A neural network model for timing control with reinforcement

Jing Wang, Yousuf El-Jayyousi and Ilker Ozden*

Department of Biomedical, Biological, and Chemical Engineering, University of Missouri, Columbia, MO, United States

How do humans and animals perform trial-and-error learning when the space of possibilities is infinite? In a previous study, we used an interval timing production task and discovered an updating strategy in which the agent adjusted the behavioral and neuronal noise for exploration. In the experiment, human subjects proactively generated a series of timed motor outputs. Positive or negative feedback was provided after each response based on the timing accuracy. We found that the sequential motor timing varied at two temporal scales: long-term correlation around the target interval due to memory drifts and short-term adjustments of timing variability according to feedback. We have previously described these two key features of timing variability with an augmented Gaussian process, termed reward-sensitive Gaussian process (RSGP). In a nutshell, the temporal covariance of the timing variable was updated based on the feedback history to recreate the two behavioral characteristics mentioned above. However, the RSGP was mainly descriptive and lacked a neurobiological basis of how the reward feedback can be used by a neural circuit to adjust motor variability. Here we provide a mechanistic model and simulate the process by borrowing the architecture of recurrent neural networks (RNNs). While recurrent connection provided the long-term serial correlation in motor timing, to facilitate reward-driven short-term variations, we introduced reward-dependent variability in the network connectivity, inspired by the stochastic nature of synaptic transmission in the brain. Our model was able to recursively generate an output sequence incorporating internal variability and external reinforcement in a Bayesian framework. We show that the model can generate the temporal structure of the motor variability as a basis for exploration and exploitation trade-off. Unlike other neural network models that search for unique network connectivity for the best match between the model prediction and observation, this model can estimate the uncertainty associated with each outcome and thus did a better job in teasing apart adjustable task-relevant variability from

unexplained variability. The proposed artificial neural network model parallels the mechanisms of information processing in neural systems and can extend the framework of brain-inspired reinforcement learning (RL) in continuous state control.

Introduction

Reinforcement learning (RL) is situated at the growing intersection between machine learning and learning rules in biological systems. RL theory has been highly influential in explaining the rules an agent applies in interacting with the environment. In the conventional RL paradigm, a discrete set of choices are presented. The agent learns the values associated with individual options and reactions in order to optimize control policies and thus maximize the reward outcome (Sutton and Barto, 1998). However, we often face choices of continuous nature, such as deciding the serving angle of a tennis ball. In such cases, it is impossible to learn the entire span of value functions with a finite number of attempts (van Hasselt and Wiering, 2007; van Hasselt, 2012; Dhawale et al., 2017). A variety of learning algorithms have been proposed to address this problem (Gullapalli, 1990; Benbrahim and Franklin, 1997; van Hasselt, 2012; Wu et al., 2014; Dhawale et al., 2017). These algorithms mainly focus on constructing models for stochastic exploration of the reward landscape in the space of model parameters.

Our previous study (Wang et al., 2020) described an updating strategy in which the agent can actively explore the space of choices by adjusting their behavioral noise (motor variability). In the experiment, test subjects proactively generated a series of internally timed motor outputs by adjusting their timing accuracy trial-by-trial in response to positive or negative feedback. We found that the time series consisting of humans' sequential motor responses varied at a slow and a fast temporal scale: the long-term serial correlations due to memory drifts and a short-term adjustment of noise due to feedback. Subsequent experiments verified the causal influence of binary feedback on humans' motor variability. We previously described the observations with an augmented Gaussian process, termed reward sensitive Gaussian process (RSGP). The RSGP is a statistical description that could produce the time varying temporal correlations and stochasticity in the data by a reward modulated Gaussian process. The covariance matrix included a static component representing the long-term correlations and a dynamic component updated according to the reward feedback. However, the RSGP, like the previous learning algorithms mentioned above, lacked a neurobiological basis of how the

reward might be used by neural circuits to adjust motor variability. The goal of this study is to incorporate some of the recent findings and ideas about learning in high-dimensional spaces (Leblois, 2013; Dhawale et al., 2019; Gershman and Ölveczky, 2020) in the framework of RL to explain the main features of behavioral time series in our study. With our model, we aim to provide a biologically plausible mechanistic model that can make testable predictions about behavioral variability. Accordingly, our model adapts the architecture of recurrent units in recurrent neural networks (RNNs). We modified the unit architecture in order to incorporate the reinforcement. In the recurrent unit, we introduced adjustable noise to the connectivity between units and linked the noise level in connectivity to the reward feedback. This architecture is consistent with the probabilistic nature of the synaptic transmission in the brain (Llera-Montero et al., 2019), and hypotheses that stochastic synapses might contribute to learning (Seung, 2003; Gershman and Ölveczky, 2020). Relevantly, the modulation of motor variability by rewarding events or performance outcomes has also been reported in other studies (Leblois, 2013; Dhawale et al., 2019; Gershman and Ölveczky, 2020), without direct insight on how motor variability can arise in a neural network. With our model, we show that reward-modulated synaptic noise can account for the observed motor variability during learning. We call our model a weight-varying autoregressive NN (vARNN). Unlike standard machine learning models that search for a unique set of connectivity, our model assumes that each connectivity strength is sampled from a distribution with fixed mean and adaptive variance due to reward modulation. This way, our model was able to generate a sequence of stochastic timing outputs that matched human behavior by integrating an external reinforcement signal with its internal variability.

In constructing our model, we aimed at an interpretable (biologically realistic and mechanistic) and simple architecture. However, our model does not describe the neural activity at the level of population or single neuron. Rather, it demonstrates in an RNN architecture that the reward-based modulation of synaptic communication can be used for generating behavioral variability. As recent literature suggests the neural circuits involved in RL, such as the striatum, the cortex, and the

dopaminergic pathways, have the infrastructure to support such a mechanism (Dhawale et al., 2019; Gershman and Ölveczky, 2020). Below, we show that (1) the time series output of the vARNN can capture both temporal dependencies and reward sensitivity of the human behavior; (2) this is a generative model where the parameters can be learned and sequential stochastic behavior can be reconstructed; and (3) by considering the uncertainty associated with the model prediction, the vARNN overall did a better job compared to alternative models in teasing apart explainable/adjustable noise from others. We will demonstrate the last point by comparing the performance of our model to two common generic neural network models used for modeling time series. These alternative network models, namely autoregressive neural network (ARNN) and the gated recurrent unit (GRU) models, are discussed in detail below.

Related work and our contribution

RNNs are designed to handle the sequential dependencies in data with temporal structure. RNN architectures with recurrent units such as Long Short Term Memory (LSTM) cells are widely used in value forecasting and natural language processing. They are powerful tools for making predictions based on historical data. In real life, the evolution of a time series is always influenced by multiple sources, and different sources can widely vary in how they affect the outcome. In neural network models, we are hoping that any input-output relationship can be learned with a sufficiently complex network structure and a large amount of training data. Here, we present a case and argue that if the aim is an interpretable and mechanistic model, we should take into account the nature of the information before applying any neural network model.

