A Context-Free Model of Savings in Motor Learning

Mahdiyar Shahbazi [1], Olivier Codol [2,3], Jonathan A. Michaels [4,5,†], and Paul L. Gribble [1,4,†,*]

[1]Dept. Psychology, Western University, London, ON, Canada
[2]Mila–Québec Artificial Intelligence Institute, Montréal, QC, Canada
[3]Dept. Neuroscience, Université de Montréal, Montréal, QC, Canada
[4]Dept. Physiology & Pharmacology, Schulich School of Medicine & Dentistry, London, ON, Canada
[5]School of Kinesiology and Health Science, Faculty of Health, York University, Toronto, ON, Canada
[†]co-senior authors
[*]corresponding author: pgribble@uwo.ca

March 26, 2025

Abstract

Learning to adapt voluntary movements to an external perturbation, whether mechanical or visual, is faster during a second encounter than during the first. The mechanisms underlying this phenomenon, known as savings, remain unclear. Recent studies propose that the high dimensionality of neural control enables the retention of learning traces that may facilitate savings. To test this idea we used MotorNet, a framework for training recurrent neural networks (RNNs) to control biomechanical models of the human upper limb. RNNs were trained to perform reaching movements with a velocity-dependent force field (FF) and without (NF) in the sequence NF1 (baseline), FF1 (adaptation), NF2 (washout), and FF2 (re-adaptation). RNNs showed behavioural signatures of savings in the absence of any explicit contextual input signalling the presence or absence of the FF. Savings was more robust in RNNs with larger numbers of units. We identified a component of RNN activity associated with savings—a shift in preparatory activity that persisted even after washout. Displacing this preparatory activity in the direction of the shift enhanced savings, whereas perturbations in the opposite direction reduced or eliminated savings. These findings suggest a potential neural basis for motor memory retention underlying savings that is reliant on the high dimensionality of neural space, and is independent of cognitive or strategic learning.

Keywords—artificial recurrent neural network; motor learning; savings; arm movement; MotorNet

1 Introduction

In studies of motor learning, *savings* commonly refers to a phenomenon in which learning is superior after previous exposure to a motor task. Savings has been demonstrated in the context of voluntary reaching movements for adaptation to novel visuomotor perturbations and for learning to counter novel mechanical environments such as velocity-dependent curl force fields (FF) [1–7]. In a typical experiment, participants are initially exposed to a novel perturbation environment and they practice reaching to targets until an asymptotic level of performance is reached, for example recovery of approximately straight-line hand paths. Following this initial learning the perturbation is removed, and participants practice again until behavioural performance in this "washout" phase returns to pre-learning baseline levels. After washout participants are re-exposed to the perturbing environment. Savings is observed as a faster learning rate when re-exposed to the perturbing environment compared to initial learning, and sometimes also as a superior initial level of performance compared to when the perturbing environment was first encountered [7, 8].

There is some debate about the mechanisms that may be responsible for savings [3, 5]. Some propose that savings is produced by explicit, cognitive or strategic processes such as a conscious memory of the action selection strategy [2], a contextual signal associated with the perturbing environment [9], meta-learning of adaptation parameters such as learning rate [10–12], or reinforcement-based memories of successful execution [13]. Others have proposed that savings may arise from implicit learning processes that are not based on conscious, strategic mechanisms, including an increased sensitivity to previously experienced errors [8], use-dependent plasticity [14], or implicit updating of internal models that predict the sensory consequences of action [15].

Recent advances have been made in the ability to record from large numbers of neurons during motor learning tasks [16] and this has resulted in new approaches to understanding the relationship between high dimensional neural population activity and sensory, motor and task parameters during, and even prior to, voluntary movement [17, 18]. These new ways of thinking have prompted us and others to consider how seemingly complex behavioural phenomena may emerge from the dynamics of populations of neurons [19]. Sun, O'Shea, and colleagues recorded neural activity in primary motor cortex of rhesus macaques during a FF reaching task and identified a neural subspace of network activity during the preparatory period prior to movement that shifted after learning [20]. This "uniform shift" persisted even after behavioural washout of FF learning. The authors proposed that this neural trace of prior learning could facilitate subsequent savings [20]. Losey and colleagues used a brain-computer interface learning paradigm to study how neural population activity in the primary motor cortex of monkeys supports motor learning of multiple tasks [21]. They identified a change in a subspace of neural population activity that supported behavioural performance of a new task without interfering with a previously learned task. They proposed that the high dimensionality

of neural activity in primary motor cortex allows for the formation of memory traces that supports multiple behaviours without interference.

In the present paper we used artificial recurrent neural network (RNN) models of upper limb motor control to test the idea that high dimensional neural control facilitates the encoding of new motor memories without interfering with previously stored motor control policies, and that these neural traces of previous learning underlie subsequent savings, without the need for additional contextual signals. We used MotorNet [22] to train RNNs to control a mathematical model of the upper limb neuromuscular system [23] in the context of a simulated FF reaching task [24, 25]. Even without any explicit contextual cue signalling the presence of absence of FFs, RNNs showed behavioural signatures of savings. In addition savings was more robust as the number of units in RNNs increased. Using approaches similar to those described in previous studies we identified a learning-related shift in neural activity in the preparatory period prior to movement onset [20, 21]. We established a causal relationship between this neural shift and savings by perturbing neural activity along the direction of this neural shift. When RNN hidden unit activity was shifted in the direction of the neural shift, savings was enhanced, whereas neural perturbations in the opposite direction reduced or abolished savings. Our findings support the hypothesis that a neural basis of motor memory retention underlies savings, one that could be independent of any cognitive or strategic learning components and that depends upon the high dimensionality of neural space.

2 Results

We trained 40 recurrent neural networks (RNNs) with 128 fully connected gated recurrent units (GRUs) to control a mathematical model of the human upper limb [22] (Figure 1a,b). Task inputs to the RNN are the Cartesian coordinates of the movement target (x, y) and a binary go signal (0 or 1) indicating when to initiate movement (Figure 1c). The RNN also receives time-delayed feedback signals corresponding to the length and velocity of each muscle, and the Cartesian coordinates of the endpoint of the limb. The output of the RNN is time-varying muscle stimulation commands to each of 6 upper limb muscles (Figure 1c). Muscle stimulation commands range between 0 and 1 and act on a musculoskeletal model of the upper limb which includes multi-joint limb dynamics and a hill-type muscle model [23].

The networks were initially trained to produce point to point reaching movements between targets located in random locations throughout the limb's workspace. No perturbing FF was applied during this initial "growing up" training phase. We refer to the absence of a perturbing FF as a "null field" (NF). RNN weights were updated using backpropagation through time, using the Adam optimizer [26] implemented in PyTorch [27]. The loss function for optimization was based on minimizing the difference between hand position and target position, and also included regularization terms that

