BIOENGINEERING DEPARTMENT, IMPERIAL COLLEGE LONDON

Neural principles underlying motor learning and adaptation

Barbara Feulner

Thesis submitted for the degree of
Doctor of Philosophy

December 2022

I hereby declare that to the best of my knowledge all work presented in this thesis was performed by me or was referenced otherwise.

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial-No Derivatives 4.0 International Licence (CC BY-NC-ND). Under this licence, you may copy and redistribute the material in any medium or format on the condition that; you credit the author, do not use it for commercial purposes and do not distribute modified versions of the work. When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Abstract

Animals, and especially humans, can learn to flexibly adjust their movements to changing environments. The neural principles underlying this remarkable capability are still not fully understood. Among the most prominent brain regions controlling movement is primary motor cortex (M1). Adapted motor behaviour can be related to a change in neural activity within this region. Yet, the rules guiding this activity change, and thus behavioural adaptation, remain unclear. The overall aim of this thesis is to investigate the learning process(es) governing the described change in activity in M1 and, with that, the change in behaviour. Computational modelling is used to study three specific aspects of learning:

1. What constrains learning to favour some neural activity patterns over others?
2. Can we identify where in a hierarchical pathway learning is happening?
3. How can sensory feedback guide the learning process?

We start by investigating *what* kind of biological constraints differentially affect learning of new neural activity that either preserves coactivation patterns between neurons (within-manifold learning), or requires learning of new coactivation patterns (outside-manifold learning). We propose a new explanation - the learnability of feedback signals - for why within-manifold activity patterns can be easier learned than outside-manifold activity patterns. In the second part we develop a hierarchical model of the motor system to investigate whether we can derive *where* learning has happened from only measuring neural activity. Lastly, we investigate *how* the brain could implement a biologically plausible learning rule which allows it to correctly assign errors and update recurrent connectivity in a goal-driven manner.

Overall, our work offers new perspectives on the role of M1 for motor learning and adaptation, challenges current beliefs, and puts a focus on the role of feedback signals for local plasticity in M1.

Acknowledgements

First and foremost I would like to thank my amazing supervisors Juan Alvaro Gallego and Claudia Clopath, who have guided me through the difficult parts, motivated me when I was loosing track, and helped me grow. I have learned so much from them during the last years, in professional as well as personal matters. Thank you!

I also want to thank my collaborators Matthew G. Perich, Raeed H. Chowdhury and Lee E. Miller, who not only shared their neural data with us, but also their wisdom and advice. Thanks goes also to all the people I talked to about my research on conferences, meetings and coffee tables. They gave me a lot of new ideas and inspiration, but also sometimes the critical input I needed.

Last but not least, I want to thank my partner and family for giving me endless support and allowing me to be open-minded to every new opportunity.

Contents

1 Introduction **11**
- 1.1 Motor control . . . 11
- 1.2 Neural modelling . . . 17
- 1.3 Learning and plasticity . . . 20
- 1.4 Aims . . . 24

2 Constraints on learning **26**
- 2.1 Abstract . . . 27
- 2.2 Introduction . . . 27
- 2.3 Results . . . 29
- 2.4 Discussion . . . 45
- 2.5 Methods . . . 48
- 2.6 Supporting information . . . 54

3 Learning location **76**
- 3.1 Abstract . . . 77
- 3.2 Introduction . . . 77
- 3.3 Results . . . 80
- 3.4 Discussion . . . 95
- 3.5 Methods . . . 99
- 3.6 Supporting information . . . 104

4 Biologically plausible learning **120**
- 4.1 Abstract . . . 121
- 4.2 Introduction . . . 121
- 4.3 Results . . . 123
- 4.4 Discussion . . . 133
- 4.5 Methods . . . 136
- 4.6 Supporting information . . . 142

5 Conclusion **144**
- 5.1 Summary . . . 144
- 5.2 Impact . . . 145
- 5.3 Future work . . . 146
- 5.4 Conclusion . . . 147

6 Bibliography **148**

List of Figures

1.1 Muscle activation . . . 12
1.2 Brain regions involved in motor control . . . 14
1.3 Two forms of control . . . 15
1.4 Combined feedback and feedforward control scheme . . . 16
1.5 Single neuron model . . . 18
1.6 From spikes to firing rates . . . 19
1.7 Synaptic transmission . . . 22
1.8 Two different approaches to learning in recurrent networks . . . 24
1.9 Thesis overview . . . 25

2.1 Recurrent neural network (RNN) performing a center-out-reach task . . . 30
2.2 RNN with correct feedback signal can learn within- and outside-manifold equally well 32
2.3 RNN with corrupted feedback signal or constraint on weight change can learn better within- than outside-manifold . . . 35
2.4 Learning the feedback signal is possible for within- but not for outside-manifold perturbations . . . 38
2.5 Feedback signal for outside-manifold perturbation can potentially be learned with incremental strategy . . . 43
2.S1 Control simulations . . . 54
2.S2 Details of initial network training . . . 55
2.S3 Simulation results for recurrent neural network trained with backpropagation-through-time using Pytorch . . . 56
2.S4 Weight change during retraining . . . 57
2.S5 Simulation with E-I network . . . 58
2.S6 Simulation with local learning rule . . . 59

2.S7 Mode correlation between before and after retraining 60
2.S8 Mode alignment during outside-manifold and within-manifold relearning 61
2.S9 Initial random decoder shapes neural manifold and final BCI mapping 62
2.S10 Feedback learning on mode level 63
2.S11 Broader Eigenvalue spectrum improves feedback learning performance for outside-manifold perturbations 64
2.S12 Inferring feedback weights is biased towards dominant neural modes 65
2.S13 Simulation results for within-manifold learning are in line with reassociation hypothesis 66
2.S14 Within-manifold contributions to incremental outside-manifold learning 67
2.S15 Difference to previous study investigating the experimental results of Sadtler et al. . . 68
2.S16 Timecourse of relearning is similar for within- and outside-manifold perturbation under the recursive-least-squares learning rule 69
2.S17 Learning feedback weights via a biological learning rule 70
2.S18 Relearning under noisy feedback signal using recursive-least-squares algorithm and local learning algorithm 71
2.S19 Simulation results are robust to how inputs are modelled 72
2.S20 Simulation results are robust to noise 73

3.1 Competing hypotheses to explain where learning happens during a visuomotor rotation task 80
3.2 Small but measurable changes in neural activity within PMd and M1 during VR adaptation 82
3.3 A modular recurrent neural network model to study VR adaptation 84
3.4 Activity changes following learning upstream (H_{input}) and within (H_{local}) the motor cortices 86
3.5 Small but coordinated connectivity changes enable motor adaptation 88
3.6 Low-dimensional connectivity changes are highly robust to noise 90
3.7 Larger visuomotor rotations allow for a better distinction between H_{input} and H_{local} . 92
3.8 A visuomotor reassociation task can clearly differentiate between H_{local} and learning through reassociation of input signals 94
3.S1 Model kinematics closely match mean target trajectories 104

3.S2 Model activity resembles neural data in an area-specific fashion . . . 105
3.S3 Actual and simulated activity changes within the neural manifold . . . 106
3.S4 Analysis of weight changes following VR adaptation under H_{input} . . . 107
3.S5 Analysis of weight changes following VR adaptation under H_{local} . . . 108
3.S6 Weight changes after initial training . . . 109
3.S7 Sensitivity of change in activity and covariance to magnitude of weight change . . . 110
3.S8 Adaptation to a visuomotor reassociation perturbation under H_{input} . . . 111
3.S9 Results are robust to parameter variations . . . 112
3.S10 Results are consistent for different plasticity combinations . . . 113
3.S11 Results are robust to variations in model architecture . . . 114
3.S12 Results are robust to variations in training algorithm . . . 115
3.S13 A modular spiking neural network replicates the main results obtained using a rate-based network . . . 116
3.S14 Analysis of weight changes for H_{all} . . . 117

4.1 Proposed recurrent neural network that controls its output based on feedback . . . 124
4.2 Feedback signals can guide local synaptic plasticity that enables successful motor adaptation . . . 125
4.3 Motor adaptation based on feedback-driven plasticity recapitulates key aspects of human and monkey behaviour . . . 127
4.4 Both feedback-based motor control and adaptation can be achieved with sparse feedback signals which are small compared to recurrent signals . . . 129
4.5 Two temporally dissociated activity changes may indicate feedback-driven plasticity in monkey M1 during visuomotor adaptation . . . 131
4.S1 Initial training protocol . . . 142
4.S2 Effective feedback-based motor control and adaptation can both be achieved with sparse feedback for varied degrees of recurrent connectivity . . . 143

List of Tables

2.1 List of parameters used in the model. 48

4.1 Simulation parameters. 137

List of Abbreviations

BCI	Brain-computer interface
M1	Primary motor cortex
PMd	Dorsal premotor cortex
RNN	Recurrent neural network
VR	Visuomotor rotation

CHAPTER 1

Introduction

"To move things is all that mankind can do, for such the sole executant is muscle, whether in whispering a syllable or in felling a forest."

*Charles Sherrington**

1.1 Motor control

Moving our body is one of the cornerstones of animal life on earth. Only through activation of different muscle fibres can we interact with the world around us. Given its importance for any living being, learning how to effectively drive goal-directed movement in our body through neural signals from our brain appears to be one of the largest accomplishments of evolution. The first attempts of goal-directed movement were naturally more elementary, meaning that the goal was merely to relocate - a necessity to escape predators or to collect food. Evolution has discovered a variety of solutions to this problem: from rhythmic contraction creating wave-like motion in limb-less worms[1], over common four-limb mammalian walking or galloping[2], to complex many-limb movement patterns of millipedes[3]. In this work we focus on the more modern, extremely sophisticated motor capabilities of higher mammals and especially humans to control their movements in a highly flexible and robust manner.

*Sherrington CS. 1979. 1924 Linacre lecture. In: JC Eccles, WC Gibson (eds). Sherrington: His Life and Thought, p. 59. New York: Springer-Verlag

Motor system

Coordinated neural activity in primary motor cortex (M1) gives rise to neural discharge in motor neurons, sitting in the spinal cord, which in turn are responsible to contract, or to shorten, muscle fibres (Figure 1.1). Those muscle fibres are bundled in hundreds of thousands to form muscles[4], which are connected to bones via tendons. Eventually, movement is initiated by a differential activation of multiple muscles, which can pull at joints in various angles, depending on the degrees of freedom of the specific joint. The richness of human motor repertoires is coming from over 600 skeletal muscles[5], which, when correctly controlled, can give rise to highly dexterous motor sequences such as the dance of a prima ballerina, or the play of a great pianist. The major challenge of motor control sending the sufficient signals to achieve high-level goals such as e.g., "reach for a glass of water".

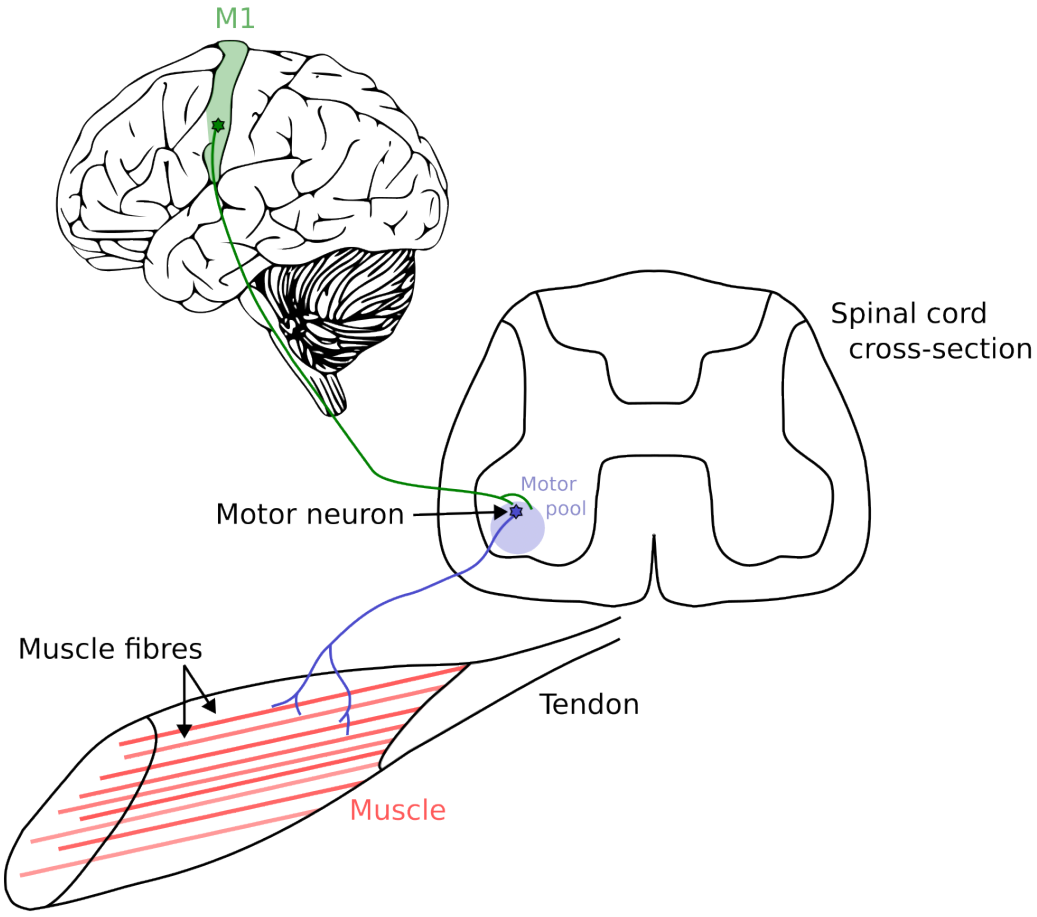


Figure 1.1: Muscle activation.

Controlling all these different knobs seems an impossible task yet research from the past 100 years, starting with Sherrington[6] among others, has begun to unravel the mysterious ways of how this elaborate control is achieved.

Primary motor cortex (M1) is the first and most important cortical region relevant for motor control, as it directly controls motor neurons in the spinal cord, which in turn control muscle fibres (Figure 1.1). Besides M1, there are two additional motor cortical areas: dorsal premotor cortex (PMd) and supplementary motor area (SMA), which both project to M1 neurons (Figure 1.2). Although they are generally thought to be more involved in motor planning than execution, it remains difficult to attach any single, specific function to any one of those areas. There are two main sensory feedback pathways that eventually reach M1. Visual feedback is mostly coming from PMd, which receives inputs from posterior parietal cortex (PPC), which in turn receives inputs from primary visual cortex (V1) (Figure 1.2). In contrast, proprioceptive feedback is conveyed more directly to M1 through primary somatosensory cortex (S1) (Figure 1.2). There are two sub-cortical regions that also play major roles for the control of movement: the basal ganglia (BG) and the cerebellum (C). Both regions form bidirectional feedback loops with M1 and can thereby indirectly modulate motor commands (Figure 1.2).

If we take a step back and consider what motor control actually entails - having a high-level goal, building knowledge about the body and its environment, taking various sources of sensory feedback, and combining all of this to eventually create sufficient motor commands - we see that basically the whole brain plays one role or another in this overarching computational effort. Yet, assigning any of those distinct computational functions to a specific brain area remains immensely hard[7].

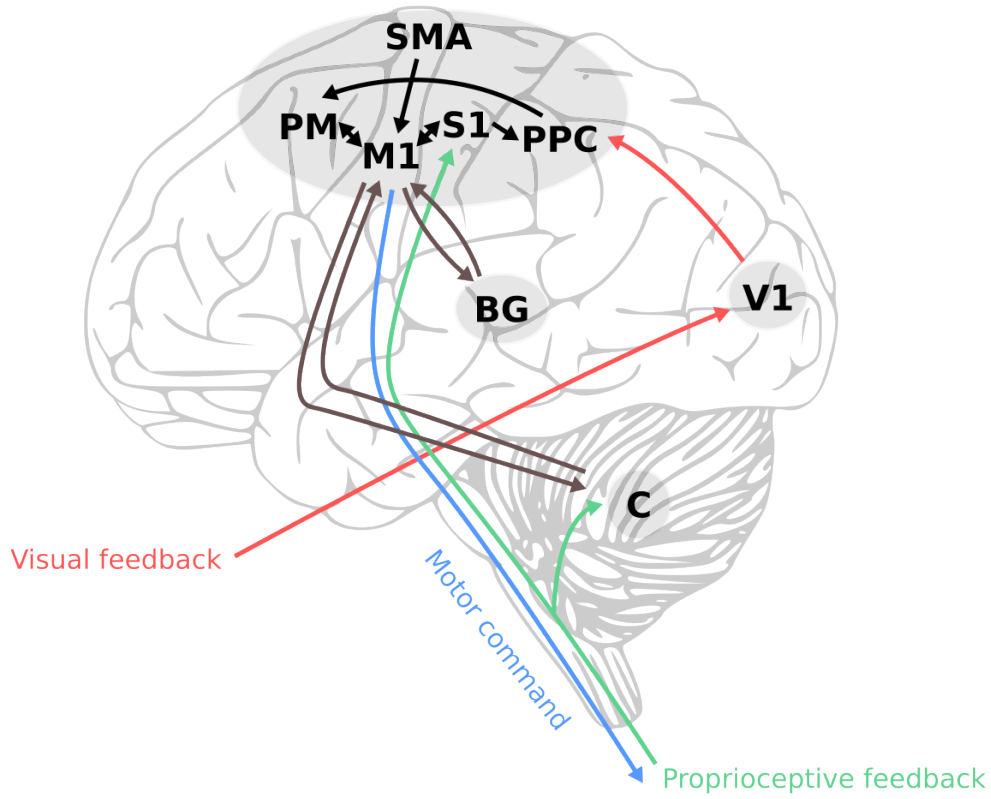


Figure 1.2: Brain regions involved in motor control[7–9]. M1, primary motor cortex; PM, premotor cortex; SMA, supplementary motor area; S1, primary somatosensory cortex; V1, primary visual cortex; PPC, posterior parietal cortex; BG, basal ganglia; C, cerebellum.

Control theory

One enormously powerful approach to study motor behaviour is control theory, which has been adopted from the field of robotics. A control problem consists of a controller and a plant, or an object/system to be controlled by the controller. The task of the controller is to send control signals to the plant in order to achieve a certain goal. In the context of motor control the brain can be seen as the controller, and the body, and its muscles, can be seen as the plant. An example goal could be reaching for a glass of water and taking a sip. Such a high-level goal is formed in more cognitive areas of the brain e.g., prefrontal cortex[10,11]. How to transform such high-level goals into a concrete activation plan for the hundreds of muscles in our body is the main task of the motor system.

There are two fundamentally different forms of control: open- and closed-loop (Figure 1.3). In *open-loop* control the controller acts according to a so-called feedforward plan - a direct transformation

from the desired output e.g., reaching the index finger from the key H to the key N, to the necessary motor signal, meaning the neural activity in M1. Such a transformation, once learned, would lead to accurate control, as long as the body and the environment is unchanged. However, as soon as there is a perturbation or an unforeseen factor e.g., a bigger keyboard, the formerly learned feedforward plan would fail. To still be able to sufficiently control the body in those scenarios *closed-loop* control takes into account the actual output, which might be perturbed, and corrects its original feedforward plan to account for the divergence.

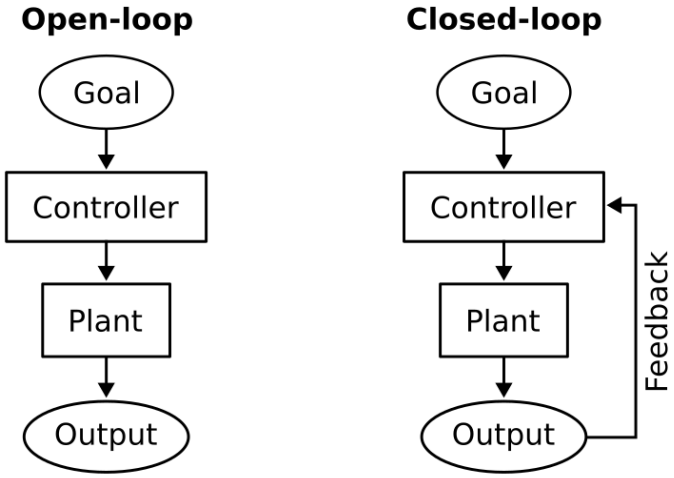


Figure 1.3: Two forms of control.

In a perfect system, without noise and delays, closed-loop control, also often referred to as feedback control, would be sufficient to control with high accuracy even in unpredictable circumstances. However, the human body is not this perfect system. The electrical signals carrying the information through the brain and the body can only be so fast, despite having invented smart tricks to accelerate signal transmission (e.g., myelination[12]). It takes around 12ms for an electrical pulse to travel from M1 via motor neurons in the spinal cord to muscle fibres in the forelimb[13]. For sensory feedback to reach back to motor cortex it takes ∼60ms for somatosensory[8,14] and ∼120ms for visual feedback[15,16]. Theory predicts and experiments show that such delayed feedback causes control to become unstable[17–20]. Open-loop control in contrast does not suffer from the time delays in the feedback, as it is completely independent. Therefore, a combination of open-loop (feedforward) and closed-loop (feedback) control is desirable[19].

One notion often used in this context of combined control schemes is that the open-loop part is denoted as "inverse model", as it basically reflects an inversion of the input-output transformation of the plant. How to learn such an inverse model to begin with is not straight-forward, as the observed errors are given in the "coordinate system" of the inputs, not the outputs. Kawato and colleagues[21,22] have proposed a "cerebellar feedback-error-learning model" (CBFELM) to resolve this problem (Figure 1.4). The basic idea is that the feedback controller acts as a teacher for the inverse model, sending it the motor command error, such that it can update its model predictions accordingly.

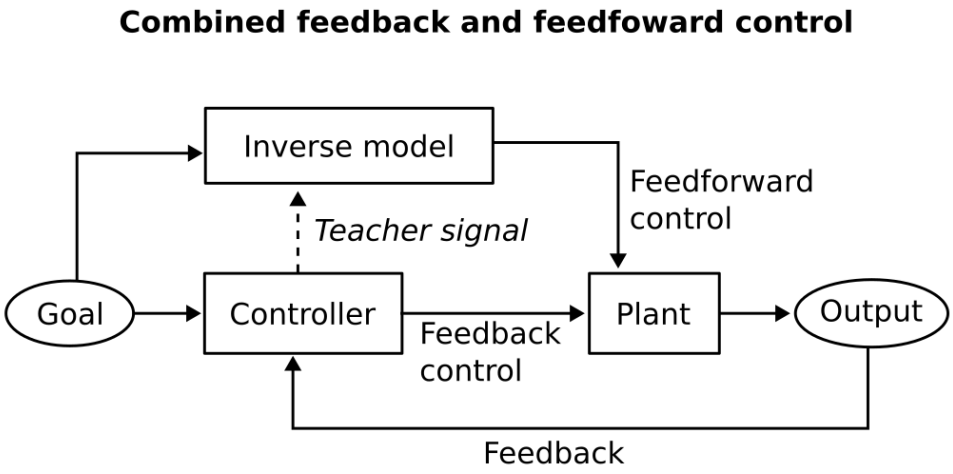


Figure 1.4: Combined feedback and feedforward control scheme[21].