In modeling the time series dynamics with long-term dependencies as in our motor timing task, a vanilla ARNN has been proposed as an efficient method (Triebe et al., 2019). However, ARNN assumes a deterministic mapping from previous values (inputs) to the predicted value (output) (Triebe et al., 2019). The GRU model, a concise version of LSTM, is another alternative for learning long-term temporal dependencies in the time series data. GRU can accommodate the reward feedback as part of the inputs, however, it is not instructed on how the past affects future observations. The error between the model prediction and actual observation is often attributed to the system noise that is identical among all samples and independent of the underlying process. In contrast to the ARNN and GRU, a weight varying autoregressive NN (vARNN) models the distribution of network parameters and uses feedback to adjust the weights of respective inputs. This is in line with the Bayesian net which takes into account the stochastic nature of the observations in updating the distribution of the network weights.

In vARNN, we intended to bridge the gap between the learning driven by scalar feedback (amount of reward or binary reward) and learning driven by gradients (regressive process). Even though we did not use complex structures such as multiple layers in deep networks, this model was able to learn the key features of human behavior. It can also be readily adapted to other processes that take scalar feedback into consideration. Moreover, this model extends the framework of RL to continuous state systems.

Materials and methods
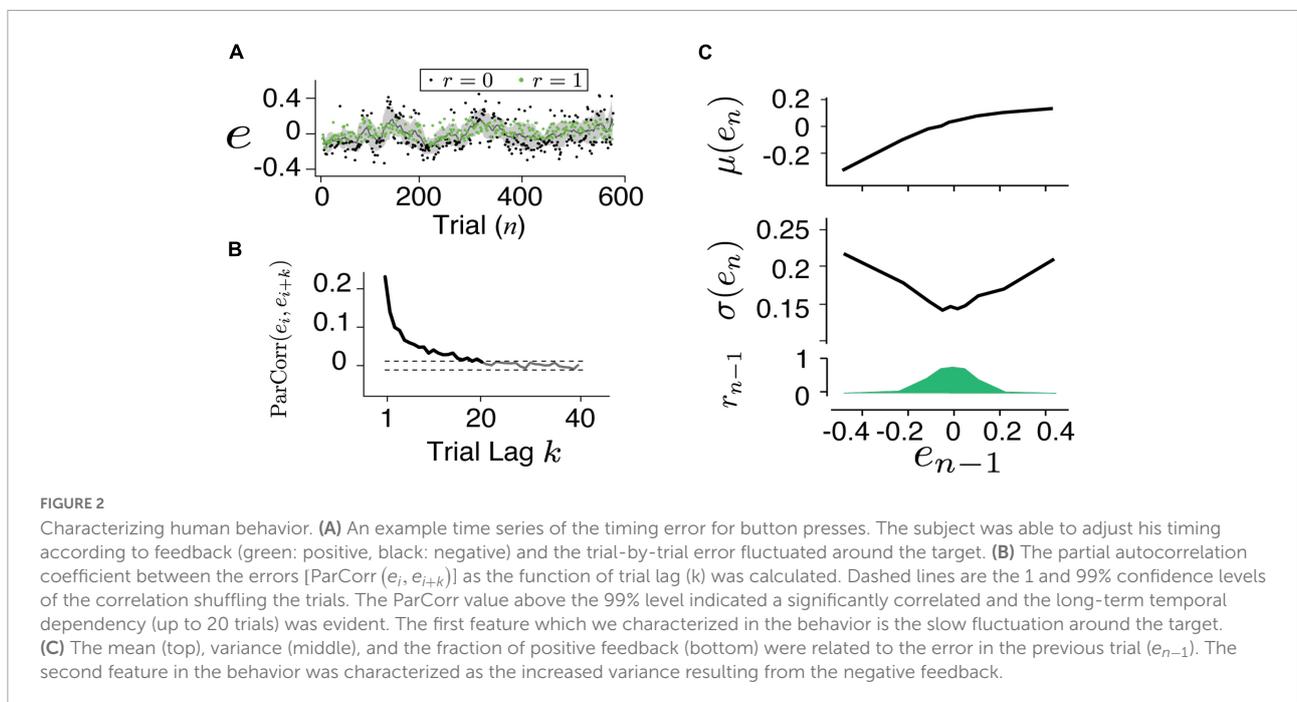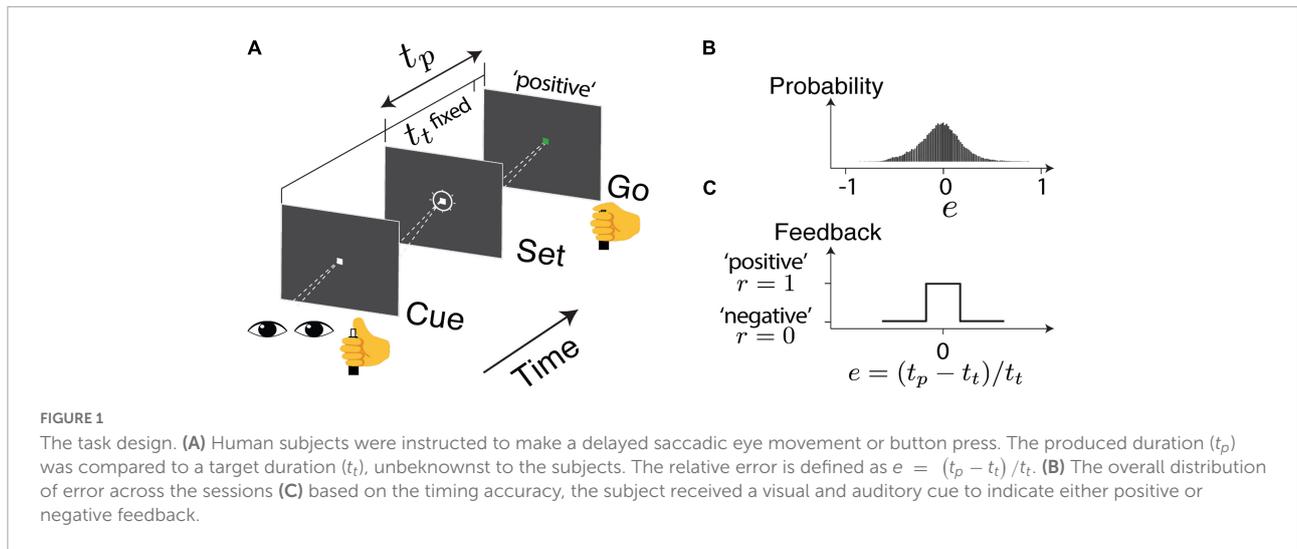
Human behavior experiment

Details of the experiment design and data collection were presented in a previous report (Wang et al., 2020). In brief, the human subjects were instructed to make delayed saccadic eye movements or button presses in response to an instruction signal (Cue-Set-Go, CSG task **Figure 1A**). The duration of the delay (production time t_p) was compared to a target duration (t_t), which was unbeknown to the subjects and was randomly sampled from a normal distribution $\sim \mathcal{N}(800ms, 80ms)$ between behavioral sessions. The relative error (e) in production time was defined as $e = (t_p - t_t)/t_t$. The subjects received positive feedback ($r = 1$) if e was within an acceptance window and negative feedback ($r = 0$) if it was outside the window (**Figure 1C**). The acceptance window was determined based on each subject's performance so that approximately half of the trials led to positive feedback. In each subject, the error distribution was centered around zero and the responses were sensitive to the feedback (**Figure 1B**). For each experimental session, we obtained two series consisting of error (e) and reward outcome (r) in the trial sequence as indicated by e_n and r_n (**Figure 2A**).