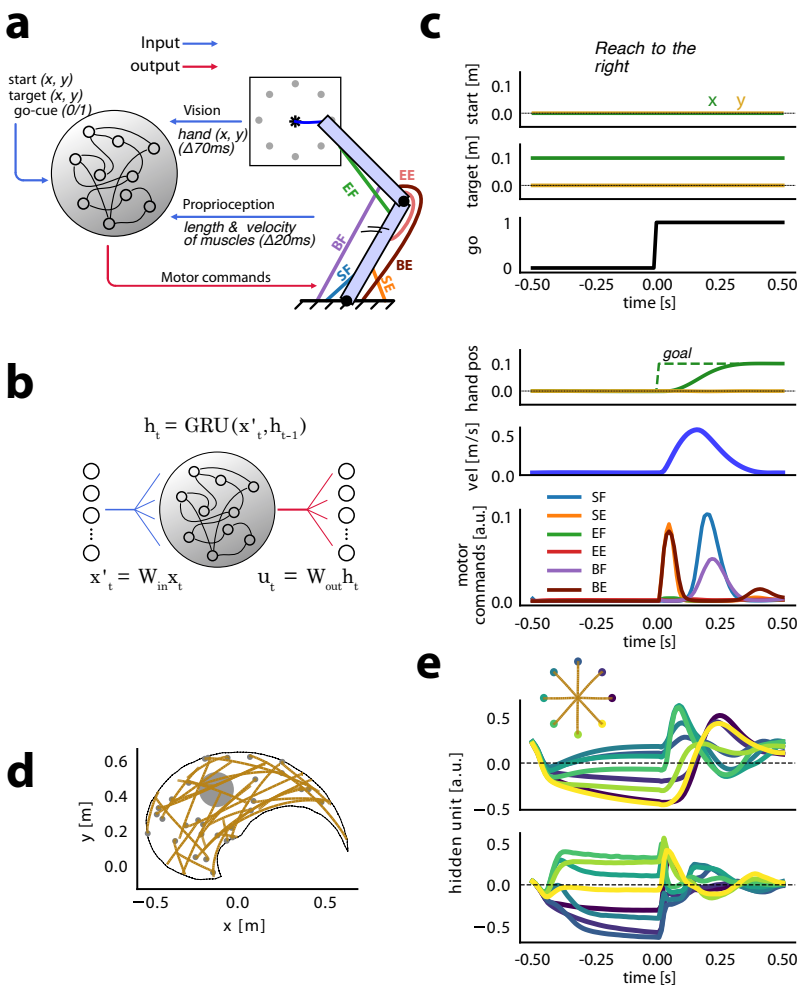

Figure 1. Recurrent neural network model inputs and outputs. (**a**) RNNs receive a 17-dimensional input signal consisting of the location of the movement target in Cartesian coordinates, a "visual" feedback signal giving the arm's endpoint position delivered with a 70 ms delay, a "proprioceptive" feedback signal consisting of the length and velocity of each of the 6 limb muscles delivered with a 20 ms delay, and a binary go cue. RNNs output 6 motor stimulation commands (between 0 and 1) to drive each muscle: SF (shoulder flexors), SE (shoulder extensors), EE (elbow extensors), EF (elbow flexors), BE (bi-articular extensors), and BF (bi-articular flexors). (**b**) The 17-dimensional input signal was mapped to the recurrent network using linear weights W_{in}. RNN output was transformed into motor commands by linear weights W_{out}. The vector h_t is the activity of hidden units at time t. (**c**) Task-related RNN inputs in a reach toward the rightmost target depicted in (a). For the purpose of illustration in this Figure, we translated the starting and target positions such that the start position is at the coordinates $(0, 0)$. The simulation duration was 1 s, with 10 ms time steps. The goal (dashed lines) was set to the hand's starting position before the go signal changed to 1, to the movement target position after that. (**d**) Sample endpoint trajectories after training RNNs on reaches to random target locations. Reaching trajectories are indicated in orange, and small gray dots show target positions. The large gray circle indicates the position of the centre-out reaches within the workspace. (**e**) Reaching trajectories and hidden unit activity over time at the end of training in the centre-out task. colours indicate each of 8 targets. The go-cue switches from 0 to 1 at time $t = 0$.

encouraged the network to produce smooth, human-like kinematics, phasic muscle commands, and stable hidden unit activity [28, 29] (see Figure 1e and Methods for details).

After training, the networks produced reaching movements with human-like characteristics including smooth, relatively straight hand paths with bell-shaped velocity profiles, and phasic activity in agonist and antagonist muscles spanning the shoulder and elbow (Figure 1c).

Figure 1d shows examples of reaching trajectories for reaches to targets located randomly across the limb's workspace. Models produced human-like reaches both when tested on reaches with random starting points and targets (Figure 1d) and when tested on centre-out reaches toward 8 equidistant targets (Figure 1e). Consistent with electrophysiological recordings in monkeys, RNN hidden units showed activity patterns that were relatively stable over time, and distinct for different movement targets during the delay period prior to the go signal (time 0 in Figure 1e). RNN hidden unit activity during movement was similarly distinct for different movement directions, and showed oscillatory activity consistent with that seen in recordings from motor cortex in non-human primates [30, 31].

2.1 Force field adaptation

After the networks were trained to perform point to point reaches in a NF, we implemented a relatively standard experimental sequence common in studies of human motor learning. We trained networks on a centre-out reaching task either in the absence of perturbing forces (null field, NF) or in the presence of a velocity-dependent curl force field (FF) (see Methods). First, networks were exposed to a NF (NF1) to characterize baseline performance. Following this, networks were trained to produce reaches in a clockwise FF (FF1, 3200 batches of training). After initial FF learning networks were re-trained in a NF (NF2, 10000 batches). Following this "washout" phase networks were re-trained in the FF (FF2, 3200 batches) (see Figure 2a).

We characterized behavioural performance of the networks by measuring the maximum deviation of each hand trajectory from a straight line connecting initial and final target positions. During centre-out reaching in the initial NF baseline tests (NF1, Figure 2b) the network produced straight hand paths with very little lateral deviation.

We then trained the networks to compensate for the effects of a clockwise force field (FF1). The first time networks encountered the FF (batch 0), they exhibited large lateral deviations from a straight line trajectory (Figure 2a,b). Over the course of training the network's hidden weights were modified so that the networks produced different muscle stimulation commands that compensated for the forces produced by the FF (Figure 2b). Only the hidden (recurrent) weights were modified after the "growing up" phase, and not the input/output weights. By the end of FF1, relatively straight hand paths were recovered, and lateral deviation was near that in the NF baseline tests (NF1).

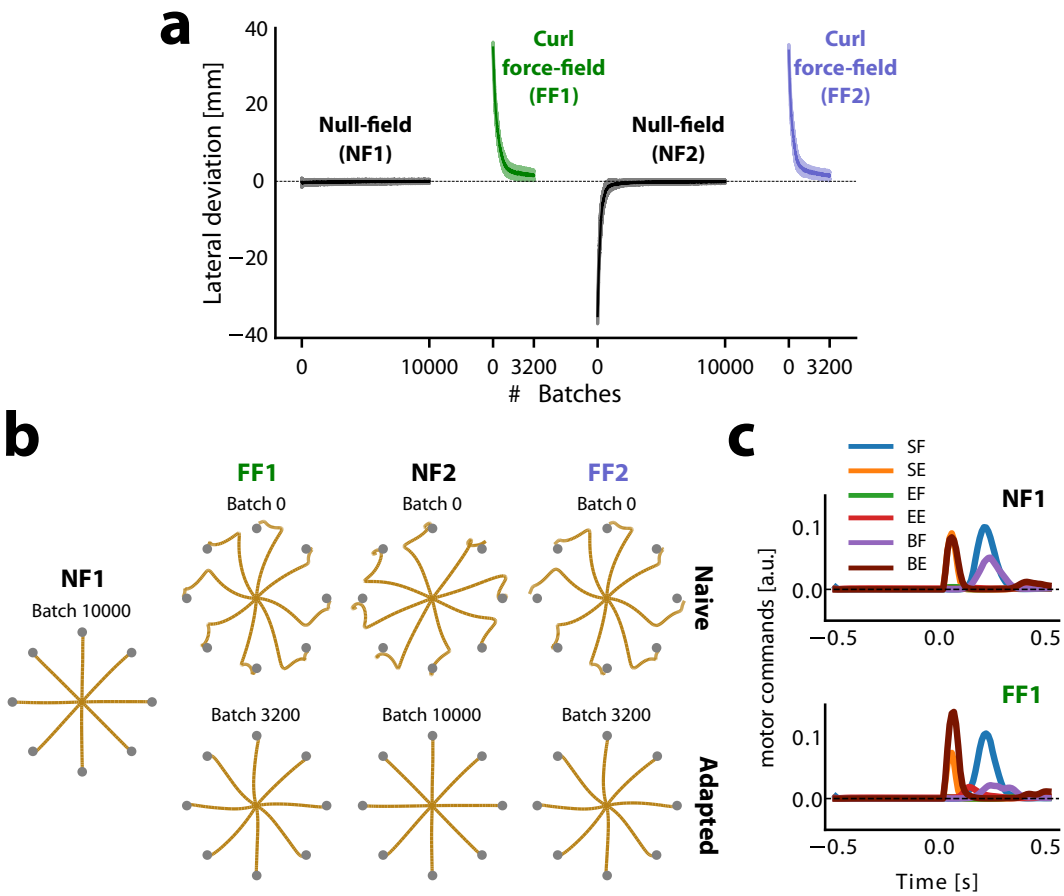

Figure 2. Networks learn to compensate for curl force fields without any contextual input. (**a**) Lateral deviation averaged over 8 centre-out reaches for each batch. Black indicates null-field phases (NF1 and NF2), green indicates the first phase of the force field (FF1), and purple indicates the second phase of the force field (FF2). Positive values indicate deviation in the direction of the force field, which is clockwise relative to the line connecting the starting and target positions. (**b**) Simulated reaching trajectories at the beginning and end of each phase, grouped in different columns. (**c**) Motor commands during reaching toward the rightmost target for NF1 and FF1.