Although there is a lot of intuition and partial experimental evidence that such an inverse model can be located in the cerebellum[7,18,21], especially for more rudimentary motor behaviour such as eye movements[23,24], it remains unclear to which extent there is a one-to-one mapping between "boxes" in the computational framework and brain regions. In chapter 4 of this thesis we address this by showing that one can implement a combined forward and feedback control scheme in a single recurrent circuit.

Neural manifold

While control theory has been extremely successful in describing motor behaviour[19,25–30], as mentioned earlier, it often lacks relation to neural circuits[7]. On the other hand, when we start from a

bottom-up view, we see thousands and thousands of neurons firing in complex time-dependent activity patterns[31], painting a picture extremely hard to decipher. One breakthrough in recent years regarding this daunting challenge is the finding that populations of neurons share common patterns of coactivation[32–36]. Mathematically, this can be found by calculating the covariance across pairs of recorded neurons and performing an *Eigendecomposition*. This method extracts distinct modes, or patterns, and thus gives an effective number of independent variables - the neural manifold -, which can be used to describe the whole population[35]. Intriguingly, the number of effective variables, or the dimensionality of the population activity, is surprisingly low - a result found across brain regions, tasks, and species[34]. In chapter 2 of this thesis we investigate how those neural manifolds can be related to learning, and why "within-manifold" learning is easier than "outside-manifold" learning[33]. A second, related idea is to describe neural activity in motor cortex as a dynamical system[37–40]. As we can describe the activity by only a handful of effective variables, we can also describe their evolution through time by simple dynamical equations. In this way, static neural activation across a population can be for example understood as a "fixed point" (a term from dynamical systems theory). This framework has been immensely successful in linking evolving neural activity with underlying function for motor control[41–44], and even recently also with the control theory framework[30].

1.2 Neural modelling

The goal in this work is to come up with a bottom-up, minimal model which can help us understand the neural implementation of motor learning and adaptation across levels: from single neurons, to neural networks, to brain areas, and eventually to motor behaviour. To start with that massive task let us first try to understand how we can describe a single neuron's behaviour in simple, mathematical terms.

Single neurons

Each neuron consists of a cell body, also called soma, a dendritic tree where inputs from other neurons arrive, and an axon which sends the cell's output to other neurons (Figure 1.5A). The key feature of neurons in the brain is that they carry a voltage signal due to their semi-permeable cell membrane which causes an ion gradient between inside and outside of the cell. Once this gradient is above a certain threshold, the neuron sends an action potential, or "spike", which is an all-or-nothing

electrical pulse created through a transient positive feedback loop (Figure 1.5B). Such a spike can travel long distances of axon, up to one metre in humans, to arrive at downstream neurons and in turn affect their voltage via a synaptic connection (cf. Synaptic plasticity section).

To describe the voltage of a neuron over time we can make use of a simple, electric circuit: the RC-circuit (Figure 1.5C). Ions can not freely pass the cell membrane thereby creating an imbalance similar to charging a capacitor (with capacitance C). On the other hand, the membrane is not completely impenetrable, thus there is also a leak current described by a resistor (with resistance R). The resulting time-dependent voltage can be described with the following linear differential equation

$$C\frac{dv}{dt} = -\frac{v}{R} + I(t) \quad \text{or} \quad \tau_m \frac{dv}{dt} = -v + RI(t)$$

where v is the voltage and $\tau_m = RC$ the membrane time constant of the neuron. $I(t)$ describes any incoming current.

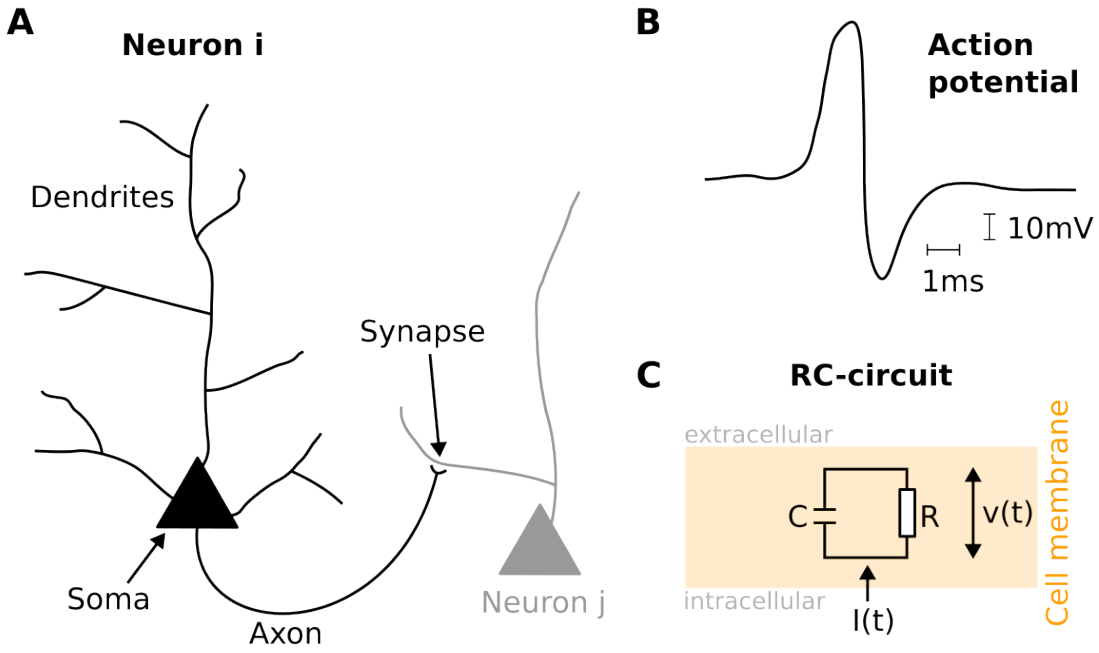


Figure 1.5: Single neuron model. **A.** Schematic of a neuron and its connection to another neuron. **B.** Typical shape of an action potential. **C.** Schematic of an RC-circuit used to model single neuron voltage.

As described earlier, neurons communicate in a stochastic, all-or-nothing manner via action potentials. Nevertheless, we can abstract this biological detail by not considering a single example of e.g., a trial in an experiment, but instead by repeating the same trial multiple times and calculating an average firing rate by summing up the spikes in an individual time bin over trials (Figure 1.6). This allows us to use so-called "rate-based" models (in comparison to "spike-based" models), which results in a similar dynamical equation, but this time, without any threshold or action potential mechanism included.

$$\tau_m \frac{dh}{dt} = -h + RI(t)$$

The dynamic variable h now describes the firing rate of the neuron. The input current $I(t)$ in a spiking model is usually given by a sum of delta functions $\sum_{t'} \delta(t - t')$, modelling the occurring incoming spikes at times t'. In a rate model the input $RI(t)$ is usually directly modelled as a function of the rate of the input neurons passed through a non-linearity $\Phi(h)$.

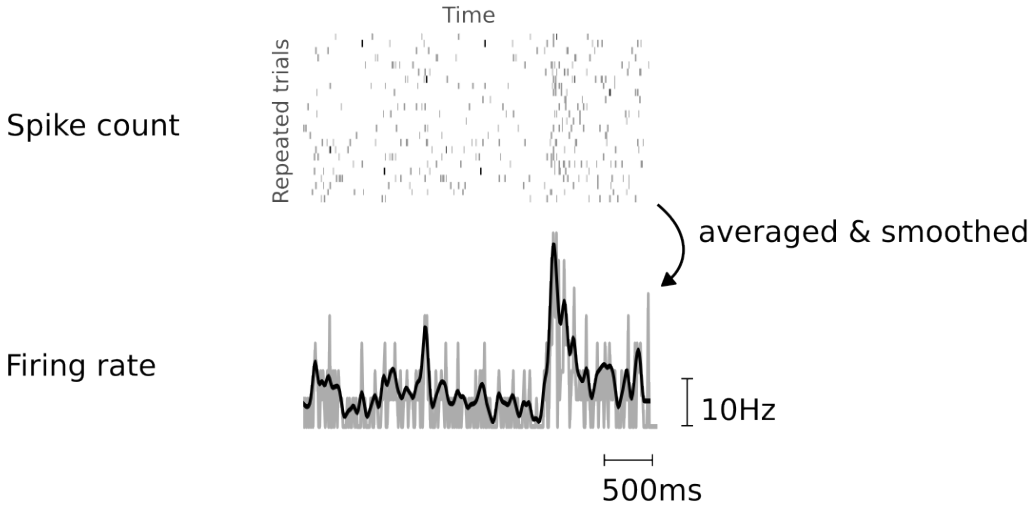


Figure 1.6: From spikes to firing rate.

Neural networks

For the next level of understanding we need to go from single neurons to networks of neurons. Key part in this endeavour will be to find an explanation for the often self-emergent properties of neural population dynamics. As briefly mentioned earlier (cf. Neural manifold section) dynamic systems theory can be extremely useful to understand those systems[43,44], especially to classify their overall

behaviour into e.g., "fixed points", "attractors", or "chaos"[45–48]. But what dictates those population dynamics? To mathematically describe such a population of neurons we can extend the single neuron firing rate formula to

$$\tau_m \frac{d\mathbf{h}}{dt} = -\mathbf{h} + \mathbf{W}\Phi(\mathbf{h})$$

where $\mathbf{h}$ is now a vector which describes the firing rates of all neurons in the population. How neurons interact with each other is described by the connectivity matrix $\mathbf{W}$, as each non-zero entry stands for a synaptic connection between a neuron i and a neuron j with the effective weight w_{ij}. Despite seminal work on the relation between connectivity and population dynamics[49–56], many details, often outside of the formalised theoretical frameworks, still remain unclear, especially in the context of learning. This is the overall topic of this thesis: pushing what we know about how changes of the connectivity matrix can be related to changes in population dynamics and eventually to changes in behaviour. We will use a "computational" approach, which consists of network simulations paralleled by experimental analyses whenever possible. We believe that this approach is viable as it allows us to bridge the gap between what we want to know and what we can directly probe experimentally[57]. Similar approaches have been successful not only in the motor domain[42,58–61], but also in more cognitive tasks[62,63], sensory processing[64], and memory formation[65,66].

1.3 Learning and plasticity

Motor learning and adaptation

> "Motor adaptation refers to a particular type of behavioral change that involves adjusting how an already well-practiced action is executed to maintain performance in response to a change in the environment or the body, either by selecting an alternative well-practiced action or modifying how the current action is executed."[67]

First of all, what kind of movements are we interested in? In this thesis we consider voluntary movements, in contrast to reflexes or locomotion. Key characteristics are that they are intentionally initiated, not triggered by external stimuli, and goal-driven (can be short- or long-term goals). Voluntary movements represent a complex interaction between cognitive aspects and pure motor functionality. Thus, in motor learning and adaptation experiments there is a cognitive, often referred

to as "explicit", and a more fundamental motor part, often referred to as "implicit"[68,69].

A commonly studied adaptation task, used in chapter 3 and 4 of this thesis, is a visuomotor rotation (VR) perturbation[70]. Users, either humans or monkeys, manipulate a joystick to control a cursor on the screen in front of them. Through this dissociation between movement (hand movement with the joystick) and visual feedback (cursor on the screen), the experimenter has the ability to manipulate the natural relation between both. In a commonly used centre-out-reaching task one can for example rotate the cursor movement by a certain degree - a VR perturbation. The user has to then correct for this perturbation by adjusting its movement accordingly. This type of learning usually happens on the order of a few dozen trials, or between thirty minutes and one hour[70], depending on the specifics of the experimental setup. This type of learning is potentially very different from long-term skill learning, such as learning to play the piano, requiring hours and hours of repeated practice[67].

In contrast, a completely different form of learning is used in chapter 2 of this thesis. Learning is still triggered by an external perturbation, yet the complex, potentially also plastic, part of the motor system downstream of M1 is completely ignored. How is this achieved? Sadtler et al.[33] used a brain-computer interface (BCI) to control the cursor on the screen not through the hand movement of the monkey, but directly through its neural activity in M1. This allowed them to enforce new activity patterns in M1, required for good control of the cursor. Despite the great insights they gained about which activity patterns can be easily learned and which not, how those patterns relate to natural learning behaviour remains unclear.

Synaptic plasticity

As already mentioned in the section about neural networks, synaptic connectivity is one of the main determinants of neural population dynamics. Thus, to learn new motor behaviours, or new neural activity patterns, one has to rewire the underlying neural circuits. In biological neurons those synaptic connections are implemented through a chemical process in which neurotransmitters are released from the pre-synaptic neuron and travel through the synaptic cleft to the post-synaptic neuron, where they cause a change in electrical potential in the post-neuron by binding to specific receptors (Figure 1.7).

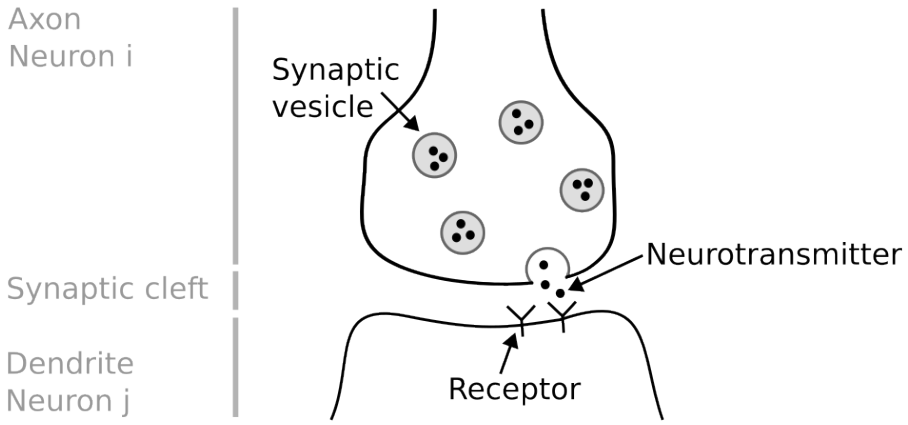


Figure 1.7: Synaptic transmission.

Interestingly, this process of synaptic transmission is not static, but plastic, meaning that the effective weight can change over time. There is a zoo of plastic mechanisms[71–73]: more short-term[74–76], more long-term[77–79], pre-synaptic[80], homeostatic[81], or structural plasticity[82–84]. All of them act on the effective weight of the synapse, with largely varying timescales between milliseconds to even weeks[81]. Despite this zoo of detailed biological mechanisms, a first heuristic approach to understand how synaptic weights change over time dates back to classic neuroscience experiments done by Donald Hebb[85], who has studied how the effective connection between neuron A and neuron B changes over time. They found that the connection strengthens when neuron A and neuron B fire at the same time, so with similar temporal patterns. Later, it has been found that the exact timing of even a single spike determines how a connection changes, which led to the notion of "spike-time-dependent plasticity"[78]. Despite being able to successfully model those biological plasticity mechanisms on a network level[73], it remains extremely difficult to use them to "learn" something useful, meaning to learn in a supervised manner, as we usually do when we train neural networks in machine learning[86]. One approach to tackle this is to introduce a third learning signal, besides the activity of the pre- and the post-synaptic neuron, which is usually some form of reward or error signal[87]. In chapter 4 of this thesis we investigate a related learning rule, where the "teacher" signal is a direct input to the neurons, thus leading the post-synaptic activity to directly represent the error.

Supervised learning

Though synaptic plasticity is how the brain eventually changes its connectivity and "learns", it is difficult to use these learning rules to train networks to do more complex tasks[86]. Thus, a more common approach to train simulated neural networks is to use some form of supervised learning rule, which abstracts away the biological detail of real biological synapses. Instead, the weights in the network are changed according to some error signal.

There are generally two different options of how to use recurrent neural networks to do something useful: 1) reservoir-style learning (Figure 1.8 left) and 2) full recurrent learning (Figure 1.8 right). In the former, the recurrent connectivity is set up in a dynamic state which produces complex time-varying patterns of activity. They can then be used to construct the required outputs by training only the decoding weights $\mathbf{W^{out}}$ (Figure 1.8 left, red arrows). This is in general easier as the correct weight update for the decoding weights can be directly calculated

$$d\mathbf{W^{out}} = -\alpha\ \epsilon\ \mathbf{h}$$

where α is the learning rate, ϵ the error in the produced output, and $\mathbf{h}$ the recurrent network activity. An example of this type of supervised learning is the classic reservoir computing approach[88], as well as the more recent FORCE algorithm[89,90].

The latter of the two options is more difficult to solve, as there is the problem of correct error assignment over time. If a recurrent weight is changing it does not only have a direct impact on the output, but it can also change the unfolding time evolution of recurrent dynamics. In machine learning, this problem is usually tackled by using "backpropagation-through-time" (BPTT); an extension to the seminal work of Rumelhart et al. developing the original idea of "error backpropagation" to train neural networks[91]. It requires to not only simulate the network dynamics "forward in time", but also to run a so-called "backward pass", where we calculate and accumulate the gradients of each network parameter with respect to the output error. It is still unclear why we can successfully "learn" with such a complex and deep neural network as the brain, given all its biological limitations, as BPTT - the golden standard in modern machine learning - can not be implemented in such circuits[86]. To tackle this open question, researchers have started to work on approximations to BPTT, taking into account different biological constraints at a time[92–94]. Chapter 4 of this thesis adds to this literature as

it shows how to use dynamic feedback currents, used for motor control, to correctly update recurrent connectivity in a learning task.

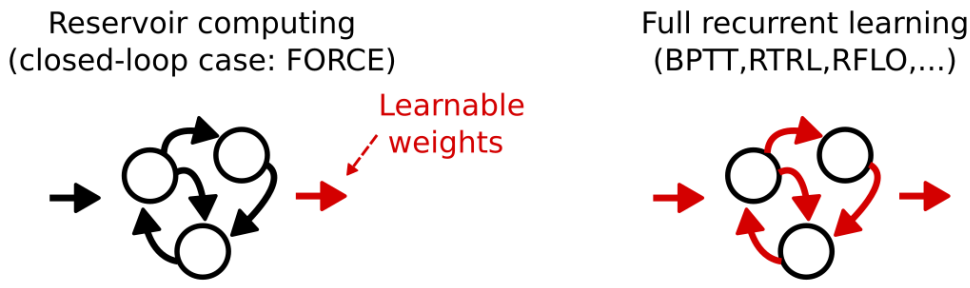


Figure 1.8: Two different approaches to learning in recurrent networks. Reservoir computing[88,95], FORCE[89], backpropagation-through-time (BPTT)[91], RTRL[92], RFLO[93].

1.4 Aims

The purpose of this thesis is to study different aspects of learning, especially trying to link observed learning behaviour with its neural implementation. Recurrent neural networks are used as a model system in which different hypotheses about learning are tested. The thesis is divided into three different self-contained chapters, each of them tackling a distinct sub-problem: *what* can we learn easily (Chapter 2), *where* in the brain do we learn (Chapter 3), and *how* do we learn (Chapter 4). All of these questions are asked in the context of motor adaptation or motor learning.

Overview of the work (Figure 1.9):

Chapter 2: Constraints on learning[96]

Chapter 3: Learning location[97]

Chapter 4: Biologically plausible learning[98]

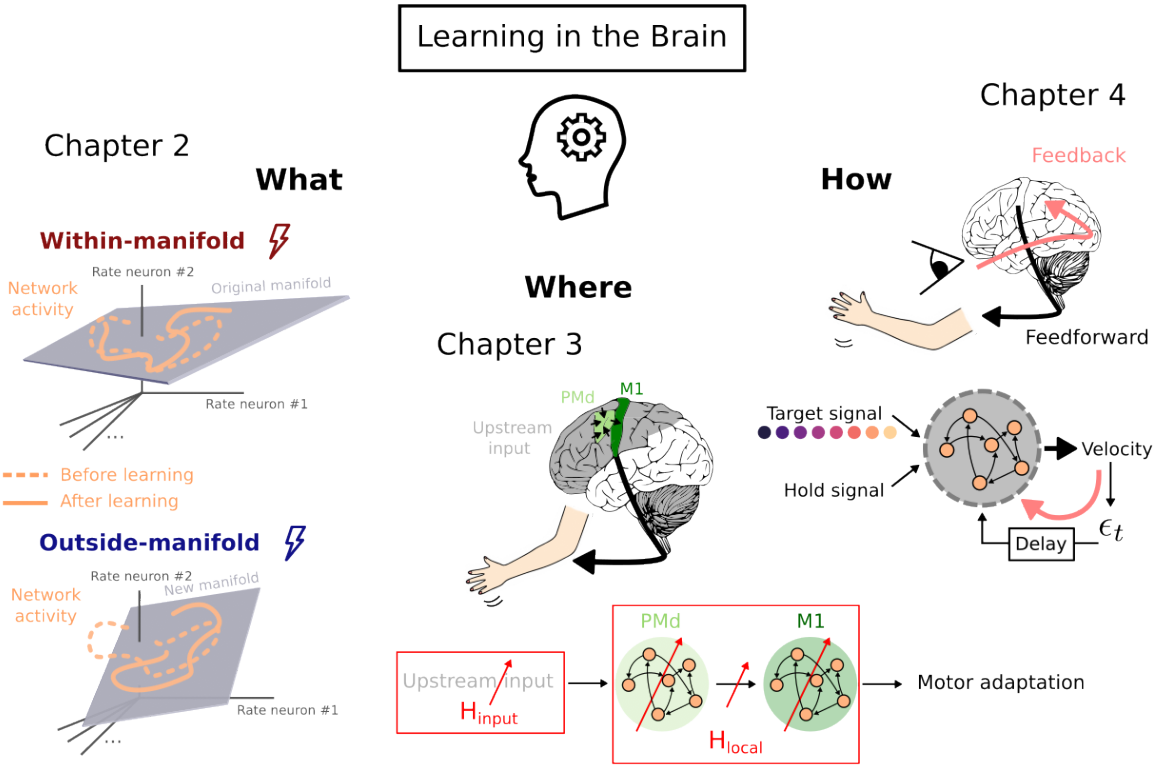


Figure 1.9: Thesis overview.

CHAPTER 2

Constraints on learning

We start out from a highly influential experimental result: Sadtler et al.[33] found that "within-manifold" learning is easier than "outside-manifold" learning. Using an elegant brain-computer interface (BCI) setup they could show that monkeys can learn more easily to adapt their neural activity in M1 when the required activity changes preserve the overall coactivation patterns between neurons. In this chapter we investigate the question of *why* this is the case. Firstly, we replicate the original BCI setup in a modelling framework allowing us to then selectively add and test different hypotheses for why within-manifold learning is easier than outside-manifold learning.

The work in this chapter is presented in an article format as published in PLOS Computational Biology in 2021[96]. I would like to thank my supervisor Claudia Clopath for her guidance and inspiration throughout this project.

Neural manifold under plasticity in a goal driven learning behaviour

2.1 Abstract

Neural activity is often low dimensional and dominated by only a few prominent neural covariation patterns. It has been hypothesised that these covariation patterns could form the building blocks used for fast and flexible motor control. Supporting this idea, recent experiments have shown that monkeys can learn to adapt their neural activity in motor cortex on a timescale of minutes, given that the change lies within the original low-dimensional subspace, also called neural manifold. However, the neural mechanism underlying this within-manifold adaptation remains unknown. Here, we show in a computational model that modification of recurrent weights, driven by a learned feedback signal, can account for the observed behavioural difference between within- and outside-manifold learning. Our findings give a new perspective, showing that recurrent weight changes do not necessarily lead to change in the neural manifold. On the contrary, successful learning is naturally constrained to a common subspace.