Autoregressive neural network model

In the classic recurrent neural network model, the input is integrated with the internal state of the recurrent unit to produce an output. Noise can be added to the output. We conceptualized the interval generating process in the recurrent unit as follows. The recurrent unit receives an update as feedback at each trial and combines it with the past history stored in the recurrent memory. The integration between univariate input and multivariate cell memory in a model of order p can be expressed as

$$e_n = h\left(W^e e_{n-1} + W^{h^T} h_{n-1} + b\right) + \epsilon_n$$

in which e_{n-1} is the input, $\overrightarrow{h}_{n-1} = [h_1, ..., h_p]$ is the cell memory, $h(.)$ is the activation function, b is the bias term and ϵ_n is the independent noise sampled from a normal distribution. The weights $\left[W^e, W^h\right] = \{w_i\}$ indicate the leverage of the current trial in the past i^{th} trial. The actual dimensionality

FIGURE 1
The task design. **(A)** Human subjects were instructed to make a delayed saccadic eye movement or button press. The produced duration (t_p) was compared to a target duration (t_t), unbeknownst to the subjects. The relative error is defined as $e = (t_p - t_t)/t_t$. **(B)** The overall distribution of error across the sessions **(C)** based on the timing accuracy, the subject received a visual and auditory cue to indicate either positive or negative feedback.

FIGURE 2
Characterizing human behavior. **(A)** An example time series of the timing error for button presses. The subject was able to adjust his timing according to feedback (green: positive, black: negative) and the trial-by-trial error fluctuated around the target. **(B)** The partial autocorrelation coefficient between the errors [ParCorr (e_i, e_{i+k})] as the function of trial lag (k) was calculated. Dashed lines are the 1 and 99% confidence levels of the correlation shuffling the trials. The ParCorr value above the 99% level indicated a significantly correlated and the long-term temporal dependency (up to 20 trials) was evident. The first feature which we characterized in the behavior is the slow fluctuation around the target. **(C)** The mean (top), variance (middle), and the fraction of positive feedback (bottom) were related to the error in the previous trial (e_{n-1}). The second feature in the behavior was characterized as the increased variance resulting from the negative feedback.

of the memory input (p) might vary and thus be determined for individual time series. Common practice is that a set of parameters is assumed to be constant throughout the time course and optimized with respect to the ensemble training data.

Weight varying autoregressive NN (vARNN) model

The weight varying autoregressive NN (vARNN) model requires the recurrent connection to be adjusted on the fly for every sample (**Figure 3A**). This is in a way similar to the stochastic nature of synaptic transmission in the

brain, which is proposed to be the substrate for motor variability and stochastic exploration during motor learning (Seung, 2003; Llera-Montero et al., 2019; Gershman and Ölveczky, 2020). Therefore, unlike the classic AR assuming constant weights, we introduced additional information provided by the feedback (r_{n-i}) for updating the weights and introducing variability. Even though negative feedback ($r = 0$) could not inform us about the size of the error (whether the interval is too long or too short), our data showed that subjects still proactively changed their motor noise. Here, we propose that the feedback is used to tune the variance, as opposed to the strength, of the associated connection w_i. To keep the relationship between feedback

and weights simple, we assumed that the weight variance is modulated by the feedback (r) in a linear fashion bounded in $[\sigma_+^2,\ \sigma_-^2]$, and this relation is applied ubiquitously for all weights:

$$w_i(r) \sim w_i \mathcal{N}\left(1, \sigma_-^2(1-r) + \sigma_+^2 r\right)$$

$$\epsilon_n \sim \mathcal{N}\left(0, \sigma_0^2\right)$$

$$h(x) = ReL(x+1)$$

As a special case, the vARNN model reduces to the classical ARNN model if σ_+^2 and σ_-^2 are set to zero. The activation is rectified linear (ReL) function that ensures that the NN generates a realistic output (i.e., $t_p > 0$ and $e > -1$). ϵ_n is the remaining unexplained (or task irrelevant) motor noise that was sampled from a normal distribution of zero mean and σ_0 standard deviation. Though more complex hidden structures can be included in the vARNN to achieve better fitting accuracy (with the danger of overfitting), we will evaluate the model with other alternatives of similar complexity.

Gated recurrent unit (GRU) model

Due to vanishing or exploding gradient problems, a sequential model cannot keep track of arbitrary long-term dependencies. In the ARNN, one can alleviate the issue of vanishing gradients by setting a cut-off for the autoregressive order p. On the other hand, the RNNs with LSTM units solve the gradient issue by controlling the sequential dependencies and the updates of internal states. The GRU is a simplified LSTM unit that merges the cell state with the hidden states. In our implementation, the GRU takes both the value and reward outcome as inputs.

The computation within the GRU is detailed as follows

Reset gate: $r_n = \sigma\left(W_r\left[X_n; h_{n-1}\right] + b_r\right)$

Forget gate: $f_n = \sigma\left(W_f\left[X_n; h_{n-1}\right] + b_f\right)$

Candidate for the hidden state: $\widetilde{h}_n = tanh\left(W_h\left[X_n; r_n \odot h_{n-1}\right] + b_h\right)$

Update of the hidden state:

$h_n = \left(1 - f_n\right) \odot h_{n-1} + f_n \odot \widetilde{h}_n$

$\odot$ symbolizes element-wise multiplication and $\sigma(.)$ indicates sigmoid function. The GRU used h hidden units to track the hidden state. We have tested networks with various sizes of hidden units, and have used $h = 32$ in the model as it reached asymptotic performance at $h = 32$. The input X_n at nth trial was from the previous trial's e_{n-1} and r_{n-1}, therefore the input has two dimensions. The parameters to be learned are the weight matrix $W_{r,f,h} \in R^{h \times (d+h)}$ and the bias $b_{r,f,h} \in R^h$ were set to 0. We also used L1 regularization to prevent overfitting.

Time series simulation and model learning

For ARNN and vARNN, we simulated the time series using designated sets of parameters $\theta = \{\omega_i, \sigma_+^2/\sigma_-^2\}$, where $i = 1, \ldots, p$. To verify that the model parameters can be learned and the optimization procedure is robust, we trained the model with simulated data generated from various assigned parameters and compared the optimized parameters with the assigned ones. Similarly, in GRU, we simulated the time series. However, the model parameters do not have interpretable correspondence like the ones (weight and variance ratio) in ARNN and vARNN.

