], and no entropy regularization during training [34]. All subsequent analyses therefore are focused on this architecture.

We next calculated cumulative regret for each agent (30 RNN instances of the best performing RNN, 31 human subjects from the placebo condition of Chakroun et al., 2020) solving the identical bandit problem (see methods). A Bayesian t-test on the mean cumulative regret on the final trial showed moderate evidence for comparable performance of RNNs and human subjects ($BF_{01} = 4.274$). This was confirmed when examining the posterior distribution of the standardized effect size, which was centered at zero ($M = -0.008$, Figure 2, B).

Analysis of switching behavior revealed that RNNs switched substantially less than human subjects (Bayesian Mann-Whitney U-Test $BF_{10} > 100$, Median switch probability: 31.5% (human), 14.5% (RNN)).

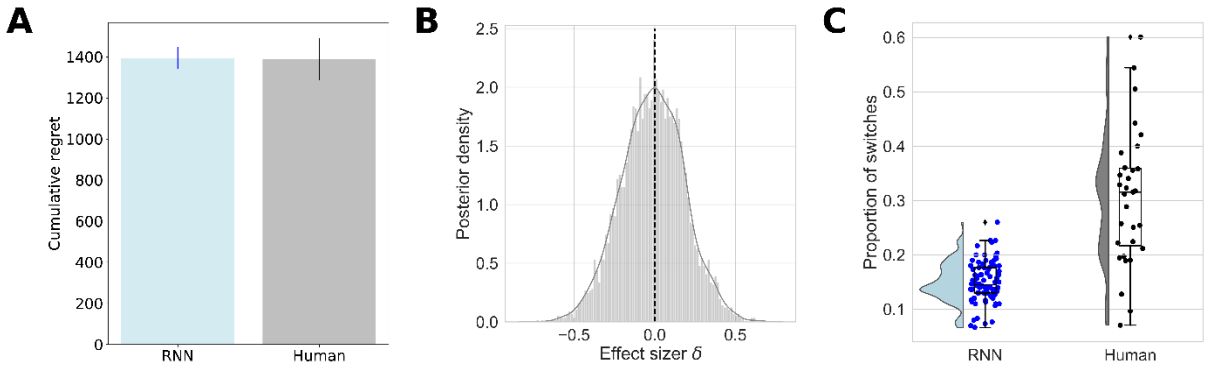


Figure 2. Behavioral data for LSTM networks with computation noise ("RNN", blue) and human learners ([11], Placebo condition, black). *(A) Mean (+/- SEM) cumulative regret over trials for RNNs (blue) and human learners (black) (B) Posterior distribution of the standardized effect size (δ, Bayesian T-Test) showing moderate evidence against a difference in cumulative regret between RNNs and human learners ($BF_{01} = 4.274$).(C) Proportion of switches for RNNs (blue) and human learners (black).*

Model comparison

To better understand human and RNN performance on this task, a total of 21 RL models (see methods section) were fitted to the behavioral data. All models were fitted to individual agent data (RNN instances from the best-performing architecture and human data from the placebo condition of [11]) via Hamiltonian monte Carlo as implemented in STAN. Model comparison was carried out using the Widely Applicable Information Criterion *WAIC* [76] by computing *ΔWAIC* scores for each model and agent (see methods section), yielding values of 0 for the best-fitting model. The best-fitting model differed for RNN and human agents. Whereas the Kalman-Filter model with higher-order perseveration (SM+DP) was the best model for RNNs, human data were better accounted for by the same model with an additional directed exploration parameter (SM+EDP, see also Supplemental Tables 1 and 2).

As can be seen from Figure 3, in RNNs, model fit of the SM+DP (without directed exploration terms) and SM+EDP model (with directed exploration terms) was highly similar. Note, the more basic SM+DP is a nested version of the SM+EDP model with the directed exploration parameter $\varphi = 0$. In such cases, a straightforward approach is to examine the more complex model, as all information about the nested parameter (here the directed exploration parameter φ) is contained in that model's posterior distribution [82]. All further analyses therefore focused on the SM+EDP model.

To quantify absolute model fit, the posterior predictive accuracy of the SM+EDP model was examined for each agent. Five-hundred data sets were simulated from the model's posterior distribution, and the proportion of trials in which the simulated and observed choices were consistent were computed and averaged across simulations. Bayesian Mann-Whitney U-Test showed very strong evidence that predictive accuracy (Figure 4A) was higher for RNNs

than for human learners (RNN: $Mdn = 0.793$, $range$: $[0.675, 0.895]$, Human: $Mdn = 0.666$, $range$: $[0.385, 0.904]$, $BF_{10} > 100$).

Analysis of model parameters

Next, we examined the model parameters (medians of individual subject posterior distributions) of the best-fitting model and compared them between human learners and RNNs, again focusing on the SM+EDP model (for corresponding results for the SM+DP model, see Supplemental Figure 2). All subsequent Bayes Factors (BF) are based on the Bayesian Mann-Whitney U-Test [83] testing for differences in median values between RNNs and human learners. Choice consistency (β) was lower for RNNs than human learners (Figure 4B, RNN: $Mdn = 0.113$, $range$: $[0.0698, 0.191]$, Human: $Mdn = 0.175$, $range$: $[0.0448, 0.305]$, $BF_{10} = 91.3$). RNNs showed substantially higher levels of perseveration (Figure 4C, RNN: $Mdn = 16.6$, $range$: $[6, 28.4]$, Human: $Mdn = 9.66$, $range$: $[-3.97, 22.6]$, $BF_{10} = 76.23$), but there was only anecdotal evidence for a greater habit step size than human learners (Figure 4D, RNN: $Mdn = 0.644$, $range$: $[0.233, 0.913]$, Human: $Mdn = 0.568$, $range$ $[0.108, 0.980]$, $BF_{10} = 2.31$). In line with the model comparison (see above) there was very strong evidence for reduced directed exploration in RNNs compared to human learners (Figure 4E, RNN: $Mdn = -0.234$, $range$: $[-3.23, 1.29]$), Human: $Mdn = 1.32$, $range$: $[-2.01, 5.40]$, $BF_{10} > 100$). Bayesian Wilcoxon Signed-Rank Tests [83] nonetheless showed strong evidence for φ estimates < 0 in RNNs ($BF_{10} > 19.98$) whereas φ estimates were reliably > 0 in human learners ($BF_{10} > 100$).

Taken together, both the model comparison and the analysis of parameter estimates show that human learners, but not RNNs, adopted a directed exploration strategy. In contrast, RNNs showed substantially increased perseveration behavior.

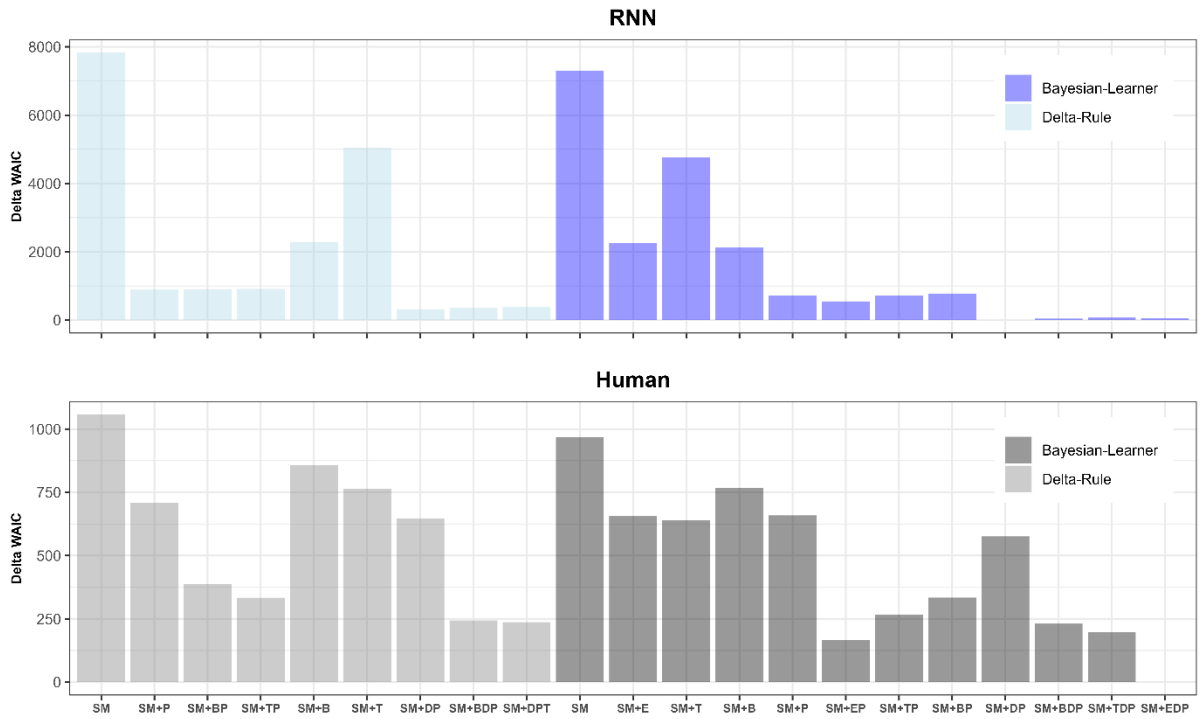


Figure 3. Model comparison via ΔWAIC (see methods section) where smaller values indicate a superior fit. *For LSTM networks with computation noise (top panel), the Bayesian learner with higher-order perseveration (SM+DP) accounted for the data best. For human learners (bottom panel), the same model, but including a uncertainty term (SM+EDP) fitted the data best.*

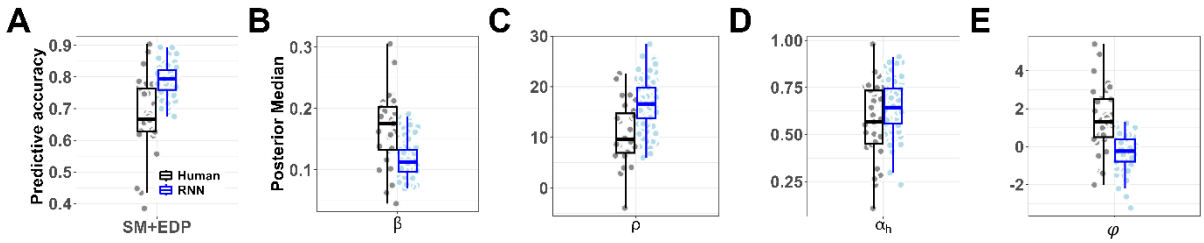


Figure 4. Median posterior values of model parameters for the best-fitting model (Bayesian learner with exploration and higher order perseveration terms, SM+EDP). *A: Predictive accuracy as the fraction of choices correctly predicted by the fitted cognitive model, B: choice stochasticity parameter β, C: perseveration parameter ρ, D: Habit step size parameter α_h, E: exploration parameter φ for human learners (black) and RNNs (blue).*

Hidden unit analysis

Finally, we investigated RNN hidden unit activity. This analysis is similar to the analysis of high dimensional neural data [35] and we first used PCA to visualize network dynamics. The first three principal components accounted for on average 73% of variance (see Supplemental Figure 3). The resulting network activation trajectories through principal component space

were then examined with respect to behavioral variables. Coding network state by behavioral *choice* revealed separated choice-specific clusters in principal component space (see Figure 5A for hidden unit data from one RNN instance, and see Supplemental Figure 4 for the corresponding visualizations for all instances). The degree of spatial overlap of the choice-specific clusters appeared to be related to the state-value estimate of the RNN (Figure 5B), with greater overlap for lower state values. Coding network state according to stay vs. switch behavior (repeat previous choice vs. switch to another bandit, as a raw metric for exploration behavior, Figure 5C) revealed that switches predominantly occurred in the region of activation space with maximum overlap in choice-specific clusters, corresponding to low state-value estimates. Highly similar patterns were observed for all RNN instances investigated (see Supplemental Figure 4).

One downside of the PCA-analysis is that components are not readily interpretable, and, due to differential rotation of these patterns in PC-space, is not straightforward to conduct analyses across all network instances. We address these issues via targeted dimensionality reduction (TDR) [29], which projects the PCA-based hidden unit data onto novel axes with specific interpretations (see methods section). We first used TDR to project the PCA-based hidden unit data onto a *value axis*, i.e., the predicted state-value estimate given the hidden unit activity on a given trial. Across all network instances, predicted state value was highly correlated with the reward obtained by the network on the previous trial ($r(17998) = .97$), Figure 5 D). We next explicitly tested the visual intuition that switching behavior predominantly occurs when state value is low, and projected the hidden unit data onto a *switch axis* via logistic regression (see methods section), corresponding to the log-odds of observing a switch. Positive log-odds indicated that a switch decision is more likely than a stay decision, and vice versa for negative log-odds. Results confirmed the results from the analysis of single network instances (e.g., Figure 5B, C): switches predominantly occurred when estimated state value was low, as reflected in a negative correlation of the *value axis* and the *switch axis* scores ($r(17998) = -.80$).

Further, we asked whether switching occurs randomly, or follows a predictable pattern. To this end, a multinomial model was fitted to predict RNN choices given the current PCA-based de-noised hidden unit activity. We then compared the accuracy of choice prediction between stay and switch trials. If RNNs follow a switching strategy, the accuracy of predicting switching decisions from de-noised hidden unit activity should be above chance level (0.25). The prediction accuracy was near perfect for stay decisions ($M = 0.996$, see Figure 5F) and markedly disrupted but still substantially above chance level for switch decisions ($M = 0.702$, see Figure 5F). This is consistent with the idea that RNNs rely on more than choice randomization to solve the exploration-exploitation dilemma.

But what determines RNN switching behavior? To address this, we compared switch targets (the option selected on a switch trial) to switch non-targets (the other bandits not selected on a switch-trial). A first comparison focused on uncertainty (based on the trial wise-heuristic, Eq. 30), and a second analysis focused on the last observed reward. Across RNN instances or human agents, we computed the mean rank of switch-targets and mean ranks of switch non-targets with respect to these two variables. A rank of 2 corresponds to selecting the bandit with the highest value on these measures, and a rank of 0 corresponds to selecting the bandit with the lowest value on these measures. Indeed, according to Bayesian Mann-Whitney U-Tests, human learners, compared to RNNs, showed a higher tendency to switch to bandits with higher uncertainty ($BF_{10} = 11.332$, Figure 6, A), whereas the tendency to switch to bandits with higher previous rewards was similar ($BF_{10} = 1.941$, Figure 6, B).