Importantly, at no time during training did the networks receive any contextual signal related to the presence or absence of the FF. Adaptation occurred during FF training because as the simulated limb is perturbed by the FF, hand position deviates away from the target, and the loss function increases. Over training the values of the RNN hidden unit weights are changed to minimize the loss function, and in turn, recover straight hand paths.

As an additional control we trained networks after the growing up phase on an opposing force field (CCW) and then as above, exposed the networks to a NF washout phase, and then to a CW force field. In this case no savings was observed in the CW force field. This underscores that the savings effect observed in the main study was learning-specific— it was due to prior learning of the CW force field, and not a general effect of learning any novel dynamics.

2.2 Washout

After force field adaptation (FF1) we trained the networks in a washout phase (NF2) in which we removed the simulated FF that the networks trained on in FF1. As is the case in empirical studies of FF learning, the networks initially showed an after-effect in movement kinematics in the opposite direction of the force field (Figure 2a,b), indicating that the networks prepared muscle commands to compensate for the (now absent) FF. After training in NF2, networks recovered straight line hand paths that were very similar to the performance in the NF1 baseline phase (only 0.1 mm difference in lateral deviation).

2.3 Re-adaptation

Following washout we re-trained networks in the same curl field (FF2) that they had been exposed to previously in the FF1 block. We observed that when networks initially encountered the FF in FF2, they exhibited smaller lateral deviations than those observed in the beginning of FF1, when they were first exposed to the simulated FF ($t(39) = 10.940$, $p = 1.9e - 13$) (Figure 3a). In addition, networks adapted to the force field in FF2 faster than they did in FF1, as measured by an increased learning rate based on an exponential fit to the learning curve (see Methods; $t(39) = 9.284$, $p = 2.0e - 11$). This pattern of improved performance in FF2 is seen in both human and monkey studies of motor learning and is typically referred to as savings.

The difference in performance between FF1 and FF2 indicates that the network weights changed in such a way that facilitated improved initial performance and faster learning rate in FF2 as compared to FF1, while also not interfering with the performance in NF2. In other words, after training FF1, and washout in NF2, the networks retained enough information about the force field to improve re-learning in FF2, and this retained information was stored in such a way that it did not interfere with the ability of the networks to perform in NF2 just as they had done in NF1, before any FF learning. The pattern of savings observed here in our RNNs occurred despite the absence of any explicit contextual signal indicating the presence or absence of the simulated FF. When contextual signals are present, neural network models often create separate representations for two different tasks [32]. We hypothesize that in our study the high dimensionality of the RNNs allows them to develop representations that effectively serve the ongoing task while also preserving information about previously learned tasks.

We tested this hypothesis by repeating all simulations using RNNs with different numbers of hidden units. We found that as the number of hidden units increases, the likelihood of networks expressing savings also increases. For models with 256 hidden units, there was an 80% chance of expressing savings based on both the learning rate (if the learning rate is faster in FF2 than in FF1) and lateral deviation (if the initial lateral deviation in FF2 is smaller than when FF1 is first

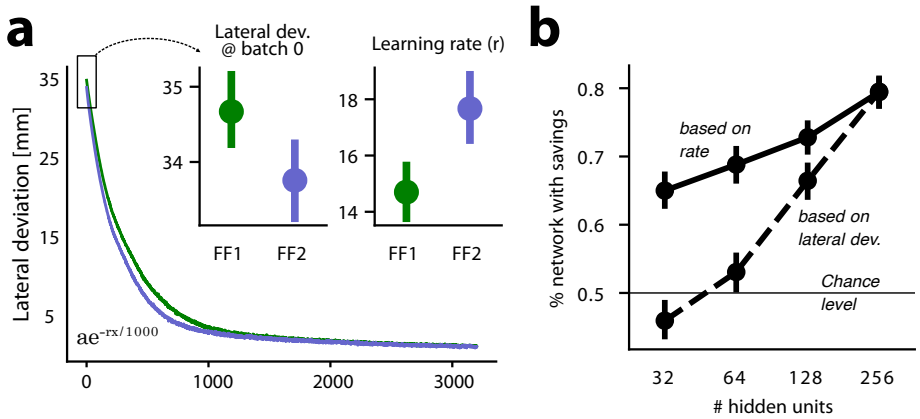

Figure 3. RNN models exhibit behavioural characteristics of savings. (**a**) Averaged learning curves shown in Figure 3a are overlaid. Sub-panels indicate (left) the lateral deviation at batch 0 (before training in the corresponding phase) and (right) the learning rate r after fitting an exponential curve to the lateral deviations over training for each network. (**b**) The percentage of networks (40 total) with savings is plotted against the number of RNN hidden units. The dashed line indicates the percentage of networks with savings, defined as the learning rate in FF2 being larger than in FF1. The solid line shows percentage of networks showing savings defined by lateral deviation at batch 0 being smaller in FF2 than FF1. Error bars indicate the 95% confidence interval.

encountered). This probability could drop to nearly chance level if the number of hidden units is smaller than 32. This supports the idea that the ability to store previously learned information in RNNs in a way that doesn't interfere with previous learning is based on the dimensionality of the network's weight space.

2.4 Learning related changes in hidden unit activity

To characterize changes in hidden unit activity after learning we followed a similar approach to that described by Sun, O'Shea, and colleagues [20], in which they investigated how neural population activity in motor cortex changed after monkeys learned to adapt to FFs in an upper limb reaching task. The focus is on hidden unit activity during the preparatory phase, prior to the go signal, as this is the primary determinant of the feed-forward motor commands to muscles [31].

We examined changes in a subspace defined by the relationship between hidden unit activity at the end of the preparatory period, just prior to the go cue, and movement-related force at the initial acceleration phase of the movement (hereafter referred to as the TDR subspace, see Methods for further details). We identified the TDR subspace by linearly regressing the preparatory hidden unit activity (340 ms before the go-cue) onto the endpoint (hand) force early during execution (90 ms after the go-cue; see Methods). Projecting the preparatory hidden unit activity associated with all 8 centre-out reaches onto this subspace revealed a ring formation (Figure 4a). This circular pattern

has also been observed in empirical studies of adaptation for motor cortical neurons [20]. This ring rotated in a counter-clockwise (CCW) following adaptation to a clockwise (CW) FF. This is consistent with the idea that that after training in the CWFF the preparatory activity of the RNN hidden units is tuned to facilitate the production of forces in the direction opposite to the upcoming FF during movement.

After washout (NF2), this rotation reverted, leaving little or no residual part of the initial CCW rotation in the preparatory hidden unit activity (Figure 4a). The preparatory hidden unit activity again rotated in a CCW direction after adaptation to the CWFF in FF2.

We probed the learning-related changes in RNN hidden unit activity that occurred outside of the TDR subspace. To do this, we calculated the extent to which the centroid of the preparatory hidden unit activity for 8 centre-out targets shifted following FF learning. Following the procedure used in Sun, O'Shea et al. [20], to isolate learning-related changes in hidden unit activity common to all reach directions we calculated the difference between the mean preparatory activity (340 ms before go cue) of NF1 and FF1 and then orthogonalized this with respect to the TDR subspace. The result is referred to as a "uniform shift" (Figure 4b). After projecting the mean preparatory activity of all experimental phases onto this uniform shift and normalizing the projection values such that the NF1 projection is 0 and the FF1 projection is 1 (see Methods), we observed that at the end of the washout phase (NF2), the RNN hidden unit activity still showed a projection onto this direction that was significantly different than zero ($t(39)=7.484$, $p=4.6e-9$; Figure 4c). This indicates that the mean hidden unit activity during the preparatory period did not fully revert to pre-adaptation levels, despite full behavioural washout by the end of the washout phase (Figure 2a). As in Sun, O'Shea et al. [20] we propose that this residual hidden unit activity captures information about the previously learned FF, and that this component can be linked to subsequent savings.