2.2 Introduction

The dynamics of single neurons within a given brain circuit are not independent, but highly correlated. Moreover, neural activity is often dominated by only a very low number of distinct correlation patterns[32,34,35,37,41,99–108]. This implies that there exists a low-dimensional manifold in the high-dimensional population activity space, to which most of the variance of the neural activity is confined. Why such low-dimensional dynamics is observed in the brain remains unclear. One potential reason

is that most experimental designs are inherently low-dimensional, and therefore bias the observed neural activity[109]. Alternatively, it has been shown that low-dimensional dynamics can arise from structured connectivity within the network[53–55,58,110–114].

The functional implications of low-dimensional neural dynamics are also unclear[115,116]. For example, preparatory and movement activity lie in orthogonal subspaces, which can serve as a gating mechanism for movement initiation[41,117]. Furthermore, the same manifold might underlie multiple skilled wrist and reach-to-grasp movements[36], as well as the same task over a long timescale[118]. These results suggest that there is one stable manifold for motor control, and that learning and performing different, but similar, movements are happening within this scaffold[35].

Using a brain-computer interface (BCI)[119] in monkeys, Sadtler et al. showed that it is possible to adapt neural activity in motor cortex, if the new activity pattern lies within the original manifold, but not if it lies outside of it[33]. They could test this by introducing two kinds of perturbations to a BCI, which was previously trained to decode two-dimensional cursor dynamics from neural activity. The monkeys had to adapt their neural activity in motor cortex to produce the correct cursor dynamics, given the perturbed BCI mapping. Interestingly, the timescale of within-manifold learning is in the order of minutes to hours[33]. In contrast, outside-manifold learning is not possible within a single session, but instead requires progressive training over multiple days[120]. Due to the timescale difference, it has been hypothesized that within-manifold learning is related to fast motor adaption and outside-manifold learning is related to skill learning. Yet, the underlying neural mechanisms for both types of learning remain unknown.

Previously, two computational studies aimed at uncovering the mechanisms underlying the constraints on learning in the described experiment. Waernberg and Kumar used a simplified implementation of the BCI task to compare the amount of weight change which is required for within- and outside-manifold learning. They found that within-manifold learning requires less weight change in their simulations and thereby concluded that within-manifold learning is faster because of smaller underlying synaptic changes[121]. In contrast, Menendez and Latham investigated the possibility that learning results from a change in input to motor cortex[122]. Although there have been these two attempts to explain the experimental observations in Sadtler et al., the search for underlying neural mechanisms remains inconclusive.

We used computational modelling to study the relationship between neural manifold and learning.

As motor learning can drive network rewiring in motor cortex on a short timescale[123–126], we wanted to test whether local network rewiring can account for within-manifold, as well as outside-manifold learning. To that end, we implemented an in-silico version of the BCI experiment in Sadtler et al., where motor cortex activity is simulated by a recurrent neural network (RNN)[53,110,113]. We showed that an ideal observer training algorithm is able to learn within- and outside-manifold perturbations equally well. One component of the algorithm is that errors in the produced cursor dynamics are translated to errors in single neuron activity. This form of feedback signal is used to adapt the local connections within the network in order to minimize the observed error. As the correct feedback signal is not a priori given in the biological system, we subsequently tested more realistic feedback scenarios. By disrupting the correct feedback signal, we found that within-manifold learning is more robust to erroneous or sparse feedback signals. Furthermore, learning the feedback signal from scratch was only possible for within-manifold perturbations. To enable feedback learning for outside-manifold perturbations we had to introduce an incremental strategy, which resembles the progressive training used in experiments[120].

2.3 Results

Recurrent neural network (RNN) performing a center-out-reach task

To investigate the potential difference between within- and outside-manifold learning, we implemented an in-silico version of a brain-computer interface (BCI) experiment previously done with monkeys[33]. Instead of measuring neural activity in monkey motor cortex, we implemented a recurrent neural network (RNN) which was trained to do a similar center-out-reach task as the monkey had to perform (Figure 2.1A). After the initial training phase, we perturbed the BCI mapping to either trigger a within- or outside-manifold change (Figure 2.1C and 2.1D). We then retrained the recurrent network. This pipeline allowed us to test to what extent reorganization in the local network can explain the behaviourally observed differences in within- and outside-manifold learning.

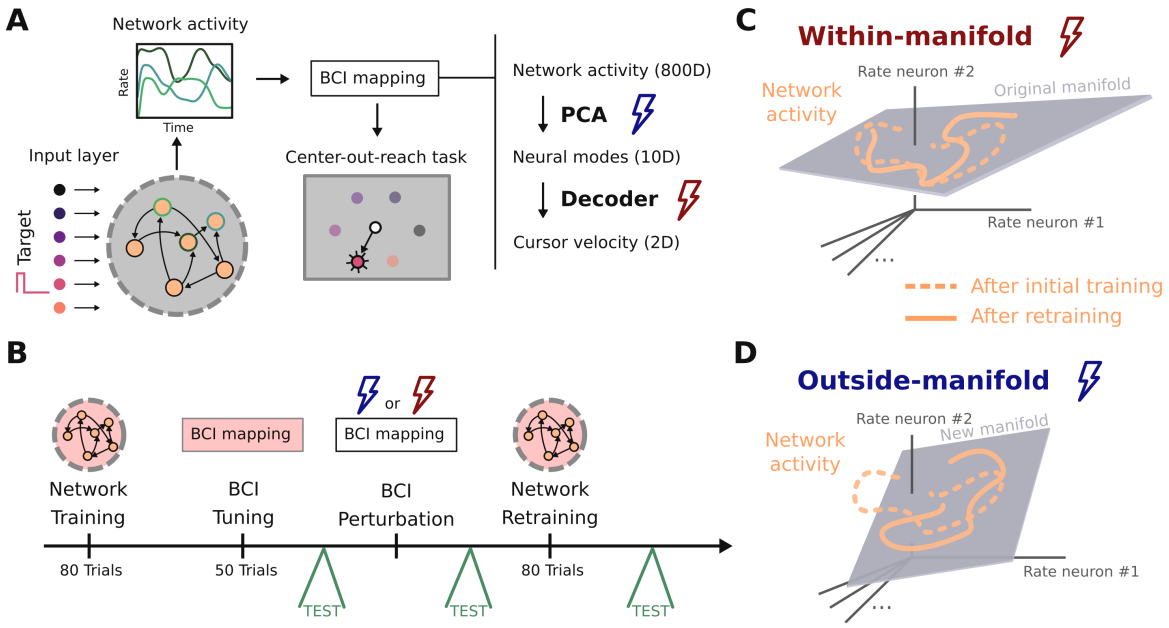


Figure 2.1: Recurrent neural network (RNN) performing a center-out-reach task. A. In order to study learning within and outside of the neural manifold, we used a brain-computer interface (BCI) setup similar to[33]. A recurrent neural network serves as generator of network activity which is then used via a BCI setup to control a cursor in a two-dimensional center-out-reach task. The key step in our simulation is to perturb the initial BCI mapping and let the recurrent network learn to compensate for that. **B.** The simulation design. **C-D.** In order to compensate the BCI perturbation the network needs to produce new dynamics. For a within-manifold perturbation (C) the new target activity is constrained to the same manifold as the original activity. For an outside-manifold perturbation (D) this is not the case, but the network activity has to explore new dimensions in state space.

RNN with correct feedback signal can learn within- and outside-manifold equally well

In order to understand constraints on within- and outside-manifold learning, we firstly simulated our network with an ideal-observer feedback signal. This means that the task error, which is defined as the difference between target and produced cursor velocity, is correctly assigned to errors on the activity of single neurons in the network. The correct credit assignment can be achieved by a matrix multiplication with the pseudo-inverse of the BCI readout. By doing this, we intentionally ignored the fact that a biological system would need to learn this credit assignment first, before it can be used for retraining. Rather, we first focused on the restructuring of the recurrent connections. With this training method, our network reached high performance in the center-out-reach task after the initial training phase (Figure 2.2A1). To probe within- versus outside-manifold learning, we then applied the corresponding perturbation to the BCI mapping, which led to impaired cursor

trajectories in both cases (Figure 2.2A2). Interestingly, using the same training method as during the initial training phase, we could retrain the network to adapt to the changed BCI mapping for within-manifold perturbation, as well as for outside-manifold perturbation. The cursor trajectories after the initial training phase and after the retraining phase looked similar for both types of perturbations (Figure 2.2A3). We quantified the performance by measuring the mean squared error between target and produced cursor velocities (Figure 2.2B). The average retraining performance confirms that within- and outside-manifold perturbations can equally well be counteracted and that this result is not constrained to a specific network initialization (Figure 2.2B), task setup (Figure 2.S1, Figure 2.S5, Figure 2.S19 and Figure 2.S20), or learning algorithm (Figure 2.S3 and Figure 2.S6). Comparing learning timescales did not show any difference for within- versus outside-manifold learning (Figure 2.S16), yet a more biologically plausible learning algorithm showed slightly slower learning for outside-manifold perturbations (Figure 2.S6). Together, these results indicate that there is no fundamental difference in learning new dynamics either within the original manifold or outside of it, as long as we focus solely on the learning of the recurrent weights.

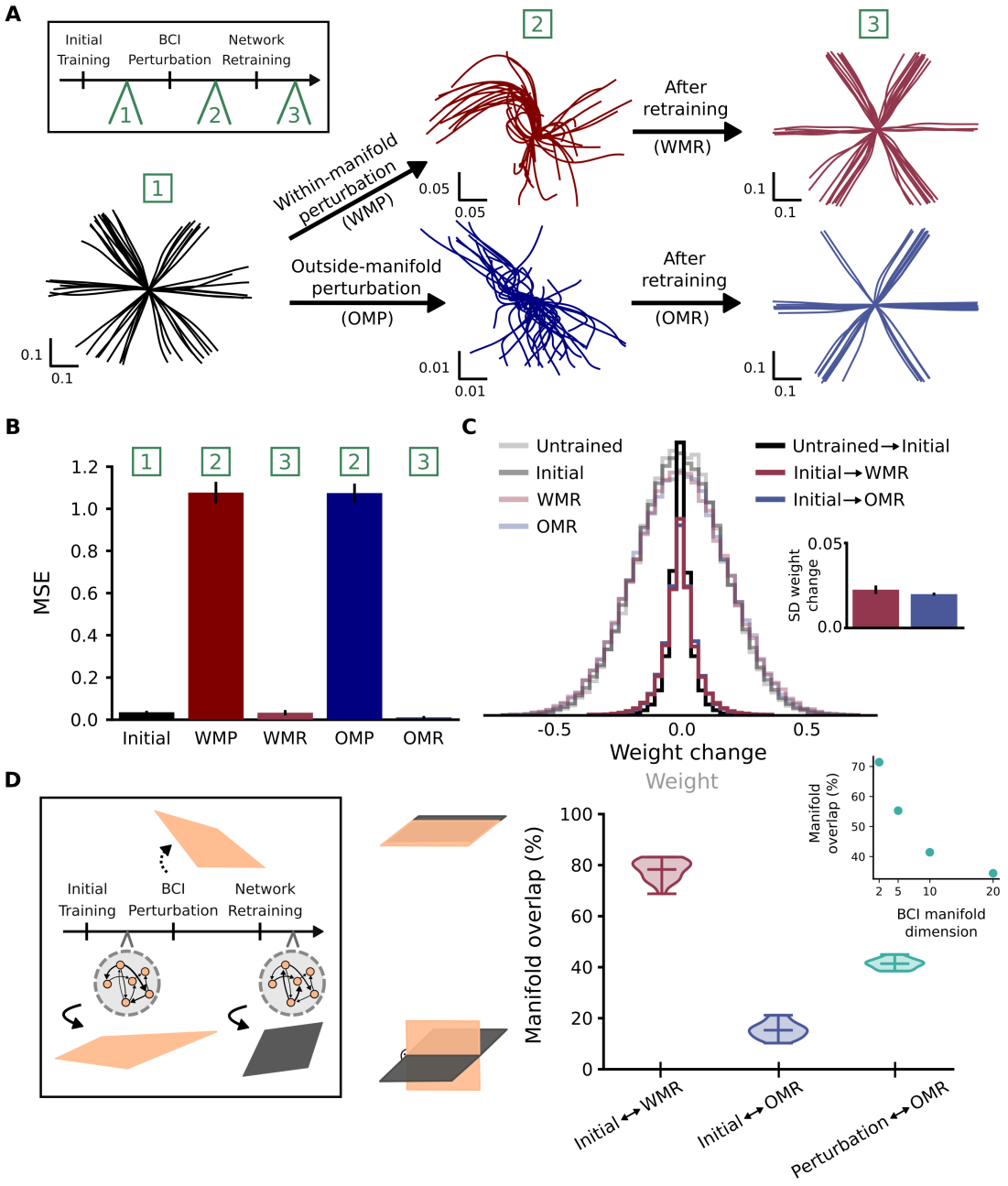


Figure 2.2: RNN with correct feedback signal can learn within- and outside-manifold equally well. A. Cursor trajectories after initial training phase (on the left), after within- (WMP) or outside-manifold (OMP) perturbation (in the middle) and after within (WMR) or outside (OMR) retraining phase (on the right) for one example simulation. **B.** Performance measured as mean squared error (MSE) between target cursor velocity and produced cursor velocity. Shown are the average simulation results for twenty randomly initialized networks, bars indicate the standard deviation across networks. **C.** Weight distribution and weight change distribution during the experiment. The inlet shows the average standard deviation of weight change, bars indicate standard deviation across networks. **D.** Manifold overlap between before and after retraining phase (red and blue), or between target manifold for outside-manifold perturbation and internal manifold after retraining phase (green). Inlet shows how the overlap between BCI target manifold and internal manifold depends on BCI manifold dimension.

Given biological constraints on the amount of affordable synaptic restructuring, we wanted to test whether there is a difference in weight change needed to compensate either within- or outside-manifold perturbations. To investigate this, we started with comparing the network connectivity before and after the retraining phase (Figure 2.2C). We found that there is no difference in the amount of weight change produced by either within- or outside-manifold retraining. In fact, the weight change distribution for both types of retraining is similar to the distribution obtained by comparing the weights before and after the initial training phase. In addition to the amount of weight change, we also compared the effective rank and dimensionality of weight change between within- and outside-manifold relearning (Figure 2.S4). We found that weight changes happening during outside-manifold learning are slightly higher dimensional compared to weight changes during within-manifold learning. These results show that retraining within or outside of the original manifold does not necessarily lead to a different amount of weight change, yet weight changes related to within-manifold learning are slightly less dimensional.

In order to understand how our network achieves high performance after the adaptation phase, we analysed the underlying dynamics before and after adaptation. To quantify the dynamical changes, we recalculated the internal manifold after the retraining phase and compared it to the original manifold (Figure 2.2D). For within-manifold relearning, we found that the internal manifolds before and after relearning have high overlap as expected (Figure 2.2D red). Interestingly, learning is not equally distributed across all ten modes in the manifold, but concentrated on the most prominent ones (Figure 2.S7). For outside-manifold, we computed two measures: 1) how much the internal manifold after retraining still overlaps with the original one (Figure 2.2D blue), and 2) how the internal manifold after retraining has aligned to the one defined by the BCI perturbation (Figure 2.2D green). We found that although the task performance is high after retraining the network following an outside-manifold perturbation, the network dynamics do not completely align with the new readout defined by the altered BCI mapping. More detailed analysis showed that outside-manifold learning focuses on the few most prominent modes, which have initially the highest decoder values (Figure 2.S8 and Figure 2.S9). This demonstrates that it is not necessary that network dynamics completely realign with the altered BCI mapping to perform well in the task. Instead it is enough to have a sufficient amount of overlap, or match the most prominent modes. The amount of necessary overlap likely depends on a combination of task dimensionality[109] and BCI manifold dimensionality

(Figure 2.2D inlet).

RNN with corrupted feedback signal or constraint on weight change can learn better within- than outside-manifold

Our results so far do not show a difference between within- and outside-manifold learning, in constrast to experimental observations[33]. To further investigate where this behaviourally observed difference potentially comes from, we next dropped the assumption of an ideal-observer feedback signal, which is biologically highly implausible, and tested the effect of different corrupted feedback versions on recurrent retraining. As neural signals are noisy and long-range projections are sparse, we tested the effect of a noisy, as well as sparse feedback signal on relearning performance. We started by analysing the effect of a noisy feedback signal (Figure 2.3B). We simulated our experiment with different strengths of noise corruption in the feedback signal and compared the task performance after retraining. Interestingly, we found a difference between within- and outside-manifold learning, as within-manifold learning shows for a certain intermediate noise level better retraining performance than outside-manifold learning (Figure 2.3B top). This difference between within- and outside-manifold learning holds when we simulated the BCI experiment using a different learning rule to modify recurrent weights (Figure 2.S18). Next, we investigated the effect of a sparse feedback signal (Figure 2.3C). In this scenario, not all neurons in the recurrent network receive a feedback signal and therefore participate in the recurrent restructuring. We again found that within-manifold learning is achieving better performance with the same portion of feedback signals available (Figure 2.3C top). Finally, we tested the effect of sparse plastic weights within the recurrent network (Figure 2.3D). Also there we found that within-manifold learning is preferential. All together, these results show that within-manifold learning outperforms outside-manifold learning in the presence of either erroneous feedback signals, sparse feedback signals, or if there is a constrained number of plastic connections within the network.

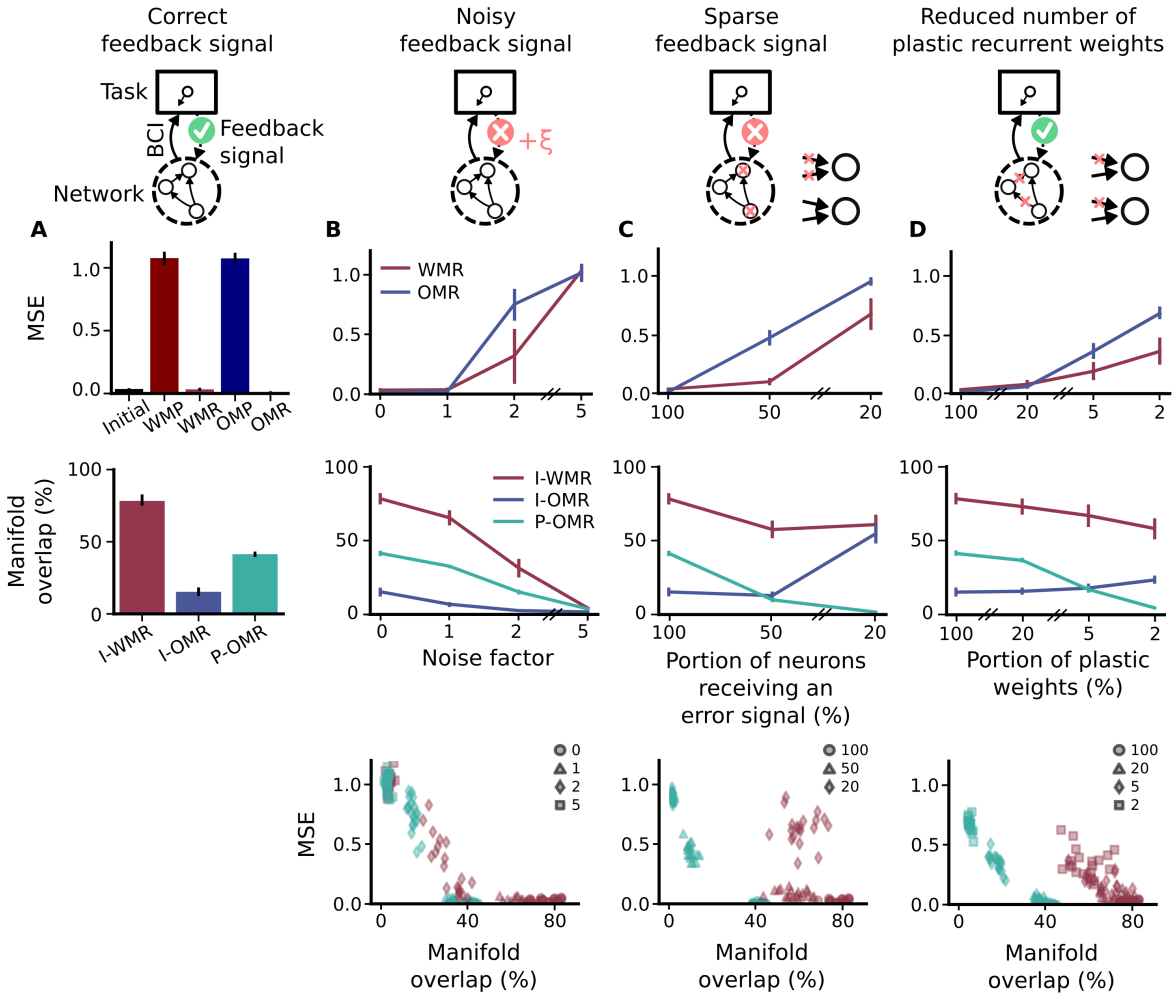


Figure 2.3: RNN with corrupted feedback signal or constraint on weight change can learn better within- than outside-manifold. A. Relearning results for RNN with correct feedback signal (same as in Figure 2.2), bars indicate here and in the following standard deviation across networks. WMP and OMP show performance after within- or outside-manifold perturbations, whereas WMR and OMR show performance after retraining of the respective BCI perturbations (top). Manifold overlap is measured between initial manifold and manifold after within learning (I-WMR), initial manifold and manifold after outside learning (I-OMR) and manifold defined by BCI perturbation and manifold after outside learning (P-OMR) (center). **B.** Relearning results for noisy feedback signal. The correct feedback transformation is distorted by adding independent noise to each entry of the matrix. The noise is drawn from a zero-mean Gaussian with standard deviation $\sigma = \alpha \cdot \sigma_{original}$, where α is the noise factor and $\sigma_{original}$ is the standard deviation of the correct feedback matrix. Top panel shows performance results, center panel shows manifold overlap and bottom panel shows the relation between both. **C.** Relearning results for sparse feedback signal. In this scenario not all neurons in the recurrent network receive a feedback signal, which leads to a portion of recurrent weights remaining static during retraining. **D.** Relearning results for sparse plastic connections. Similar to (C) a portion of recurrent weights remain static during retraining. But in contrast to (C), these static connections are not clustered to specific neurons. Instead, all neurons keep at least one plastic incoming connection.

Although the three different corrupted training scenarios show similar results with respect to

retraining performance, the specific kind of disturbance could yet be highly different. Therefore, we analysed in more detail the network dynamics before and after retraining. We compared how well the internal manifold after retraining aligns with either the initial manifold in the case of within-learning, or with the dictated manifold coming from the BCI perturbation in the case of outside-learning. Interestingly, we found that, for noise in the feedback signal, the network dynamics are driven off the target manifold (Figure 2.3B center). This makes sense as the feedback signal is disturbed in random directions. During the retraining phase, the network then tries to align its internal manifold to these random directions, resulting in reduced overlap to the correct manifold dimensions. We found that task performance is high as long as manifold overlap is more than 40% (Figure 2.3B bottom), which is in line with what we found for outside manifold learning with the correct feedback signal (Figure 2.2D). In contrast, reducing the number of plastic connections has a different effect on recurrent restructuring. It primarily disturbs the manifold alignment for outside-manifold perturbations (Figure 2.3C and 2.3D center). This is due to the fact that the static connections in the network constrain the dynamics to stay within the original manifold. Therefore, for within-manifold learning the manifold overlap between before and after retraining is always relatively high (Figure 2.3C and 2.3D center). Nevertheless, the task performance is reduced in these cases as within-manifold perturbations can not correctly be counteracted (Figure 2.3C and 2.3D top). Furthermore, we found that not only the ratio of plastic connections affects retraining, but also the level of disturbance. As long as each neuron in the network receives a feedback signal, retraining performance is relatively high even if only 20% of the connections are plastic (Figure 2.3D top). In summary, although manifold overlap and task performance correlate in some regimes (e.g. Figure 2.3D bottom), we can find cases with high task performance and low manifold overlap (outside-manifold learning with correct feedback signal), and we can find cases with low task performance and high manifold overlap (within-manifold learning with sparse plastic connections), showing that one does not necessarily imply the other.