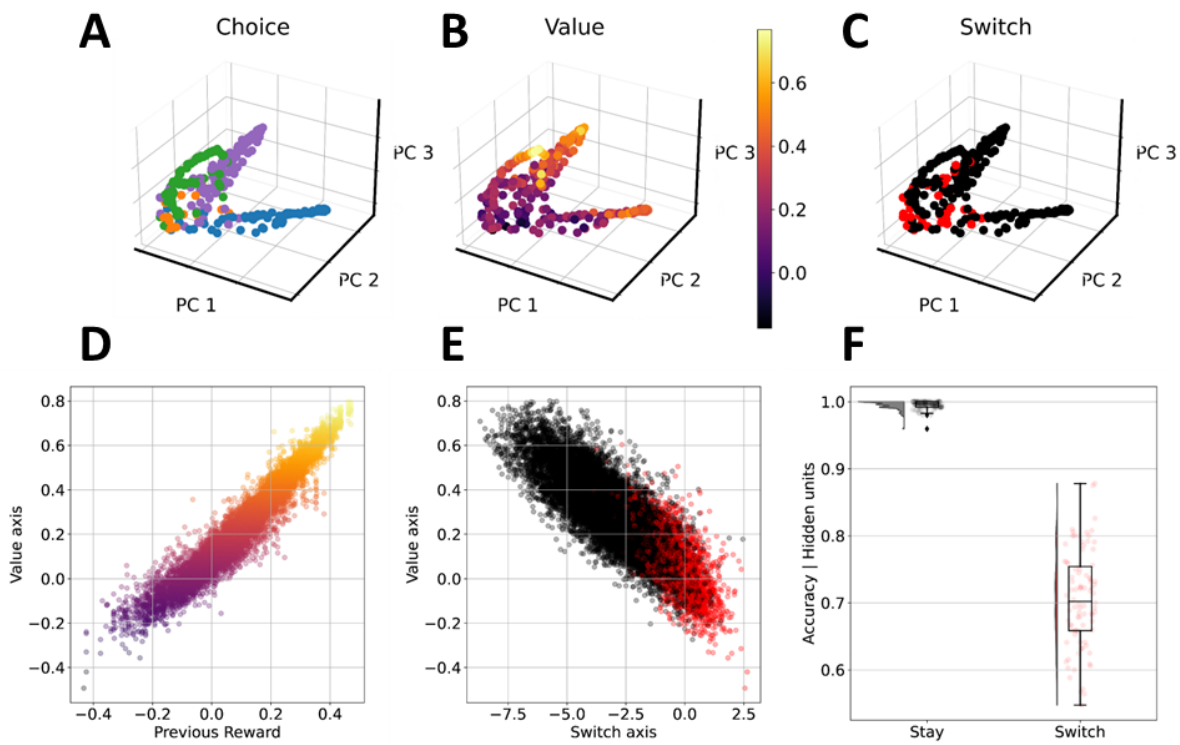


Figure 5. Hidden unit activation dynamics for a single network instance (A-C) and across all instances (D, E). A-C: *Hidden unit dynamics (first three principal components) of an example RNN agent color coded by choice (A), state-value estimate (B) and switching behavior (C, switch – red, stay - black). D, E: Targeted dimensionaility reduction. D: The state-value axis (y-axis) was highly correlated with previous reward (x-axis). Note that previous reward is a continuous variable in the range -0.5 and 0.5 (as rewards in the range: [0,1] were mean centered). E: Lower state-value (y-axis) was linked to greater log-odds of switching (x-axis). F: Accuracy of choice prediction given the PCA-based de-noised hidden unit activation state using a multinomial model revealed almost perfect accuracy for stay decisions (99%) and reduced, but above chance-level accuracy for switch decisions (70%).*

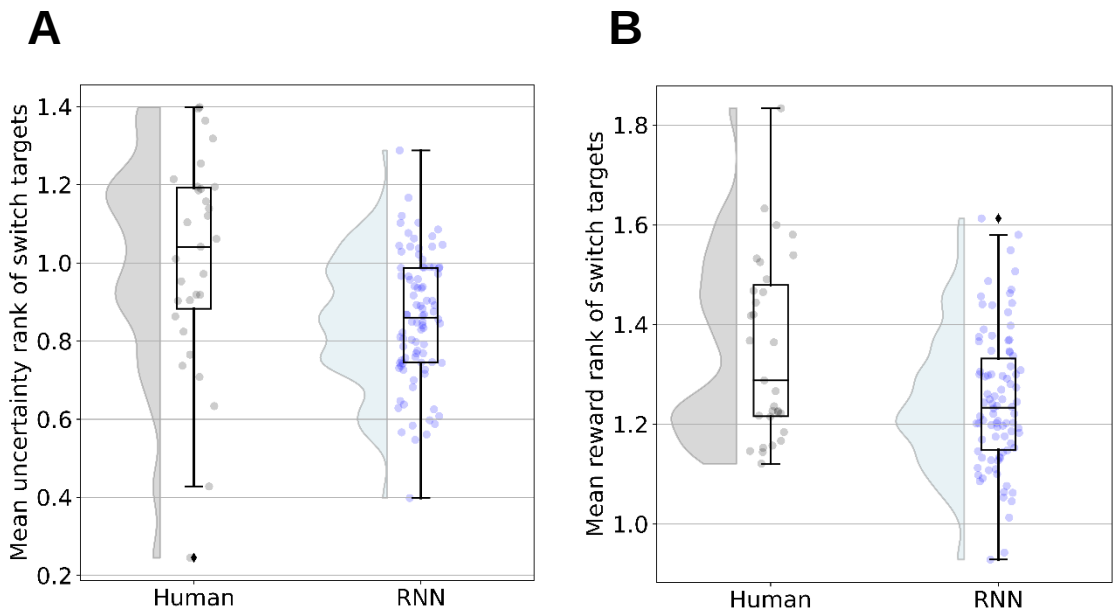


Figure 6. Switch-target analysis regarding uncertainty (A) and previous reward (B). *On a given switch trial the uncertainty and last observed reward of all three potential switch targets were ranked (0=lowest, 2 = highest). Points denote the mean rank of the switch-target across all switch trials per Human agent or RNN instance. Higher values indicate the tendency to switch to targets with more uncertainty (A) or more reward (B), whereas lower values indicate the opposite. Human agents (black) show evidence to switch to more uncertain options, whereas RNNs (blue) show uncertainty aversion (A). Both agents are comparable regarding the effect of reward rank on switch targets (B).*

Discussion

Here we comprehensively investigated exploration mechanisms in recurrent neural network (RNN) models trained to solve restless bandit problems, reinforcement learning tasks commonly used in cognitive and systems neuroscience. We expanded upon previous work in four ways. First, in contrast to earlier work [32,34] we focused on four-armed restless bandit problems, allowing for a more comprehensive analysis of exploration behavior. Second, we systematically investigated a range of RNN design choices and resulting impacts on performance. Third, we directly compared human and RNN behavior, both in terms of performance (cumulative regret) and using computational modeling, when solving the exact same task problem. Finally, we investigated exploration mechanisms in the best-performing network architecture via a comprehensive analysis of hidden unit activation dynamics.

We extensively tested and transparently report upon a total of twenty-four RNN design factor combinations. The architecture exhibiting the best performance was an LSTM network, combined with computation noise as previously suggested [32], no entropy regularization and trained with the advantage-actor-critic (A2C) algorithm [69]. The superior performance of the LSTM versus the vanilla RNN is not surprising. LSTMs are endowed with a more sophisticated memory process (Equations 5 - 9) where different gating mechanisms regulate the impact of past experiences (previous actions and rewards) on current choices. These mechanisms allow LSTM networks to learn dependencies over longer time scales than vanilla RNNs [66]. The

performance benefit of LSTMs also resonates with a well-known computational model of learning and action selection mechanisms in prefrontal cortex (PFC) and basal ganglia [84]. This model is characterized by a combination of LSTM-like gating mechanisms that are combined with an Actor-Critic architecture. Here, the *Actor* (i.e., the basal ganglia) updates actions by gating working memory updating processes in the PFC. The *Critic* (i.e., midbrain DA neurons) estimates reward values of possible actions, thereby adjusting future actions to maximize reward. Similar to Findling & Wyart [32] our results show the benefit of biologically-inspired computation noise ("Weber Noise", [32]) added to hidden unit activity that scales with the degree of recurrent activity reconfiguration between subsequent trials. Entropy regularization, in contrast, adds noise to the policy during training to discourage premature convergence to a suboptimal policy [69]. This "update-dependent" noise mechanism might thus entail exploration-specific noise that contrasts with the introduction of general stochasticity as implemented in entropy regularization schemes. This result further reinforces the performance-enhancing effect of computation noise observed in human reward-guided decision making [45] which is thought to be modulated by reciprocal connections of the locus coeruleus-norepinephrine system and the anterior cingulate cortex [27,37]. This also resonates with results from deep neural networks, where various noise-based schemes have been implemented to improve network performance and/or to increase the resilience of the networks under sparse information, e.g. noise at the input level, similar to dataset augmentation methods [85], at the level of the weight update [86] or during computation [41–43].

To clarify differences between human and RNN behavior, we applied comprehensive cognitive modeling [36]. Model comparison according to WAIC revealed that human behaviour was best accounted for by a Bayesian learning model (Kalman Filter) with an uncertainty-based directed exploration term and higher-order perseveration (SM+EDP). This model is an extension of a previously applied model (SM+EP) [11,13,24] with higher-order perseveration according to a formalism suggested by Miller et al. [71], and similar to Lau & Glimcher 2005. RNN behaviour, on the other hand, was best accounted for by the same model without a directed exploration term, although the numerical differences in WAIC to models with directed exploration were numerically very small.

In such situations where model comparison yields somewhat inconclusive results for a full model (SM+EDP) and nested versions of this model (e.g. SM+DP corresponds to SM+EDP with $\varphi = 0$), there are different possible ways to proceed. The first would be to only examine the best-fitting model (SM+DP). Alternatively, one could examine the full model (SM+EDP), as it contains the nested model as a special case, and allows to directly quantify the degree of evidence that e.g. $\varphi = 0$ [82]. After all, the posterior distribution of φ contains all information regarding the value of this parameter, given the data and the priors. Following

suggestions by Kruschke [82], we therefore focused on the full model. Indeed, this analysis revealed strong evidence that $\varphi < 0$ in RNNs, highlighting the caveats of solely relying on categorical model comparison in cases of nested models.

Both RNNs and human learners exhibited higher-order perseveration behavior, i.e., perseveration beyond the last trial. This tendency was substantially more pronounced in RNNs than in human learners, and is conceptually linked to exploration, because perseveration can be conceived of as an uncertainty-avoiding strategy. However, similar to analysis using first-order perseveration [11], accounting for higher-order perseveration did not abolish the robust evidence for directed exploration in our human data. This is in line with the idea that human learners may apply perseveration (uncertainty avoiding) and directed exploration (information gain) strategies in parallel [87]. RNNs showed substantially higher levels of perseveration than human learners. Perseveration is often thought to be maladaptive, as learners "stick" to choices regardless of reward or task demands [88–90] and it is a hallmark of depression and substance use disorders [91], behavioral addictions [92], obsessive compulsive disorder [93] and tightly linked to PFC functioning [94,95]. In the light of these findings, it might appear surprising that RNNs show a pronounced tendency to perseverate. But perseveration might support reward accumulation by enabling the network to minimize losses due to excessive exploration (see below).

Analysis of model parameters also revealed strong evidence for uncertainty aversion in RNNs (even when accounting for higher-order perseveration), reflected in an overall *negative* exploration bonus parameter φ (i.e., an "exploration malus"). In contrast, human learners showed the expected positive effect of uncertainty on choice probabilities (directed exploration) [8–11,13,96]. Importantly, this divergence between human and RNN mechanisms demonstrates that directed exploration is not required for human-level task performance. Overall, human learners perseverate less and explore more (Figure 2C), thereby potentially avoiding the costs of prolonged perseveration and finding the optimal bandit faster by continuously exploring the environment. Both strategies converge on comparable performance. Finally, RNNs showed a lower inverse temperature parameter (β) than human learners. All things being equal, lower values of β reflect more random action selection, such that choices depend less on the terms included in the model. A β-value of zero would indicate random choices, and as β increases, the policy approaches a deterministic policy in which choices depend completely upon the model terms. However, the absolute level of choice stochasticity reflected in a given value of β also depends on the absolute values of all terms included in the model. Whereas the absolute magnitude of value and exploration terms was comparable between human and RNNs, the perseveration term was about twice the magnitude in RNNs, which explains the lower β-values in RNNs. The results from the analysis

of predictive accuracy also confirmed that a greater proportion of choices was accounted for by the best-fitting computational model in RNNs compared to humans, showing that these differences in β do not reflect a poorer model fit.

To investigate the computational dynamics underlying RNN behaviour, we initially applied dimensionality reduction of hidden unit activations patterns via Principal Component Analysis (PCA) [29,32,34]. The first three principal components accounted for on average 73% of variance in hidden unit activity (see Supplemental Figure 3). Visual inspection of activation patterns in principal component space then revealed three effects: First, coding network state by behavioral *choice* revealed clearly separated choice-specific clusters in principal component space, an effect that was observed across all RNN instances examined (see Supplemental Figure 4). Second, the degree of spatial overlap of the choice-specific clusters directly related to the state-value estimate of the network. Action representations on trials with higher state-value estimates were more separated than during trials with lower state-value estimates. Again, this pattern was observed across all RNN instances examined (see Supplemental Figure 4) and resonates with systems neuroscience work showing neural populations are more predictive for high-value actions than for low-value actions [7]. Oculomotor regions like the frontal eye field (FEF) [97–100] and the lateral intraparietal area (LIP) [101,102] show more pronounced choice-predictive activation patterns during saccades to high vs. low value targets. Third, to investigate the link between RNN dynamics and exploration, we coded network state in PC-space by stay vs. switch behavior. This revealed that switches predominantly occurred in the region of activation space with maximum overlap in choice predictive clusters, corresponding to low state-value estimates. Again, this effect was observed across all network instances examined. Generally, these observations show that 1) switches occurred predominantly during trials with low state value, 2) low state value was associated with less pronounced choice-predictive activation patterns. Although these patterns were qualitatively highly similar across RNN instances (see Supplemental Figure 4), the geometrical embedding of these effects in principal component space differed. This illustrates one downside of PCA - the components as such are not directly interpretable, and the different rotations of patterns in principal component space complicate the aggregation of analyses across network instances.

To address this issue, and to obtain interpretable axes, we applied targeted dimensionality reduction (TDR) [7,29]. TDR projects the PCA-based de-noised hidden unit activation patterns onto novel axes with clear interpretations (see methods section), allowing for a quantification of the intuitions gained from PCA. We projected the de-noised hidden unit data onto a *value axis*, i.e., the predicted state-value estimate given the de-noised hidden unit activity on a given trial. Across all network instances, this measure was highly correlated with the reward obtained on the previous trial (Figure 5D). Likewise, we projected the de-noised

hidden unit data onto a *switch-axis*, i.e., the predicted log-odds of observing a switch, given the de-noised hidden unit activity on a given trial. Across all network instances, this axis showed a strong negative correlation with the value-axis, confirming that indeed the log-odds of switching increased with decreasing state value, and decreased with increasing state value, resembling a Win-Stay-Lose-Shift (WSLS) strategy [103] that accounts for substantial choice proportions also in human work [21]. However, pure WSLS would predict much higher switch rates than observed in RNNs, suggesting that RNNs show a mixture of a WSLS-like strategy in conjunction with high perseveration.

Finally, we decoded choices from de-noised hidden unit activation dynamics, and compared prediction accuracy for stay vs. switch decisions. The decoder showed near perfect accuracy for stay decisions, which resonates with animal work showing that neural choice decoding is improved during perseveration [7,104]. Importantly, performance of the decoder was lower, but still substantially above chance-level for switches. The de-noised hidden units therefore represent an activation pattern that can be utilized to correctly predict switch-targets, suggesting that switching behavior is not entirely based on choice randomization.

Further analyses revealed that RNNs and human agents tended to switch to bandits with higher previous reward, but human agents showed a stronger preference to switch to more uncertain options than RNN agents (See Figure 6). Switching to bandits with higher previous rewards could be computationally explained by value-based exploration ("Boltzman Exploration", [1]) as implemented in standard RL-models with a softmax action selection function, where choice probabilities are proportional to trial-by-trial value estimates. As probabilities in the softmax function sum to 1, decreases in the value of one bandit increase the choice probabilities of other bandits proportional to their value estimates. The difference in uncertainty-based preference in switch targets resonates well with our cognitive modeling results, where human agents show strong evidence for an uncertainty bonus (positive φ estimate). In contrast RNN agents show an uncertainty malus (negative φ estimate) exploring in an uncertainty averse manner. Generally, these findings confirm that RNNs and human agents explore similarly in a value-based manner, but show a dissociation regarding uncertainty-based exploration.

One caveat of this work is that, although often applied in the context of RL in volatile environments [28,105,106], the comparison between stay and switch trials does not unequivocally map onto the exploitation vs. exploration distinction. For example, stay decisions can be due to greedy choices (choosing the option with the highest expected reward) but also due to perseveration. In contrast, switch decisions can be due to random or strategic exploration [10] and may involve more complex model-based strategies and/or simpler heuristics like following motor patterns such as exploring by choosing each available

option once and then exploiting [63]. We nonetheless applied the stay vs. switch distinction, as it makes by far the least assumptions regarding what constitutes exploration vs. exploration.

Several limitations of this work need to be addressed. First, although our final network model space resulted in a total of 24 different RNN architectures, the impact of additional design choices such as network size, learning rate, discount factor, type of activation function or values for the Weber fraction (noise) were not systematically explored. Although the combination of LSTM with the A2C algorithm is robust to different hyperparameter settings [69], a different RNN architecture or hyperparameter combination could have yielded even better performance or could have produced a form of directed exploration. Future research could benefit from the use of other architectures such as transformer models [107–109] or explore the role of these additional factors. Second, a general limitation of this approach more generally is that neural network models, although roughly based on neuronal computations, suffer from a number of biological implausibilities [110]. These include the backpropagation algorithm used to update the parameters of the network [111], the lack of separate modules analogous to different brain regions, and lack of neuromodulation mechanisms [110]. However, some recent work has begun to address these shortcomings [112,113]. Last, it can be argued that the restless bandit task is a suboptimal paradigm to investigate directed exploration, because reward and uncertainty are confounded, unlike other tasks, such as the horizon task [9]. But the episodic character of this task (a series of discrete games where reward-related information is reset for every episode) has the caveat that explore-exploit behaviour can only unfold over a relatively small time frame (1-6 trials), whereas in the restless bandit task it can evolve over much longer time periods (e.g. 300 trials).