All models made one step forward forecasts ($\hat{e}_n$) based on their corresponding graph in **Figure 3**. In all models, we assumed an additive noise (ϵ_n) at the last step was irreducible and was sampled from the independent identical distribution (i.i.d.). The residual error, defined as the difference between the estimated mean $\hat{e}_n$ and actual output, was what we aimed to minimize on the training data with respect to the set of chosen parameters. In real practice, the loss function (*Loss*) consists of the residual error (*Res*), and a regularization (*P*) to penalize the model complexity. The regularization prevents overfitting and helps strike a balance between the accuracy and generalizability of the model. It has two parts, the first part aims to relax the constraint of knowing the exact AR order for each time series. The models could have included inputs from far back in time, and this term ensures setting small weights to zeros. The regularization function achieves this by having a large gradient when the weight is close to zero and a vanishing gradient when it is far away from zero. The second term keeps the ratio of weight variance in a reasonable range.

$$Loss(\theta) = Res + P$$

$$[P(\theta) = \lambda_1\left[\frac{1}{p}\sum_{i=1}^{p}\frac{2}{1 + \exp\left(-c_1|\omega_i|^{1/c_2}\right)} - 1\right]$$
$$+ \lambda_2\left[\sigma_-^2/\sigma_+^2 + \sigma_+^2/\sigma_-^2\right]$$

In which λ_1 and λ_2 are the regularization coefficients. It relates to the data noise and we use $\lambda_1, \lambda_2 = 0.01\sigma^2(e_n)$. c_1 and c_2 are the regularization curve parameters. With AR coefficients in the range $[0, 1]$, $c_i, c_2 = 3$ works well for the data (Triebe et al., 2019).

$$Res(\theta)^{\text{wMSE}} \equiv \frac{1}{N}\sum_{n=1}^{N}g_n[\hat{e}_n(\theta) - e_n]^2$$

$$= \frac{1}{N}\sum_{n=1}^{N}\frac{1}{\sigma_n^2}[\hat{e}_n(\theta) - e_n]^2$$

In a deterministic NN model, the optimization objective is conventionally defined by the average of all prediction errors, so
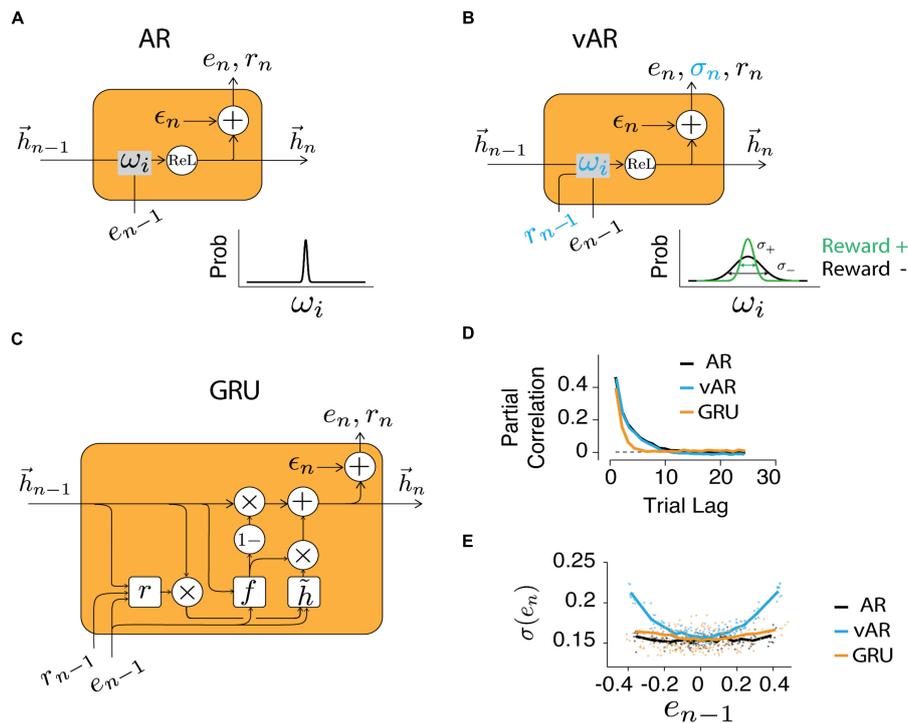
FIGURE 3
Neural network models. Structure of the recurrent unit in AR **(A)**, vAR **(B)**, and GRU NN models **(C)**. The output of NN (e_n) was retrieved and recursively served as its input. In vAR and GRU, the feedback (r_n) also served as input. In AR, the input weights were kept constant. In vAR, the input weights were adjusted according to the respective feedback. Specifically, positive feedback reduces the variance of weight (green). The hidden unit (h_n) implements the rectified linear function. In GRU, the hidden state was updated based on the computed node values (r: reset gate, f: forget gate, $\tilde{h}$: candidate state). The private and independent noise (ϵ_n) was sampled from a normal distribution. **(D)** Time series were simulated for each model, and the correlation showed the long-term temporal dependency of the output. **(E)** The output error variance was modulated by the feedback in vAR similar to the way in human behavior, but not in AR or GRU model.

called mean squared error (*MSE*). It assumes equal contribution from each data sample. In our case, vARNN directly provides a non-trivial output variance which varies trial-by-trial. Aitken showed that the best linear estimator is when a weighted sum of residuals is minimized, and the weight is proportional to the reciprocal of the individual variances (Aitken, 1936). The intuition is similar to Bayesian inference in that the error from an unreliable observation (large variance) should be weighed less than the error of more reliable ones. We therefore used this weighted MSE (*wMSE*) for computing the objective function. The error weight for each trial sample is indicated by g_n, not to be confused with the connection weights (ω_i) of the NN. A set of parameters were obtained for each simulation/experimental time series by minimizing the loss by combining the *wMSE* of residual and the penalty.

As the null estimation for the weighted residual function, we computed it with random weights:

$$Res(\theta)^{\text{rwMES}} = \sum_{n=1}^{N} g_n [\hat{e}_n(\theta) - e_n]^2 / \sum_{n=1}^{N} g_n$$

In which g_n is the random number sampled uniformly from a positive value set, e.g., [0, 1]. As a special case, when all g_n are equal, *wMSE* reduces to the *MSE*. All three models were trained on segments of the time series. For ARNN and vARNN, the dimensionality (p) of the cell state variable might vary between individual sessions, however, we initialized it to be $p = 20$ as inferred from the results of partial correlation analysis of the data. For GRU, a sequence of inputs was provided, and the corresponding final outputs were evaluated. In both the simulated and experimental data, models were trained on 80% of data sampled randomly without replacement, and the rest 20% were used for validation. We used a stochastic gradient descent optimizer and checked the performance on the validation set for every epoch. In the GRU, a stopping criterion was used to prevent overfitting. The criterion was that the loss on the validation set needed to be larger than the training set for at least 10 consecutive steps.

In the vARNN model training, backpropagation through a random node is seemingly impossible. We implemented a method called the reparameterization trick, applied in the variational auto-encoders (VAE) for capturing latent features

of noisy and high dimensional inputs (Kingma and Welling, 2013). In this method, a separated parameter takes care of the randomness and allows backpropagation to flow through the deterministic nodes.