2.5 Perturbing preparatory hidden unit activity along the uniform shift

The fact that the preparatory activity of RNN hidden units did not fully revert to the NF1 levels in the uniform shift direction suggests that this residual activity might underlie savings [20, 21]. We tested this idea directly by perturbing hidden unit activity along the direction of the uniform shift, and we probed the effect of these perturbations on behavioural measures of savings.

We added a proportion of the uniform shift to the preparatory hidden unit activity of networks performing centre-out reaches and we measured the resulting changes in the lateral deviation of simulated hand trajectories. Importantly, the perturbations to preparatory hidden unit activity resulted in little or no changes in muscle activity prior to movement, and during movement itself no perturbations were delivered. We used models at batch 0 of the FF2 phase, when they had not yet been trained on FF adaptation a second time. We have already shown that in FF2, hand

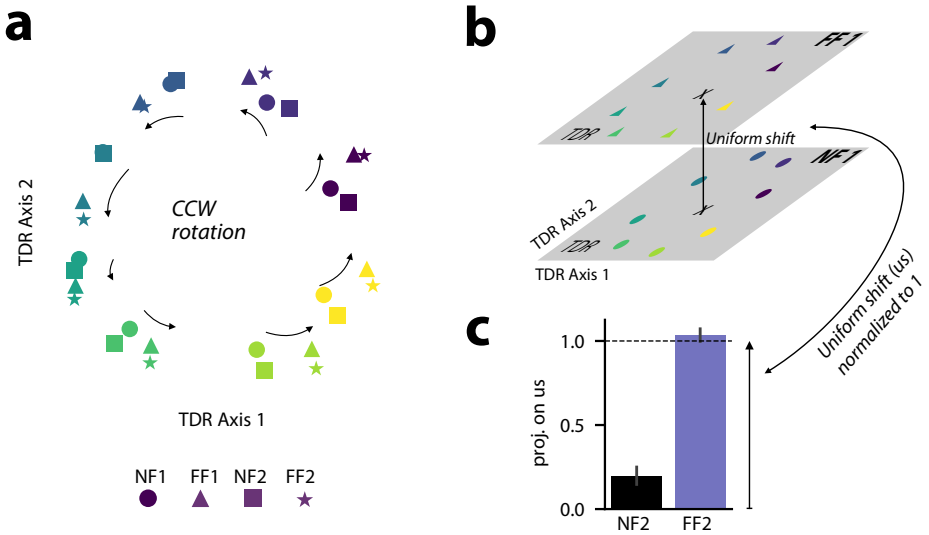

Figure 4. Changes in the preparatory activity of RNN hidden units following FF learning. **a**: Projection of the hidden preparatory activity (340 ms before the go-cue) of an example trained model performing 8 centre-out reaches on the force-predictive subspace acquired with targeted dimensionality reduction (TDR). Different reaching targets are indicated with different colours, and different adaptation phases are indicated with different shapes: circle for NF1, triangle for FF1, square for NF2, and star for FF2. **b**: A schematic illustration of the uniform shift. Each cross indicates the centre of the hidden preparatory activity for NF1 and FF1, and the arrow indicates the uniform shift. **c**: Projection of the hidden preparatory activity of all phases onto the uniform shift after orthogonalizing the uniform shift with respect to TDR. The data are scaled so that the projection of NF1 onto the uniform shift is zero and the projection of FF1 is one.

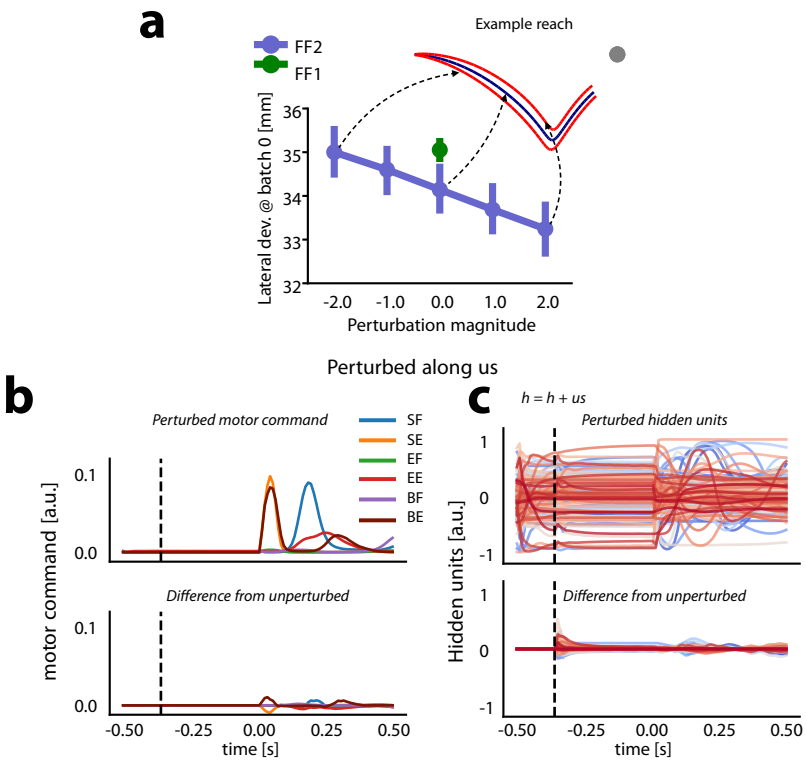

Figure 5. Uniform shift in RNN hidden unit activity is related to savings. **A**: Lateral deviation in FF2 (purple) when the hidden preparatory activity was perturbed in the positive (+) and negative (-) directions of the uniform shift with different magnitudes. Arrows point to the trajectories of an example model when the hidden unit activity was perturbed (red trajectories) or not (blue trajectories). The lateral deviation hand trajectories for FF1 are illustrated in green. **B**: Motor commands of the same model performing the same reach as shown in panel A when the hidden unit activity was perturbed with a magnitude of 2.0. The vertical dashed line indicates the time at which the perturbation was delivered. The lower sub-panel shows the difference from the unperturbed motor command. **C**: The activity of 128 RNN hidden units after being perturbed (dashed line). The lower sub-panel shows the difference between the unperturbed and perturbed RNN hidden unit activity.

trajectory lateral deviation is smaller than in FF1, and we used this as a metric to characterize savings (Figure 3a). We examined how much the lateral deviation at batch 0 would change following perturbations to RNN hidden unit activity in the direction of the uniform shift. Details about how the perturbations to hidden unit activity were implemented may be found in Methods.

When we perturbed the preparatory hidden unit activity in the opposite direction of the uniform shift (negative magnitudes in Figure 5a), lateral deviation of hand trajectories increased. By increasing the magnitude of perturbation further, we could effectively abolish savings altogether. In contrast, if we perturbed the RNN hidden unit preparatory activity in the same direction as the uniform shift, lateral deviation of hand trajectories was reduced, thus enhancing savings. Example trajectories toward the right-most target are shown in Figure 5a for each uniform shift perturbation. These results support the idea that the hidden unit activity in the direction of the uniform shift

that remains after washout represents a neural trace of the initial FF learning, one that supports subsequent savings when the models adapt to the FF a second time.

The perturbations changed the activity of hidden units (Figure 5c). These changes were large early after the delivery of the perturbation, and then they reached a steady state. However, is it important to note that these changes in the preparatory hidden unit activity did not result in substantive changes in the motor commands (Figure 5b), which emphasizes that the uniform shift resides in the null space of motor output.

In summary, the activity of networks along the direction of the uniform shift did not revert fully to pre-adaptation levels after the washout phase, despite the behaviour (hand trajectories and muscle activity) being fully washed out, the same as pre-adaptation baseline performance. Perturbing RNN hidden unit activity along the direction of the uniform shift enhanced savings while perturbations which reduced this residual hidden unit activity reduced or eliminated savings. RNN models with higher dimensionality (more hidden units) were more likely to exhibit evidence of savings while smaller models were less likely to show savings.