Taken together, we identified a novel RNN architecture (LSTM with computation noise) that solved restless four-armed bandit tasks with human-level accuracy. Cognitive modeling showed both human learners and RNNs exhibit higher-order perseveration behavior on this task, which was substantially more pronounced for RNNs. Human learners, on the other hand, but not RNNs, exhibited a directed (uncertainty-based) exploration. Analyses of the networks' exploration behavior confirmed that exploratory choices in RNNs were primarily driven by rewards and choice history. Hidden-unit dynamics revealed that exploration behavior in RNNs was driven by a disruption of choice predictive signals during states of low estimated state value, reminiscent of computational mechanisms in monkey PFC. Overall, our results highlight how computational mechanisms in RNNs can at the same time converge with and diverge from findings in human and systems neuroscience.

Declarations Section

Author Contribution

Conceptualization: DT, IP, JP. Methods: DT, AB, IP, JP. Implementation: DT, IP. Writing (first draft): DT. Writing (review and editing): AB, IP, JP. Supervision: JP. Funding acquisition: JP.

Ethics Approval and Consent to Participate

Not applicable

Funding

This work was funded by Deutsche Forschungsgemeinschaft (Code 496990750 to J.P.).

Conflict of Interests

All authors declare that they have no conflicts of interest.

References

1. Sutton RS, Barto AG. Reinforcement learning: an introduction. Second edition. Cambridge, Massachusetts: The MIT Press; 2018.

2. Thorndike EL. The Law of Effect. Am J Psychol. 1927;39: 212–222. doi:10.2307/1415413

3. Huys QJM, Maia TV, Frank MJ. Computational psychiatry as a bridge from neuroscience to clinical applications. Nat Neurosci. 2016;19: 404–413. doi:10.1038/nn.4238

4. Maia TV, Frank MJ. From reinforcement learning models to psychiatric and neurological disorders. Nat Neurosci. 2011;14: 154–162. doi:10.1038/nn.2723

5. Yahata N, Kasai K, Kawato M. Computational neuroscience approach to biomarkers and treatments for mental disorders. Psychiatry Clin Neurosci. 2017;71: 215–237. doi:10.1111/pcn.12502

6. Daw ND, O'Doherty JP, Dayan P, Seymour B, Dolan RJ. Cortical substrates for exploratory decisions in humans. Nature. 2006;441: 876–879. doi:10.1038/nature04766

7. Ebitz RB, Albarran E, Moore T. Exploration Disrupts Choice-Predictive Signals and Alters Dynamics in Prefrontal Cortex. Neuron. 2018;97: 450-461.e9. doi:10.1016/j.neuron.2017.12.007

8. Schulz E, Gershman SJ. The algorithmic architecture of exploration in the human brain. Curr Opin Neurobiol. 2019;55: 7–14. doi:10.1016/j.conb.2018.11.003

9. Wilson RC, Geana A, White JM, Ludvig EA, Cohen JD. Humans Use Directed and Random Exploration to Solve the Explore–Exploit Dilemma. J Exp Psychol Gen. 2014;143: 2074–2081. doi:10.1037/a0038199

10. Wilson RC, Bonawitz E, Costa VD, Ebitz RB. Balancing exploration and exploitation with information and randomization. Curr Opin Behav Sci. 2021;38: 49–56. doi:10.1016/j.cobeha.2020.10.001

11. Chakroun K, Mathar D, Wiehler A, Ganzer F, Peters J. Dopaminergic modulation of the exploration/exploitation trade-off in human decision-making. Gershman SJ, Frank MJ, Gershman SJ, Averbeck BB, Pearson J, editors. eLife. 2020;9: e51260. doi:10.7554/eLife.51260

12. Speekenbrink M, Konstantinidis E. Uncertainty and exploration in a restless bandit problem. Top Cogn Sci. 2015;7: 351–367. doi:10.1111/tops.12145

13. Wiehler A, Chakroun K, Peters J. Attenuated directed exploration during reinforcement learning in gambling disorder. J Neurosci. 2021;41: 2512–2522. doi:10.1523/JNEUROSCI.1607-20.2021

14. Wu CM, Schulz E, Speekenbrink M, Nelson JD, Meder B. Generalization guides human exploration in vast decision spaces. Nat Hum Behav. 2018;2: 915–924. doi:10.1038/s41562-018-0467-4

15. Lau B, Glimcher PW. Dynamic Response-by-Response Models of Matching Behavior in Rhesus Monkeys. J Exp Anal Behav. 2005;84: 555–579. doi:10.1901/jeab.2005.110-04

16. Ito M, Doya K. Validation of Decision-Making Models and Analysis of Decision Variables in the Rat Basal Ganglia. J Neurosci. 2009;29: 9861–9874. doi:10.1523/JNEUROSCI.6157-08.2009

17. Balcarras M, Ardid S, Kaping D, Everling S, Womelsdorf T. Attentional Selection Can Be Predicted by Reinforcement Learning of Task-relevant Stimulus Features Weighted by Value-independent Stickiness. J Cogn Neurosci. 2016;28: 333–349. doi:10.1162/jocn_a_00894

18. Gershman SJ. Origin of perseveration in the trade-off between reward and complexity. Cognition. 2020;204: 104394. doi:10.1016/j.cognition.2020.104394

19. Palminteri S. Choice-confirmation bias and gradual perseveration in human reinforcement learning. Behav Neurosci. 2023;137: 78–88. doi:10.1037/bne0000541

20. Seymour B, Daw ND, Roiser JP, Dayan P, Dolan R. Serotonin Selectively Modulates Reward Value in Human Decision-Making. J Neurosci. 2012;32: 5833–5842. doi:10.1523/JNEUROSCI.0053-12.2012

21. Worthy DA, Hawthorne MJ, Otto AR. Heterogeneity of strategy use in the Iowa gambling task: A comparison of win-stay/lose-shift and reinforcement learning models. Psychon Bull Rev. 2013;20: 364–371. doi:10.3758/s13423-012-0324-9

22. Badre D, Doll BB, Long NM, Frank MJ. Rostrolateral prefrontal cortex and individual differences in uncertainty-driven exploration. Neuron. 2012;73: 595–607. doi:10.1016/j.neuron.2011.12.025

23. Payzan-LeNestour E. Learning to Choose the Right Investment in an Unstable World: Experimental Evidence Based on the Bandit Problem. Rochester, NY; 2012. doi:10.2139/ssrn.1628657

24. Beharelle AR, Polanía R, Hare TA, Ruff CC. Transcranial Stimulation over Frontopolar Cortex Elucidates the Choice Attributes and Neural Mechanisms Used to Resolve Exploration–Exploitation Trade-Offs. J Neurosci. 2015;35: 14544–14556. doi:10.1523/JNEUROSCI.2322-15.2015

25. Cremer A, Kalbe F, Müller JC, Wiedemann K, Schwabe L. Disentangling the roles of dopamine and noradrenaline in the exploration-exploitation tradeoff during human decision-making. Neuropsychopharmacology. 2023;48: 1078–1086. doi:10.1038/s41386-022-01517-9

26. Dubois M, Habicht J, Michely J, Moran R, Dolan RJ, Hauser TU. Human complex exploration strategies are enriched by noradrenaline-modulated heuristics. Kahnt T, Büchel C, Warren C, editors. eLife. 2021;10: e59907. doi:10.7554/eLife.59907

27. McClure S, Gilzenrat MS, Cohen J. An exploration-exploitation model based on norepinepherine and dopamine activity. 2005.

28. Swanson K, Averbeck BB, Laubach M. Noradrenergic regulation of Win-Stay/Lose-Shift policy and choice determinism in a two-armed bandit task. bioRxiv; 2020. p. 2020.11.13.382069. doi:10.1101/2020.11.13.382069

29. Mante V, Sussillo D, Shenoy KV, Newsome WT. Context-dependent computation by recurrent dynamics in prefrontal cortex. Nature. 2013;503: 78–84. doi:10.1038/nature12742

30. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521: 436–444. doi:10.1038/nature14539

31. Botvinick M, Wang JX, Dabney W, Miller KJ, Kurth-Nelson Z. Deep Reinforcement Learning and Its Neuroscientific Implications. Neuron. 2020;107: 603–616. doi:10.1016/j.neuron.2020.06.014

32. Findling C, Wyart V. Computation noise promotes cognitive resilience to adverse conditions during decision-making. bioRxiv; 2020. doi:10.1101/2020.06.10.145300

33. Tsuda B, Tye KM, Siegelmann HT, Sejnowski TJ. A modeling framework for adaptive lifelong learning with transfer and savings through gating in the prefrontal cortex. Proc Natl Acad Sci. 2020;117: 29872–29882. doi:10.1073/pnas.2009591117

34. Wang JX, Kurth-Nelson Z, Kumaran D, Tirumala D, Soyer H, Leibo JZ, et al. Prefrontal cortex as a meta-reinforcement learning system. Nat Neurosci. 2018;21: 860–868. doi:10.1038/s41593-018-0147-8

35. Cunningham JP, Yu BM. Dimensionality reduction for large-scale neural recordings. Nat Neurosci. 2014;17: 1500–1509. doi:10.1038/nn.3776

36. Farrell S, Lewandowsky S. Computational Modeling of Cognition and Behavior. Cambridge: Cambridge University Press; 2018. doi:10.1017/CBO9781316272503

37. Findling C, Wyart V. Computation noise in human learning and decision-making: origin, impact, function. Curr Opin Behav Sci. 2021;38: 124–132. doi:10.1016/j.cobeha.2021.02.018

38. Song HF, Yang GR, Wang X-J. Reward-based training of recurrent neural networks for cognitive and value-based tasks. Behrens TE, editor. eLife. 2017;6: e21492. doi:10.7554/eLife.21492

39. Dezfouli A, Griffiths K, Ramos F, Dayan P, Balleine BW. Models that learn how humans learn: The case of decision-making and its disorders. PLoS Comput Biol. 2019;15: e1006903. doi:10.1371/journal.pcbi.1006903

40. Dasgupta I, Wang J, Chiappa S, Mitrovic J, Ortega P, Raposo D, et al. Causal Reasoning from Meta-reinforcement Learning. arXiv; 2019. doi:10.48550/arXiv.1901.08162

41. Dong Z, Oktay D, Poole B, Alemi AA. On Predictive Information in RNNs. arXiv; 2020. doi:10.48550/arXiv.1910.09578

42. Fortunato M, Azar MG, Piot B, Menick J, Osband I, Graves A, et al. Noisy Networks for Exploration. arXiv; 2019. doi:10.48550/arXiv.1706.10295

43. Qin M, Vucinic D. Training Recurrent Neural Networks against Noisy Computations during Inference. arXiv; 2018. doi:10.48550/arXiv.1807.06555

44. Drugowitsch J, Wyart V, Devauchelle A-D, Koechlin E. Computational Precision of Mental Inference as Critical Source of Human Choice Suboptimality. Neuron. 2016;92: 1398–1411. doi:10.1016/j.neuron.2016.11.005

45. Findling C, Skvortsova V, Dromnelle R, Palminteri S, Wyart V. Computational noise in reward-guided learning drives behavioral variability in volatile environments. Nat Neurosci. 2019;22: 2066–2077. doi:10.1038/s41593-019-0518-9

46. Renart A, Machens CK. Variability in neural activity and behavior. Curr Opin Neurobiol. 2014;25: 211–220. doi:10.1016/j.conb.2014.02.013

47. Kumar S, Dasgupta I, Marjieh R, Daw ND, Cohen JD, Griffiths TL. Disentangling Abstraction from Statistical Pattern Matching in Human and Machine Learning. arXiv; 2022. doi:10.48550/arXiv.2204.01437

48. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, et al. Human-level control through deep reinforcement learning. Nature. 2015;518: 529–533. doi:10.1038/nature14236

49. Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, et al. Mastering the game of Go without human knowledge. Nature. 2017;550: 354–359. doi:10.1038/nature24270

50. Silver D, Hubert T, Schrittwieser J, Antonoglou I, Lai M, Guez A, et al. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. Science. 2018;362: 1140–1144. doi:10.1126/science.aar6404

51. Wauthier ST, Mazzaglia P, Çatal O, De Boom C, Verbelen T, Dhoedt B. A learning gap between neuroscience and reinforcement learning. arXiv; 2021. doi:10.48550/arXiv.2104.10995

52. Botvinick M, Ritter S, Wang JX, Kurth-Nelson Z, Blundell C, Hassabis D. Reinforcement Learning, Fast and Slow. Trends Cogn Sci. 2019;23: 408–422. doi:10.1016/j.tics.2019.02.006

53. Lake BM, Salakhutdinov R, Tenenbaum JB. Human-level concept learning through probabilistic program induction. Science. 2015;350: 1332–1338. doi:10.1126/science.aab3050

54. Marcus G. Deep Learning: A Critical Appraisal. arXiv; 2018. doi:10.48550/arXiv.1801.00631

55. Tsividis PA, Loula J, Burga J, Foss N, Campero A, Pouncy T, et al. Human-Level Reinforcement Learning through Theory-Based Modeling, Exploration, and Planning. arXiv; 2021. Available: http://arxiv.org/abs/2107.12544

56. Hao J, Yang T, Tang H, Bai C, Liu J, Meng Z, et al. Exploration in Deep Reinforcement Learning: A Comprehensive Survey. arXiv; 2023. Available: http://arxiv.org/abs/2109.06668

57. Ladosz P, Weng L, Kim M, Oh H. Exploration in deep reinforcement learning: A survey. Inf Fusion. 2022;85: 1–22. doi:10.1016/j.inffus.2022.03.003

58. Binz M, Schulz E. Using cognitive psychology to understand GPT-3. arXiv; 2022. doi:10.48550/arXiv.2206.14576

59. Hamid AA, Pettibone JR, Mabrouk OS, Hetrick VL, Schmidt R, Vander Weele CM, et al. Mesolimbic dopamine signals the value of work. Nat Neurosci. 2016;19: 117–126. doi:10.1038/nn.4173

60. Mohebi A, Pettibone JR, Hamid AA, Wong J-MT, Vinson LT, Patriarchi T, et al. Dissociable dopamine dynamics for learning and motivation. Nature. 2019;570: 65–70. doi:10.1038/s41586-019-1235-y

61. Behrens TEJ, Woolrich MW, Walton ME, Rushworth MFS. Learning the value of information in an uncertain world. Nat Neurosci. 2007;10: 1214–1221. doi:10.1038/nn1954

62. Kalman RE. A New Approach to Linear Filtering and Prediction Problems. J Basic Eng. 1960;82: 35–45. doi:10.1115/1.3662552

63. Fintz M, Osadchy M, Hertz U. Using deep learning to predict human decisions and using cognitive models to explain deep learning models. Sci Rep. 2022;12: 4736. doi:10.1038/s41598-022-08863-0

64. Wicherts JM, Veldkamp CLS, Augusteijn HEM, Bakker M, van Aert RCM, van Assen MALM. Degrees of Freedom in Planning, Running, Analyzing, and Reporting Psychological Studies: A Checklist to Avoid p-Hacking. Front Psychol. 2016;7: 1832. doi:10.3389/fpsyg.2016.01832

65. Elman JL. Finding structure in time. Cogn Sci. 1990;14: 179–211. doi:10.1016/0364-0213(90)90002-E

66. Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Comput. 1997;9: 1735–1780. doi:10.1162/neco.1997.9.8.1735

67. Rehmer A, Kroll A. On the vanishing and exploding gradient problem in Gated Recurrent Units. IFAC-Pap. 2020;53: 1243–1248. doi:10.1016/j.ifacol.2020.12.1342

68. WILLIAMS RJ, PENG J. Function Optimization using Connectionist Reinforcement Learning Algorithms. Connect Sci. 1991;3: 241–268. doi:10.1080/09540099108946587

69. Mnih V, Badia AP, Mirza M, Graves A, Harley T, Lillicrap TP, et al. Asynchronous methods for deep reinforcement learning. Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48. New York, NY, USA: JMLR.org; 2016. pp. 1928–1937.

70. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. 2010. Available: https://www.semanticscholar.org/paper/Understanding-the-difficulty-of-training-deep-Glorot-Bengio/b71ac1e9fb49420d13e084ac67254a0bbd40f83f

71. Miller KJ, Shenhav A, Ludvig EA. Habits without values. Psychol Rev. 2019;126: 292–311. doi:10.1037/rev0000120

72. Stan Development Team. RStan: the R interface to Stan. 2022. Available: http://mc-stan.org/

73. R Core Team. R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing; 2022. Available: https://www.R-project.org/

74. Gelman A, Rubin DB. Inference from Iterative Simulation Using Multiple Sequences. Stat Sci. 1992;7: 457–472. doi:10.1214/ss/1177011136

75. Vehtari A, Gabry J, Magnusson M, Yao Y, Bürkner P-C, Paananen T, et al. loo: Efficient leave-one-out cross-validation and WAIC for Bayesian models. 2022. Available: https://mc-stan.org/loo/

76. Vehtari A, Gelman A, Gabry J. Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. Stat Comput. 2017;27: 1413–1432. doi:10.1007/s11222-016-9696-4

77. Agrawal S, Goyal N. Analysis of Thompson Sampling for the multi-armed bandit problem. arXiv; 2012. doi:10.48550/arXiv.1111.1797

78. Auer P, Cesa-Bianchi N, Fischer P. Finite-time Analysis of the Multiarmed Bandit Problem. Mach Learn. 2002;47: 235–256. doi:10.1023/A:1013689704352

79. Sussillo D, Barak O. Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. Neural Comput. 2013;25: 626–649. doi:10.1162/NECO_a_00409

80. Haarnoja T, Ha S, Zhou A, Tan J, Tucker G, Levine S. Learning to Walk via Deep Reinforcement Learning. arXiv; 2019. doi:10.48550/arXiv.1812.11103

81. Henderson P, Islam R, Bachman P, Pineau J, Precup D, Meger D. Deep Reinforcement Learning that Matters. arXiv; 2019. doi:10.48550/arXiv.1709.06560

82. Kruschke JK, editor. Doing Bayesian Data Analysis (Second Edition). Second Edition. Doing Bayesian Data Analysis (Second Edition). Second Edition. Boston: Academic Press; 2015. pp. i–ii. doi:https://doi.org/10.1016/B978-0-12-405888-0.09999-2

83. van Doorn J, Ly A, Marsman M, Wagenmakers E-J. Bayesian rank-based hypothesis testing for the rank sum test, the signed rank test, and Spearman's ρ. J Appl Stat. 2020;47: 2984–3006. doi:10.1080/02664763.2019.1709053

84. O'Reilly RC, Frank MJ. Making working memory work: a computational model of learning in the prefrontal cortex and basal ganglia. Neural Comput. 2006;18: 283–328. doi:10.1162/089976606775093909

85. Goodfellow I, Bengio Y, Courville A. Deep learning. MIT Press; 2016.

86. An G. The Effects of Adding Noise During Backpropagation Training on a Generalization Performance. Neural Comput. 1996;8: 643–674. doi:10.1162/neco.1996.8.3.643

87. Payzan-LeNestour E, Dunne S, Bossaerts P, O'Doherty JP. The neural representation of unexpected uncertainty during value-based decision making. Neuron. 2013;79: 191–201. doi:10.1016/j.neuron.2013.04.037

88. Dehais F, Hodgetts HM, Causse M, Behrend J, Durantin G, Tremblay S. Momentary lapse of control: A cognitive continuum approach to understanding and mitigating perseveration in human error. Neurosci Biobehav Rev. 2019;100: 252–262. doi:10.1016/j.neubiorev.2019.03.006

89. Hauser MD. Perseveration, inhibition and the prefrontal cortex: a new look. Curr Opin Neurobiol. 1999;9: 214–222. doi:10.1016/S0959-4388(99)80030-0

90. Hotz G, Helm-Estabrooks N. Perseveration. Part I: A review. Brain Inj. 1995;9: 151–159. doi:10.3109/02699059509008188

91. Zuhlsdorff K. Investigating reinforcement learning processes in depression and substance use disorder: translational, computational and neuroimaging approaches. Thesis, Cambridge University. 2022. doi:10.17863/CAM.91233

92. de Ruiter MB, Veltman DJ, Goudriaan AE, Oosterlaan J, Sjoerds Z, van den Brink W. Response Perseveration and Ventral Prefrontal Sensitivity to Reward and Punishment in Male Problem Gamblers and Smokers. Neuropsychopharmacology. 2009;34: 1027–1038. doi:10.1038/npp.2008.175

93. Apergis-Schoute A, Ip HYS. Reversal Learning in Obsessive Compulsive Disorder: Uncertainty, Punishment, Serotonin and Perseveration. Biol Psychiatry. 2020;87: S125–S126. doi:10.1016/j.biopsych.2020.02.339

94. Goldberg E, Bilder RM. The Frontal Lobes and Hierarchical Organization of Cognitive Control. The Frontal Lobes Revisited. Psychology Press; 1987.

95. Munakata Y, Morton JB, Stedron JM. The role of prefrontal cortex in perseveration: Developmental and computational explorations. Connectionist models of development: Developmental processes in real and artificial neural networks. New York, NY, US: Psychology Press; 2003. pp. 83–114.

96. Schulz E, Wu CM, Ruggeri A, Meder B. Searching for Rewards Like a Child Means Less Generalization and More Directed Exploration. Psychol Sci. 2019;30: 1561–1572. doi:10.1177/0956797619863663

97. Ding L, Hikosaka O. Comparison of Reward Modulation in the Frontal Eye Field and Caudate of the Macaque. J Neurosci. 2006;26: 6695–6703. doi:10.1523/JNEUROSCI.0836-06.2006

98. Glaser JI, Wood DK, Lawlor PN, Ramkumar P, Kording KP, Segraves MA. Role of expected reward in frontal eye field during natural scene search. J Neurophysiol. 2016;116: 645–657. doi:10.1152/jn.00119.2016

99. Roesch MR, Olson CR. Impact of expected reward on neuronal activity in prefrontal cortex, frontal and supplementary eye fields and premotor cortex. J Neurophysiol. 2003;90: 1766–1789. doi:10.1152/jn.00019.2003

100. Roesch MR, Olson CR. Neuronal activity related to anticipated reward in frontal cortex: does it represent value or reflect motivation? Ann N Y Acad Sci. 2007;1121: 431–446. doi:10.1196/annals.1401.004

101. Platt ML, Glimcher PW. Neural correlates of decision variables in parietal cortex. Nature. 1999;400: 233–238. doi:10.1038/22268

102. Sugrue LP, Corrado GS, Newsome WT. Matching behavior and the representation of value in the parietal cortex. Science. 2004;304: 1782–1787. doi:10.1126/science.1094765

103. Herrnstein RJ. The matching law: Papers in psychology and economics. Cambridge, MA, US: (H. Rachlin & D. I. Laibson, Eds.). Russell Sage Foundation; Harvard University Press.; 1997.

104. Coe B, Tomihara K, Matsuzawa M, Hikosaka O. Visual and Anticipatory Bias in Three Cortical Eye Fields of the Monkey during an Adaptive Decision-Making Task. J Neurosci. 2002;22: 5081–5090. doi:10.1523/JNEUROSCI.22-12-05081.2002

105. Domenech P, Rheims S, Koechlin E. Neural mechanisms resolving exploitation-exploration dilemmas in the medial prefrontal cortex. Science. 2020;369: eabb0184. doi:10.1126/science.abb0184

106. Kovach CK, Daw ND, Rudrauf D, Tranel D, O'Doherty JP, Adolphs R. Anterior Prefrontal Cortex Contributes to Action Selection through Tracking of Recent Reward Trends. J Neurosci. 2012;32: 8434–8442. doi:10.1523/JNEUROSCI.5468-11.2012

107. Chen L, Lu K, Rajeswaran A, Lee K, Grover A, Laskin M, et al. Decision Transformer: Reinforcement Learning via Sequence Modeling. Advances in Neural Information Processing Systems. Curran Associates, Inc.; 2021. pp. 15084–15097. Available: https://proceedings.neurips.cc/paper/2021/hash/7f489f642a0ddb10272b5c31057f0663-Abstract.html

108. Parisotto E, Song F, Rae J, Pascanu R, Gulcehre C, Jayakumar S, et al. Stabilizing Transformers for Reinforcement Learning. Proceedings of the 37th International Conference on Machine Learning. PMLR; 2020. pp. 7487–7498.

109. Upadhyay U, Shah N, Ravikanti S, Medhe M. Transformer Based Reinforcement Learning For Games. arXiv; 2019. doi:10.48550/arXiv.1912.03918

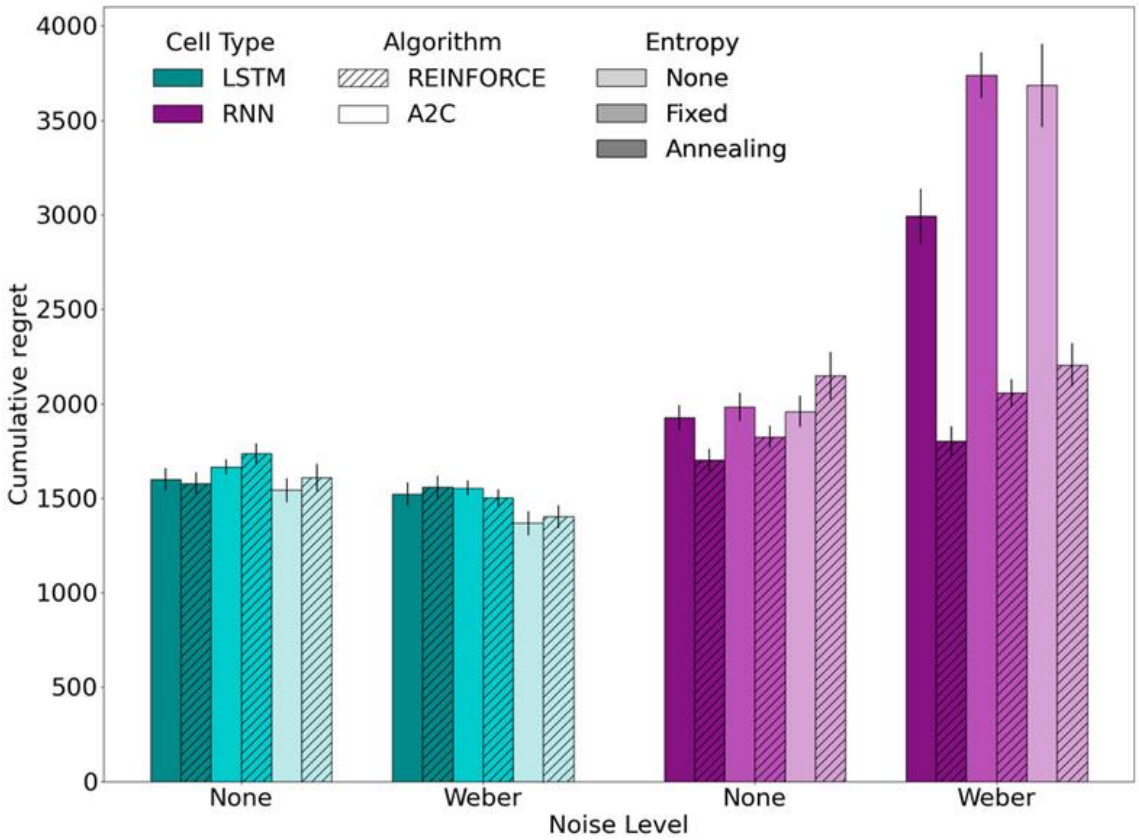
110. Pulvermüller F, Tomasello R, Henningsen-Schomers MR, Wennekers T. Biological constraints on neural network models of cognitive function. Nat Rev Neurosci. 2021;22: 488–502. doi:10.1038/s41583-021-00473-5

111. Lillicrap TP, Santoro A, Marris L, Akerman CJ, Hinton G. Backpropagation and the brain. Nat Rev Neurosci. 2020;21: 335–346. doi:10.1038/s41583-020-0277-3

112. Mei J, Muller E, Ramaswamy S. Informing deep neural networks by multiscale principles of neuromodulatory systems. Trends Neurosci. 2022;45: 237–250. doi:10.1016/j.tins.2021.12.008

113. Robertazzi F, Vissani M, Schillaci G, Falotico E. Brain-inspired meta-reinforcement learning cognitive control in conflictual inhibition decision-making task for artificial agents. Neural Netw. 2022;154: 283–302. doi:10.1016/j.neunet.2022.06.020

Supplement



Supplemental Figure 1. Mean final cumulative regret of all rnn architectures in the restless bandit task by different design factors*. X-axis denote whether no computation noise ("None") or weber noise ("Weber") is added to hidden units. Error bars denote SEM.*

Supplemental Table 1. *Model Comparison for the best performing RNN architecture (LSTM network with computational noise)*

Model	Delta Rule	Bayesian Learner Rule

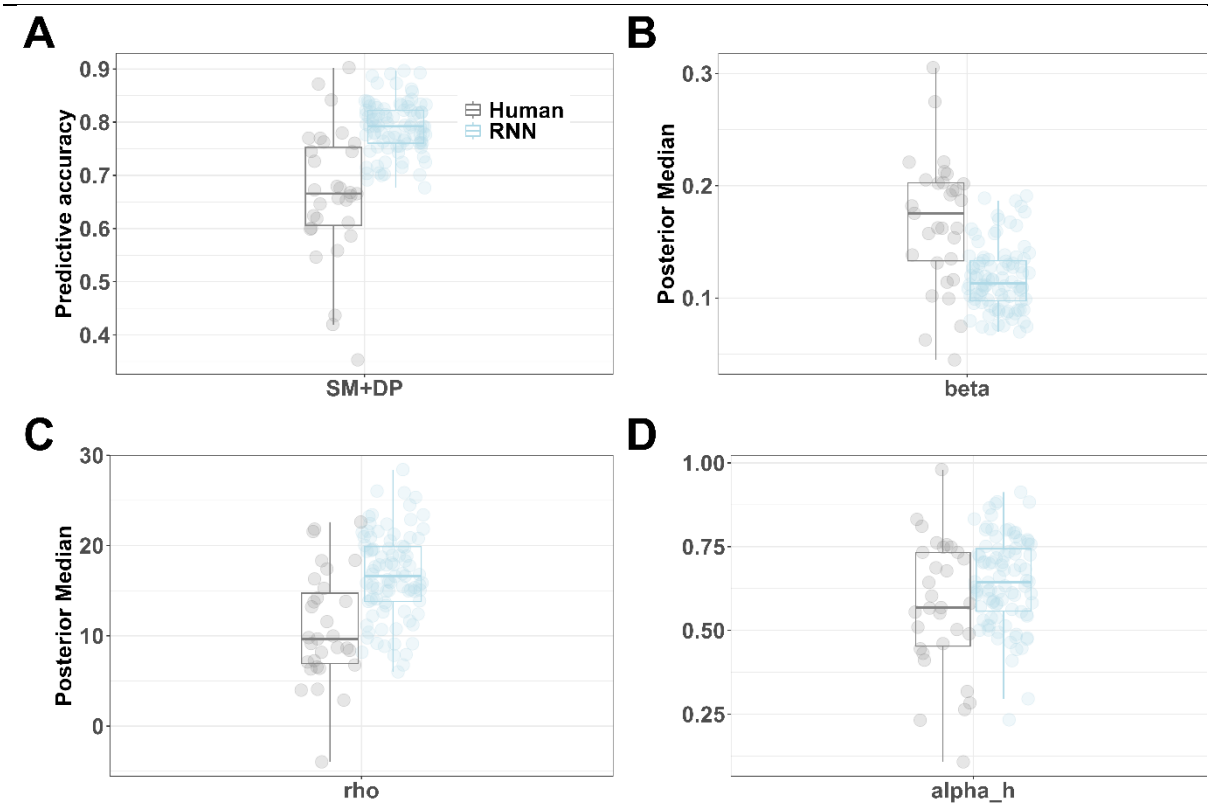
SM	7835.40	7305.10
SM + P	887.77	719.23
SM + BP	906.35	779.40
SM + TP	912.02	711.96
SM + B	2284.64	2126.84
SM + T	5046.36	4761.29
SM + DP	319.12	0 *
SM + BDP	361.78	43.82
SM + TDP	394.04	74.73
SM + E	-	2254.24
SM + EP	-	550.37
SM + EDP	-	49.96

Note: This table shows Delta WAIC values and standard deviations for each cognitive model in the model space (see Table 1) for the RNN data. The cognitive model with the lowest Delta WAIC value shows the best model fit.