Model evaluation metrics

We tested whether vARNN was a better model for predicting the sequential structure by comparing the residuals of ARNN, vARNN, and GRU. All three models were tested on the simulated and human behavioral data. Specifically, we first trained each model and compared the residual error $Res(\theta^*)$ on the validation set. In order to compare across subjects, sessions, and models, we have defined the residual as the ratio between the unexplained variance and total data variance. Note that this metric is similar to $1-R^2$, with non-uniform weights on the sample errors. $Res = 1$ indicates complete structurelessness, and $Res = 0$ indicates a fully predictable process. We also computed Res from optimization with respect to the randomly weighted error ($rwMSE$) and obtained a null distribution of Res value for the vARNN.

Note that the weighted error ($wMSE$) is related to the log-likelihood under the grand assumption of Gaussian distribution. Therefore, minimizing this $wMSE$ is equivalent to finding the posterior updated with observation on each trial. We refer to $wMSE$ to be consistent with the metrics used in other models. Also note that all models, tested here or reported elsewhere, can only account for approximately 20% of the total variance in the human data, the remaining 80% of the noise is temporally independent (**Figure 4A**). Other sources of variability can be due to downstream motor noise. Analysis of correlation and reward modulated variability has shown that the noise of saccadic movement and hand pressing was decoupled (Wang et al., 2020), suggesting that the majority likely pertained to the output stage of the motor system.

Figures 4A, **5C** show the comparison between models. There were 62 behavioral sessions collected from five human subjects (two females and three males, 18–65 years) who participated in the CSG task. Each behavioral session provided two time series one from eye movement and one from hand movement. This amounted to a total 124 time series and 59,410 trials in the data set. In the model simulation, we repeatedly generated 10 sessions for each parameter set and 1,000 trials for each session. For finding the model parameters, we also repeated the optimization 10 times with randomized initialization.
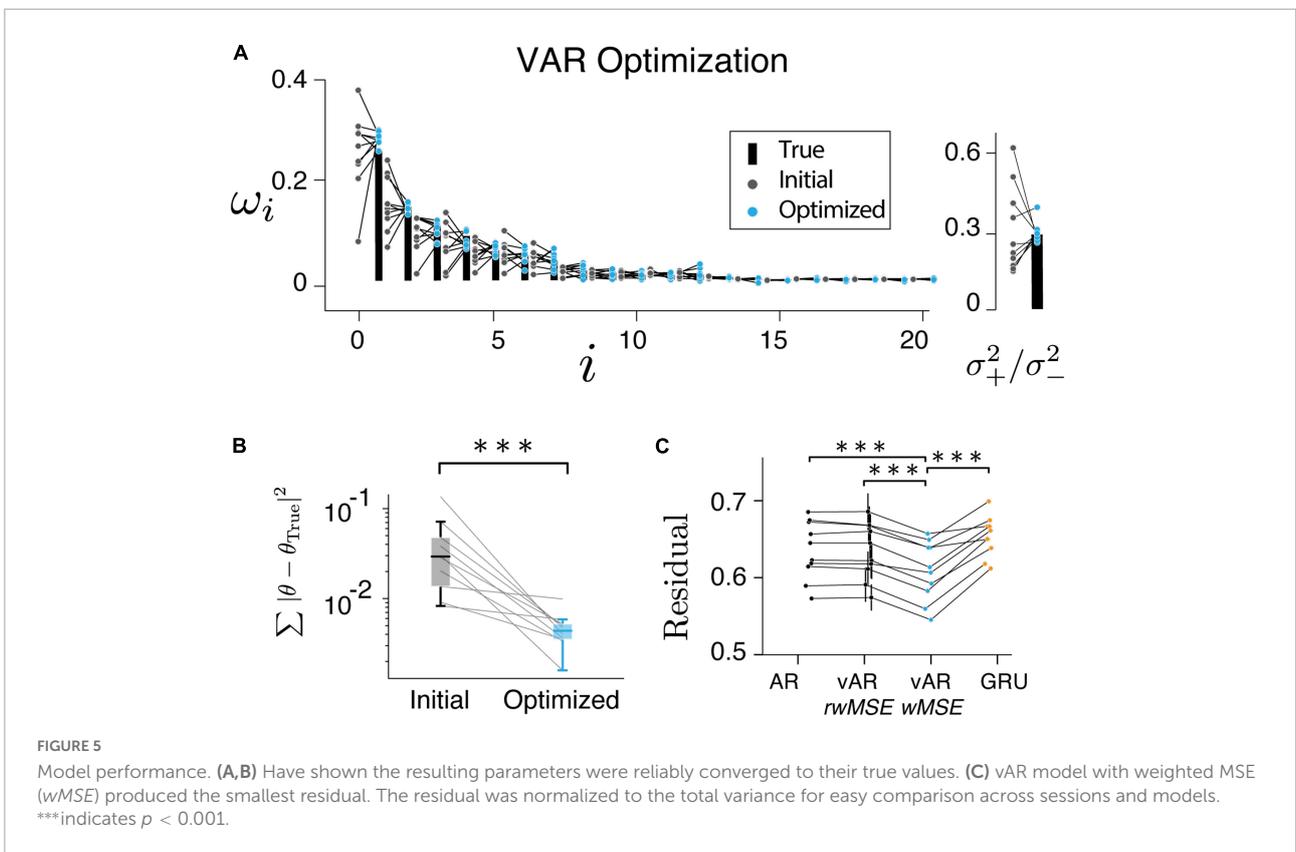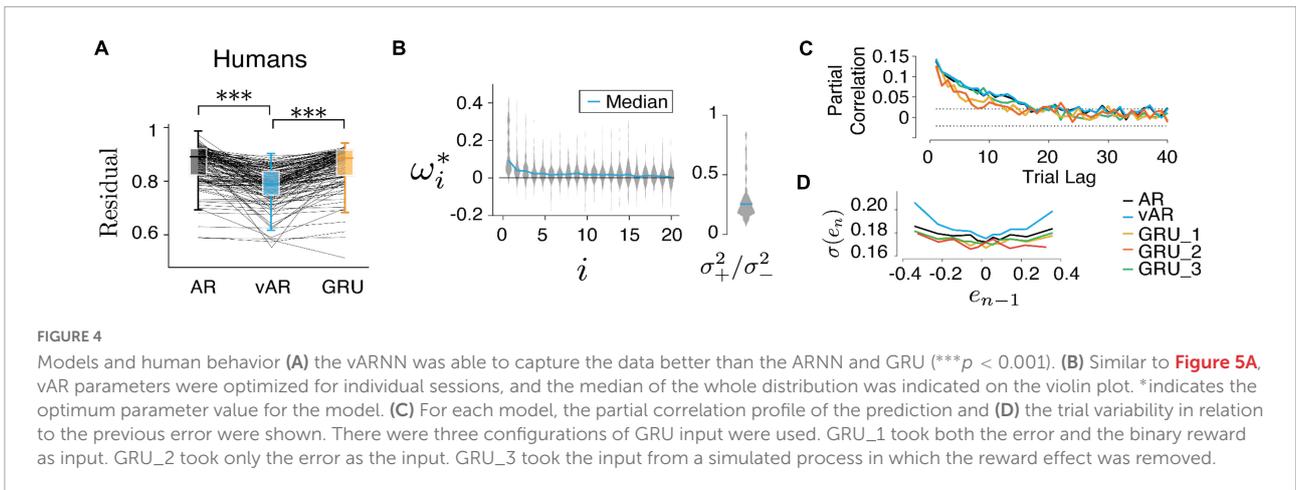
Results

Behavioral observations

An example session data of a human subject performing the button pressing task (**Figure 1**) is shown in **Figure 2A**. On average, human subjects received positive feedback ($r = 1$, green) in half of the trials (50.2%). A running mean with a window size of 20 trials has shown that the performance fluctuated around the targeted interval ($e = 0$). Analyzing the partial autocorrelation (*ParCorr*) as the function of trial lag showed that the temporal correlation of the behavioral sequence lasted up to 20 trials (**Figure 2B**). Analyzing the statistics of the two adjacent trials (e_{n-1} and e_n) revealed a non-trivial way the feedback influenced the timing variability (**Figure 2C**). It's shown that positive reinforcement ($r > 0$) reduced the motor variability while negative ones ($r = 0$) led to increased variability. Additional control experiments that randomly delivered the feedback verified that the feedback was *causally* involved in the adjustment of motor variability (Wang et al., 2020).