3 Discussion

We have shown here that RNNs trained to control a computational model of the upper limb show behavioural signatures of savings when learning multiple FFs in succession. We also found that savings is enhanced when the dimensionality of the control network is higher. Following the approach described in a recent electrophysiological study of monkey motor cortex [20], we identified a subspace of RNN hidden unit activity during the preparatory period after target presentation but prior to movement that shifts after initial FF learning, and subsequently retreats after behavioural washout in a NF. Importantly, this "uniform shift" did not retreat all the way back to pre-FF baseline levels after washout (see Figure 4c). Despite this residual trace of FF learning after washout, in a NF the RNN produced reaches to targets that were the same as those produced prior to initial learning. Importantly, alternating network training on opposing fields (CW and CCW) did not produce savings, as was the case in [20].

We interpret this residual hidden unit activity as a neural trace of the initial learning that remains after washout, and we suggest that this persistent signal contributes to savings [20]. We tested this hypothesis directly by perturbing RNN hidden unit activity during the preparatory period along the direction of the uniform shift—an approach that is possible in a computational model but is not presently feasible in a biological neural network. After NF washout when we re-exposed networks to a previously learned FF, perturbations that amplified the residual trace of learning (increasing activity along the uniform shift) resulted in increased savings. When we perturbed hidden unit activity in the other direction to reduce activity along the uniform shift, savings was reduced and

in some cases abolished altogether. These results provide evidence of a causal link between the identified neural trace of learning that remains after washout and subsequent savings when RNNs are re-exposed to the previously learned FF.

Our results are compatible with the proposal by Sun, O'Shea, et al. [20] that the activity of neurons in primary motor cortex during the preparatory period prior to movement contains a component that tracks previously learned motor behaviour. They propose that these residual traces of prior learned behaviour are encoded in a way that separates the associated motor memories in neural state space and facilitates recall of the appropriate control policies. Losey et al. [21] describe a similar account in the context of a brain-machine interface in which new motor learning is encoded in a neighbouring region of neural state space such that it solves the motor control task but doesn't interfere with prior learning. In our RNNs this was achieved through learning-related changes in the recurrent weights, such that after NF washout the component of the uniform shift that remained didn't interfere with NF motor behaviour, but did produce savings when networks were re-exposed to the previously learned FF.

Our finding that higher-dimensional RNNs are more likely to produce savings supports the idea that encoding newly learned control policies so that they do not interfere with previously learned motor memories depends upon the availability of adequate dimensionality in neural state space. This implies that multiple motor memories can be encoded in neural subspaces so that they do not interfere with each other. A number of recent empirical studies support this idea. Kim et al. [33] recorded from anterior lateral motor cortex of mice over several months and tracked how neurons represented different learned motor tasks. They found that learning produced new neural representations that did not modify existing representations, and re-exposure to a previously learned motor task re-activated the previous neural activity patterns. In a recent paper Bernardi et al. [34] recorded neural activity in prefrontal cortex and hippocampus of monkeys during cognitive tasks, and found that multiple abstract task-related variables were encoded in neural state space using a geometry that allowed separability using linear classifiers. Neural recordings in monkey motor cortex show that this kind of task representation emerges prior to movement, in preparatory activity after a movement target is shown but before a go signal instructs the animal to begin a movement [20, 30]. A similar time course emerged in our RNN simulations here (Figure 1e, Figure 4a).

In primates presumably high-level contextual cues can aid in indexing the appropriate previously learned control policy by activating populations of neurons in a neural direction appropriate for the previously learned task. Indeed a number of theoretical accounts exist that position contextual cues as a driver of motor memory encoding and selection [9, 35, 36]. Similar indexing is likely occurring in accounts where the learning of FFs that would normally interfere is avoided by associating each with the planning [37] or imagination [38] of different follow-through movements. In our RNNs no such contextual signal was provided, and so the question arises, how are residual traces of previously

learned FFs activated? One possibility is that the error signals encountered when our RNNs are re-exposed to a previously learned FF activate neurons within the previously learned subspace. This kind of scheme in which re-exposure to previously encountered errors produces savings is consistent with accounts in which a history of errors or corrections to errors plays a role in motor memory formation [3, 8].

In a recent computational modelling study Dubreuil et al. [18] proposed that non-random neural population connectivity structures involving multiple subpopulations that play functionally distinct roles encode multiple tasks better than random connectivity structures. This seems compatible with the idea presented here and in other recent work that low-dimensional recurrent subspaces embedded within a high-dimensional neural control space are used to encode features of movements such as target directions [30], anticipated sensory consequences of perturbations [39], and motor skills [20, 21]. The idea that motor memories are encoded in a distributed sensorimotor network and that features of motor adaptation emerge as a result of the dynamical properties of recurrent neural circuits have also been discussed in other computational modelling studies using recurrent neural networks. Ajemian et al. [40, 41] proposed a theoretical framework in which error signals prompt a reorganization of synaptic connectivity to encode motor memories within a high dimensional neural space.

The work presented here adds to the growing literature documenting how complex features of behaviour can arise due to the dynamics of recurrent neural circuits. Fuelner et al. [42] show that a number of features of motor adaptation emerge as a result of the dynamical properties of recurrent neural circuits in which sensory feedback modulates motor output. In a recent paper Smoulder et al. [19] show that neural activity in monkey motor cortex scales with reward magnitude, and that this reward signal interacts with movement preparation signals in such a way that high rewards disrupt movement preparation and result in poor motor performance compared to moderate rewards. They propose that these interactions constitute a neural basis of "choking under pressure".

The phenomenon of savings in motor learning implies that motor memory associated with an initial bout of training leads to faster subsequent relearning. The nature of the memory that is stored and how it influences subsequent relearning has been a topic of some debate in the recent literature. One account of savings emphasizes the effect of explicit, strategic components of motor learning [2, 43, 44]. Another line of work focuses on the idea that faster relearning may also be driven by implicit learning processes that result from previously experienced errors [3, 5, 7, 45].

The results of our work here with RNNs and the related electrophysiological studies of populations of motor cortical neurons of non-human primates [20, 21] point to a neural basis of savings. We showed here that by increasing the number of hidden units in our RNNs, and hence increasing the dimensionality of the control space, RNNs were more likely to produce behavioural signatures of savings (Figure 3b). The high dimensionality of neural space enables a new motor memory to be

encoded in such a way that it doesn't interfere with other previously learned information, while still facilitating savings when the network is re-exposed to the newly learned skill. This neural basis of savings can be characterized as implicit, since in our RNN simulations we did not provide the network with any contextual input that would signal the presence or absence of any given FF.

4 Methods

4.1 RNN model

Recurrent neural networks (RNNs) were trained to control movements of a simulated two degree of freedom arm that included rotations about a shoulder joint and an elbow joint in a horizontal plane. The model includes six rigid-tendon Hill-type muscle actuators comprising mono-articular flexors and extensors spanning the shoulder and elbow, as well as a pair of bi-articular muscles producing flexion or extension forces about both shoulder and elbow joints [23]. RNN models are implemented in PyTorch [27] and receive a 17-dimensional input signal to a linear input layer, which is fully connected to a recurrent layer consisting of gated recurrent units (GRUs) [46]. The GRU layer is connected to a 6-dimensional linear output layer which provides stimulation commands over time to each of the 6 muscles in the arm model (see Figure 1). Simulations were carried out in Python 3.10 using our open-source MotorNet toolbox [22].

Input-hidden (W_{in}) and hidden-hidden recurrent (W_{r}) weights (see Figure 1) were initialized using Glorot initialization [47] and orthogonal initialization [48], respectively, with biases set to 0. The output layer used a sigmoid activation function. The hidden-output weights (W_{out}) were initialized with the Glorot initialization scheme, and its biases were set to -5.0. The sigmoid activation function ensured the controller's output remained close to 0 at the start of training, promoting a stable initialization state. We set the initial hidden unit activity of the network ($\mathbf{h}_0$) as a learnable parameter and initialized it to 0.