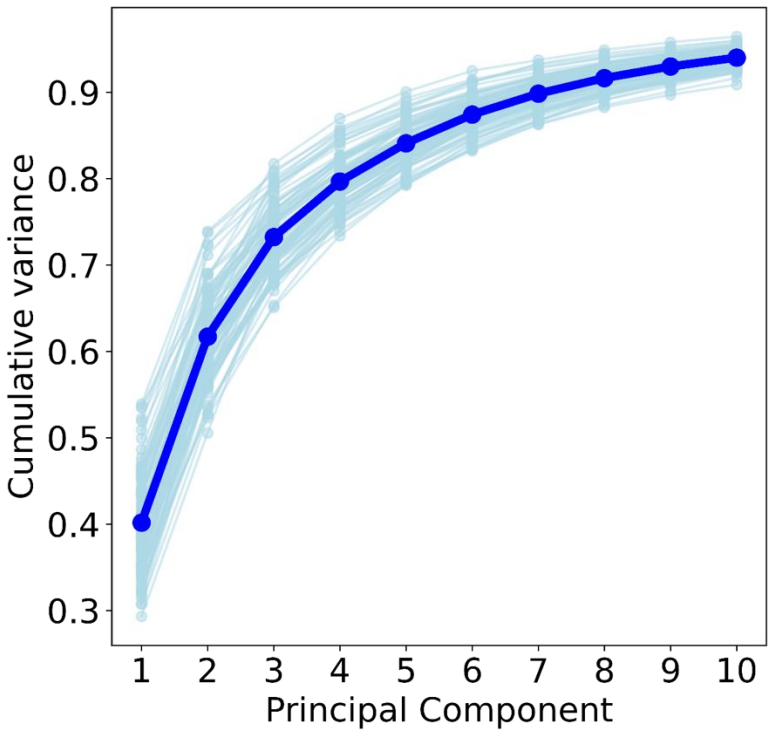
Supplemental Table 2. *Model Comparison for human data*

Model	**Delta Rule**	**Bayesian Learner Rule**
SM	1057.70	967.71
SM + P	708.45	658.87
SM + BP	388.13	333.93
SM + TP	333.39	267.68
SM + B	858.05	767.47
SM + T	762.96	639.96
SM + DP	646.96	575.79
SM + BDP	243.94	230.92
SM + TDP	236.94	196.55
SM + E	-	655.61
SM + EP	-	165.43
SM + EDP	-	0 *

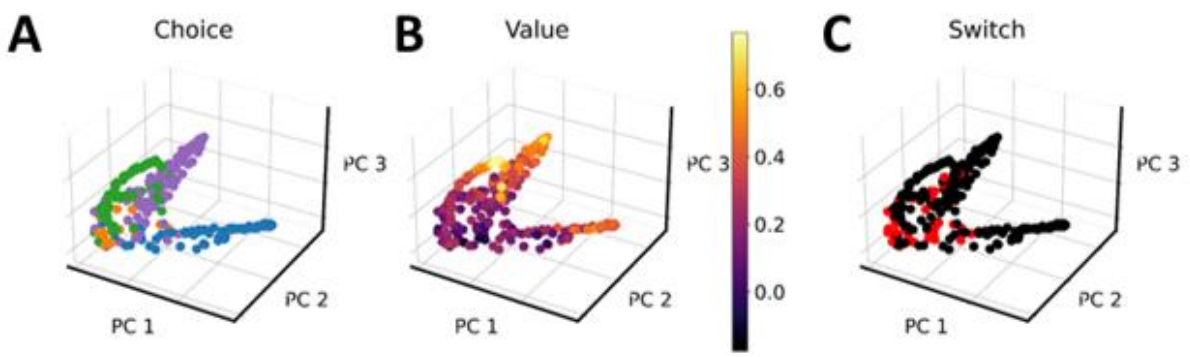
Note: This table shows Delta WAIC values and standard deviations for each cognitive model in the model space (see Table 1) for the RNN data. The cognitive model with the lowest Delta WAIC value shows the best model fit.



Supplemental Figure 2. Median posterior values of model parameters for the best-fitting model for RNN agents (Bayesian learner with higher order perseveration term, SM+DP). *A: Predictive accuracy as the fraction of choices correctly predicted by the cognitive model, B: choice stochasticity parameter β, C: perseveration parameter ρ, D: Habit step size parameter* α_h.

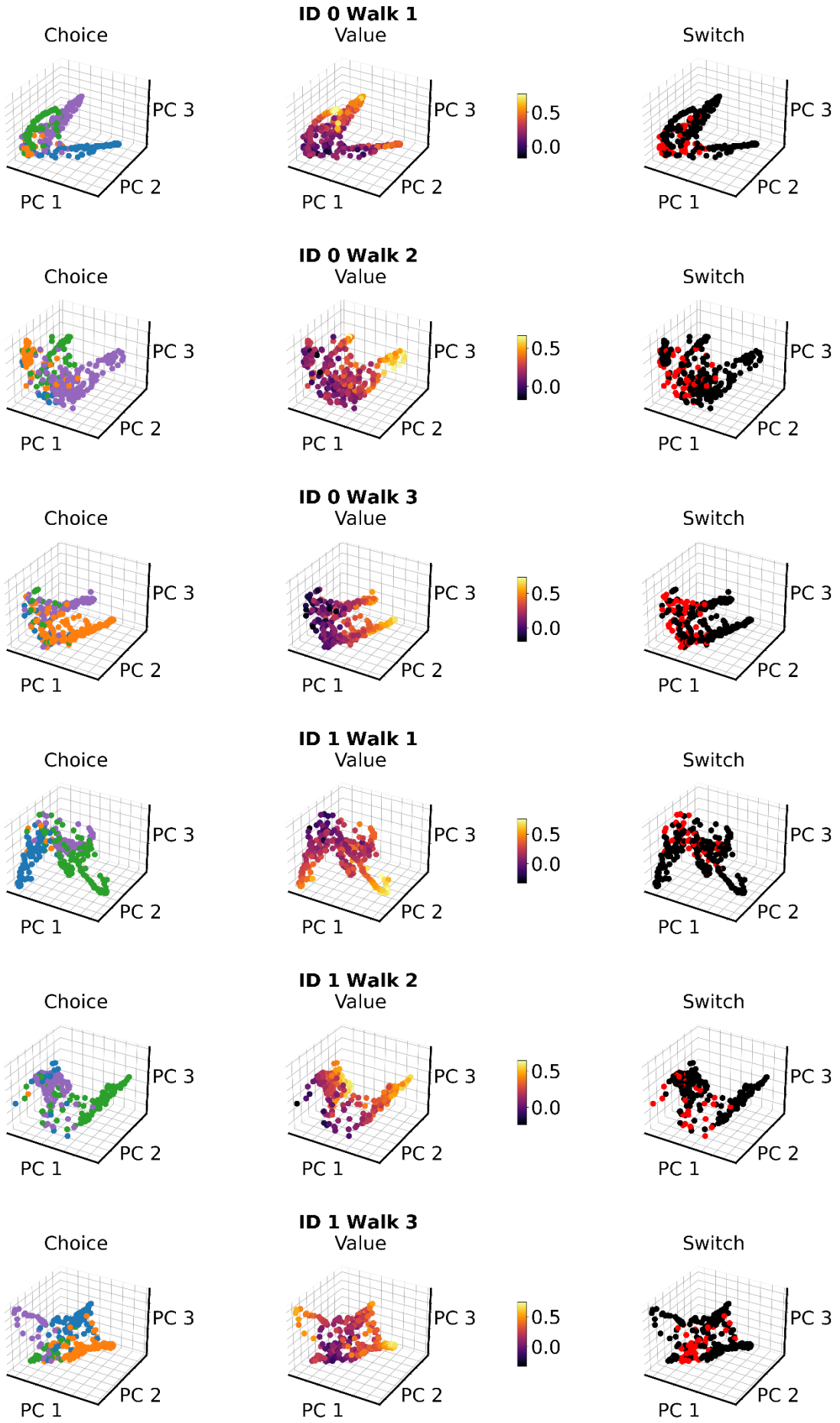


Supplemental Figure 3. Cumulative variance explained in hidden unit activity by principal components of the best RNN architecture (LSTM cell with weber noise, A2C algorithm and no entropy regularization): *Light blue lines denote cumulative variance explained for each RNN instance. Dark blue line denotes mean cumulative variance explained over all RNN instances.*

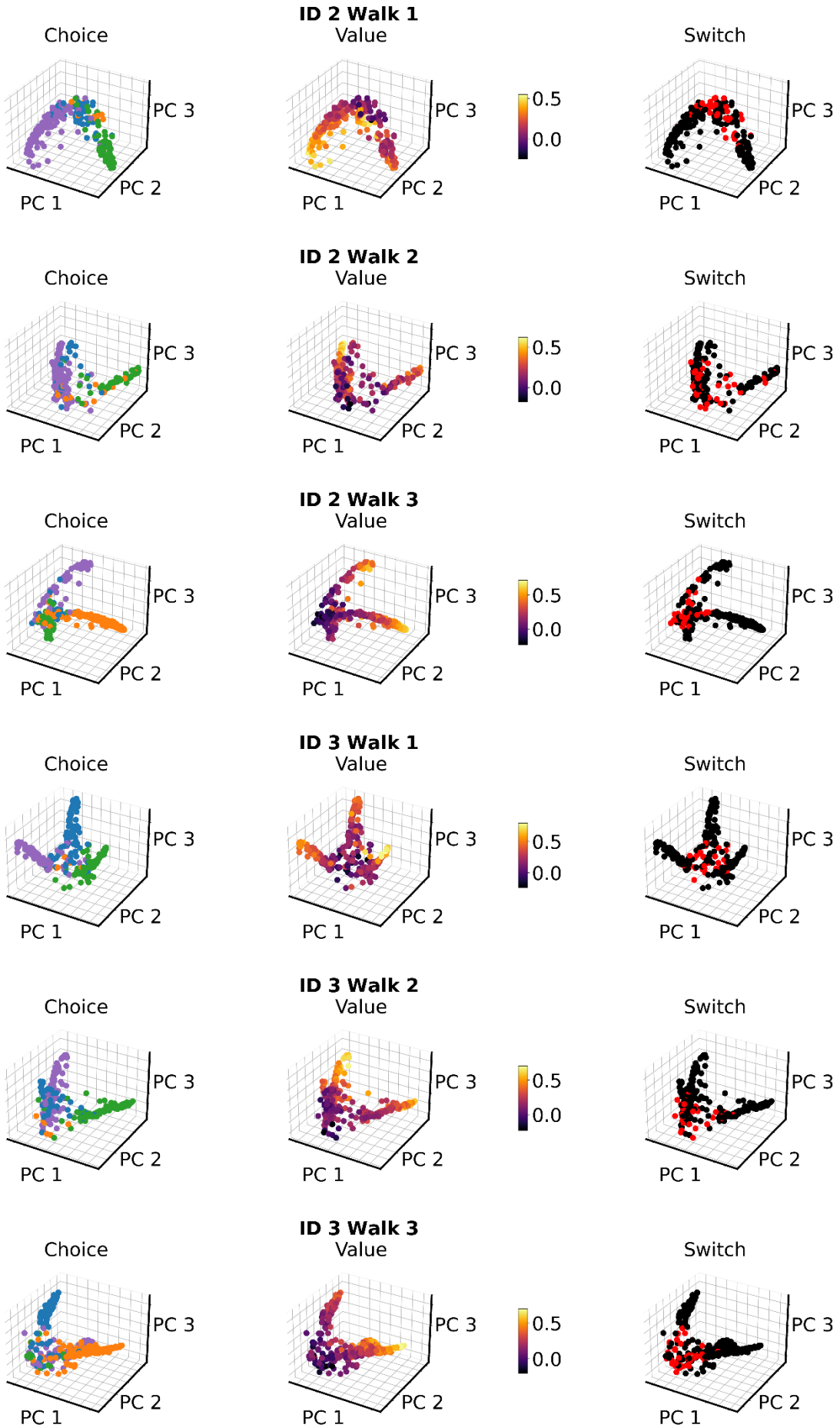


Supplemental Figure 4. *Hidden unit dynamics (first three principal components) of an example RNN agent color coded by choice (A), state-value estimate (B) and switching behavior (C, switch – red, stay - black). The pattern of choice-specific clusters (A), more overlap for actions corresponding to low state-value estimates (B) and switches occuring predominantly in this region of overlap (C) generalizes to all RNN instances under consideration (see below)*

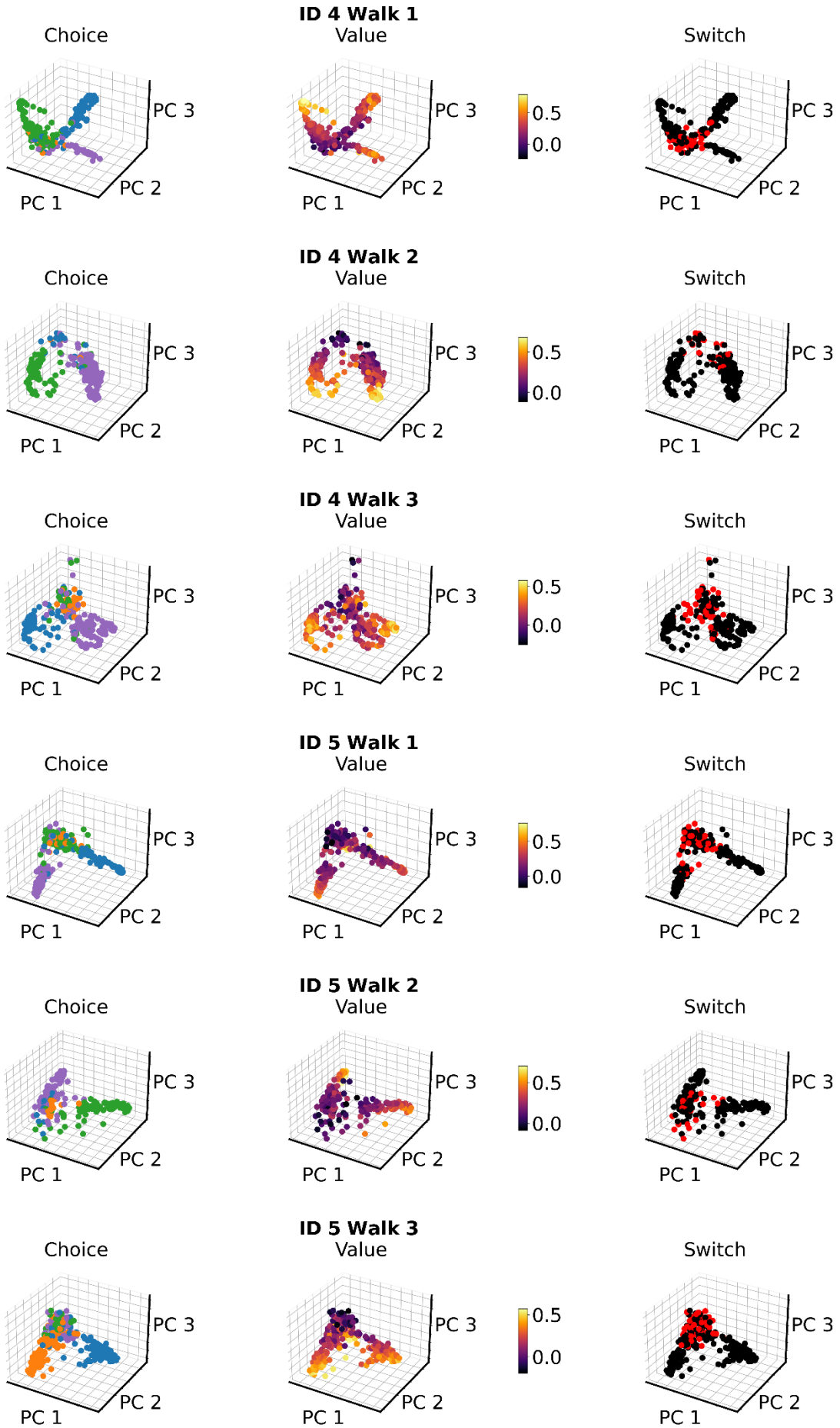
1266


ID 0 Walk 1
Choice
Value
Switch
ID 0 Walk 2
Choice
Value
Switch
ID 0 Walk 3
Choice
Value
Switch
ID 1 Walk 1
Choice
Value
Switch
ID 1 Walk 2
Choice
Value
Switch
ID 1 Walk 3
Choice
Value
Switch
PC 1
PC 2
PC 3
0.5
0.0