Behavioral characterization pointed to a strategy that humans were planning the action by incorporating the past actions and the corresponding feedback in sequential order. A neural network with recurrent units (RNN) provides the generic architecture for this task. We explicitly designed RNNs such that (1) the past performance and feedback were stored in the memory and could be retrieved later; (2) the feedback was used to assess the desirability of corresponding performance and (3) the model used a recurrent unit to generate this process recursively. Based on these requirements, we have tested three types of recurrent units that take the outcomes of previous trials as inputs for current trial prediction: ARNN, vARNN, and GRU (**Figure 3**). While all three structures are suitable for modeling time series data, ARNN represents a model that does not use reward feedback. However, the model can still access the feedback information partly and indirectly through the error input. On the other hand, vARNN and GRU use reward feedback information directly. GRU takes the previous reward feedback as input to generate a prediction, while the vARNN uses the reward for modulating the connectivity. Therefore, each model represents a different approach for simulating the time series data.

Model captures the temporal characteristics of human behavior

We first verified that models could give rise to salient characteristics of observed human behavior. The long-range correlation was recaptured (**Figure 3D**) by all models in the generated time series (simulated data with long- and short-term correlations similar to human data). Next, we asked how scalar feedback can be used to improve the estimation of production times t_p, a variable in vector form. One straightforward approach could be that the circuit rewires or redistributes the strengths of connectivity among the memory components. That is to increase the connection (ω_i) associated with the ones that lead to positive feedback and to reduce it for the negative ones. In this thread of thought, GRU was designed to take feedback as

FIGURE 4
Models and human behavior **(A)** the vARNN was able to capture the data better than the ARNN and GRU (***$p < 0.001$). **(B)** Similar to **Figure 5A**, vAR parameters were optimized for individual sessions, and the median of the whole distribution was indicated on the violin plot. *indicates the optimum parameter value for the model. **(C)** For each model, the partial correlation profile of the prediction and **(D)** the trial variability in relation to the previous error were shown. There were three configurations of GRU input were used. GRU_1 took both the error and the binary reward as input. GRU_2 took only the error as the input. GRU_3 took the input from a simulated process in which the reward effect was removed.

FIGURE 5
Model performance. **(A,B)** Have shown the resulting parameters were reliably converged to their true values. **(C)** vAR model with weighted MSE (*wMSE*) produced the smallest residual. The residual was normalized to the total variance for easy comparison across sessions and models. ***indicates $p < 0.001$.

input (**Figure 3C**) and modify the connectivity in a way so that the model would learn from behavioral data. On the other side, vARNN was proposed to solve this problem by adjusting the noise level of the weight (**Figure 3B** inset) instead of adjusting the weight itself.

In human data, the variance increased significantly with the increasing size of the error in the previous trial (**Figure 2C**). This variance adjustment was most evident in the time series generated with the vARNN, but less prominent in the ARNN and GRU (**Figure 3E**). This might seem a trivial finding given that the proposed vARNN deliberately introduces varying levels

of noise, however, as it is demonstrated in **Supplementary Material**, vARNN actually provides a close approximation to the theoretical optimal given the constraint of the system (**Supplementary Figure 1**).
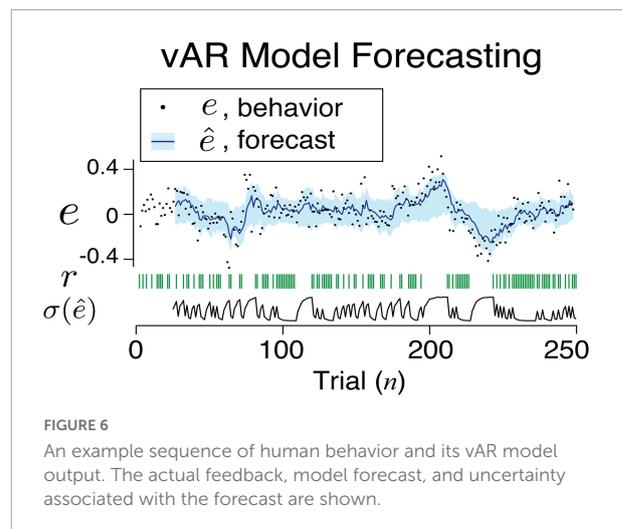
Model can learn from data

In the next step, we verified that the vARNN model can be trained. We found that the resulting parameters converged to the true parameters reliably (**Figures 5A,B**, pairwise *t*-test

p = 2.78e−4). We also found that the weighted residual error (*wMSE*) provided a better objective function for optimization as it led to a significantly smaller error than the conventional *MSE* (**Figure 5C**, pairwise *t*-test, one tail, p = 3.06e−5). It also deviated significantly from the null distribution computed with random weights (*rwMSE*) (**Figure 5C**, P = 6.70e−5). In theory, there can be an unbounded number of hidden units in GRU to capture any complex temporal dependencies. We have tested a single GRU layer with varying hidden units, which has shown asymptotic performance. The vARNN's objective takes into account the uncertainty in addition to the magnitude of error, we think this is one of contributing factors to why vARNN outperformed the GRU model (**Figure 5C**, P = 9.02e−4).

Comparing the vARNN model to its alternatives and model prediction error to its null distribution

Finally, we tested all three models on human behavior data. The residual errors of the test set were computed, respectively, and compared between models after optimization. Across all human behavioral sessions, vARNN stood out as the best network model in terms of its accuracy of prediction based on past history (**Figure 4A**, pairwise *t*-test, p = 1.36e−23 between model ARNN and vARNN, and p = 4.47e−29 between GRU and vARNN). For each model, we also characterized their temporal dependency and adjustable variability (**Figures 4C,D**) based on predicted response after each was trained on the human data. We also tested variants of GRU. There were 3 configurations of input were used. GRU_1 took both the response and the binary reward as input. GRU_2 took only the response as the input. Similar to the GRU model fit to artificial data in **Figure 3D**, the temporal correlation showed bias toward a shorter time scale when the process has a time-changing temporal correlation like ours (**Figure 4C**). In general, GRU can learn the desired long-term dependency as shown in GRU_3, which was from a simulated process with the reward effect removed. The network weights ω_i and variance ratio (σ_+^2/σ_-^2) in vARNN were obtained for each behavioral session, and their distributions are shown in **Figure 4B**. The median value of the variance ratio was 0.226 and significantly smaller (p = 1.5e−18) than the null hypothesis where the ratio should approximate 1.