The RNN models received a 17-dimensional input vector consisting of task-related inputs along with time-delayed feedback representing visual and proprioceptive signals. The task-related input consisted of a 2-element vector of (x, y) Cartesian coordinates for the target position, and a binary go-cue signal that switched from 0 to 1 when the movement should be initiated. The visual feedback was a Cartesian coordinate of the arm's endpoint (x, y), and the proprioceptive feedback was the lengths and velocities of all 6 muscles. The time step for simulations was set to 10 ms, the visual delay (Δ_v) was 70 ms, and the proprioceptive delay (Δ_p) was 20 ms. We also treated the go cue as a visual signal, meaning that at each time step the network received the 70 ms time delayed value. At each time step the RNN model transformed the above described 17-dimensional input into a 6-dimensional muscle stimulation command.

4.2 Growing up training phase

During an initial "growing up" phase new initialized RNN models were trained to move the arm from random starting positions to random target positions, both drawn from a uniform distribution across the joint space of the arm model. Note that due to muscle lengths and joint geometry, only a subset of the workspace was reachable for the model (Figure 1d). In 50% of simulations, no go-cue was provided (a catch trial). This was done to ensure that the network avoided producing anticipatory muscle stimulation commands. In the other 50% of cases the time of the go-cue switch from 0 to 1 was drawn from a random uniform distribution between 100 ms and 300 ms after the start of each simulated trial.

The loss function for training was mainly comprised of position loss, the Euclidean distance between the arm endpoint position $\mathbf{x}_t$ and the desired position $\mathbf{x}_t^*$. The desired position was set to be equal to the starting position of the limb's endpoint before the go cue, and after that the target position. We also included terms in the loss function that punished large and oscillatory hidden and muscle activity, and the jerk (the second derivative of acceleration) of the endpoint trajectory [49]. The full form of the loss function is shown in Equation 1:

$$
\begin{aligned}
L &= \frac{\sum_{t=1}^{N} L_t}{N} \\
L_t &= 10^3 L_t^p + 10^5 L_t^j + 10^{-1} L_t^m + 10^{-5} L_t^h \\
L_t^p &= \left\| \mathbf{x}_t^* - \mathbf{x}_t \right\|_1 \\
L_t^j &= \dddot{\mathbf{x}}_t^T \dddot{\mathbf{x}}_t \\
L_t^m &= \mathbf{f}_t^T \mathbf{f}_t + 3 \times 10^{-3} \dot{\mathbf{f}}_t^T \dot{\mathbf{f}}_t \\
L_t^h &= \mathbf{h}_t^T \mathbf{h}_t + 10^2 \dot{\mathbf{h}}_t^T \dot{\mathbf{h}}_t
\end{aligned}
\tag{1}
$$

where the subscript t indicates time step, N is the total number of time steps in the simulation, T is the transpose operation, and $\|\|_1$ is the L_1 norm. L_t^p, L_t^j, L_t^m, and L_t^h indicate position, jerk, muscle, and hidden loss, respectively. $\mathbf{h}_t$ is a n-element vector of hidden unit activity, and $\dot{\mathbf{h}}_t$ is its derivative. $\mathbf{f}_t$ is a 6-element vector of muscle forces, and $\dot{\mathbf{f}}_t$ is its derivative. Note that muscle forces are different from muscle stimulation commands (Figure 1A,C), which are inputs to the Ordinary Differential Equation that produces muscle forces [23].

The RNN models were initially trained on 20,000 batches with a batch size of 32 on simulations of 1 s (100 time steps). RNN weights were adjusted using the Adam optimization scheme [26] with a learning rate $lr = 0.003$.

4.3 Motor learning phases

Once the networks were trained to perform reaches to random targets in a null-field (NF), we fixed the input-to-hidden weights (W_{in}) and the hidden-to-output weights (W_{out}) and their biases. This allowed us to isolate subsequent learning-related changes in hidden unit activity resulting from our experimental manipulations to the recurrent connectivity of hidden units [42].

We trained networks on centre-out reaches from a start position to 8 equidistant targets around the circumference of a circle (see Figure 1). The start position corresponded to external joint angles of 60 degrees at the shoulder and 90 degrees at the elbow. We exposed models sequentially to FFs or NFs and in each phase we continued to adjust hidden recurrent weights to optimize the loss function described in Equation 1. During these experimental phases we used a stochastic gradient descent optimization scheme with learning rate parameter $lr = 0.005$ [50]. This ensures batch-local learning, and thus provides greater control and transparency over the course of learning. It also results in gradual learning over batches, better resembling learning curves from empirical studies of force field learning in humans and non-human primates.

In the NF1 experimental phase the models were trained on centre-out reaches only. We trained the models for 10,000 batches of size 32 (4 repetitions of each of the 8 targets). As in the growing-up phase, 50% of trials were catch-trials in which the go-cue did not change from 0 to 1. After NF1 training, we continued to train the models to perform centre-out reaches but we introduced a clockwise curl force field (CWFF) for all movements (FF1). The external force F_x, F_y applied at the arm's endpoint that produced a CWFF is described by the following equation:

$$\begin{bmatrix} F_x \\ F_y \end{bmatrix} = b \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \tag{2}$$

where $\dot{x}$ and $\dot{y}$ are the velocity of the arm's endpoint in Cartesian coordinates and $b = 8$ Ns/m is a scalar constant defining the strength of the FF. In the null field (NF), $b = 0$. We trained the models for 3,200 batches of size 32 in the FF1 experimental phase, with 50% catch-trials.

Following FF1, the models were trained again in a null field (NF2), using the same procedures as in NF1 (10,000 batches of size 32). After NF2, the models were again trained in the presence of a CWFF (FF2), exactly as in FF1.

4.4 Lateral deviation

We evaluated the behavioural performance of the models during NF1, FF1, NF2, and FF2 by calculating the maximum lateral deviation of the endpoint trajectory from straight lines connecting the starting and target positions. We will refer to this measure as "lateral deviation", and it was considered positive if it was in the clockwise direction from the straight line, and negative otherwise.

For each batch of training we calculated the mean lateral deviations across all 8 targets.

For each model, we characterized the learning rate during FF1 and FF2 by fitting an exponential function of the following form to the mean lateral deviation across training batches:

$$\hat{y} = \alpha e^{-r x_n / 1000} \tag{3}$$

where $\hat{y}$ is the modelled lateral deviation at batch number x_n, α is a scaling factor that determines the initial lateral deviation, and r is the rate at which lateral deviation decays over time (indicating the learning rate). Before fitting we smoothed learning rates over batches by window-averaging with a kernel size of 5 batches.

4.5 Targeted dimensionality reduction

Following the procedure described in [20] we identified a subspace of RNN hidden unit activity that predicts the arm's endpoint initial forces based on the preparatory activity of hidden units (hidden unit activity before the go-cue). To do this we applied targeted dimensionality reduction (TDR) using model data at the end of the NF1 experimental phase. The subspace is defined as:

$$\mathbf{H}^{\text{NF1}}_{\text{-340 ms}} = \left[\mathbf{F}^{\text{NF1}}_{\text{+90 ms}} \quad \mathbb{1} \right] W \tag{4}$$

where $\mathbf{H}^{\text{NF1}}_{\text{-340 ms}}$ is the matrix of hidden unit activity of size (targets $\times$ units) 340 ms before the go-cue, $\mathbf{F}^{\text{NF1}}_{\text{+90 ms}}$ is the matrix of endpoint forces of size (targets $\times$ 2) 90 ms after go-cue, $\mathbb{1}$ is the targets-element vector concatenated to the force matrix, and W is the matrix of size (3 $\times$ units). For the $\mathbf{F}^{\text{NF1}}_{\text{+90 ms}}$ parameter 90 ms was chosen because it coincides with peak acceleration. For the $\mathbf{H}^{\text{NF1}}_{\text{-340 ms}}$ parameter, 340 ms was chosen because it ensures that hidden unit activity is stabilized.

We then calculated the pseudo-inverse of W, resulting in a (units$\times$3) matrix W^+. To find the force-predictive subspace, we took the first two columns of W^+ (ignoring the intercept) and orthogonalized them using the Gram-Schmidt orthogonalization scheme, resulting in $\hat{W}^+$. We projected the preparatory hidden unit activity of all experimental phases ($\mathbf{H}^{\text{NF1}}_{\text{-340 ms}}$,$\mathbf{H}^{\text{FF1}}_{\text{-340 ms}}$,$\mathbf{H}^{\text{NF2}}_{\text{-340 ms}}$,$\mathbf{H}^{\text{FF2}}_{\text{-340 ms}}$) onto $\hat{W}^+$ after removing their global mean.