Learning the feedback signal is possible for within- but not for outside-manifold perturbations

Although we have shown that within-manifold learning is more robust to corrupted feedback signals, the question remains of how the biological system could solve the error assignment and learn the

feedback weights. A very naive hypothesis would be that it can infer the feedback weights by simply knowing its own internal dynamics and regressing them against the observed cursor dynamics (Figure 2.4A). We tested this hypothesis and surprisingly found that this simple strategy gives good estimates of the feedback weights in the case of a within-manifold perturbation (Figure 2.4B), but fails completely in the case of an outside-manifold perturbation (Figure 2.4C). To quantify the quality of feedback learning, we calculated the correlation coefficient between the learned and the correct feedback weights. We found that the feedback learning performance is worse compared to inferring the feedback after the initial training phase for a within-manifold perturbation, but it is definitely better compared to an outside-manifold perturbation (Figure 2.4D). Furthermore, the estimate is almost as good after 6 trials as after 50 trials (Figure 2.4D). These results suggest that feedback learning is possible for within-manifold perturbations, at least from a computational point of view, and that observing only a few trials is in principle enough to obtain a good estimate. In contrast, the feedback weights can not be learned for an outside-manifold perturbation.

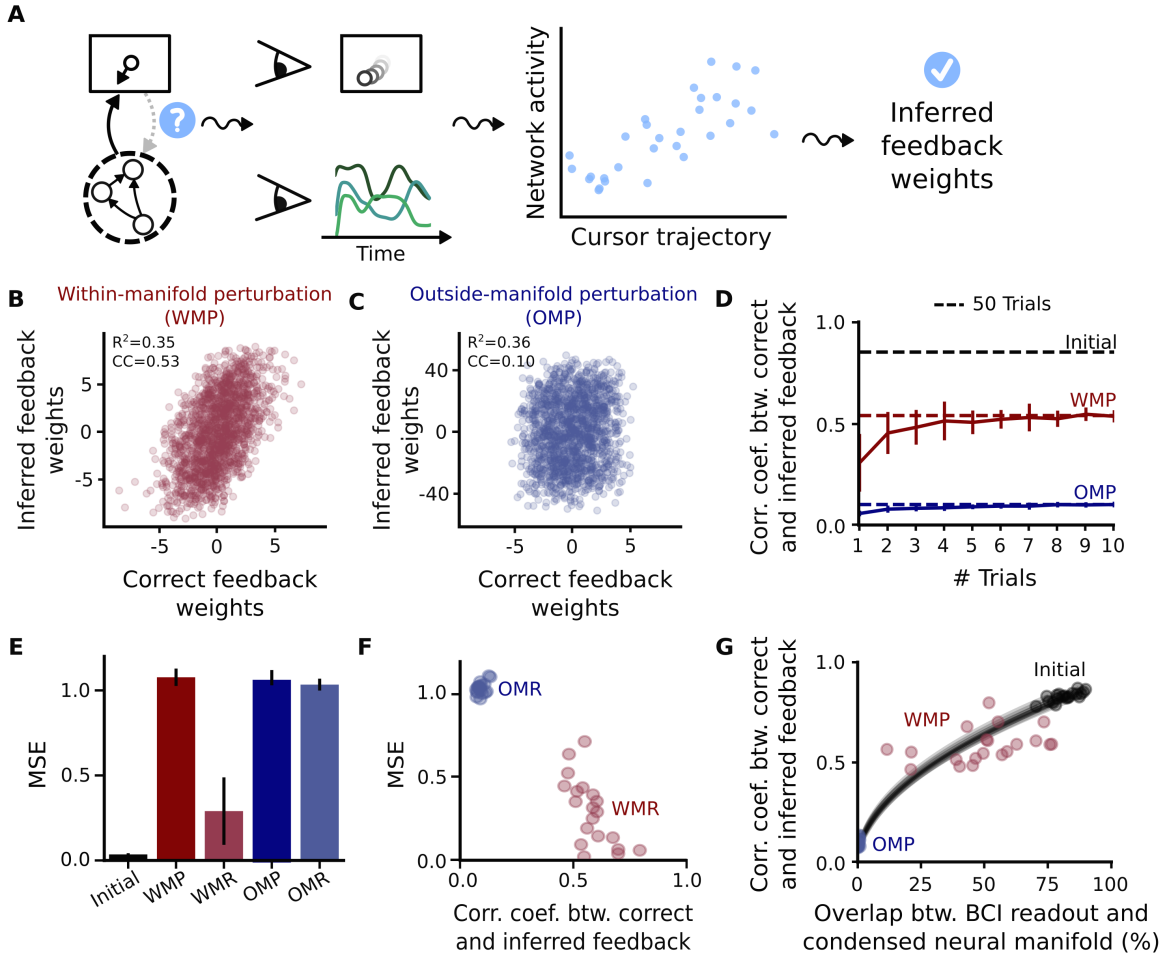


Figure 2.4: Learning the feedback signal is possible for within- but not for outside-manifold perturbations. A. Regression is used to learn feedback weights. **B-C.** Result of feedback learning for a within-manifold perturbation (WMP) (B) and an outside-manifold perturbation (OMP) (C). Accuracy of predicting neural dynamics from cursor dynamics is measured by R^2. Similarity between inferred and correct feedback weights is measured by calculating the correlation coefficient between both. **D.** Feedback learning results depend on the number of trials used for regression. Dashed lines show the feedback learning result for taking 50 trials. Compared is feedback learning after the initial training phase (where BCI readout and internal manifold completely overlap) (black), after within-manifold perturbation (red) and after outside-manifold perturbation (blue). **E.** Recurrent relearning using the learned feedback weights. As before, task performance is quantified by calculating the mean squared error (MSE) between target cursor velocity and produced cursor velocity. Compared is task performance after within- (WMP) or outside-manifold (OMP) perturbation to performance after relearning with inferred feedback weights (WMR and OMR respectively) **F.** Accuracy in the feedback learning, measured by the correlation coefficient between correct and inferred feedback weights, affects recurrent relearning performance, measured by mean squared error. **G.** Alignment of internal manifold and readout determines feedback learning performance. To take into account variance in feedback learning for within-manifold perturbations, we calculate the manifold overlap only up to a network-specific dimension, not up to 10 dimensions as done in the rest of the paper. The specific dimension is defined as $1 + (\sum \lambda_i)^2 / \sum \lambda_i^2$ where λ_i are the Eigenvalues of the covariance matrix. The data points show feedback learning results for 20 simulations, whereas the black lines are interpolations between OMP and initial BCI mappings for which we evaluated feedback learning.

To get intuition for this result, we firstly analysed a reduced model, where we synthesized data for any given Eigenvalue spectrum. For this reduced model, the output transformation only consists of the projection to the neural manifold, as this seemed to be the relevant part of the BCI mapping contrasting feedback learning for within- versus outside-manifold perturbations (Figure 2.S10). To learn feedback signals in this case, we regressed neural activity against 10-D neural mode activity. By measuring feedback learning performance for different Eigenvalue spectra, we found that high dimensional dynamics would allow feedback learning even for outside-manifold projections (Figure 2.S11). To understand this further, we returned to the simulation data and analysed feedback learning performance in more detail (Figure 2.S12). We used the fact that the feedback weight matrix can be described as two separate feedback vectors, one for the cursor's x-dimension and one for its y-dimension, each having the same number of entries as neurons in the network. Each entry in this vector is telling how neural activity of a specific neuron in the network should change in response to an observed error in cursor dynamics (either in x- or y-direction). Instead of using the basis system where each axis represents a neuron, we can instead look at these feedback vectors in the basis system given by principal component analysis (Figure 2.S12C). In this view, each entry in the feedback vector tells how a neural mode should change according to an observed cursor error. As expected, for a within-manifold perturbation most of the entries of a feedback vector in this transformed coordinate system are zero (Figure 2.S12D). As the BCI readout only takes into account the first ten prominent neural modes, also the correct feedback vector is confined to these first ten modes (Figure 2.S12D,F,G). In contrast, an outside-manifold perturbation results in a BCI readout, and thereby a feedback vector, which has components normally distributed across principal component dimensions (Figure 2.S12E). This means that basically every mode of the system has to adapt in order to counteract an outside-manifold perturbation. In this case it is not possible to infer the correct feedback weights for all of these modes (Figure 2.S12F). Interestingly, we found that weights obtained from a linear regression are dominated by components of dominant neural modes (Figure 2.S12H-J). This means that regressing neural dynamics against cursor dynamics does not put equal weight on predicting all neural modes, but instead focuses on predicting the dominant modes. That is also the reason why we observed equal accuracy in predicting neural dynamics from cursor dynamics under both a within-manifold perturbation and an outside-manifold perturbation (Figure 2.4B-C R^2 values). In both cases, feedback learning is able to predict the dominant modes

with some accuracy, and to infer the corresponding feedback weight components (Figure 2.S12G). For within-manifold perturbations, the feedback weight components of the prominent modes are the ones which solely constitute the feedback signal. The weight inference is therefore successful. In contrast, for the outside-manifold case, the feedback weight components of the dominant modes constitute only a minor fraction of all feedback weights which would need to be inferred (Figure 2.S12F) Weight inference therefore fails in this case. Projecting the feedback weights into the principal component axis of neural dynamics allowed us to understand why feedback learning is possible for a within-manifold perturbation, but impossible for an outside-manifold perturbation.

In a next step, we wanted to test whether the learned feedback weights are good enough to correctly drive recurrent relearning. For this, we estimated the feedback weights offline as described above and then used them during recurrent relearning to propagate the error signal from cursor to neuron level (Figure 2.S17C). As expected, we found that outside-manifold learning will be completely impaired if we use the learned feedback weights (Figure 2.4E blue). In contrast, task performance increased during recurrent relearning for within-manifold perturbations (Figure 2.4E red). Interestingly, we found that there is a relatively large amount of variation in relearning performance if we compare different network initializations. However, this large amount of variation is only observed for the task performance after within-manifold relearning. We hypothesized that the accuracy of the learned feedback weights is responsible for the successful retraining of recurrent weights during the adaptation phase. This is indeed what we found when we plotted the retraining performance against the feedback learning accuracy (Figure 2.4F). In summary, these results suggest that the accuracy of the feedback learning determines whether or not the BCI perturbation can be counteracted by recurrent restructuring. Together with the fact that it is not possible to infer feedback weights for outside-manifold perturbations, this could explain why outside-manifold learning was not observed experimentally[33].

Although we have shown that the accuracy of the feedback signal is responsible for either success or failure of recurrent retraining, it is still unclear why feedback learning works better for some BCI perturbations than others, and especially why it works for within-manifold perturbations, but completely fails for outside-manifold perturbations. As the main difference between within- and outside-manifold perturbations is the alignment of the readout to the internal manifold, we hypothesized that this alignment controls whether feedback learning is successful or not. To test this,

we carried out simulations where we interpolated between the initial BCI readout and the one used for an outside-manifold perturbation. This allowed us to sweep the whole spectrum of alignments, as the initial BCI readout is completely aligned to the internal manifold of the network, whereas the readout for outside-manifold perturbations is completely orthogonal to it. We then inferred the feedback weights for all of these different readouts and quantified the result by calculating the correlation coefficient between the learned and the correct feedback weights. As hypothesized, we found that there is a specific monotonic relationship between feedback learning performance and alignment of BCI readout and internal manifold (Figure 2.4G). This also explains why we observed a large variation for within-manifold relearning performance. In our setup, we fix the number of manifold dimensions used in the BCI readout, but for each individual network the internal manifold dimension can deviate from this. This means that within-manifold perturbations can be in fact slightly outside-manifold for individual networks, so that the BCI readout in these cases is not hundred percent aligned to the true internal manifold. In summary, by looking at the underlying requisites enabling successful feedback learning, we could understand why it is impossible to infer correct feedback weights in the case of an outside-manifold perturbation and why we would expect variations in within-manifold learning in the case that the readout dimension does not exactly match the internal manifold dimension.

Although we have shown so far that it is in principle possible to infer feedback weights, the question remains how this procedure could be implemented in the brain. Obviously, linear regression is a statistical method which is unlikely to be directly implemented by the brain. However, a biologically plausible learning rule can be used to learn to predict neural dynamics from cursor dynamics, and thereby successfully learn feedback weights (Figure 2.S17A-B). Another major concern is the multiple use of the feedback weights. Firstly, they are learned using the neural dynamics, but then they are used to propagate the error signal. To tackle this problem we have alternatively tried to learn feedback signals not by regression neural dynamics against cursor dynamics, but instead by regressing neural dynamics directly against cursor errors. We found that it is possible to infer correct feedback weights in this scenario, as long as the produced cursor dynamics are larger than the target cursor dynamics (Figure 2.S17D-F). Thereby, we can imagine a neural implementation where neurons representing the error in cursor dynamics are connected to neurons in motor cortex, responsible for moving the cursor. The weight between them would then be adjusted over time to learn to signal the correct

error feedback. An open question which remains is how neurons in motor cortex can simultaneously represent two signals, the error and the internal dynamics. Interestingly, this could be implemented by neural multiplexing[127], for example by dendritic and somatic signalling[128]. Taken together, we showed that our approach of learning feedback signals can be realised using local, biologically plausible, synaptic plasticity.

Feedback signal for outside-manifold perturbation can potentially be learned with incremental strategy

Besides the original experimental study showing that monkeys can not adapt to outside-manifold perturbations on the timescale of a single experiment[33], there is new evidence showing that outside-manifold learning is indeed possible if the training lasts for several days and follows a specific incremental protocol[120]. We have shown so far that outside-manifold learning should not be possible, as the correct feedback weights needed for successful retraining can not be learned. We reasoned that to nevertheless be able to learn outside of the original manifold, one would have to insert intermediate perturbation steps, where for each perturbation step there is a certain amount of overlap between the current internal manifold and the BCI readout. This would assure that the system has the chance to learn constructive feedback weights in each step, which could then lead to successful recurrent restructuring. With this strategy, one could potentially build up the learning by shifting the internal manifold step-by-step closer to the one defined by the outside-manifold perturbation. To test this idea we implemented the incremental strategy by inserting three intermediate perturbations which interpolate between the original BCI readout and the readout for an outside-manifold perturbation (Figure 2.5A).

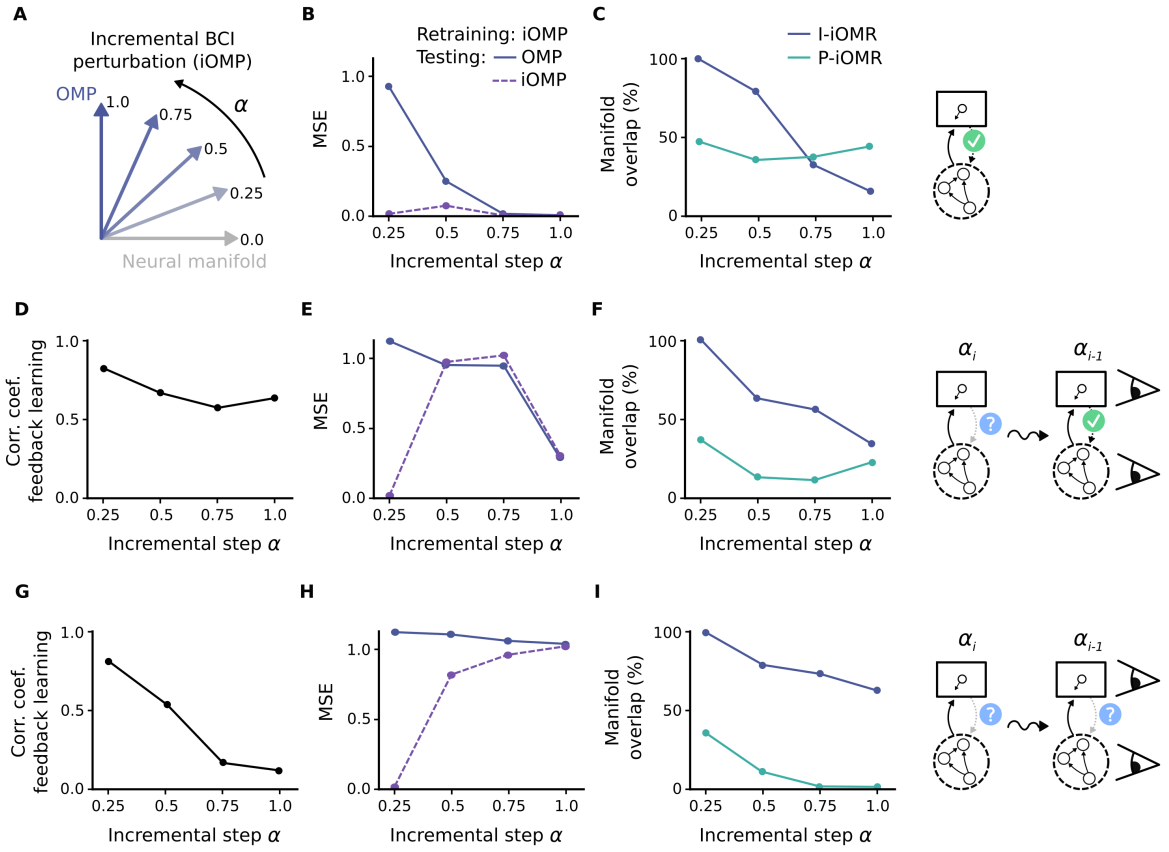


Figure 2.5: Feedback signal for outside-manifold perturbation can potentially be learned with incremental strategy. A. Incremental outside-manifold perturbation, where with each incremental step α the BCI readout is less aligned with the original neural manifold. **B-C.** Retraining results for incremental perturbation steps using correct feedback signal. **B.** Task performance after retraining, measured as mean squared error (MSE), achieved with current incremental BCI readout (iOMP), which is the readout used during retraining, as well as achieved under the full outside-manifold perturbation (OMP). **C.** Manifold overlap between initial manifold and manifold after relearning (blue line), as well as between target manifold, defined by the current BCI readout, and manifold after relearning (green line). **D-F.** Retraining results using incremental strategy and assuming, that each consecutive incremental learning step starts from a correctly trained network. **D.** Feedback learning performance, quantified by calculating the correlation coefficient between correct and inferred feedback weights, for each incremental learning step. **E.** Task performance after recurrent retraining. **F.** Manifold overlap for each incremental step. **G-I.** Retraining results using full incremental strategy. Here, each consecutive learning step starts from the network state obtained in the previous step. Feedback learning performance (G), task performance after recurrent retraining (H) and manifold overlap (I).

To test whether the inclusion of the extra BCI perturbations is enough to enable feedback learning, we firstly performed a simplified simulation, where the feedback learning relies on a correctly trained previous network state (Figure 2.5D-F). To obtain the correctly trained network state, we ran a separate retraining simulation with an ideal-observer feedback signal for every intermediate perturbation step. As expected, each of the incremental BCI perturbations could correctly be counteracted by recurrent

retraining with an ideal-observer feedback signal (Figure 2.5B). By looking again at the manifold overlap obtained during retraining, we found that each retraining simulation achieved between 40% - 50%, in line with our previous results (Figure 2.5C). To test feedback learning, we next tried to learn the correct feedback weights for each perturbation step by observing the correctly trained network from the preceding perturbation step. We found that the overlap between the network state of one perturbation step and the BCI readout of the next step is high enough to allow successful feedback learning (Figure 2.5D). To finally test whether the incremental learning strategy would work in this simplified setup, we used the learned feedback weights to retrain the recurrent network. Interestingly, we found that retraining drastically improves by using an incremental training strategy (Figure 2.5E). However, the manifold overlap achieved during recurrent retraining is lower compared to retraining with an ideal-observer feedback signal (Figure 2.5E). This makes sense as the learned feedback signal does not completely point in the new desired directions, as it would do for the correct feedback signal. Instead, it also has components pointing in the dimensions of the manifold orientation before retraining. With this simplified scenario we could show that incremental learning can potentially be used to counteract an outside-manifold perturbation, assuming that the new incremental perturbation always starts from a well-trained network.

To finally see whether our network can be trained incrementally, without relying on correctly trained intermediate steps, we implemented a similar simulation as before, but now the feedback learning uses the preceding network states which also have been trained with learned feedback, which is not necessarily the correct feedback we have used before (Figure 2.5G-I). We found that for this scenario the accuracy of the feedback learning drops for each intermediate perturbation step (Figure 2.5G). As expected, this deficiency led to poor performance in the recurrent retraining (Figure 2.5H) and the incremental strategy is therefore not helping in this case. To understand why incremental learning fails here, we compared what happens to the overlap between internal manifold and BCI readout with each consecutive step. In the simplified simulations before (Figure 2.5B-F), we found that the internal manifold does not align as well with the current readout if we use learned feedback weights instead of the correct ones (compare Figure 2.5C and 2.5F). In the following incremental step, this reduced overlap leads to a larger misalignment between current network state and BCI readout, which in turn leads to a decrease in feedback learning accuracy. The problem is therefore that the network can not catch up with the BCI perturbation and falls behind more and more for every incremental learning

step. In summary, we demonstrated that a progressive schedule could in principle lead to outside manifold learning, but we also identified limitations associated with this schedule.

2.4 Discussion

To investigate which factors influence learning within and outside of the neural manifold, we used a recurrent neural network trained with a machine learning algorithm. Although this method of modelling neural dynamics lacks biological details, using RNNs has been surprisingly useful to understand neural phenomena[42,55,57,66,129]. Using this approach, we could identify feedback learning as a potential bottleneck differentiating between learning within versus outside the original manifold. As feedback learning might not be the only relevant factor, we also tested the impact of further biological constraints. We found that neither separation of excitatory and inhibitory neural population (Figure 2.S5), nor local learning rule (Figure 2.S6)[93,130], qualitatively change the fact that within- and outside-manifold perturbations can equally well be learned, given an ideal-observer feedback signal. We also took into account the amount of weight change produced in order to assess learnability of within- and outside-manifold perturbations. Of course, the solutions the recursive-least-squares learning algorithm found are not guaranteed to be accessible through a biological learning algorithm and should be considered as an upper bound on weight change, as RLS is known to produce relatively large weight changes in order to keep the observed error small. Investigating the weight change under a more biologically plausible learning algorithm (Figure 2.S6) showed indeed much smaller weight changes with almost equal learning performance. We also saw a minor difference between within- and outside-manifold learning with respect to the learning timescale (Figure 2.S6A) and the amount of weight change (Figure 2.S6C), but not the relearning performance (Figure 2.S6B). This could potentially also contribute to the experimental observation that within-manifold learning is easier than outside-manifold learning.

In principle, there are two alternative scenarios which would lead to altered network dynamics in motor cortex. Assuming motor cortex is highly controlled by other brain regions, such as for example the cerebellum and pre-motor cortex, the first scenario would be to alter the control signals coming from these regions, and to therefore trigger different activity patterns[122]. Nevertheless, how the inputs are learned remains unclear. The second scenario is rewiring in the local circuit. In this study we focused on the latter and tested to what extent it can explain experimental observations[33,120].