These findings confirmed the strategy that a learning agent reduces the amount of noise in response to positive feedback. In simulating the vAR model, we assumed that the two levels of noise were shared among all connections between the input and the AR unit. We think that this is a strong assumption given the observed range of variance across behavioral sessions. Ideally, a connection might have its private and optimized range of noise adjustment. Once vARNN was trained for a behavioral session, we made a forward prediction and estimated the variance using

FIGURE 6

An example sequence of human behavior and its vAR model output. The actual feedback, model forecast, and uncertainty associated with the forecast are shown.

the outcomes in the past trials. **Figure 6** illustrates the one-step forward estimation of mean and standard error ($\sigma(\hat{e})$). The uncertainty of estimation was dynamically adjusted according to the recent history of feedback (r). This additional information is the key feature of our model in comparison to others.

Discussion

The nervous system is constantly calibrating itself in response to the ever-changing environment using external feedback. The trial-by-trial variations in a motor timing task we observed in humans carried both feedback sensitive and memory related components. This observation is consistent with the idea that animals and humans actively use the motor noise as a means to establish a trade-off between exploitation and exploration, i.e., reducing the noise when the outcome is better than expected (less exploration) and increasing the noise when the outcome becomes worse. Reward-based modulation of motor variability for stochastic exploration during motor learning has been reported or suggested by various previous studies (Wu et al., 2014; Pekny et al., 2015; Dhawale et al., 2017, 2019; Wang et al., 2020). However, the neural circuit mechanisms governing this process are still to be revealed.

In constructing our model, we assumed that the long-term temporal dependency, which is unlikely to be adjusted continuously, and therefore, can be modeled by an autoregressive process. This leads to observed long-term stationary temporal correlations observed in our and other paradigms (Chen et al., 1997; Chaisanguanthum et al., 2014; Fischer and Whitney, 2014; Liberman et al., 2014; Murakami et al., 2017). We further assumed that the synaptic connections have fixed values on average, corresponding to the fixed coefficient in the AR model. The short-term adjustment of timing responses was realized by injecting

various levels of noise into the synaptic weights in proportion to the strength of the connections leading to the vARNN model. The model output provides a probabilistic estimation for the next trial. Note that this model does not apply to a volatile environment where the reward contingency constantly changes, under which circumstance the learning presumably relies on the fast reconfiguration of synaptic strengths. We found that the modeling result is strikingly similar to the optimal control. This reductionist approach allows us to capture the essence of the behavior in a simple recurrent form and focus on the interpretability of the model.

We evaluated the vARNN and its alternatives, ARNN and GRU, according to their prediction accuracy. The vARNN captured both the temporal correlation and reward-modulated variability that the ARNN and GRU failed to. We suggest that the modular design of the vARNN model allows these two features to be learned effectively. While the autoregressive backbone of the vARNN was able to adjust network connections to cover the long-term correlations, the reward-based variations of connectivity independently took care of the reward adjustment of the short-term variations. In both ARNN and GRU, the network tries to learn the two temporal features simultaneously through the same mechanism and network units. We propose that the employment of separate mechanisms for different processes, for example, the neuromodulatory system for modulating the connectivity and activation of neurons, is very suitable for the brain in a variety of computations.

As we mentioned, in both ARNN and GRU models, the parameters were deterministic, but the added external noise at the output controlled the variability of response. vARNN model used a probabilistic approach. At the input stage, the network takes the previous outcomes and corresponding feedback as new evidence. The new evidence is incorporated stochastically by the weight whose variance was modulated by reward feedback. This point estimation is to assimilate the probabilistic estimation of posterior, in which the distribution of both prior and evidence are provided. At the output stage, the network generates a sample from the posterior and additive external noise. The updated internal state will be used as a new prior for the next trial.

There are other algorithms turning the deterministic regression model into a probabilistic generator. For example, a technique called thinning the network can achieve this by randomly and repeatedly dropping out the nodes or weight updating during training (Srivastava et al., 2014). By randomly turning on and off the nodes, it can significantly reduce overfitting and serve as regularization. Presumably, it is applied to redundantly connected networks. If we replace each node in the vARNN with a fully connected ensemble of units, our network can effectively implement the sampling algorithm similar to the thinning.

In this task, the only feedback the subjects received was a binary signal about the accuracy of timing at the end of each trial. The dopamine (DA) signaling in the brain provides a candidate infrastructure for such a signal, as elaborated in RL algorithms. DA is involved in the representation of reward and motivation in the brain and therefore, has been suggested as the neurobiological substrate for the feedback signal in RL. However, unlike that in RL algorithms, in our model, the state of the system is not fully available in order to update the corresponding value function. We here proposed a slightly different mechanism, which requires the brain to adjust its neural noise or, more specifically, the signal-to-noise level. Numerous studies pointed toward the effect of DA on signal and synaptic transmission in the prefrontal cortex (PFC) and PFC-striatum pathways (Law-Tho et al., 1994; Nicola and Malenka, 1997; Tritsch and Sabatini, 2012). This provides the neural substrate for the trial-by-trial adjustment of synaptic noise at the cellular level. Using acute slice preparations, it has been shown that DA can affect the intrinsic membrane excitability and synaptic transmission in the PFC (Kroener et al., 2009). Appropriate levels of DA receptor (D1R) stimulation can suppress the processing of neural noise and improve animals' performance in a working memory task (Vijayraghavan et al., 2007). It's highly possible that a similar mechanism was at play in our task. The DA projection carrying feedback signal is sent to PFC and the downstream basal ganglia areas. Another relevant example of controlling neuronal noise is attention in perception and cognition. It's been shown that the sensory cortical neurons can vary their correlated spiking noise depending on the relationship between the receptive field and the spatial cues (Ruff and Cohen, 2014; Ni et al., 2018). It's also possible that, by tuning noise among PFC neurons, the brain can harness the correlated variability in the population to achieve optimal behavioral control.

We note that in the behavioral experiments, the motor noise was harnessed differently during different phases of learning. The same neural mechanisms and neurochemical signaling can lead to apparently different strategies implemented at the level of behavior. Intriguingly, we found that in the naive human subjects where a new target was assigned during each test session, their baseline motor variability was maintained at a high level. Positive reinforcement effectively reduces variability. On the other hand, the well-trained subjects, where the baseline variability was kept at a minimal level, actively increased their motor noise such that exploration led to fast descent toward the rewarding place. This important observation lends insight into how we should regularize the neural network at different stages of training and suggests that in the initial phase, the model weight might respond to the positive reinforcement with variance reduction. However, at the expert learning stage where continuous minor adjustment can be accomplished by actively increasing the variance in response to the negative reinforcement.

Conclusion

Overall, we proposed a neural network model that implements the observed behavioral process by feedback-dependent tuning of the noise of connectivity. With its simplicity and biological plausibility, our model is an important supplement to the existing understanding of motor control. Our model also provides an opportunity to link models of continuous state control in artificial intelligence systems and RL in biological systems.