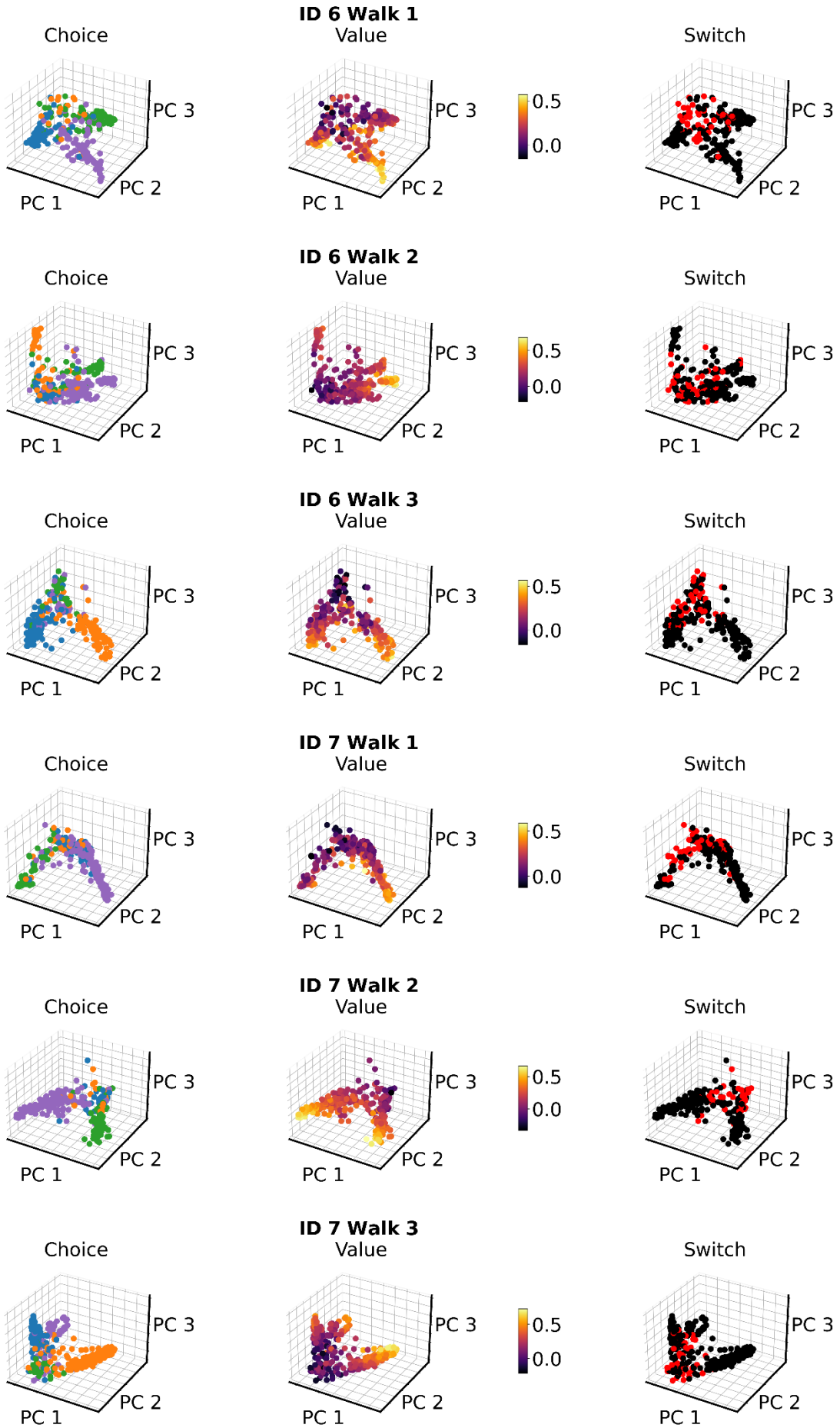
1267

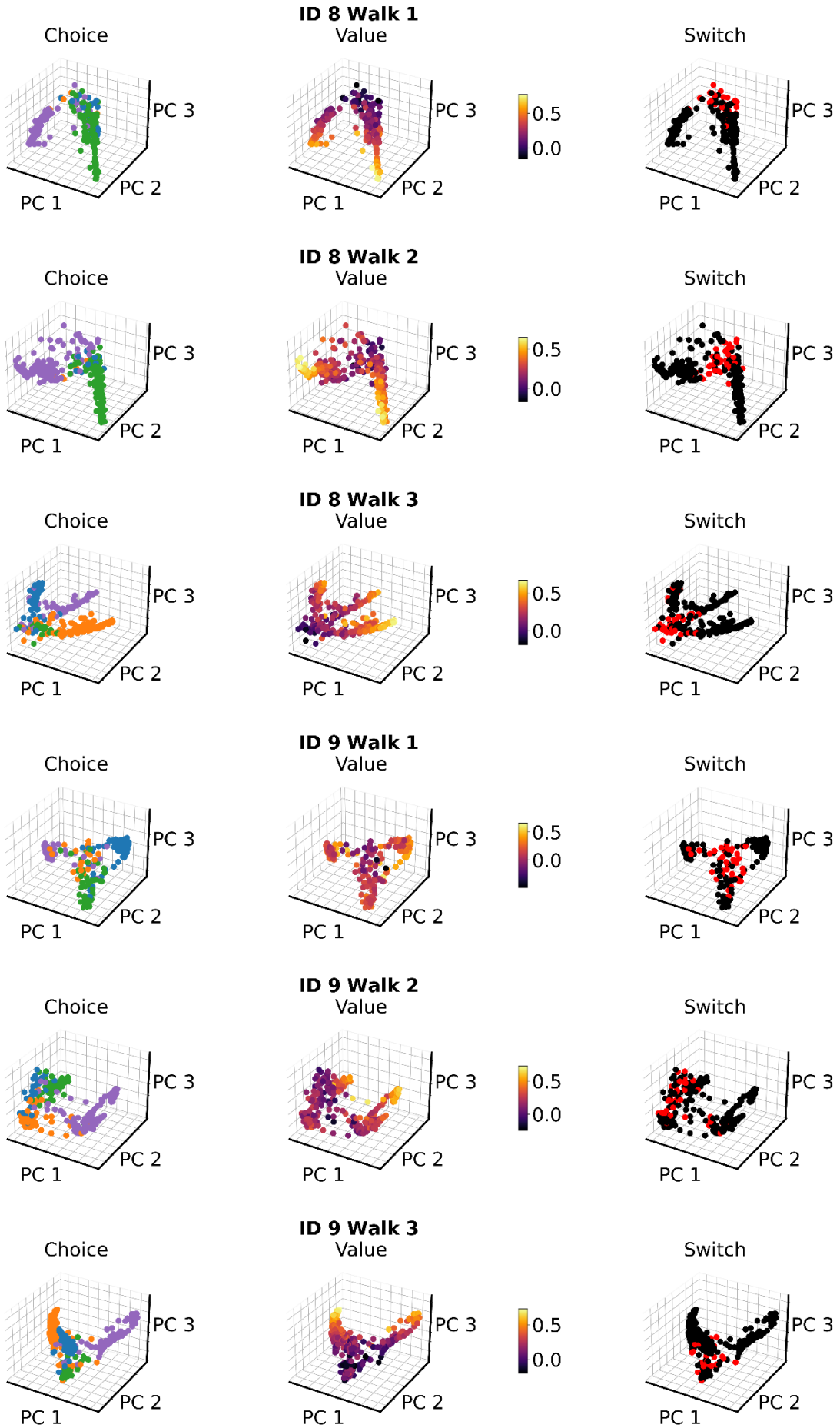

ID 2 Walk 1
Choice
Value
Switch
ID 2 Walk 2
Choice
Value
Switch
ID 2 Walk 3
Choice
Value
Switch
ID 3 Walk 1
Choice
Value
Switch
ID 3 Walk 2
Choice
Value
Switch
ID 3 Walk 3
Choice
Value
Switch
PC 1
PC 2
PC 3
0.5
0.0

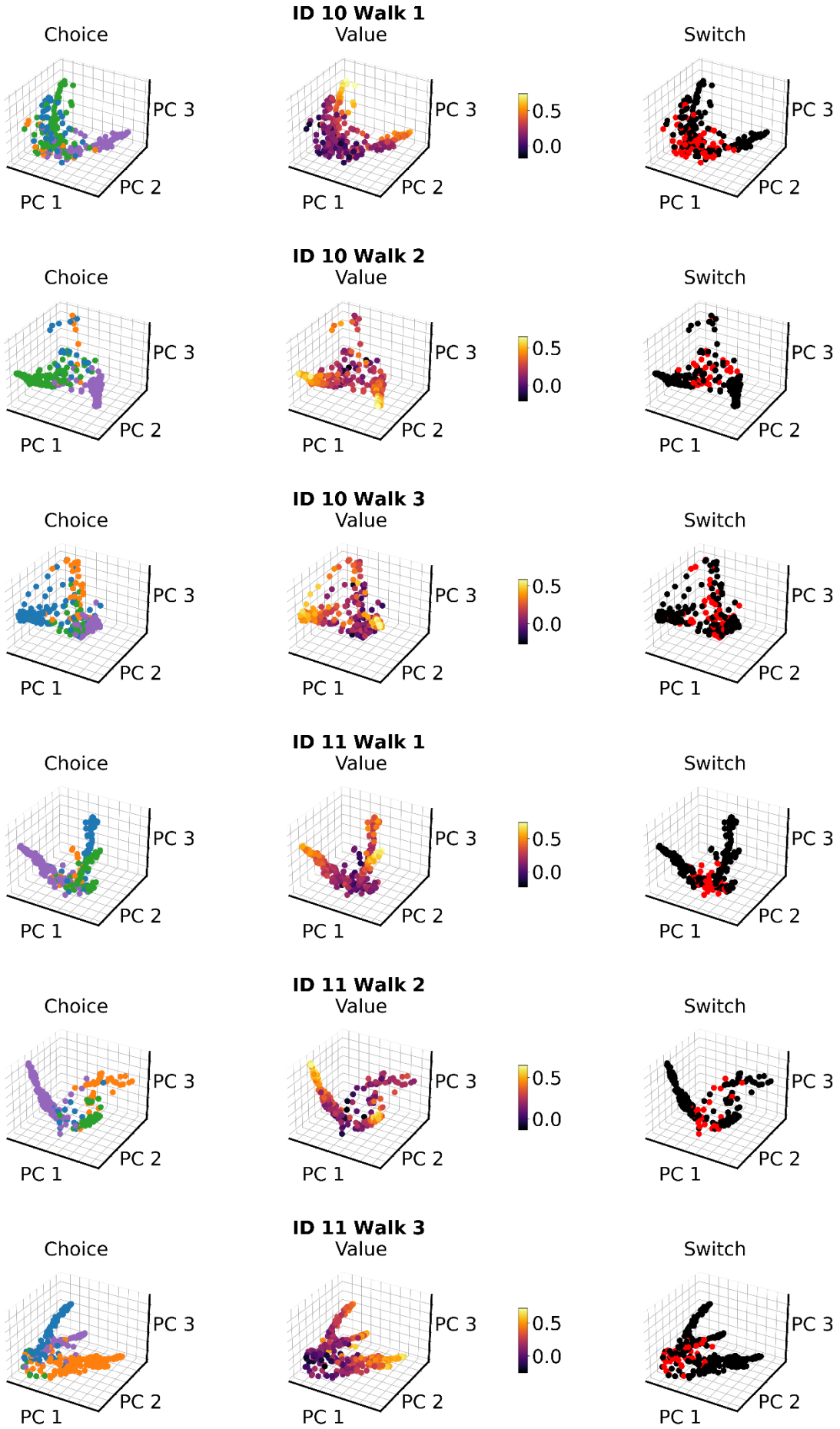
1268


ID 4 Walk 1
Choice
Value
Switch
ID 4 Walk 2
Choice
Value
Switch
ID 4 Walk 3
Choice
Value
Switch
ID 5 Walk 1
Choice
Value
Switch
ID 5 Walk 2
Choice
Value
Switch
ID 5 Walk 3
Choice
Value
Switch
PC 1
PC 2
PC 3
0.5
0.0

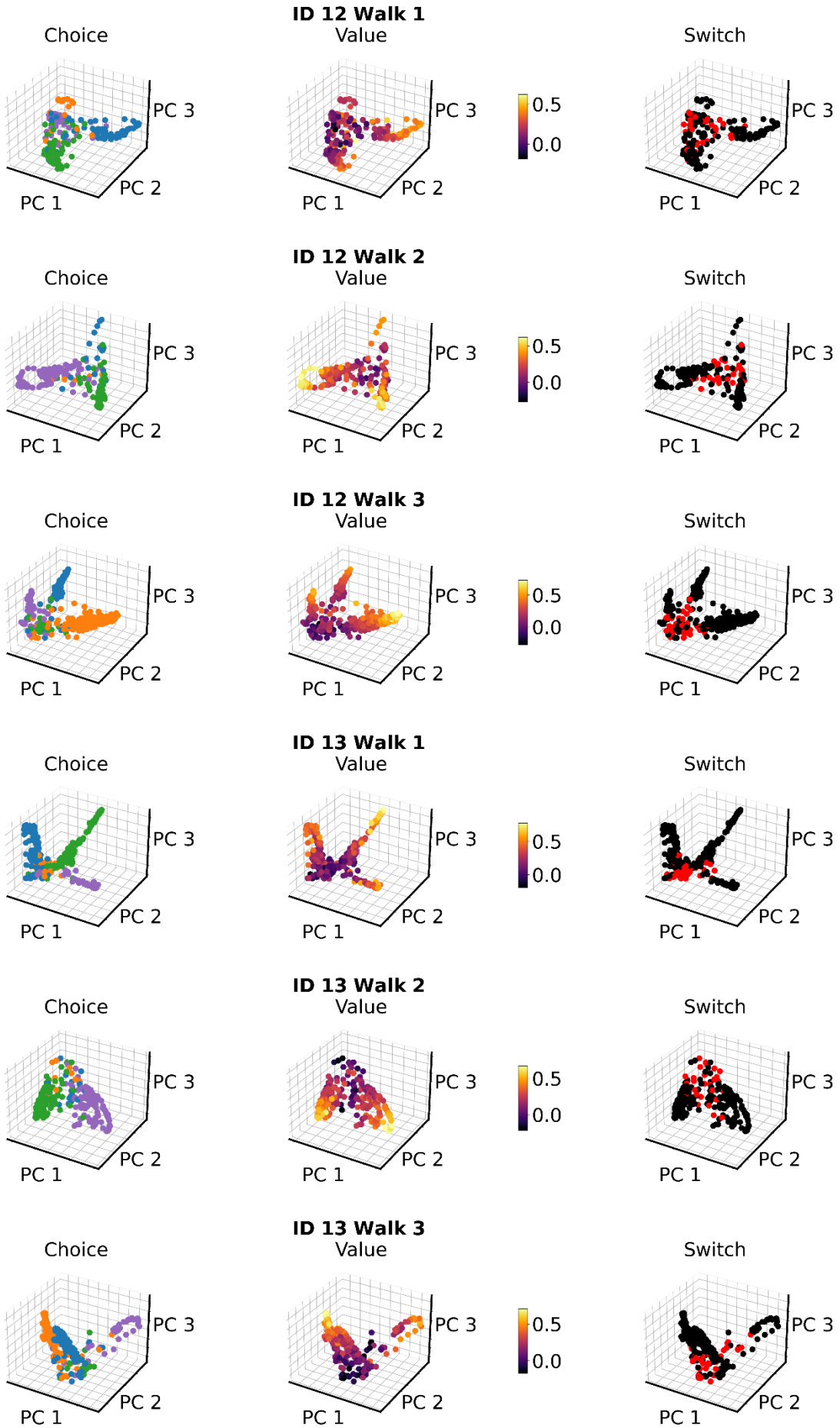
1269


ID 6 Walk 1
Choice
Value
Switch
ID 6 Walk 2
Choice
Value
Switch
ID 6 Walk 3
Choice
Value
Switch
ID 7 Walk 1
Choice
Value
Switch
ID 7 Walk 2
Choice
Value
Switch
ID 7 Walk 3
Choice
Value
Switch
PC 1
PC 2
PC 3
0.5
0.0