We found that reorganization of local connectivity does not necessarily lead to a change in neural manifold. This shows that globally coordinated changes in neural activity can not only be realized by adaptation of input drive, but also via local rewiring driven by a learned feedback signal. Interestingly, our results on feedback learning are not necessarily constrained to the hypothesis of local plasticity, but could be also relevant for adapting neural activity by changes upstream. It remains to be investigated to what extent one can contrast both frameworks, given experimental data, as biology potentially uses multiple mechanisms in parallel.

Beside the main result of Sadtler et al., showing that within-manifold learning is easier than outside-manifold learning, they made further experimental observations which we tested in our model. Firstly, Sadtler et al. observed a differential after-effect for within- versus outside-manifold learning. For a within-manifold perturbation, the monkey had to relearn the original mapping in order to achieve high task performance. In the case of an outside-manifold perturbation however, after switching back to the original mapping the monkey was immediately able to perform the task with high accuracy. Our current modelling approach does not prevent learning in the case of an outside-manifold perturbation. There are still changes in neural dynamics under an outside-manifold perturbation, even if they do not improve task performance, as the feedback signal is misaligned (Figure 2.4C and 2.4E). However, it seems plausible that there would be another mechanism to prevent learning under invalid feedback in reality, similar to the complex interaction of feedforward and inverse model relevant in motor control[131]. Secondly, in a follow up study Golub et al.[132] discovered that the monkeys learned to counteract a within-manifold perturbation by reassociating existing neural activity patterns. By adapting their analysis we found that our simulation data for within-manifold learning was also best described by the reassociation hypothesis (Figure 2.S13). Thirdly, Oby et al. found that there are two different types of contributions to outside-manifold learning: 1) components which lie outside of the original set of activity patterns but still within the manifold and 2) true outside-manifold components which are orthogonal to the original manifold[120]. We found that our model too is using within-manifold components to learn outside-manifold perturbations (Figure 2.S14). Together, these results show qualitative agreement between our simulation data and a set of relevant experimental observations.

A recent computational study has shown that weight changes necessary for within-manifold learning are much smaller compared to outside-manifold learning and proposed this finding as a potential

reason for the experimental observation in Sadtler et al.[121]. In contrast, we could not observe a difference in weight change during retraining (Figure 2.2C). However, we did observe that, if we constrain the number of plastic weights, within-manifold learning is more successful compared to outside-manifold learning (Figure 2.3C-D) and also that weight changes related to outside-manifold learning are slightly higher-dimensional (Figure 2.S4C-D). One fundamental difference compared to Waernberg and Kumar is that they used an alternative way of implementing the BCI perturbation, which led to the finding that the weight changes are bigger during outside-manifold learning (Figure 2.S15). Instead of setting new target readout dynamics, which could be derived given a certain within- or outside-manifold perturbation, they implemented a perturbation by changing the encoder matrix (called K in their case). As the encoder matrix injects the readout dynamics back into the system, this already changes the effective recurrent weights in the network, even before retraining. To counteract the perturbation, the readout weight matrix Φ is retrained to produce the same readout dynamics as initially. The overall weight change which Waernberg and Kumar reported in their paper is given by an outer product of both matrices $K\Phi$. By reimplementing their simulations, we replicated their finding that within-manifold learning leads to smaller weight changes than outside-manifold learning (Figure 2.S15C,E,G). Interestingly though, when we implemented the BCI perturbation by setting new target readout dynamics, instead of changing K itself, the observed weight changes were small both for within- and for outside-manifold learning (Figure 2.S15D,F,H). Thereby, we could identify the main difference between both studies and replicate our finding that within- and outside-manifold learning produces similar amount of weight change if retraining starts from the initial connectivity, as it would be for a BCI monkey experiment.

As having a feedback signal is a general concept of motor control[131,133,134], it would be interesting to study whether one could also infer correct feedback signals in an upstream region exclusively for the case of a within-manifold perturbation. We hypothesize that, as long as the, potentially non-linear, output transformation via motor cortex is aligned with the internal manifold of the upstream region, feedback learning should be viable. Thus, our study gives a new perspective on the general problem of credit assignment, proposing that correct feedback learning critically depends on the alignment between output transformation and internal manifold in a specific neural pathway (Figure 2.4G).

Our results can also be related to a recent finding, reporting that a variety of different movements are controlled by neural activity lying in the same subspace across movements[36]. As our results

suggest that the readout - used to produce a movement - indirectly defines the accessible neural activity space for motor learning (Figure 2.4F and 2.4G), we would expect that a stable readout implies a stable manifold across different movements. One could test this hypothesis by estimating the muscle output transformation for each specific movement. We would predict that less overlap in the output transformation implies less overlap in neural manifolds, measured during the two different movements.

Finally, our work might be relevant for future BCI designs. Our results support the idea that a subject can potentially learn to interact with any kind of static BCI mapping, as long as its dimensions are aligned to the internal dynamics of the local circuit it is connected to. It remains to be experimentally tested to what extent this holds true, but if so, it could impact development of brain-computer interfaces for real-world applications.

In summary, our model makes the following predictions, 1) for within-manifold perturbations, it is possible to infer useful feedback weights by regressing neural activity against cursor dynamics, 2) learning performance during a BCI task can be predicted by the alignment between BCI readout and internal manifold of the connected brain region, 3) silencing local network rewiring would impair learning of within- as well as outside-manifold perturbations.

In conclusion, we showed that local network rewiring can account for the observed behavioural differences between within- and outside-manifold learning, assuming that the feedback signal driving local reorganization has to be learned.

2.5 Methods

Variable	Definition	Value
N_T	number of targets in center-out-reach task	6
τ	unit time constant	0.1 s
g	connection strength	1.5
p	connection probability	10%
N	number of neurons in the network	800
d_M	predefined manifold dimension	10
L	trial duration	2 s
dt	time discretization	0.01 s

Table 2.1: List of parameters used in the model.

Task

In order to compare our learning results to experimental studies, we implemented a standard center-out-reach task, which is regularly used to experimentally probe motor learning. In our case, the task consists of six targets, evenly distributed on a circle. In an experiment, the subject would get a visual cue for example at the beginning of each trial to indicate which of the six targets is the target of the current trial. The subject should then reach this target in a given amount of time. We simulated the target cue by having an input pulse at the beginning of each trial, which is distinct for each of the six different targets. To simulate activity in motor cortex we used a recurrent neural network (RNN). The output is produced by a mapping, which we called brain-computer interface (BCI) for didactic reasons. It transformed the rates measured in the RNN to x and y velocity of a cursor, where the cursor simulated the center-out-reach movement. At the beginning of each trial, the cursor was set to the center of the workspace, and after the target cue ends, its movement was controlled by the RNN dynamics. The target speed for the cursor was 0.2 and the target directions were evenly distributed between 0 and 2π. To measure performance we calculated the mean squared error between the target cursor velocity and the produced cursor velocity and summed it over all 50 test trials. As the target cursor velocity was constant and did not change depending on where the cursor is at a given moment, we used an open loop setup. The target for each specific trial was randomly chosen among the six.

Recurrent neural network (RNN)

The recurrent neural network was simulated by the following dynamical equation:

$$\tau \dot{x}_i = -x_i + \sum_j W_{ij} r_j + s_{ik}$$

The recurrent connections W_{ij} were sparse, with connection probability $p = 10\%$, and drawn from a Gaussian $\sim \mathcal{N}(0, g^2/(Np))$ (cf. list of parameters in Table 2.1). To simulate the target cue our network received a pulsed input s_{ik} at the beginning of each trial. The input is given by

$$s_{ik} = W_{ik}\, s^{\text{pulse}}$$

where s^{pulse} is a $0.2s$ long pulse with amplitude 1 and the input weights W_{ik} were drawn randomly

from a uniform distribution $\sim \mathcal{U}(-1, 1)$.

To implement learning in the network we used an adapted form of the recursive-least-squares algorithm, which was inspired from[53]. If learning is on, it started after the target cue period. Then, at every second time point (which corresponds to $\Delta t = 20ms$) an update step was made. Firstly, the performance error e^P, given by the difference between the produced cursor velocity and the target cursor velocity, was calculated. Via a linear transformation this two-dimensional performance error was then assigned to errors e on the activity of single units in the network.

$$e = W^{fb} e^P$$

This transformation W^{fb} is what we defined as feedback weights. In the ideal-observer case, this feedback matrix was given by the pseudo-inverse of the BCI transformation. Note that we calculated the error for each neuron through these feedback weights, which was used to adapt recurrent weights in the network, but there was no additional input for each neuron due to this error signal. The recurrent weight update is given by

$$W_{ij}(t) = W_{ij}(t - \Delta t) - \frac{e_i(t) \sum_k P^i_{jk}(t) r_k(t)}{1 + \sum_m \sum_n r_m(t) P^i_{mn}(t - \Delta t) r_n(t)}$$

where e_i is the error for unit i and P^i is a matrix which estimates the inverse of the correlation matrix of the input to neuron i. P^i was also updated in every update step and the update rule is given by

$$P^i_{jk}(t) = P^i_{jk}(t - \Delta t) - \frac{\sum_m \sum_n P^i_{jm}(t - \Delta t) r_m(t) r_n(t) P^i_{nk}(t - \Delta t)}{1 + \sum_m \sum_n r_m(t) P^i_{mn}(t - \Delta t) r_n(t)}$$

At the start of learning, the matrices P^i were initialized as diagonal with 0.05 on the diagonal. For each training period the network was trained for 80 trials. Note that the training algorithm did not change the structural connectivity, so only connections which were non-zero at the beginning were updated.

Brain-computer interface (BCI)

The brain-computer interface consisted of two linear transformations which sequentially transformed network rates to cursor velocities. The first transformation is given by a projection to principal

components of the network dynamics. To calculate this projection, we measured the dynamics over 50 trials during which the network is performing the task. Here, and at every other point, we took network dynamics into account only from the end of the cue phase. Then, we used principal component analysis to calculate the projection matrix C. The second transformation is given by a decoder matrix D which was optimized offline by doing regression of target cursor velocities against the first 10 principal components, taking the same 50 trials as before. The full BCI transformation T is then given by

$$T = D\ C$$

For the initial training period, T was set up randomly. Each entry of the matrix was drawn from a standard normal distribution, normalized to have matrix norm 1 and multiplied with a factor 0.04. This setup showed to drive initial learning sufficiently well, and assured good task performance after the initial network training period, followed by the BCI tuning described above (Figure 2.S2). Note that the initial random mapping had high impact on the produced neural manifold and therefore also on the optimized BCI mapping (Figure 2.S9).

For a within-manifold perturbation, we inserted a permutation matrix η_{WM} in between the first and the second linear transformation, which shuffled the decoder weights related to the principal components.

$$T_{WM} = D\ \eta_{WM}\ C$$

In contrast, for an outside-manifold perturbation, we inserted a permutation matrix η_{OM} before the first linear transformation, which shuffled the projection matrix and therefore resulted in not extracting the correct principal components of the network dynamics.

$$T_{OM} = D\ C\ \eta_{OM}$$

In order to assure that both types of perturbations cause approximately the same performance error we calculated the performance, measured as mean squared error, for 200 random perturbations of each type. Then we calculated the mean performance over all perturbations, ranked them according to their difference to the mean and selected the perturbations closest to the mean. This ensured that the selected perturbations had similar mean squared error. Note that although the performance error was matched through this procedure, the cursor velocities for outside-manifold perturbations

were scaled down compared to within-manifold perturbations. This is a consequence of the readout transformation to non-dominant directions in neural state space which naturally contain less variance and therefore lead to lower cursor velocities.

Analysis

To quantify manifold overlap we adapted the measurement used in[132]. Firstly, we calculated the covariance matrix of the network dynamics S, given 50 trials of performing the task. This was done for the initial network (subscript $_1$) and for the network after retraining (subscript $_2$). Then, both of these matrices were projected onto the neural manifold defined by the initial network. This projection is given by the same projection matrix C as used in the BCI transformation. To quantify the amount of explained variance in the first 10 dimensions, defined as β_1 for the initial network and β_2 for the retrained network, we calculated the trace of the projected matrices (taking into account only the first 10 dimensions) and divided it by the trace of the original matrices (full-dimensional).

$$\beta_2 = \frac{Tr(CS_2C^T)}{Tr(S_2)} \quad \beta_1 = \frac{Tr(CS_1C^T)}{Tr(S_1)}$$

The manifold overlap is then defined as the ratio β_2/β_1. For calculating the overlap between the perturbed BCI readout and the network manifold after retraining we replaced the projection into the Eigenbasis of the initial covariance matrix, given by C, with a projection given by the new readout transformation, which is $C\ \eta_{OM}$.

To calculate the overlap between BCI readout and condensed network manifold (Figure 2.4G), we summed up the normalized projections of the readout T into the first k dimensions of the Eigenspace of the network dynamics, given by C,

$$\frac{\sum_i^k (T_x C_i)^2}{2|T_x|^2} + \frac{\sum_i^k (T_y C_i)^2}{2|T_y|^2}$$

where T_x is the first row of the BCI transformation, decoding the cursor velocity in x-direction, and T_y is the second row of the BCI transformation, decoding the cursor velocity in y-direction. The number of dimensions k is defined by

$$1 + \frac{(\sum_i \lambda_i)^2}{\sum_i \lambda_i^2} = 1 + PR$$

where λ_i are the Eigenvalues of S_1 and PR is participation ratio, a commonly used measure of dimensionality.

Feedback learning (Figure 2.4)

To learn the feedback weights for a given BCI perturbation, we created a dataset of 50 trials where the network is performing the task with the perturbed BCI readout. During this, there are no plastic changes yet in the recurrent connections. To infer feedback weights we then regressed the network rates, measured during the 50 trials, against the x and y cursor velocities. The accuracy of the regression is about the same for within- and outside-manifold perturbations, whereas the match between inferred and true feedback weights strongly differs. To obtain Figure 2.4D, we only used a randomly selected subset of the 50 trials. To obtain Figure 2.4G, we analysed not only the given BCI perturbations, but added intermediate perturbations, which interpolate between the original BCI transformations and the outside manifold perturbations. The analysis of the interpolated perturbations was shown in Figure 2.4G (black lines).

Incremental strategy (Figure 2.5)

In each incremental step, the learning starts from the weight configuration after initial training. We did this to insure that the network stays in a physiological state, as the recursive-least-squares algorithm tends to increase the weights during training. Without this reset, the weight distribution would grow with every incremental training and could potentially lead to a network state where most of the neurons are saturated.

Each incremental BCI perturbation T_α is defined by interpolating between the original readout T and the full outside-manifold perturbation T_{OM}, where the incremental factor α defines the ratio between both.

$$T_\alpha = (1 - \alpha)\ T + \alpha\ T_{OM}$$

2.6 Supporting information

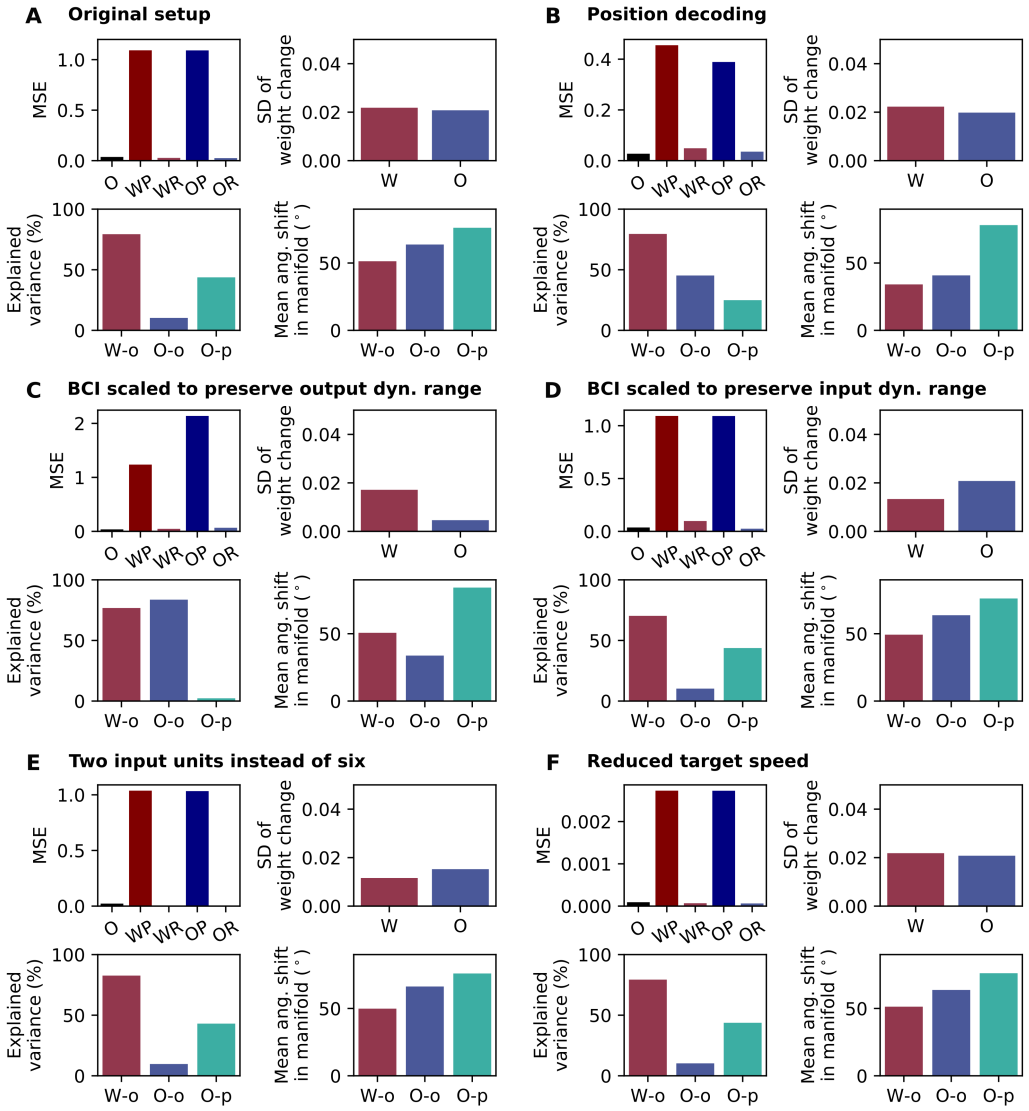


Figure 2.S1: Control simulations. A. Original setup as in Figure 2.2. Upper left panel shows task performance, measured as mean squared error (MSE), after initial training (O), after within-manifold perturbation (WP), after within-manifold retraining (WR), after outside-manifold perturbation (OP) and after outside-manifold retraining (OR). Upper right panel shows the standard deviation of the weight change distribution between before and after retraining for within- (W) and outside-manifold (O) retraining. Lower left panel shows the manifold overlap between initial and retrained manifold for within- (W-o) and outside-manifold (O-o) perturbation, as well as the overlap between retrained and target manifold for outside-manifold perturbations (O-p). Lower right panel shows the mean principal angle between the same manifold as in lower left panel. **B.** Position decoding instead of velocity decoding. **C.** BCI perturbations are scaled in order to preserve the range of velocities after perturbation. This especially affects outside-manifold perturbations as in this case the velocity values are normally only 10% of the original ones, due to a signal loss caused by the outside-manifold projection. **D.** BCI perturbations are scaled to assure that the theoretical approximation for the neural activity after retraining, given the initial training state, does not exceed the dynamic range of the neurons. **E.** Instead of having one input unit for each target, here, there are only two input units. They signal x and y position of the targets. **F.** Reduced target speed of 0.01, instead of 0.2.

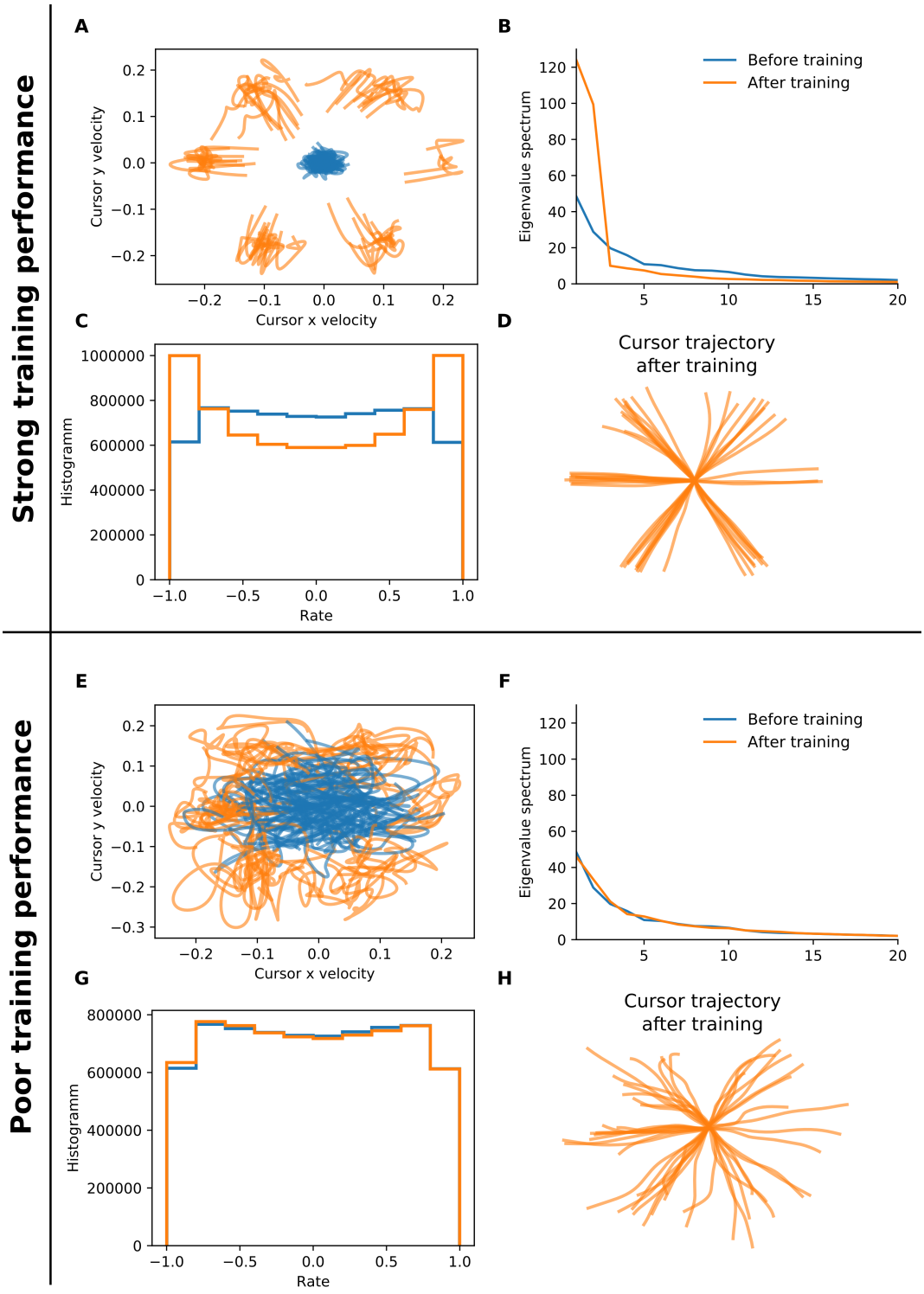


Figure 2.S2: Details of initial network training. A-D. Initial training with random decoder, normalized and scaled with factor 0.04 (the default version used in the paper). **E-H.** Initial training with random decoder, normalized and scaled by a factor 0.2. **A,E.** Cursor velocities before and after training. **B,F.** Eigenvalue spectrum of network dynamics before and after training. **C,G.** Rate distribution before and after training. **D,H.** Reconstructed cursor trajectory after training.