Data availability statement

The original contributions presented in this study are publicly available. This data can be found here: GitHub, https://github.com/wangjing0/vARNN.

Author contributions

JW and IO designed the study. JW developed the code and analysis. YE-J contributed to the code and analysis. All authors contributed to the article and approved the submitted version.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fncom. 2022.918031/full#supplementary-material

References

Aitken, A. C. (1936). IV.—On least squares and linear combination of observations. *Proc. R. Soc. Edinburgh* 55, 42–48. doi: 10.1017/S0370164600014346

Benbrahim, H., and Franklin, J. A. (1997). Biped dynamic walking using reinforcement learning. *Rob. Auton. Syst.* 22, 283–302. doi: 10.1016/S0921-8890(97)00043-2

Chaisanguanthum, K. S., Shen, H. H., and Sabes, P. N. (2014). Motor variability arises from a slow random walk in neural state. *J. Neurosci.* 34, 12071–12080. doi: 10.1523/JNEUROSCI.3001-13.2014

Chen, Y., Ding, M., and Kelso, J. A. S. (1997). Long memory processes (1/fαType) in human coordination. *Phys. Rev. Lett.* 79, 4501–4504. doi: 10.1103/PhysRevLett.79.4501

Dhawale, A. K., Smith, M. A., and Ölveczky, B. P. (2017). The role of variability in motor learning. *Annu. Rev. Neurosci.* 40, 479–498. doi: 10.1146/annurev-neuro-072116-031548

Dhawale A. K., Miyamoto Y. R., Smith M. A., and Ölveczky B. P. (2019). Adaptive Regulation of Motor Variability. *Curr. Biol.* 29, 3551–3562.e7. doi: 10.1016/j.cub.2019.08.052

Fischer, J., and Whitney, D. (2014). Serial dependence in visual perception. *Nat. Neurosci.* 17, 738–743. doi: 10.1038/nn.3689

Gershman, S. J., and Ölveczky, B. P. (2020). The neurobiology of deep reinforcement learning. *Curr. Biol.* 30, R629–R632. doi: 10.1016/j.cub.2020.04.021

Gullapalli, V. (1990). A stochastic reinforcement learning algorithm for learning real-valued functions. *Neural Netw.* 3, 671–692. doi: 10.1016/0893-6080(90)90056-Q

Kingma, D. P., and Welling, M. (2013). Auto-encoding variational bayes. *arXiv [Preprint]* doi: 10.48550/arXiv.1312.6114

Kroener, S., Chandler, L. J., Phillips, P. E. M., and Seamans, J. K. (2009). Dopamine modulates persistent synaptic activity and enhances the signal-to-noise ratio in the prefrontal cortex. *PLoS One* 4:e6507. doi: 10.1371/journal.pone.0006507

Law-Tho, D., Hirsch, J. C., and Crepel, F. (1994). Dopamine modulation of synaptic transmission in rat prefrontal cortex: An in vitro electrophysiological study. *Neurosci. Res.* 21, 151–160. doi: 10.1016/0168-0102(94)90157-0

Leblois A. (2013). Social modulation of learned behavior by dopamine in the basal ganglia: Insights from songbirds. *J. Physiol. Paris* 107, 219–229. doi: 10.1016/j.jphysparis.2012.09.002

Liberman, A., Fischer, J., and Whitney, D. (2014). Serial dependence in the perception of faces. *Curr. Biol.* 24, 2569–2574. doi: 10.1016/j.cub.2014.09.025

Llera-Montero, M., Sacramento, J., and Costa, R. P. (2019). Computational roles of plastic probabilistic synapses. *Curr. Opin. Neurobiol.* 54, 90–97. doi: 10.1016/j.conb.2018.09.002

Murakami, M., Shteingart, H., Loewenstein, Y., and Mainen, Z. F. (2017). Distinct sources of deterministic and stochastic components of action timing decisions in rodent frontal cortex. *Neuron* 94, 908–919.e7.

Ni, A. M., Ruff, D. A., Alberts, J. J., Symmonds, J., and Cohen, M. R. (2018). Learning and attention reveal a general relationship between population activity and behavior. *Science* 359, 463–465. doi: 10.1126/science.aao0284

Nicola, S. M., and Malenka, R. C. (1997). Dopamine depresses excitatory and inhibitory synaptic transmission by distinct mechanisms in the nucleus accumbens. *J. Neurosci.* 17, 5697–5710. doi: 10.1523/JNEUROSCI.17-15-05697.1997

Pekny, S. E., Izawa, J., and Shadmehr, R. (2015). Reward-dependent modulation of movement variability. *J. Neurosci.* 35, 4015–4024. doi: 10.1523/JNEUROSCI.3244-14.2015

Ruff, D. A., and Cohen, M. R. (2014). Attention can either increase or decrease spike count correlations in visual cortex. *Nat. Neurosci.* 17, 1591–1597. doi: 10.1038/nn.3835

Seung, H. S. (2003). Learning in spiking neural networks by reinforcement of stochastic synaptic transmission. *Neuron* 40, 1063–1073. doi: 10.1016/S0896-6273(03)00761-X

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15, 1929–1958. doi: 10.1109/TCYB.2020.3035282

Sutton, R. S., and Barto, A. G. (1998). *Reinforcement Learning: An Introduction.* Cambridge, MA: MIT Press. doi: 10.1109/TNN.1998.712192

Triebe, O., Laptev, N., and Rajagopal, R. (2019). AR-Net: A simple auto-regressive neural network for time-series. *arXiv [Preprint]* doi: 10.48550/arXiv. 1911.12436

Tritsch, N. X., and Sabatini, B. L. (2012). Dopaminergic modulation of synaptic transmission in cortex and striatum. *Neuron* 76, 33–50. doi: 10.1016/j.neuron. 2012.09.023

van Hasselt, H. (2012). "Reinforcement learning in continuous state and action spaces," in *Reinforcement Learning: State-of-the-Art*, eds M. Wiering and M. van Otterlo (Berlin: Springer), 207–251. doi: 10.1007/978-3-642-276 45-3_7

van Hasselt, H., and Wiering, M. A. (2007). "Reinforcement learning in continuous action spaces," in *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, (Honolulu, HI), 272–279. doi: 10.1109/ADPRL.2007.368199

Vijayraghavan, S., Wang, M., Birnbaum, S. G., Williams, G. V., and Arnsten, A. F. T. (2007). Inverted-U dopamine D1 receptor actions on prefrontal neurons engaged in working memory. *Nat. Neurosci.* 10, 376–384. doi: 10.1038/nn 1846

Wang, J., Hosseini, E., Meirhaeghe, N., Akkad, A., and Jazayeri, M. (2020). Reinforcement regulates timing variability in thalamus. *Elife* 9:e55872. doi: 10. 7554/eLife.55872

Wu, H. G., Miyamoto, Y. R., Gonzalez Castro, L. N., Ölveczky, B. P., and Smith, M. A. (2014). Temporal structure of motor variability is dynamically regulated and predicts motor learning ability. *Nat. Neurosci.* 17, 312–321. doi: 10.1038/nn. 3616