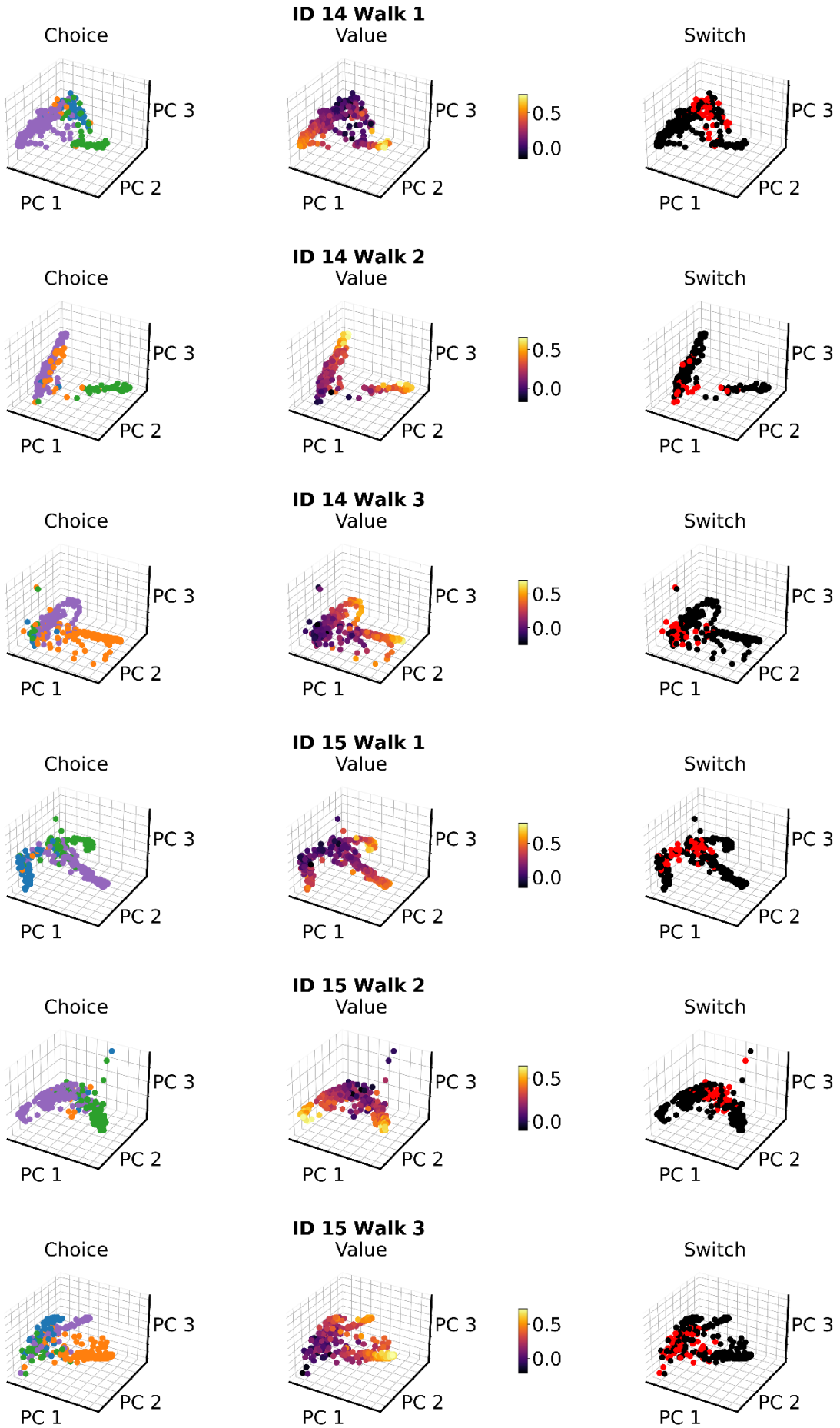

ID 8 Walk 1
Choice
Value
Switch
ID 8 Walk 2
Choice
Value
Switch
ID 8 Walk 3
Choice
Value
Switch
ID 9 Walk 1
Choice
Value
Switch
ID 9 Walk 2
Choice
Value
Switch
ID 9 Walk 3
Choice
Value
Switch
PC 1
PC 2
PC 3
0.5
0.0


ID 10 Walk 1
Choice
Value
Switch
ID 10 Walk 2
Choice
Value
Switch
ID 10 Walk 3
Choice
Value
Switch
ID 11 Walk 1
Choice
Value
Switch
ID 11 Walk 2
Choice
Value
Switch
ID 11 Walk 3
Choice
Value
Switch
PC 1
PC 2
PC 3
0.5
0.0

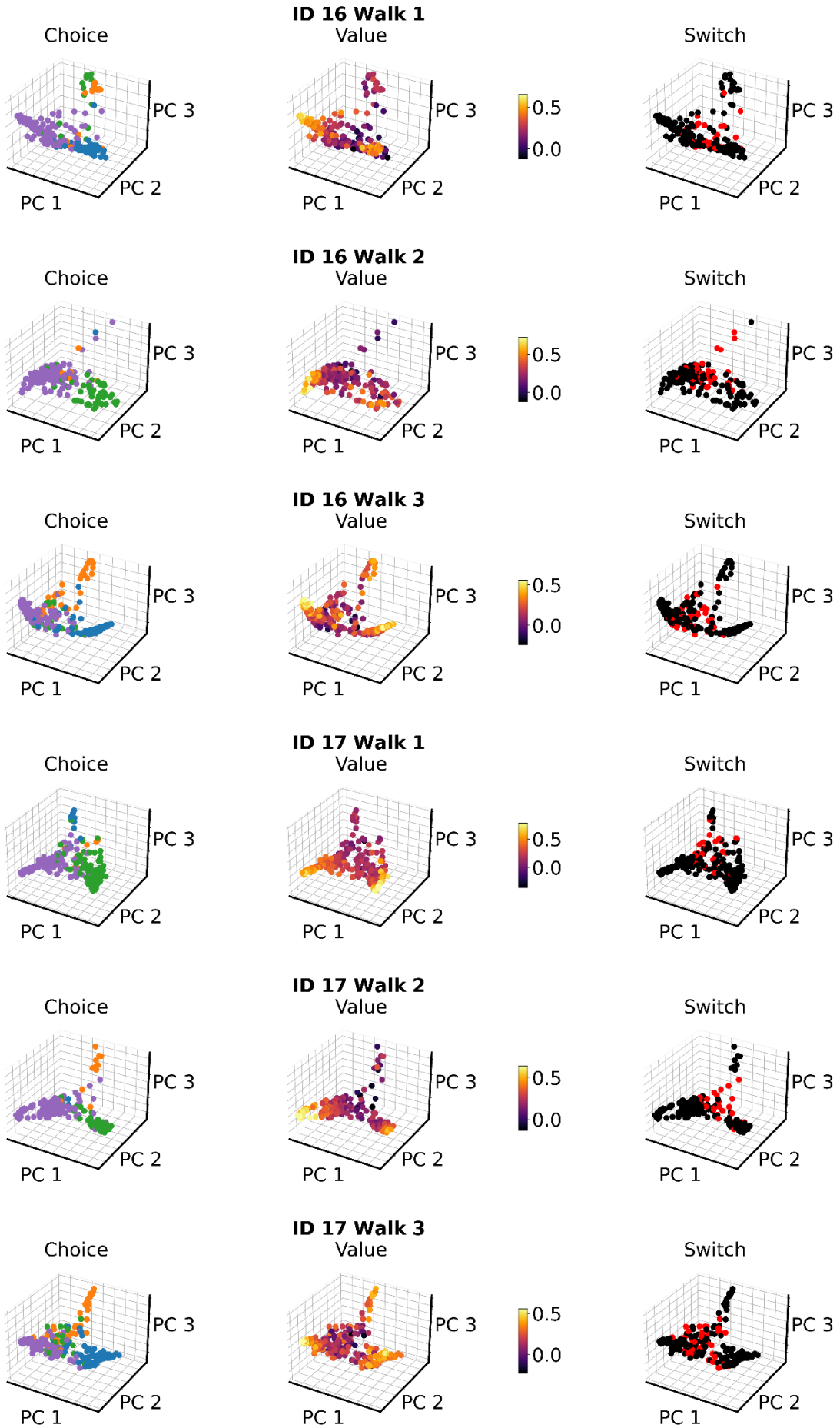
1272

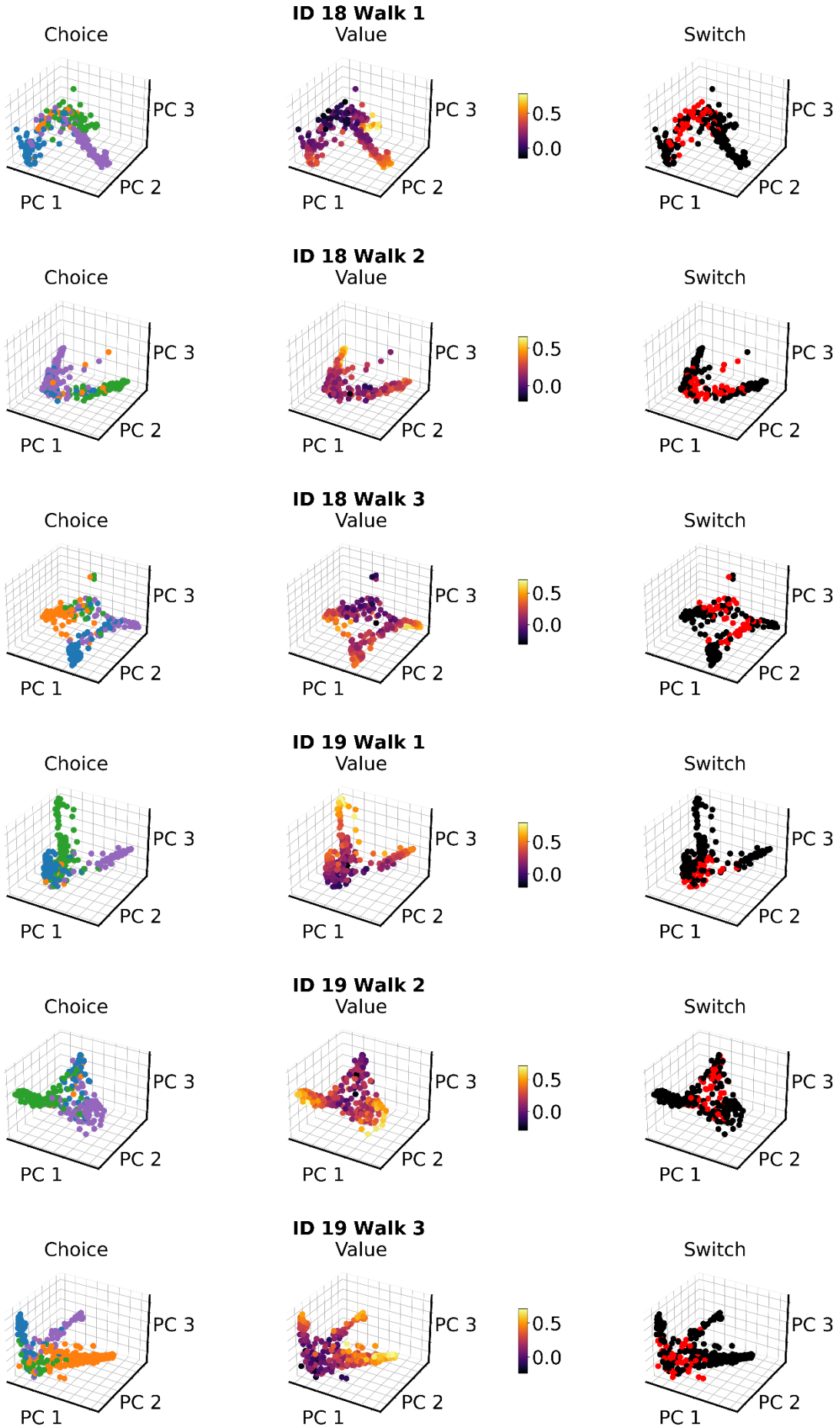

ID 12 Walk 1
Choice
Value
Switch
ID 12 Walk 2
Choice
Value
Switch
ID 12 Walk 3
Choice
Value
Switch
ID 13 Walk 1
Choice
Value
Switch
ID 13 Walk 2
Choice
Value
Switch
ID 13 Walk 3
Choice
Value
Switch
PC 1
PC 2
PC 3
0.5
0.0

1273

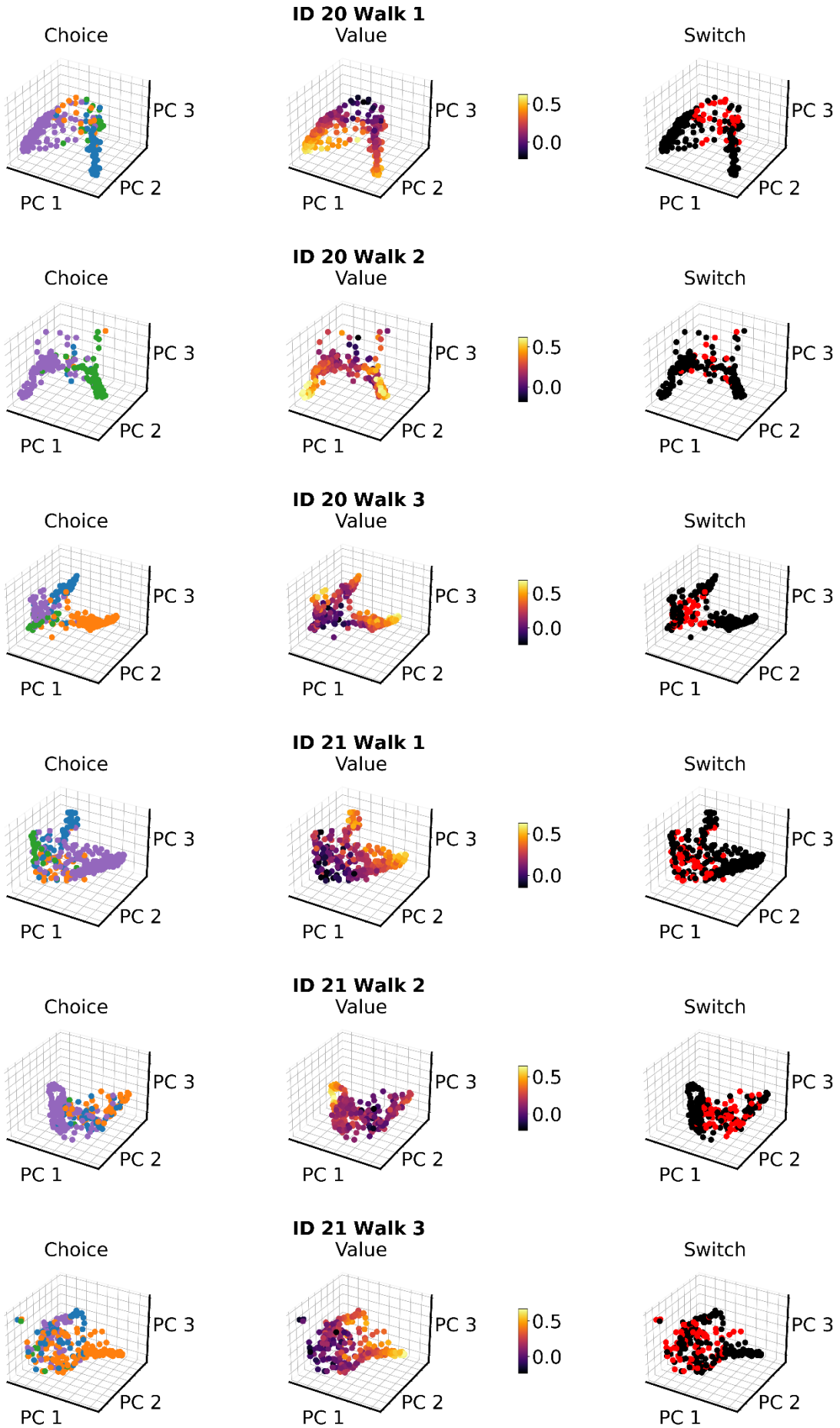

ID 14 Walk 1
Choice
Value
Switch
ID 14 Walk 2
Choice
Value
Switch
ID 14 Walk 3
Choice
Value
Switch
ID 15 Walk 1
Choice
Value
Switch
ID 15 Walk 2
Choice
Value
Switch
ID 15 Walk 3
Choice
Value
Switch
PC 1
PC 2
PC 3
0.5
0.0