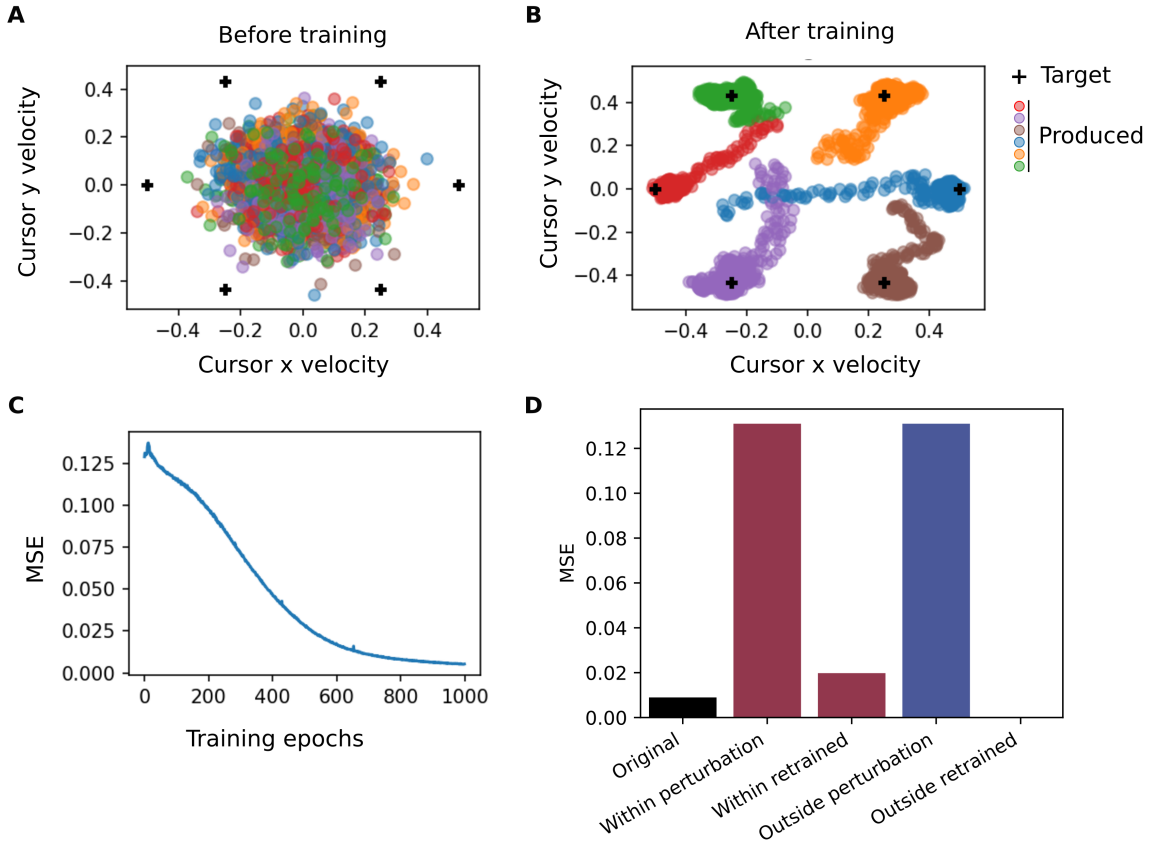


Figure 2.S3: Simulation results for recurrent neural network trained with backpropagation-through-time using Pytorch. Here, the network dynamics are given by $h_t = \tanh(W_{ih}x_t + W_{hh}h_{(h-1)})$. Each trial has 100 time steps and the target cue is given during the first 5 steps. The input x_t is modelled similar to our main setup. The stimulus amplitude is 50. The incoming weights W_{ih} are fixed and randomly drawn from a uniform distribution between $-\sqrt{1/N}$ and $\sqrt{1/N}$, where N is the number of neurons in the network. The recurrent weight matrix is initialized in a similar fashion as in our main setup, except that here, we use a fully connected network. For gradient descent we use Adam optimizer[135] with learning rate 0.001. The loss and the fixed output decoder is the same as in our main setup. The training uses an ideal-observer feedback signal to propagate the error in cursor velocities to errors on single neurons in the recurrent network. For training, we use a batch size of 30. **A.** Cursor velocities before initial training. **B.** Cursor velocities after initial training. **C.** Learning curve for initial training. **D.** Performance results for within- and outside-manifold retraining. Number of training epochs for initial training, as well as retraining, is 1000. The MSE value in (C-D) is not summed over time or trials.

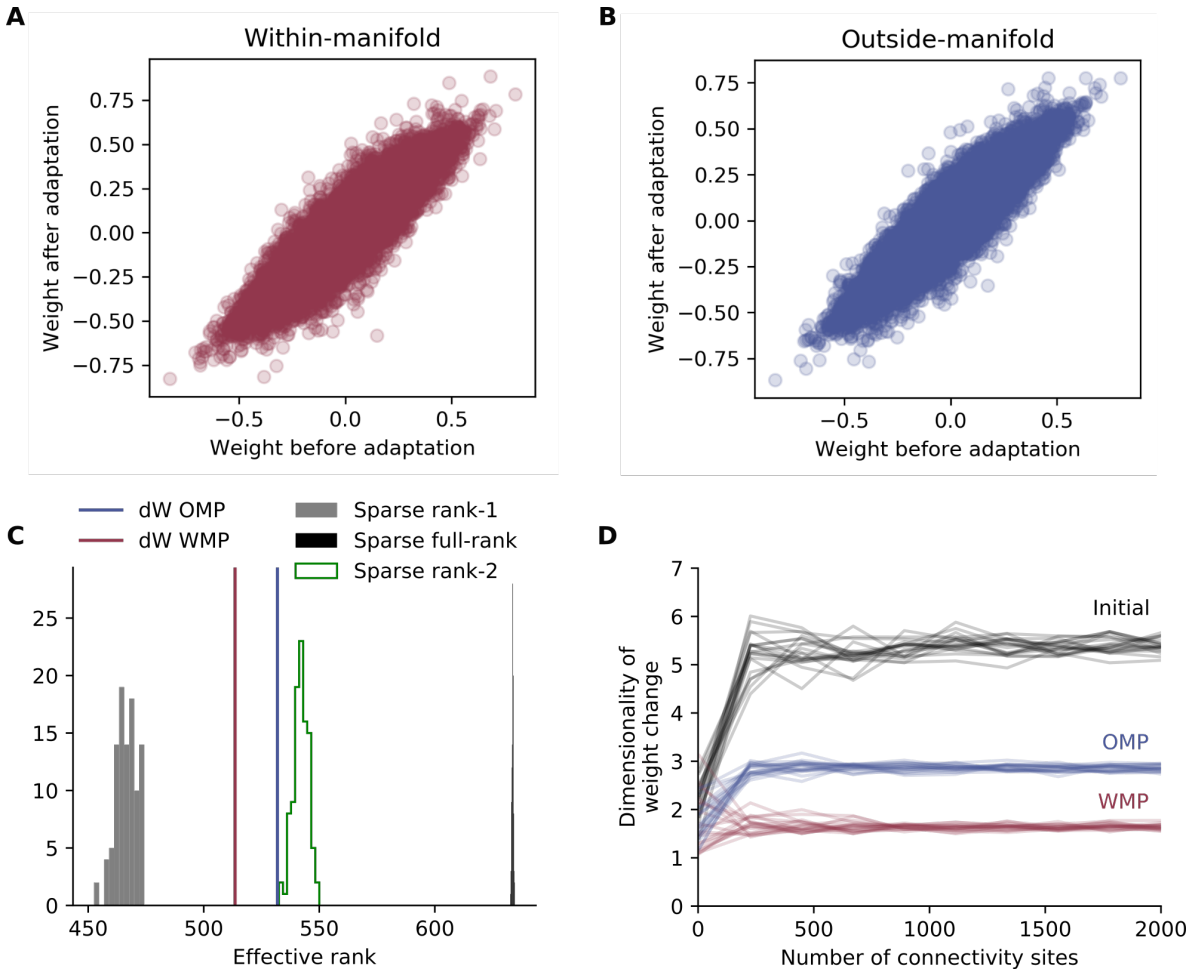


Figure 2.S4: Weight change during retraining. A. Relation between weight before retraining and after retraining for within-manifold perturbation. **B.** Relation between weight before retraining and after retraining for outside-manifold perturbation. **C.** Measurement of the effective rank[136] of the weight change matrix for within-manifold and outside-manifold retraining. Both values are compared to values obtained for random rank-1, random rank-2 and full-rank matrices, having the same sparsity as the network weight matrix. **D.** Dimensionality of weight change dynamics during retraining (dW measured at the end of each trial). Since it is not feasible to calculate the full covariance matrix for all plastic connections, as the number is too high ($\sim$ 64000), we calculated the dimensionality from randomly chosen subsets of connections and checked for convergence.

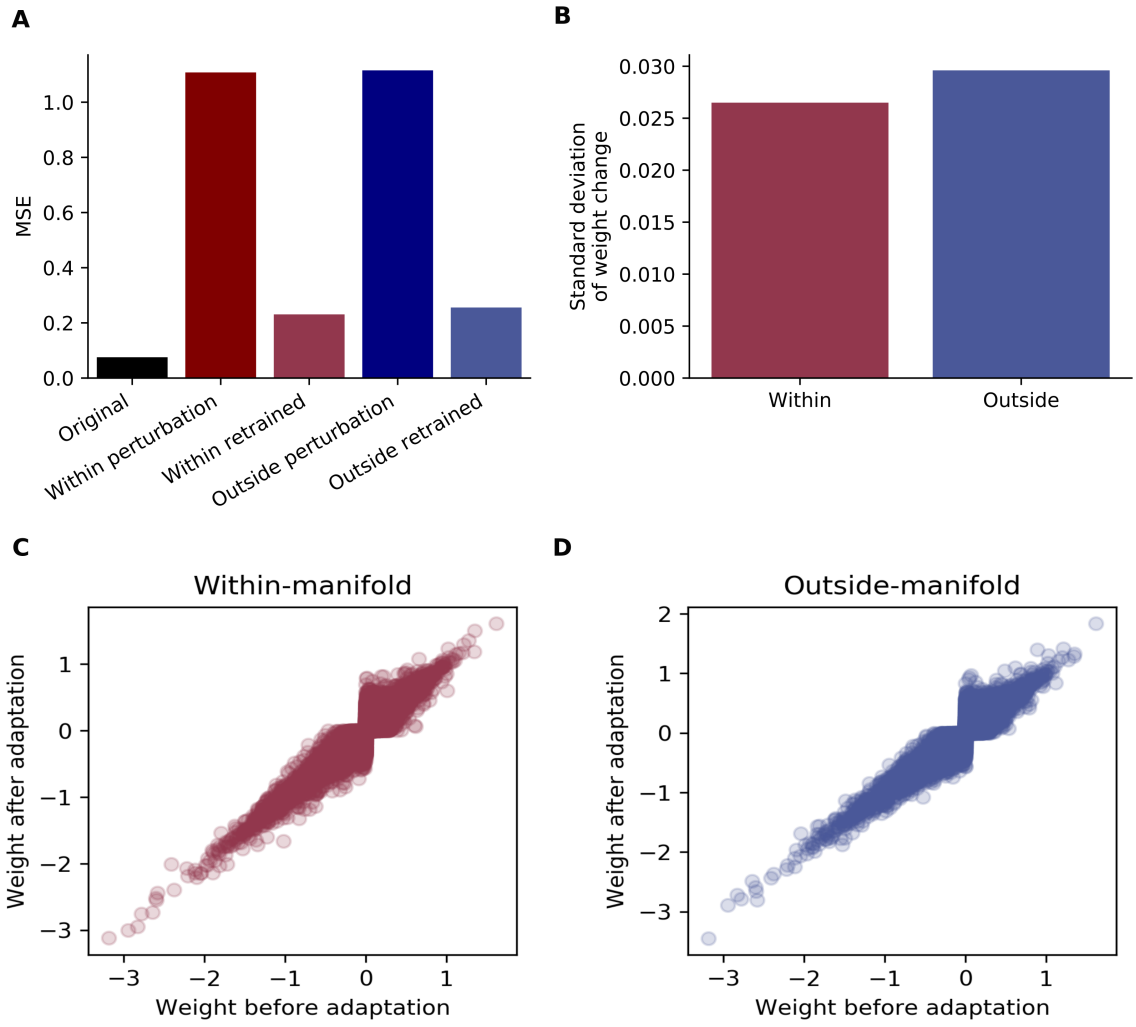


Figure 2.S5: Simulation with E-I network. Simulation results for a network with Daleian connectivity. Each neuron has either positive (excitatory) or negative (inhibitory) outgoing weights. During learning, the sign of these weights is preserved. If a learning step would produce a sign flip we instead clip the weight to zero. **A.** Training performance measured as mean squared error (MSE). **B.** Standard deviation of weight change during within- and outside-manifold training. **C-D.** Weights before versus after within- (C) or outside- (D) manifold training.

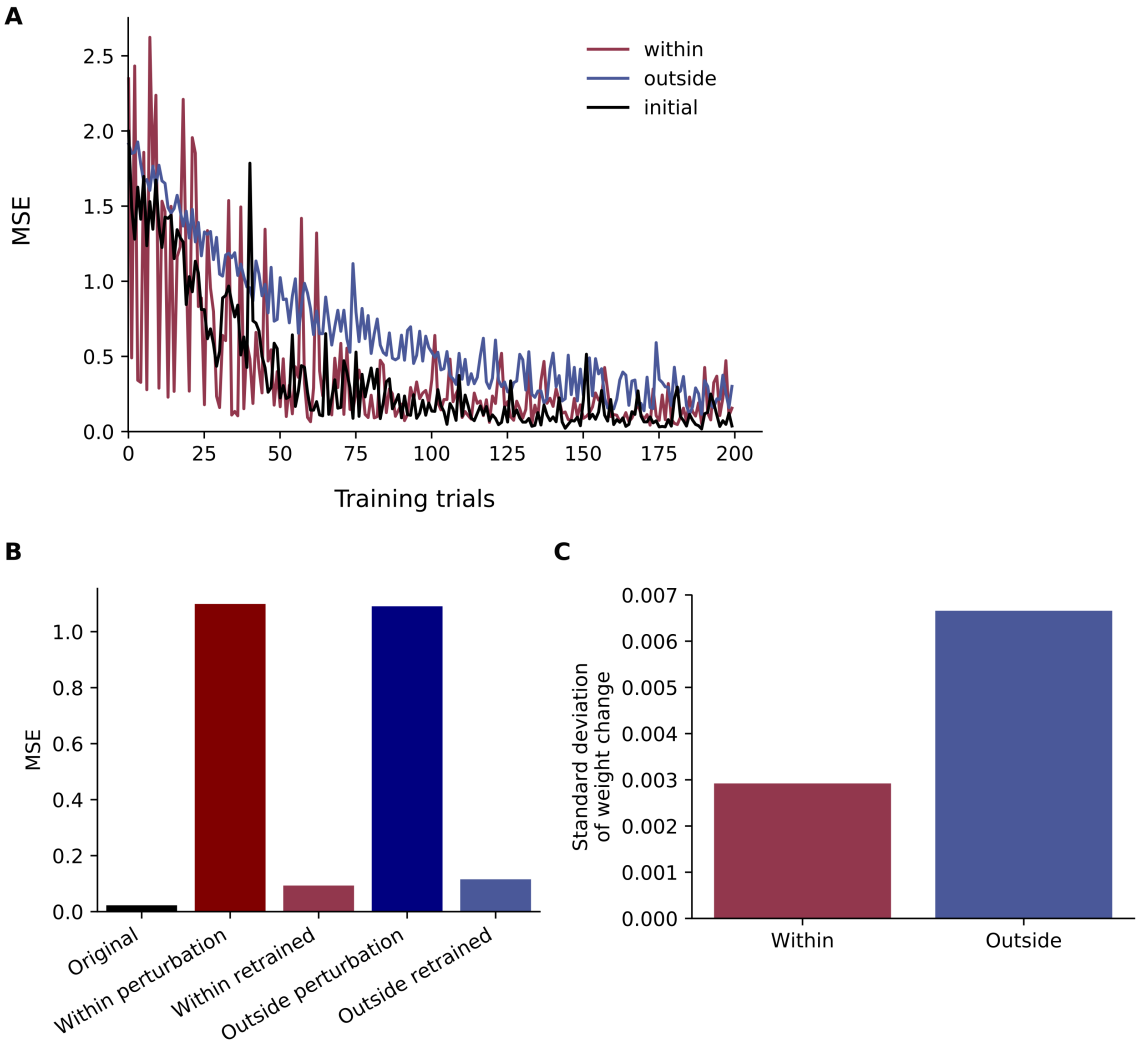


Figure 2.S6: Simulation with local learning rule. Implementation of recently proposed local learning rule approximating backpropagation-through-time algorithm[130]. The weight update is given by $dW_{ij}^t = -e_j^t(1 - \tanh^2(x_j^t)) \sum_{t' \leq t-1} r_i^{t'}$, where e_j is the error for neuron j, x_j is the activity of neuron j and $r_i = \tanh(x_i)$ is the rate of neuron i. **A.** Learning curves for initial, within- and outside-manifold training period. Here, the mean squared error (MSE) is summed over update steps per trial, which are 90 steps. **B.** Performance results after training measured as MSE. **C.** Standard deviation of weight change during within- and outside-manifold training.

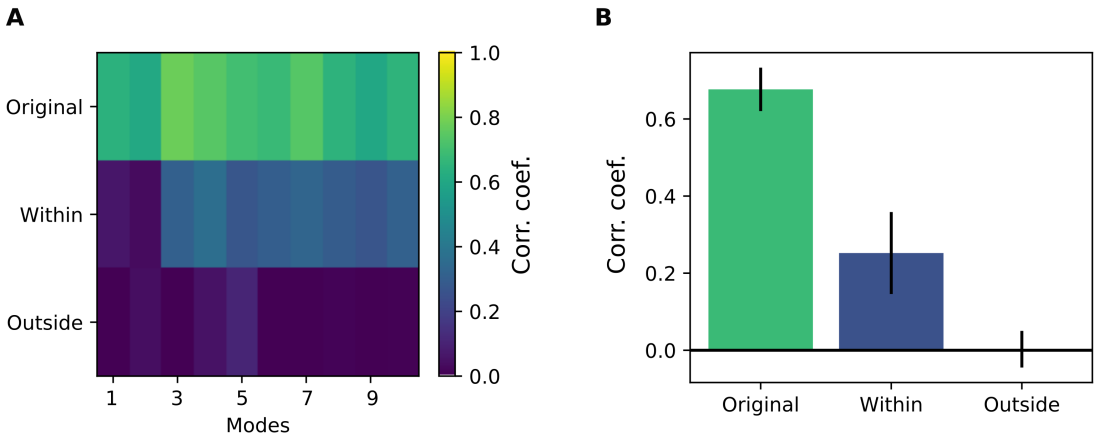


Figure 2.S7: Mode correlation between before and after retraining. A. Correlation between the neural modes before and after retraining for each of the six targets. Note, the neural manifold is in this case not the actual internal manifold (which is different after retraining), but the static one defined by the initial BCI mapping. We averaged over 20 simulations and the 6 targets. **B.** Average mode correlation. Error bars are standard deviation across the ten modes.

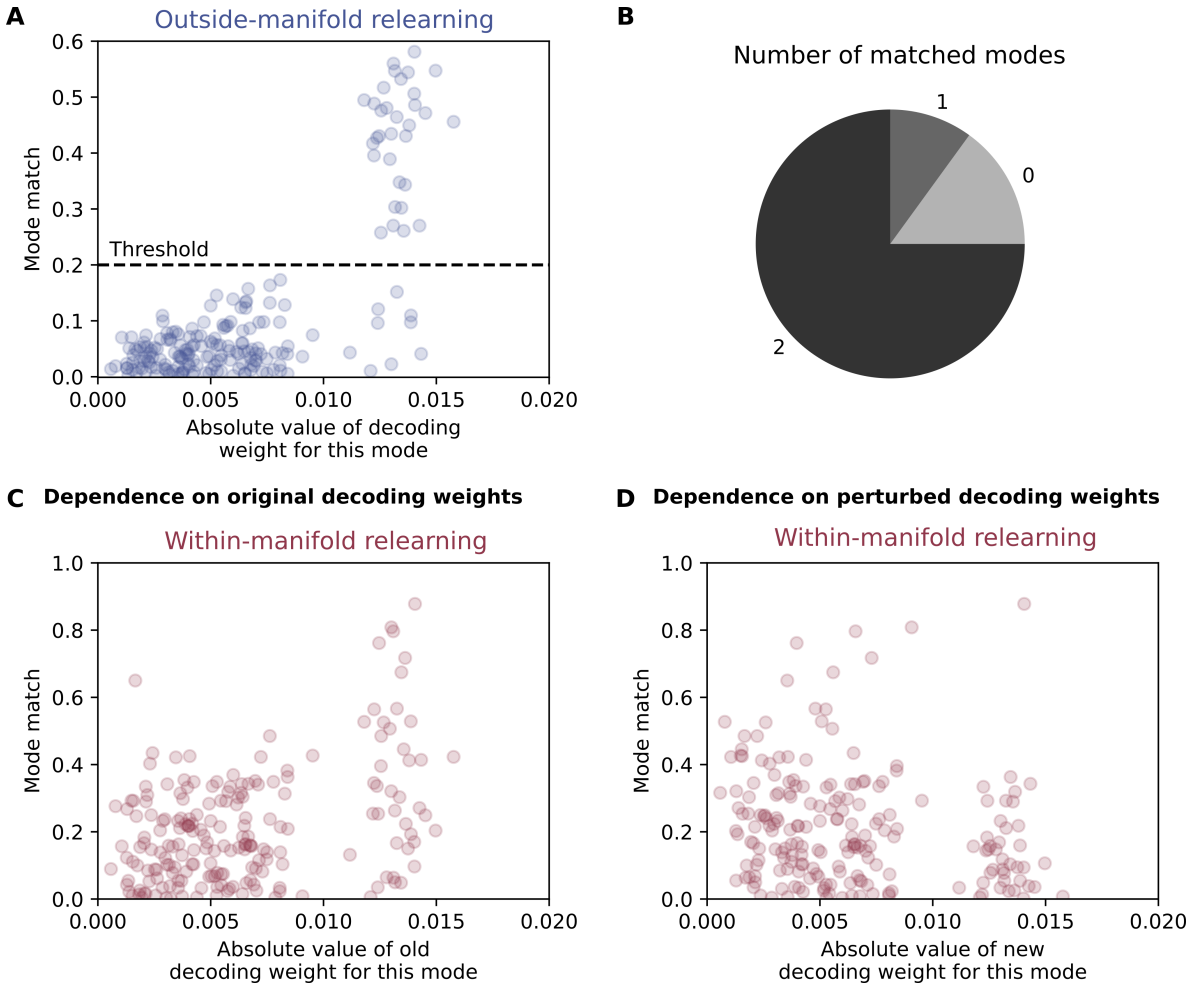


Figure 2.S8: Mode alignment during outside-manifold and within-manifold relearning. A. Mode match is defined as scalar product between the Eigenvector of the specific Eigenvalue before and after retraining. Each dot represents the mode match of one of ten Eigenvalues considered, and shows the result for one of 20 simulation runs (total 200 points shown). We defined a mode as matched if the scalar product is bigger than 0.2. **B.** We quantified how many modes are matched per simulation run, taking the definition from (A). **C.** Same as in (A), but now for within-manifold retraining. As the decoding weights are shuffled during a within-manifold perturbation one can compare the mode match to the decoding weights before and after perturbation. Here, we compared to before perturbation. **D.** Mode match for within-manifold retraining compared to decoding weights after perturbation.