4.6 Uniform shift

Following the experimental phases FF1 and FF2 we calculated the direction in which the RNN hidden unit activity during the preparatory period shifted, averaged across movement directions. We averaged the preparatory hidden unit activity over the 8 targets in FF1 and NF1, and then calculated the difference. Following the convention in [20] this shift is referred to as a "uniform shift" (us):

$$\mathbf{us} = \bar{\mathbf{H}}^{\text{FF1}}_{\text{-340 ms}} - \bar{\mathbf{H}}^{\text{NF1}}_{\text{-340 ms}}, \tag{5}$$

where the bar indicates averaging over 8 movement directions. We orthogonalized the uniform shift with respect to $\hat{W}^{+}$. This allowed us to test for changes outside the force-predictive subspace.

We then projected the preparatory hidden unit activity (340 ms before the go-cue) of all experimental phases after removing the global mean. Next, we normalized the projection values for each model so that the projection of $\mathbf{H}^{\text{NF1}}$-340 ms onto the uniform shift is zero, and the projection of $\mathbf{H}^{\text{FF1}}$-340 ms onto the uniform shift is one.

4.7 Perturbing hidden unit activity

To conduct a causal test of the idea that the non-zero uniform shift activity that remained following NF2 is related to savings, we perturbed the activity of hidden units at the end of the preparatory period by adding to each hidden unit a proportion $(-2, -1, 0, 1, 2)$ of the projection of that unit onto the uniform shift. We conducted these perturbations separately for each movement direction. The perturbation was applied 340 ms prior to the go cue and the duration of the perturbation was one simulation time step.

5 Grants

This work was supported by the Natural Sciences and Engineering Research Council of Canada through a Discovery Grant RGPIN/05458-2018 to P.L.G., and Discovery Grant RGPIN-2022-04421 to J.A.P. J.A.M. was partially supported by a Banting Postdoctoral Fellowship and a BrainsCAN Postdoctoral Fellowship.

6 Code Availability

Python code to reproduce the simulations and analyses described here is available on GitHub at the following repository: `https://github.com/mshahbazi1997/MotorSavingModel.git`

7 Disclosures

No conflicts of interest, financial, or otherwise, are declared by the author(s).

8 Author contributions

M.S., O.C., J.A.M., and P.L.G. conception and design of research; M.S. performed simulations; M.S. and P.L.G. analyzed data; M.S., P.L.G., and J.A.M. interpreted results of experiments; M.S. prepared figures; M.S., P.L.G., and J.A.M. drafted manuscript; M.S., P.L.G., and J.A.M. edited and revised manuscript; M.S., O.C.,P.L.G., and J.A.M. approved final version of manuscript.

9 Acknowledgements

The authors wish to thank Mehrdad Kashefi for useful discussions about this study.

References

[1] A. M. Haith, D. M. Huberdeau, and J. W. Krakauer. The influence of movement preparation time on the expression of visuomotor learning and savings. *J. Neurosci.*, 35(13):5109–5117, 2015. DOI: 10.1523/jneurosci.3869-14.2015.

[2] J. R. Morehead, S. E. Qasim, M. J. Crossley, and R. Ivry. Savings upon re-aiming in visuomotor adaptation. *J. Neurosci.*, 35(42):14386–14396, 2015. DOI: 10.1523/jneurosci.1046-15.2015.

[3] L.-A. Leow, A. de Rugy, W. Marinovic, S. Riek, and T. J. Carroll. Savings for visuomotor adaptation require prior history of error, not prior repetition of successful actions. *J. Neurophysiol.*, 116(4):1603–1614, 2016. DOI: 10.1152/jn.01055.2015.

[4] K. P. Nguyen, W. Zhou, E. McKenna, K. Colucci-Chang, L. C. J. Bray, E. A. Hosseini, L. Alhussein, M. Rezazad, and W. M. Joiner. The 24-h savings of adaptation to novel movement dynamics initially reflects the recall of previous performance. *J. Neurophysiol.*, 122(3):933–946, 2019. DOI: 10.1152/jn.00569.2018.

[5] C. Yin and K. Wei. Savings in sensorimotor adaptation without explicit strategy. *J. Neurophysiol.*, 2020. DOI: 10.1152/jn.00524.2019.

[6] A. M. Hadjiosif, J. R. Morehead, and M. A. Smith. A double dissociation between savings and long-term memory in motor learning. *PLoS Biol.*, 21(4):e3001799, 2022. DOI: 10.1371/journal.pbio.3001799.

[7] S. K. Coltman, J. G. A. Cashaback, and P. L. Gribble. Both fast and slow learning processes contribute to savings following sensorimotor adaptation. *J. Neurophysiol.*, 121(4):1575–1583, 2019. DOI: 10.1152/jn.00794.2018.

[8] D. J. Herzfeld, P. A. Vaswani, M. K. Marko, and R. Shadmehr. A memory of errors in sensorimotor learning. *Science*, 345(6202):1349–1353, 2014. DOI: 10.1126/science.1253138.

[9] J. B. Heald, M. Lengyel, and D. M. Wolpert. Contextual inference underlies the learning of sensorimotor repertoires. *Nature*, 600(7889):489–493, 2021. DOI: 10.1038/s41586-021-04129-3.

[10] E. Zarahn, G. D. Weston, J. Liang, P. Mazzoni, and J. W. Krakauer. Explaining savings for visuomotor adaptation: linear time-invariant state-space models are not sufficient. *J. Neurophysiol.*, 100(5):2537–2548, 2008. DOI: 10.1152/jn.90529.2008.

[11] S. D. McDougle, K. M. Bond, and J. A. Taylor. Explicit and implicit processes constitute the fast and slow processes of sensorimotor learning. *J. Neurosci.*, 35(26):9568–9579, 2015. DOI: 10.1523/jneurosci.5061-14.2015.

[12] S. T. Albert and R. Shadmehr. Estimating properties of the fast and slow adaptive processes during sensorimotor adaptation. *J. Neurophysiol.*, 119(4):1367–1393, 2018. DOI: 10.1152/jn.00197.2017.

[13] V. S. Huang, A. Haith, P. Mazzoni, and J. W. Krakauer. Rethinking motor learning and savings in adaptation paradigms: model-free memory for successful actions combines with internal models. *Neuron*, 70(4):787–801, 2011. DOI: 10.1016/j.neuron.2011.04.012.

[14] J. Diedrichsen, O. White, D. Newman, and N. Lally. Use-dependent and error-based learning of motor behaviors. *J. Neurosci.*, 30(15):5159–5166, 2010. DOI: 10.1523/jneurosci.5406-09.2010.

[15] D. M. Wolpert, Z. Ghahramani, and M. I. Jordan. An internal model for sensorimotor integration. *Science*, 269(5232):1880–1882, 1995. DOI: 10.1126/science.7569931.

[16] E. M. Trautmann, J. K. Hesse, G. M. Stine, R. Xia, S. Zhu, D. J. O'Shea, B. Karsh, J. Colonell, F. F. Lanfranchi, S. Vyas, A. Zimnik, N. A. Steinmann, D. A. Wagenaar, A. Andrei, C. M. Lopez, J. O'Callaghan, J. Putzeys, B. C. Raducanu, M. Welkenhuysen, M. Churchland, T. Moore, M. Shadlen, K. Shenoy, D. Tsao, B. Dutta, and T. Harris. Large-scale high-density brain-wide neural recording in nonhuman primates. *bioRxivorg*:2023.02.01.526664, 2023. DOI: 10.1101/2023.02.01.526664.

[17] D. Kobak, W. Brendel, C. Constantinidis, C. E. Feierstein, A. Kepecs, Z. F. Mainen, X.-L. Qi, R. Romo, N. Uchida, and C. K. Machens. Demixed principal component analysis of neural population data. *Elife*, 5:9424, 2016. DOI: 10.7554/eLife.10989.

[18] A. Dubreuil, A. Valente, M. Beiran, F. Mastrogiuseppe, and S. Ostojic. The role of population structure in computations through neural dynamics. *Nat. Neurosci.*, 25(6):783–794, 2022. DOI: 10.1038/s41593-022-01088-4.