1274

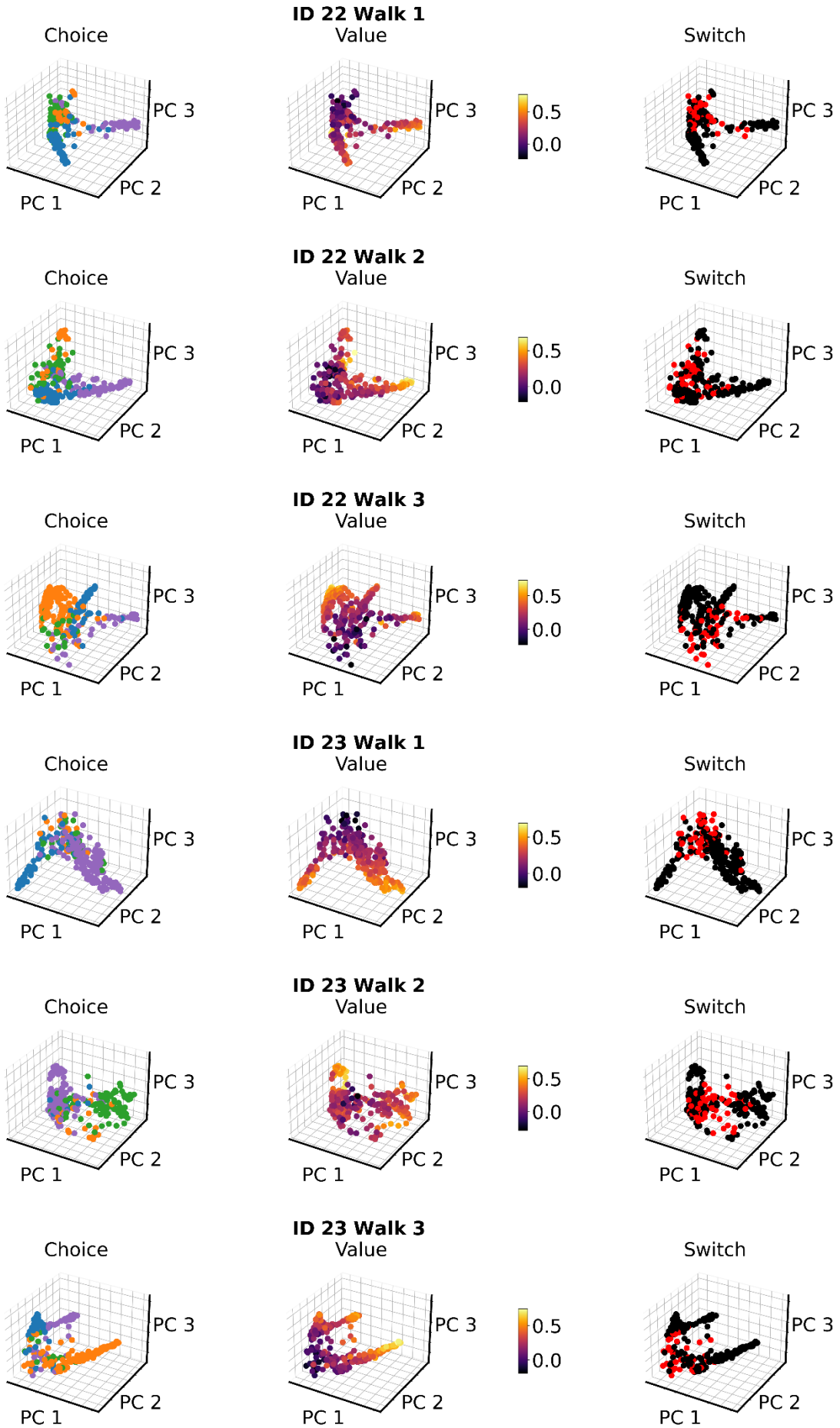

ID 16 Walk 1
Choice
Value
Switch
ID 16 Walk 2
Choice
Value
Switch
ID 16 Walk 3
Choice
Value
Switch
ID 17 Walk 1
Choice
Value
Switch
ID 17 Walk 2
Choice
Value
Switch
ID 17 Walk 3
Choice
Value
Switch
PC 1
PC 2
PC 3
0.5
0.0


ID 18 Walk 1
Choice
Value
Switch
ID 18 Walk 2
Choice
Value
Switch
ID 18 Walk 3
Choice
Value
Switch
ID 19 Walk 1
Choice
Value
Switch
ID 19 Walk 2
Choice
Value
Switch
ID 19 Walk 3
Choice
Value
Switch
PC 1
PC 2
PC 3
0.5
0.0

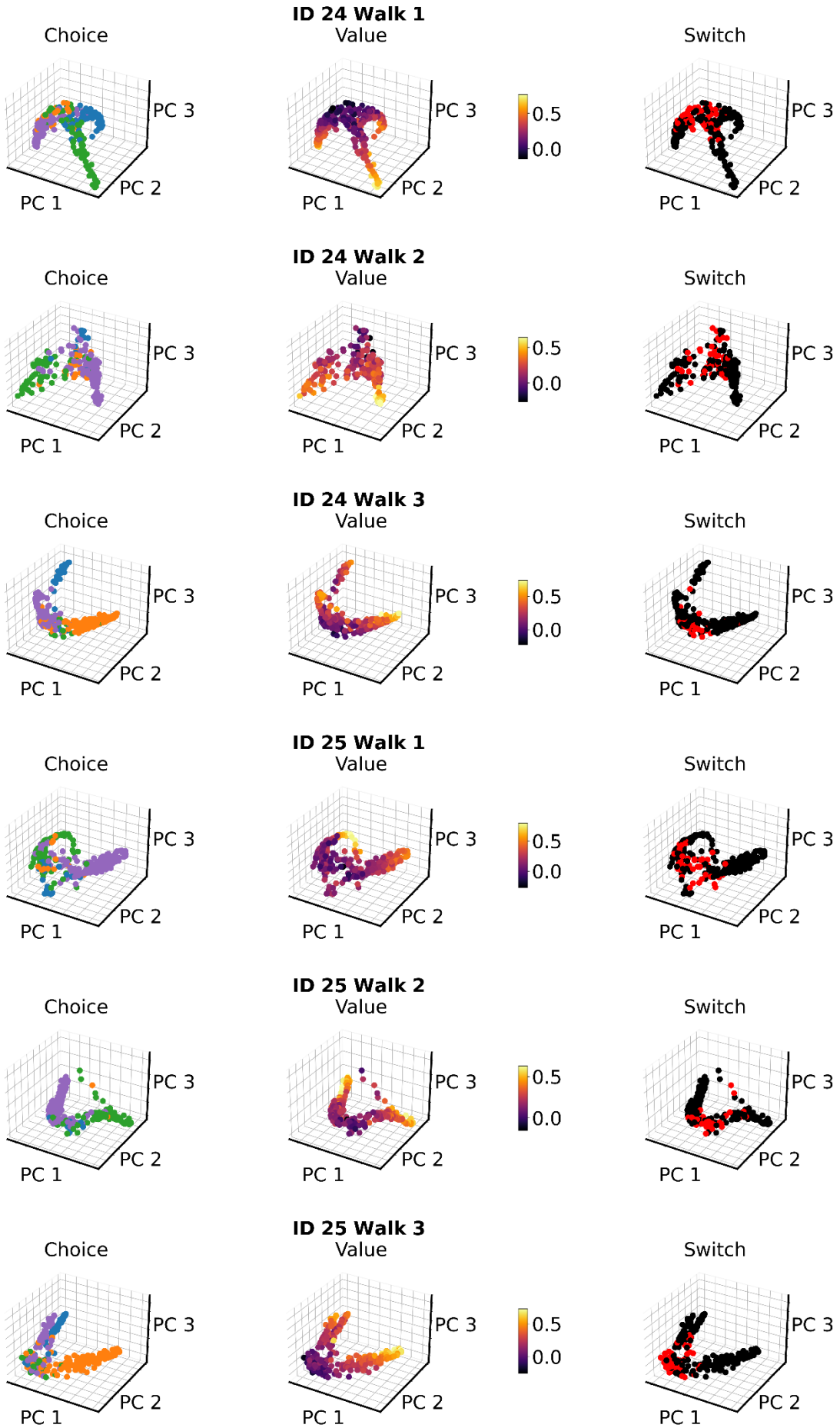
1276


ID 20 Walk 1
Choice
Value
Switch
ID 20 Walk 2
Choice
Value
Switch
ID 20 Walk 3
Choice
Value
Switch
ID 21 Walk 1
Choice
Value
Switch
ID 21 Walk 2
Choice
Value
Switch
ID 21 Walk 3
Choice
Value
Switch
PC 1
PC 2
PC 3
0.5
0.0

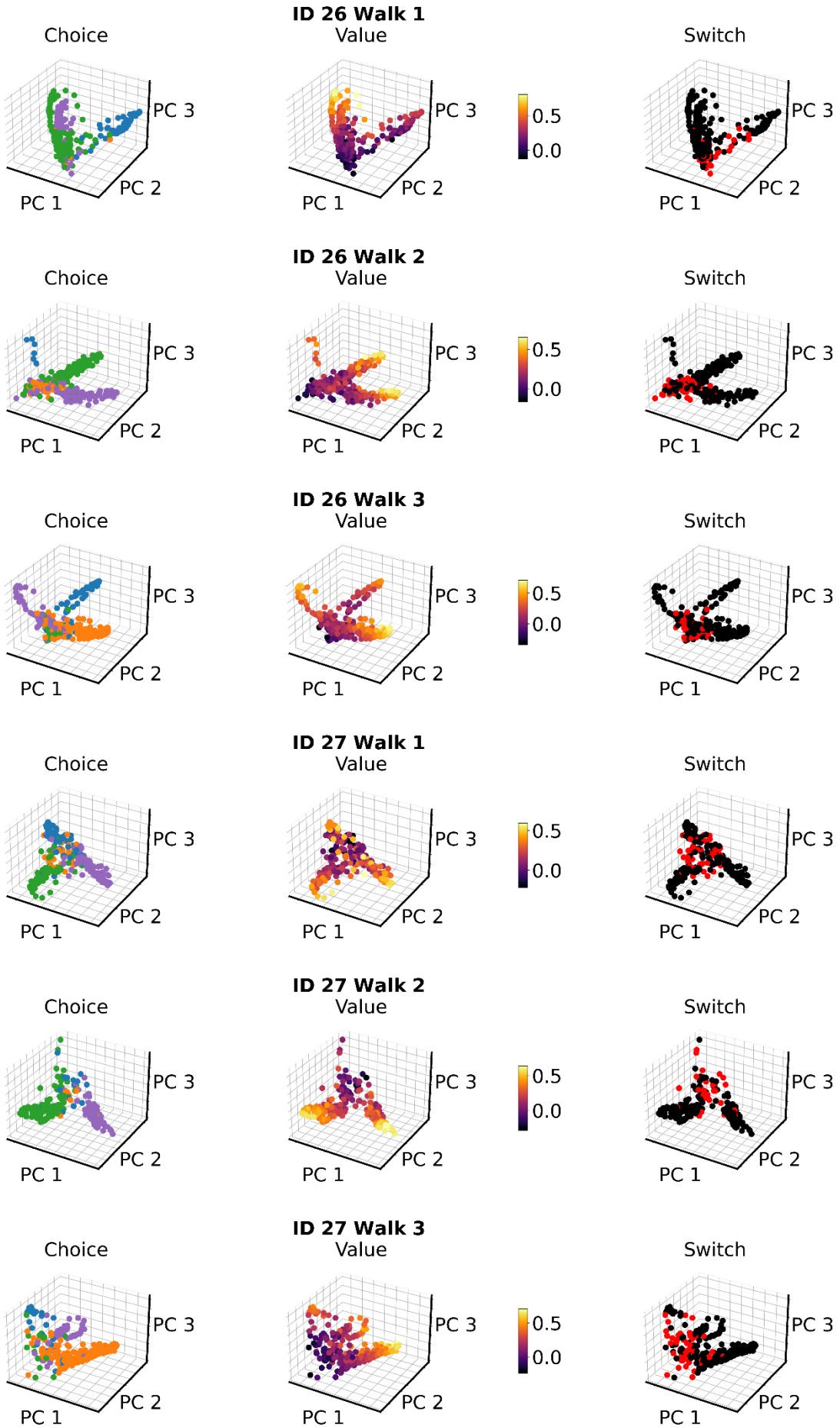
1277

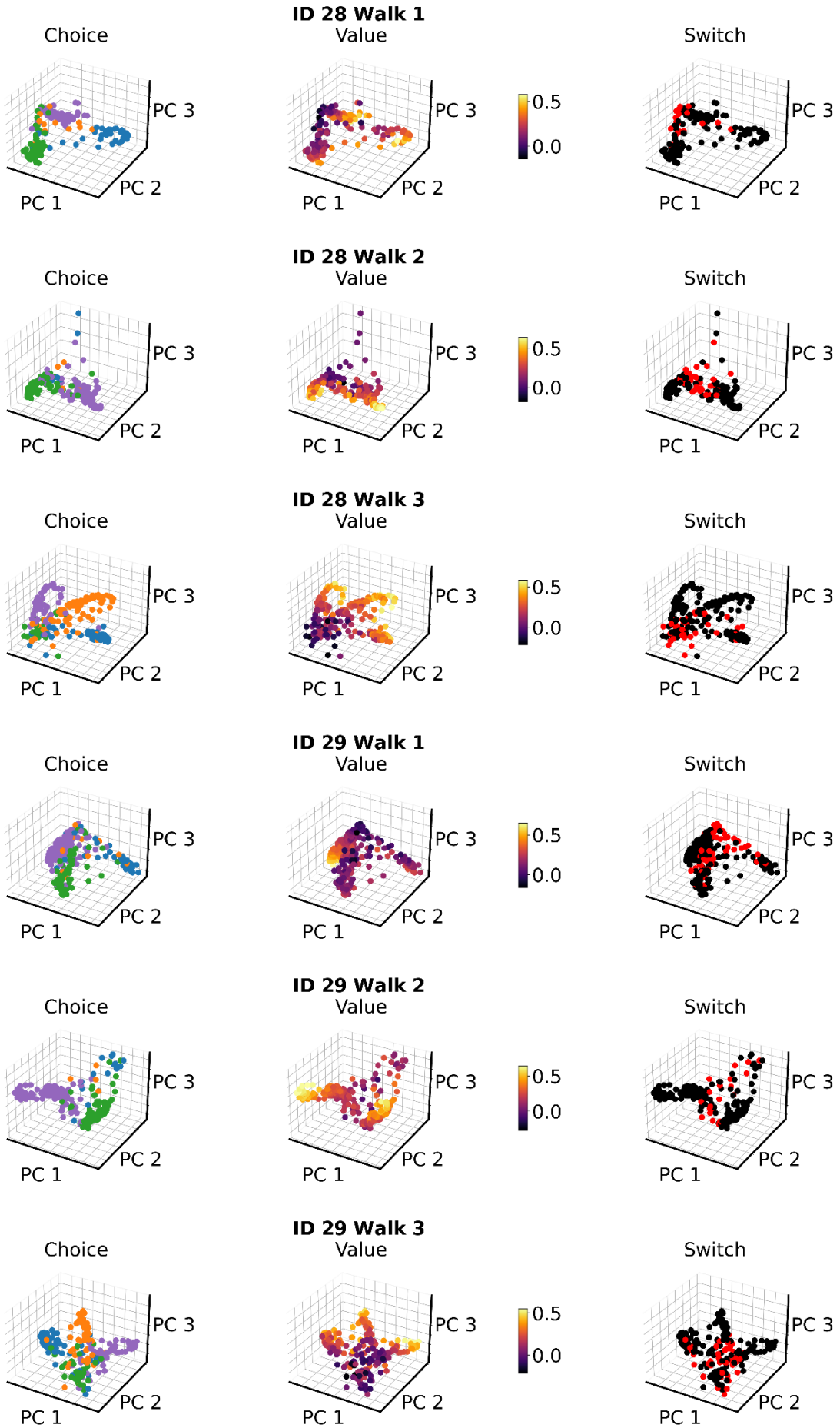

ID 22 Walk 1
Choice
Value
Switch
ID 22 Walk 2
Choice
Value
Switch
ID 22 Walk 3
Choice
Value
Switch
ID 23 Walk 1
Choice
Value
Switch
ID 23 Walk 2
Choice
Value
Switch
ID 23 Walk 3
Choice
Value
Switch
PC 1
PC 2
PC 3
0.5
0.0

1278


ID 24 Walk 1
Choice
Value
Switch
ID 24 Walk 2
Choice
Value
Switch
ID 24 Walk 3
Choice
Value
Switch
ID 25 Walk 1
Choice
Value
Switch
ID 25 Walk 2
Choice
Value
Switch
ID 25 Walk 3
Choice
Value
Switch
PC 1
PC 2
PC 3
0.5
0.0

1279


ID 26 Walk 1
Choice
Value
Switch
ID 26 Walk 2
Choice
Value
Switch
ID 26 Walk 3
Choice
Value
Switch
ID 27 Walk 1
Choice
Value
Switch
ID 27 Walk 2
Choice
Value
Switch
ID 27 Walk 3
Choice
Value
Switch
PC 1
PC 2
PC 3
0.5
0.0


ID 28 Walk 1
Choice
Value
Switch
ID 28 Walk 2
Choice
Value
Switch
ID 28 Walk 3
Choice
Value
Switch
ID 29 Walk 1
Choice
Value
Switch
ID 29 Walk 2
Choice
Value
Switch
ID 29 Walk 3
Choice
Value
Switch
PC 1
PC 2
PC 3
0.5
0.0