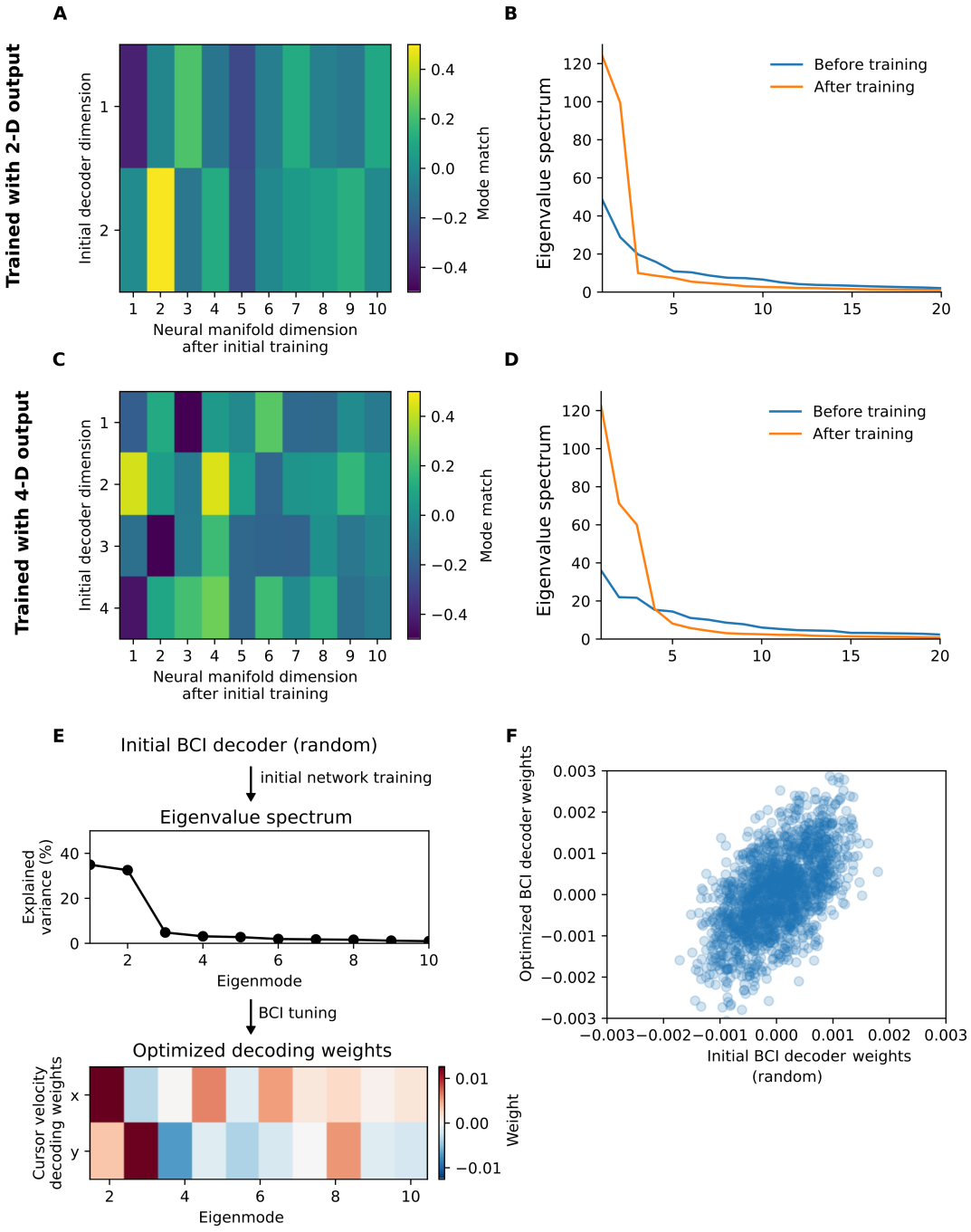


Figure 2.S9: Initial random decoder shapes neural manifold and final BCI mapping. A-B. Standard case where initial random decoder output is two-dimensional. **A.** Mode match is calculated as scalar product between vectors. The initial decoder induces the first two strong neural modes, which are strongly amplified in the Eigenvalue spectrum after initial training (B). **C-D.** Test case where initial random decoder output is four-dimensional. Here, the assignment of initial decoder and neural manifold modes is less prominent, meaning that there is no one mode specifically responsible for one output dimension. (C). In this case the number of amplified modes does not correspond to the dimensionality of the initial decoder (D) **E.** Initial random decoder shapes Eigenvalue spectrum during initial network training phase, and Eigenvalue spectrum in turn shapes optimized BCI decoding weights *D*. **F.** Relation between optimized BCI decoder and initial random one used for initial network training.

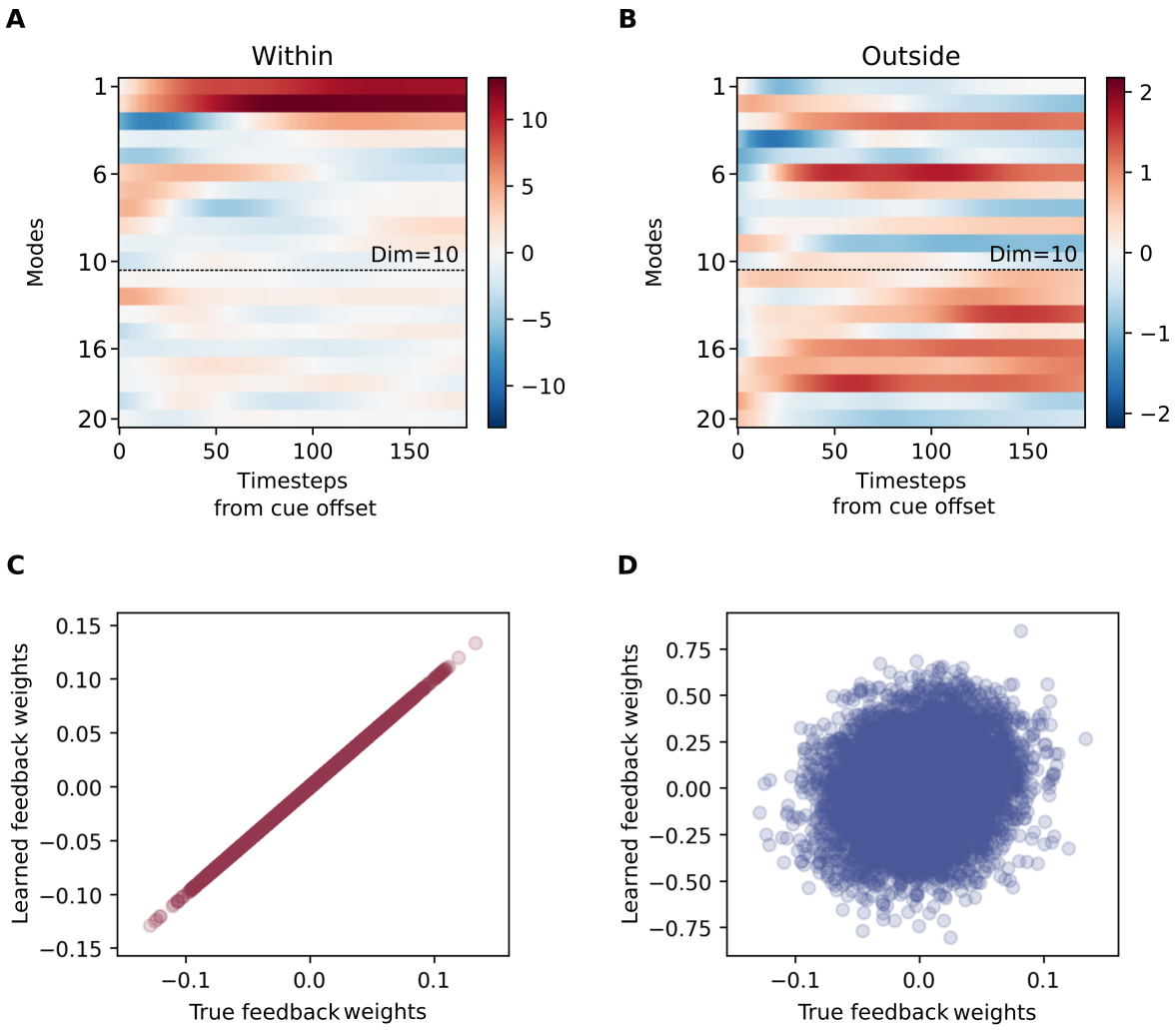


Figure 2.S10: Feedback learning on mode level. A-B. Mode activation during one example trial for a within-manifold perturbation (A) and an outside-manifold perturbation (B). Mode activation is obtained by projecting measured neural dynamics onto the original neural manifold, given by the transformation matrix C. **C-D.** Results of linear regression to infer inverse of transformation matrix C for a within-manifold perturbation (C) and an outside-manifold perturbation (D).

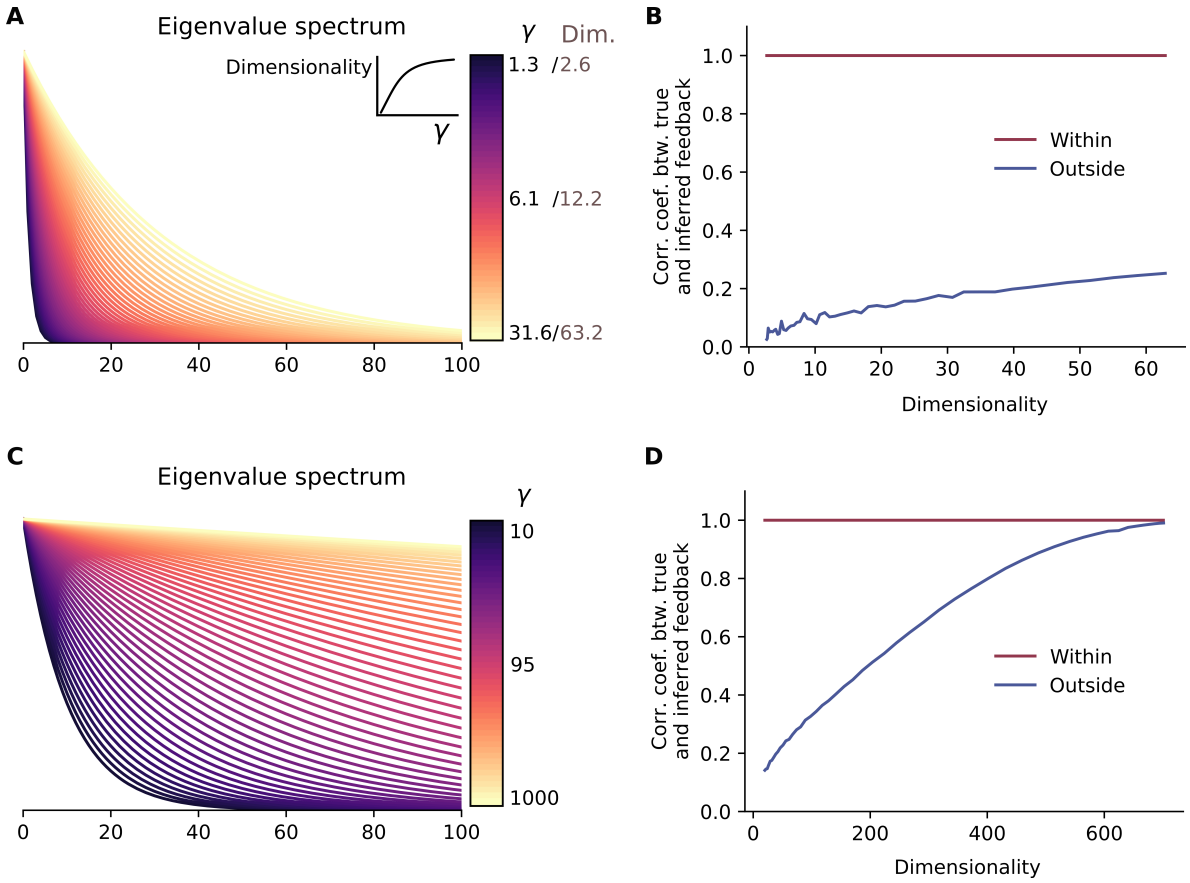


Figure 2.S11: Broader Eigenvalue spectrum improves feedback learning performance for outside-manifold perturbations. A,C. Imposed Eigenvalue spectra, defined by $f(x;\gamma) = \exp^{-x/\gamma}$. Dimensionality is calculated by $(\sum_i \lambda_i)^2 / \sum_i (\lambda_i^2)$ where λ_i is the ith Eigenvalue. Detailed methods are described at the end of this section. **B,D.** Feedback learning results measured by correlation coefficient between inferred and true feedback weights, dependent on the dimensionality of the imposed Eigenvalue spectrum.

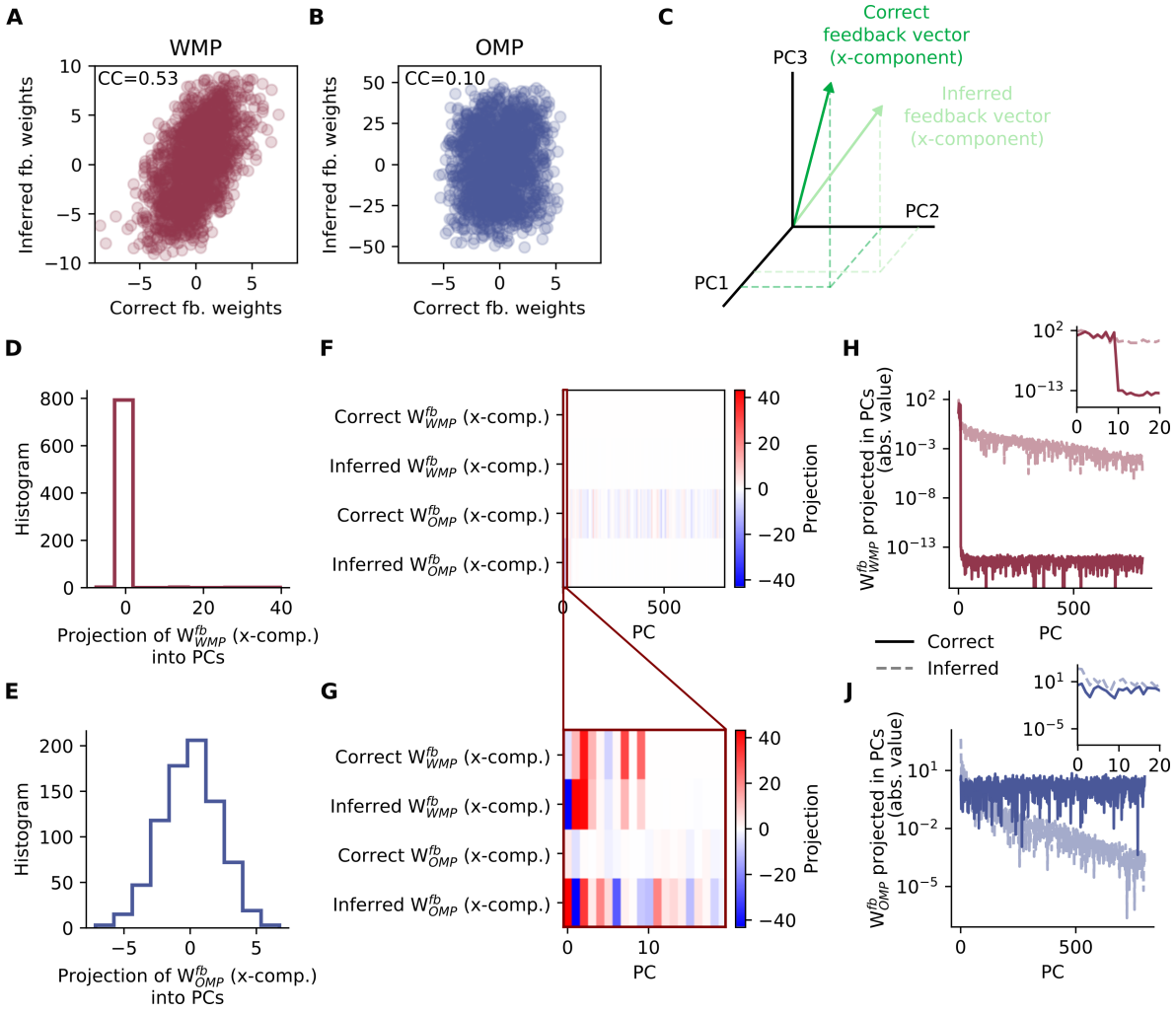


Figure 2.S12: Inferring feedback weights is biased towards dominant neural modes. A-B. Feedback learning results for within-(WMP) (A) and outside-manifold perturbation (OMP) (B). CC describes the correlation coefficient between inferred and correct feedback weights. Same data as in Figure 2.4B-C. **C.** Feedback weight vectors for x- and y-component of cursor dynamics can be expressed in terms of principal components of neural dynamics. With this, we can compare feedback learning performance, dependent on the related neural mode. Per definition, the most prominent neural modes in the system lie in the first principal component axes. **D-E.** Distribution of projections into the principal component basis for correct feedback (x-component) for within-manifold perturbation (D) and outside-manifold perturbation (E). **F.** Comparison of projections into the principal component basis for correct and learned feedback vectors (x-component). **G.** Same as (F) but zoomed in to show only the first 20 principal components. **H-J.** Absolute value of projection into principal component basis for inferred and correct feedback weights for within- (H) and outside-manifold perturbation (J). The inlets show the same plot as (H-J), but focussing on principal components 1-20.

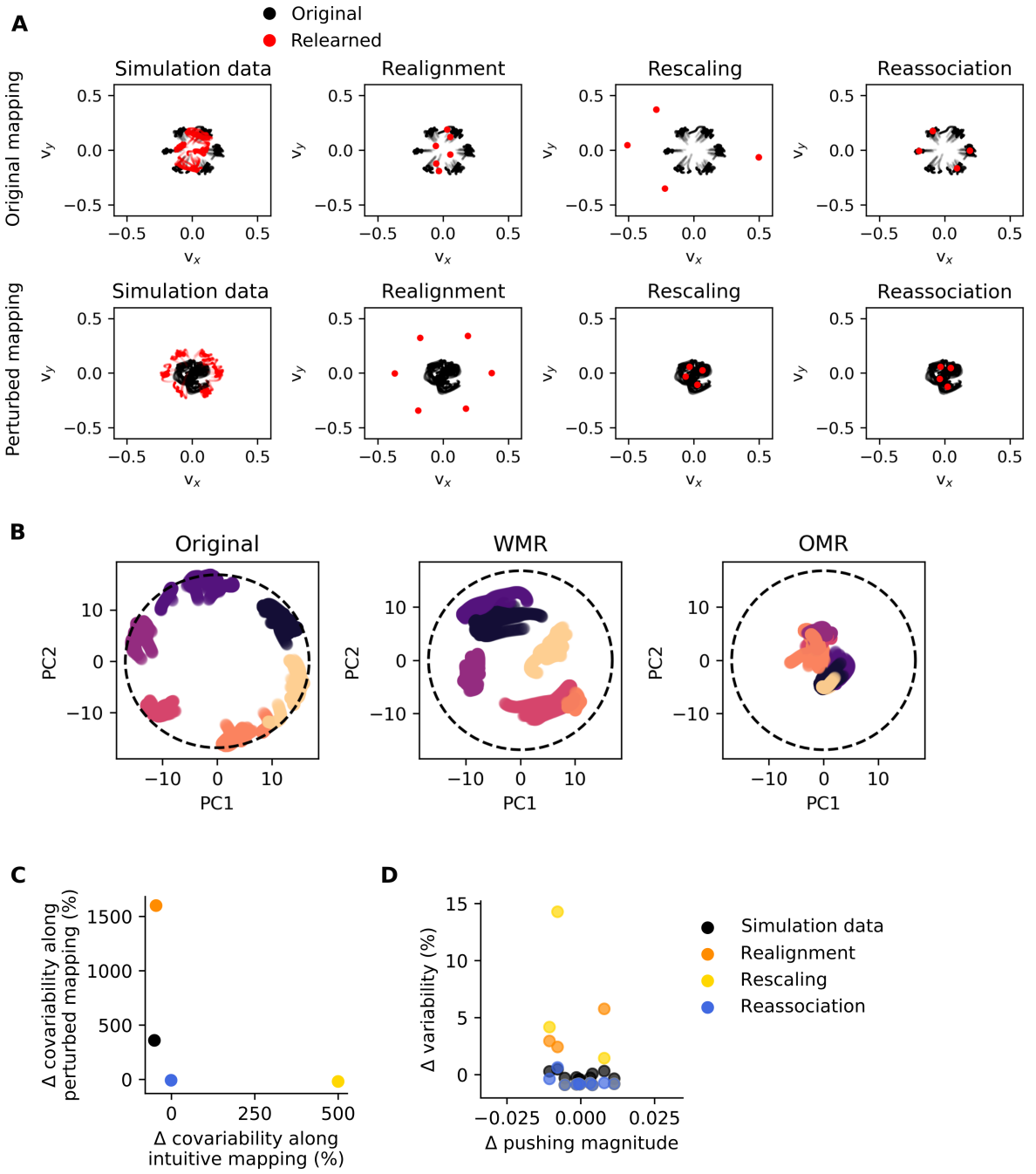


Figure 2.S13: Simulation results for within-manifold learning are in line with reassociation hypothesis[132]. **A.** Cursor velocities under original (upper row) and perturbed mapping (lower row) for original (black) and retrained network (red). Comparison between simulation data (first column), realignment hypothesis (second column), rescaling hypothesis (third column) and reassociation hypothesis (fourth column)[132]. Detailed methods are described at the end of this section. **B.** Original neural activity (left), neural activity after within-manifold relearning (middle) and neural activity after outside-manifold relearning (right) (cf. Figure 2.2). Neural activity is projected into the original neural manifold and the first two principal components are shown. Different colors correspond to different target locations. Dashed black circle represents boundary of initial repertoire. **C.** Change in covariability along original and perturbed mapping. **D.** Change in variability of neural modes dependent on the imposed change in the BCI readout (pushing magnitude).