[19] A. L. Smoulder, P. J. Marino, E. R. Oby, S. E. Snyder, H. Miyata, N. P. Pavlovsky, W. E. Bishop, B. M. Yu, S. M. Chase, and A. P. Batista. A neural basis of choking under pressure. *Neuron*, 0(0), 2024. DOI: 10.1016/j.neuron.2024.08.012.

[20] X. Sun, D. J. OShea, M. D. Golub, E. M. Trautmann, S. Vyas, S. I. Ryu, and K. V. Shenoy. Cortical preparatory activity indexes learned motor memories. *Nature*, 602(7896):274–279, 2022. DOI: 10.1038/s41586-021-04329-x.

[21] D. M. Losey, J. A. Hennig, E. R. Oby, M. D. Golub, P. T. Sadtler, K. M. Quick, S. I. Ryu, E. C. Tyler-Kabara, A. P. Batista, B. M. Yu, and S. M. Chase. Learning leaves a memory trace in motor cortex. *Current Biology*, 34(7):1519–1531.e4, 2024. ISSN: 0960-9822. DOI: 10.1016/j.cub.2024.03.003.

[22] O. Codol, J. A. Michaels, M. Kashefi, J. A. Pruszynski, and P. L. Gribble. MotorNet: a python toolbox for controlling differentiable biomechanical effectors with artificial neural networks. *eLife*, 2024. DOI: 10.7554/eLife.88591.

[23] D. A. Kistemaker, J. D. Wong, and P. L. Gribble. The central nervous system does not minimize energy cost in arm movements. *J. Neurophysiol.*, 104(6):2985–2994, 2010. DOI: 10.1152/jn.00483.2010.

[24] R. Shadmehr and F. A. Mussa-Ivaldi. Adaptive representation of dynamics during learning of a motor task. *J. Neurosci.*, 14(5 Pt 2):3208–3224, 1994. DOI: 10.1523/jneurosci.14-05-03208.1994.

[25] M. A. Conditt, F. Gandolfo, and F. A. Mussa-Ivaldi. The motor system does not learn the dynamics of the arm by rote memorization of past experience. *J. Neurophysiol.*, 78(1):554–560, 1997. DOI: 10.1152/jn.1997.78.1.554.

[26] D. P. Kingma and J. Ba. Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. DOI: 10.48550/arXiv.1412.6980.

[27] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: an imperative style, high-performance deep learning library. *arXiv [cs.LG]*, 2019.

[28] J. A. Michaels, S. Schaffelhofer, A. Agudelo-Toro, and H. Scherberger. A goal-driven modular neural network predicts parietofrontal neural dynamics during grasping. *Proc. Natl. Acad. Sci. U. S. A.*, 117(50):32124–32135, 2020. DOI: 10.1073/pnas.2005087117.

[29] D. Sussillo, M. M. Churchland, M. T. Kaufman, and K. V. Shenoy. A neural network that finds a naturalistic solution for the production of muscle activity. *Nat. Neurosci.*, 18(7):1025–1033, 2015. DOI: 10.1038/nn.4042.

[30] M. M. Churchland, J. P. Cunningham, M. T. Kaufman, J. D. Foster, P. Nuyujukian, S. I. Ryu, and K. V. Shenoy. Neural population dynamics during reaching. *Nature*, 487(7405):51–56, 2012. DOI: 10.1038/nature11129.

[31] M. M. Churchland and K. V. Shenoy. Preparatory activity and the expansive null-space. *Nat. Rev. Neurosci.*, 25(4):213–236, 2024. DOI: 10.1038/s41583-024-00796-z.

[32] L. Driscoll, K. Shenoy, and D. Sussillo. Flexible multitask computation in recurrent networks utilizes shared dynamical motifs. *Biorxiv*:2022–08, 2022. DOI: 10.1038/s41593-024-01668-6.

[33] J.-H. Kim, K. Daie, and N. Li. A combinatorial neural code for long-term motor memory. *Nature*, 637(8046):663–672, 2025. DOI: 10.1038/s41586-024-08193-3.

[34] S. Bernardi, M. K. Benna, M. Rigotti, J. Munuera, S. Fusi, and C. D. Salzman. The geometry of abstraction in the hippocampus and prefrontal cortex. *Cell*, 183(4):954–967.e21, 2020. DOI: 10.1016/j.cell.2020.09.031.

[35] D. M. Wolpert and M. Kawato. Multiple paired forward and inverse models for motor control. *Neural Netw.*, 11(7-8):1317–1329, 1998. DOI: 10.1016/s0893-6080(98)00066-5.

[36] M. Haruno, D. M. Wolpert, and M. Kawato. Mosaic model for sensorimotor learning and control. *Neural Comput.*, 13(10):2201–2220, 2001. DOI: 10.1162/089976601750541778.

[37] H. R. Sheahan, D. W. Franklin, and D. M. Wolpert. Motor planning, not execution, separates motor memories. en. *Neuron*, 92(4):773–779, November 2016. ISSN: 0896-6273.

[38] H. R. Sheahan, J. N. Ingram, G. M. Žalalytė, and D. M. Wolpert. Imagery of movements immediately following performance allows learning of motor skills that interfere. en. *Scientific reports*, 8(1):14330, September 2018. ISSN: 2045-2322.

[39] J. A. Michaels, M. Kashefi, J. Zheng, O. Codol, J. Weiler, R. Kersten, P. L. Gribble, J. Diedrichsen, and J. A. Pruszynski. Sensory expectations shape neural population dynamics in motor circuits. *bioRxiv*:2024.12.22.629295, 2024. DOI: 10.1101/2024.12.22.629295.

[40] R. Ajemian, A. D'Ausilio, H. Moorman, and E. Bizzi. Why professional athletes need a prolonged period of warm-up and other peculiarities of human motor learning. *J. Mot. Behav.*, 42(6):381–388, 2010. DOI: 10.1080/00222895.2010.528262.

[41] R. Ajemian, A. D'Ausilio, H. Moorman, and E. Bizzi. A theory for how sensorimotor skills are learned and retained in noisy and nonstationary neural circuits. *Proc. Natl. Acad. Sci. U. S. A.*, 110(52):E5078–87, 2013. DOI: 10.1073/pnas.1320116110.

[42] B. Feulner, M. G. Perich, L. E. Miller, C. Clopath, and J. A. Gallego. A neural implementation model of feedback-based motor learning. *Nat. Commun.*, 16(1):1–14, 2025. DOI: 10.1038/s41467-024-54738-5.

[43] D. M. Huberdeau, A. M. Haith, and J. W. Krakauer. Formation of a long-term memory for visuomotor adaptation following only a few trials of practice. *J. Neurophysiol.*, 114(2):969–977, 2015. DOI: 10.1152/jn.00369.2015.

[44] G. Avraham, J. Ryan Morehead, H. E. Kim, and R. B. Ivry. Re-exposure to a sensorimotor perturbation produces opposite effects on explicit and implicit learning processes. *bioRxiv*:2020.07.16.205609, 2020. DOI: 10.1101/2020.07.16.205609.

[45] S. K. Coltman, R. J. van Beers, W. P. Medendorp, and P. L. Gribble. Sensitivity to error during visuomotor adaptation is similarly modulated by abrupt, gradual, and random perturbation schedules. *J. Neurophysiol.*, 126(3):934–945, 2021. DOI: 10.1152/jn.00269.2021.

[46] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv [cs.CL]*, 2014. DOI: 10.3115/v1/D14-1179.

[47] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In Y. W. Teh and M. Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. Pmlr, 2010.

[48] W. Hu, L. Xiao, and J. Pennington. Provable benefit of orthogonal initialization in optimizing deep linear networks. *arXiv [cs.LG]*, 2020. DOI: 10.48550/arXiv.2001.05992.

[49] T. Flash and N. Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *J Neurosci*, 5(7):1688–1703, 1985. DOI: 10.1523/jneurosci.05-07-01688.1985.

[50] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In S. Dasgupta and D. McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 (3) of *Proceedings of Machine Learning Research*, pages 1139–1147, Atlanta, Georgia, USA. Pmlr, 2013.