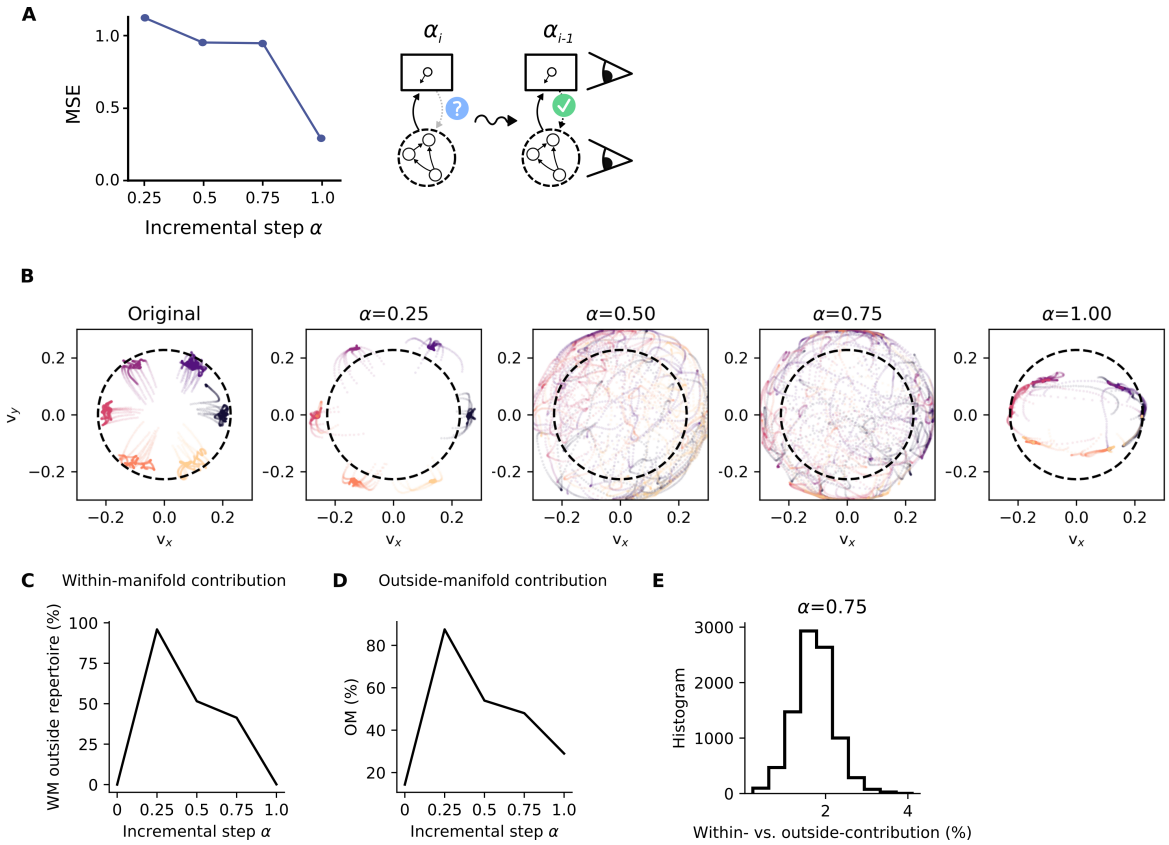


Figure 2.S14: Within-manifold contributions to incremental outside-manifold learning[120]. **A.** Relearning performance under incremental strategy (Figure 2.5E). **B.** Cursor velocities under the original mapping for each incremental training step. Dashed black circle shows boundary of initial repertoire. Colours represent different targets. **C.** Percentage of within-manifold patterns which lie outside of the original repertoire (dashed black circle in (B)). **D.** Percentage of variability measured outside of original manifold as a function of incremental step α. **E.** Histogram of percentage of within- versus outside- manifold contributions. For each relearned neural activity pattern which was outside of the original manifold or repertoire we calculated the projection onto the original manifold d_{WM} (within-manifold contribution) and subtracted this projection from the total neural activity vector to obtain the residual vector d_{OM} (outside-manifold contribution). This gave two distances in the high dimensional space of neural activity. To calculate the percentage of within-manifold contribution we calculated $d_{WM}^2/(d_{WM}^2 + d_{OM}^2)$.

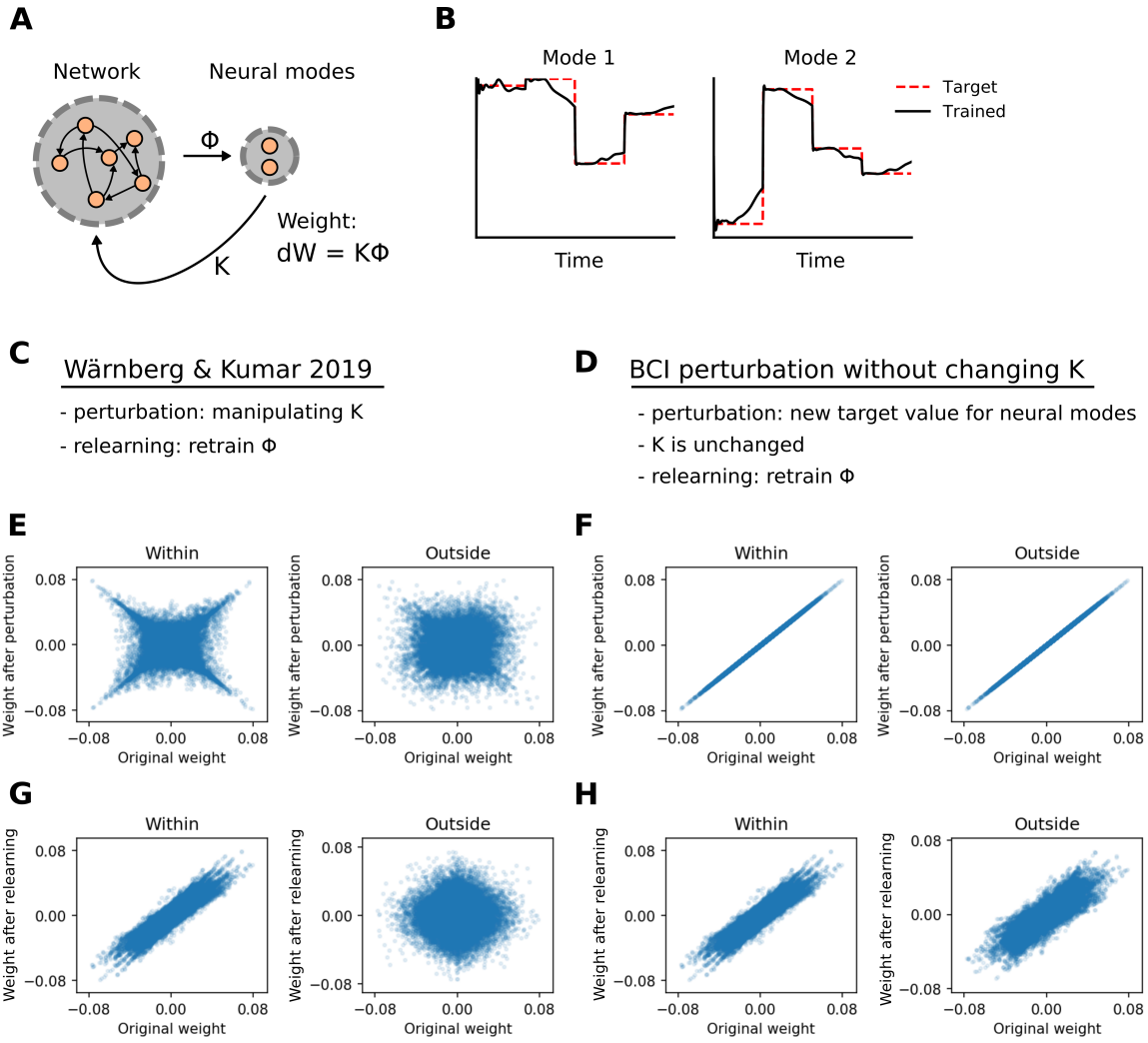


Figure 2.S15: Difference to previous study investigating the experimental results of Sadtler et al.[121]. A. Reimplementation of Waernberg and Kumar[121]. Neural modes are decoded from a random recurrent network with static weights. The neural mode signal is fed back into the recurrent network via a feedback matrix K. Network training consists of finding the decoding weights ϕ which produce a given target signal at the neural mode level. FORCE is used to adapt the decoding weights accordingly[110]. **B.** Resulting neural mode dynamics after initial training. **C.** To implement the BCI experiment from Sadtler et al. Waernberg and Kumar manipulated the feedback matrix K and retrained the decoding weights ϕ in order to produce the original target dynamics on the mode level[121]. **D.** Alternative implementation of the BCI perturbation which is closer to our simulations. Here, the BCI perturbation is implemented by setting a new target signal on the mode level, without manipulating K. **E.** Weight change after perturbation, without retraining, for implementation of Waernberg and Kumar. **F.** Weight change after perturbation, without retraining, for alternative implementation. **G.** Weight change after retraining for implementation of Waernberg and Kumar. **H.** Weight change after retraining for alternative implementation.

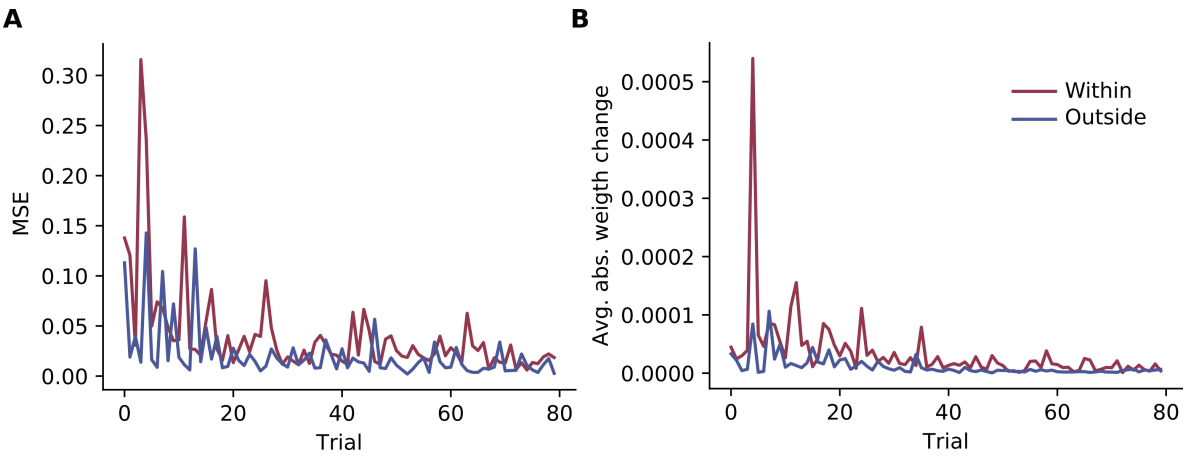


Figure 2.S16: Timecourse of relearning is similar for within- and outside-manifold perturbation under the recursive-least-squares learning rule. A. Mean squared error (MSE) during relearning for within- and outside-manifold perturbation. The MSE is summed over all update steps in a trial, which constitutes 90 steps. **B.** Amount of weight change measured during relearning for within- and outside-manifold perturbation.

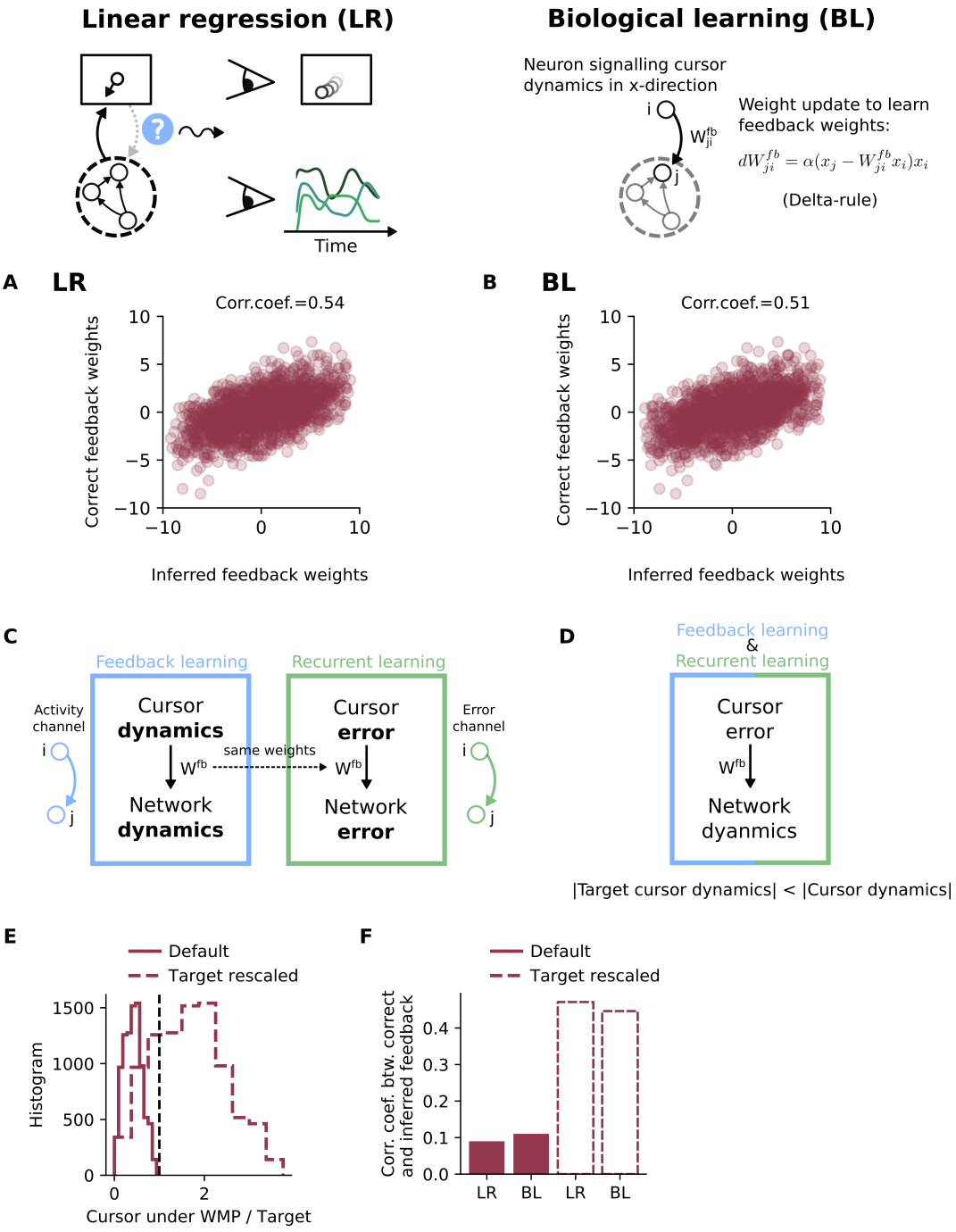


Figure 2.S17: Learning feedback weights via a biological learning rule. **A.** Inferred feedback weights using linear regression (LR) are compared to correct feedback weights, which are given by the pseudo-inverse of the BCI mapping. In linear regression, weight factors are estimated to predict neural dynamics from cursor dynamics. **B.** Linear regression can be implemented by a simple, biologically plausible, learning rule (BL). Learned feedback weights, using a biological learning rule, are compared to correct feedback weights. **C.** Illustration of relearning steps: 1) Feedback weights are inferred from observing neural and cursor dynamics. 2) The learned feedback weights (W^{fb}) are then used to propagate the error form cursor to network level. **D.** Alternative way of inferring feedback weights. Instead of predicting neural dynamics from cursor dynamics, neural dynamics could also be predicted by the observed error signal in cursor dynamics. **E.** Cursor dynamics under within-manifold perturbation are in general smaller than the target values. Rescaling target values by 0.25 (dashed line) reverses the relation. **F.** Feedback learning performance, measured as correlation coefficient between correct and inferred feedback weights, when cursor error signal is used to predict neural dynamics.

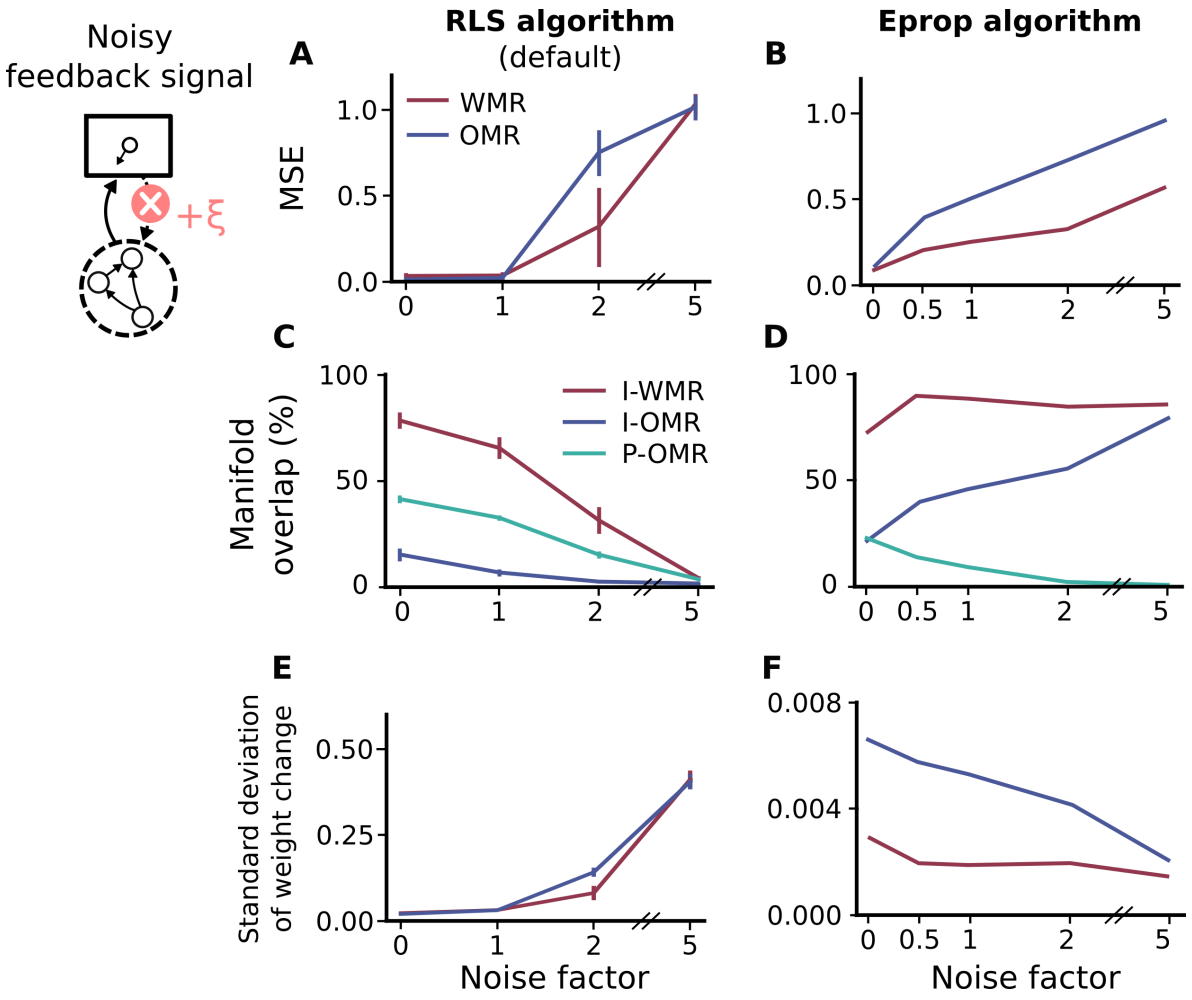


Figure 2.S18: Relearning under noisy feedback signal using recursive-least-squares algorithm and local learning algorithm[130]. A-B. Relearning performance, measured as mean squared error (MSE), as a function of the amplitude of the noise in the feedback signal using recursive-least-squares (RLS) algorithm (A) and an alternative implementation with a local learning algorithm (Eprop) (B). **C-D.** Manifold overlap between original manifold and manifold after within learning (I-WMR), original manifold and manifold after outside learning (I-OMR) and manifold defined by BCI perturbation and manifold after outside learning (P-OMR). **E-F.** Weight change during relearning. **A,C,E.** Simulations with RLS (same as Figure 2.3B in the main paper). **B,D,F.** Simulations with Eprop learning rule.

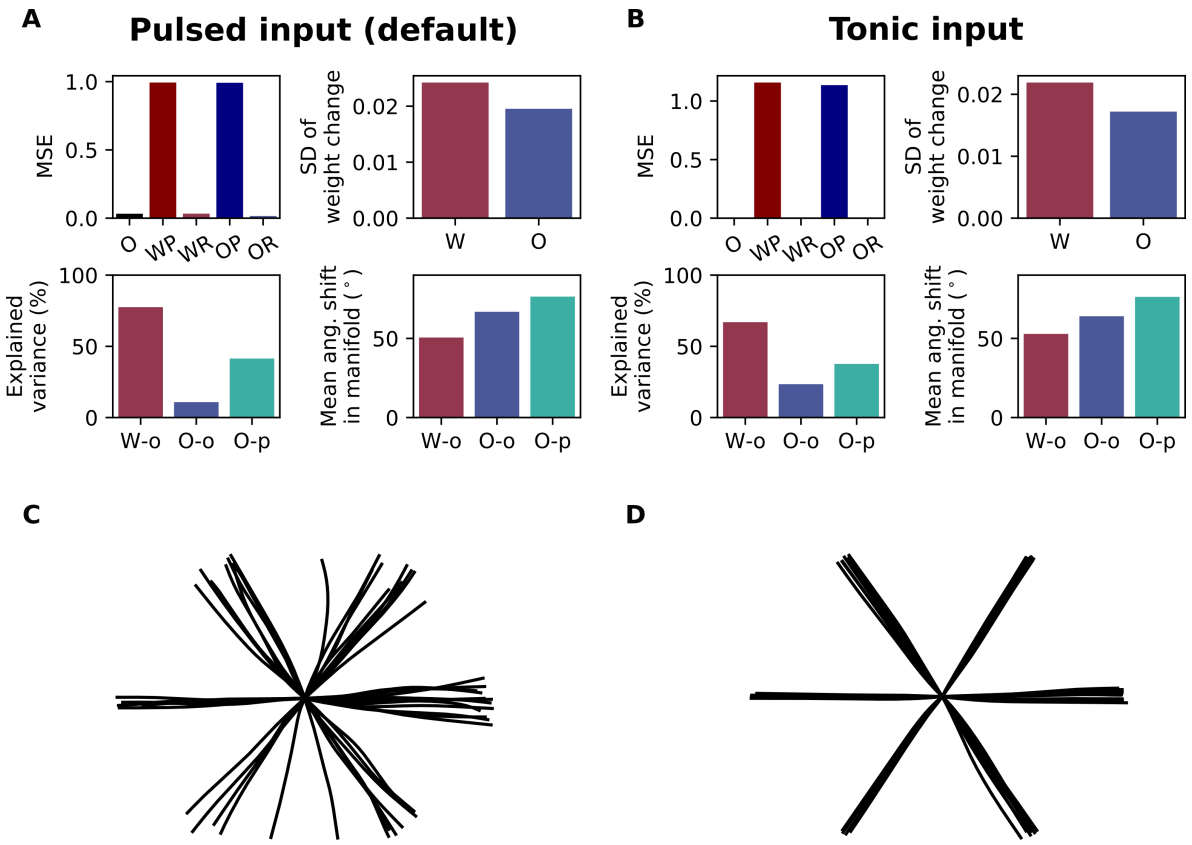


Figure 2.S19: Simulation results are robust to how inputs are modelled. **A-B.** Upper left panel shows task performance, measured as mean squared error (MSE), after original training (O), after within-manifold perturbation (WP), after within-manifold retraining (WR), after outside-manifold perturbation (OP) and after outside-manifold retraining (OR). Upper right panel shows the standard deviation of the weight change distribution between before and after retraining for within- (W) and outside-manifold (O) retraining. Lower left panel shows the manifold overlap between original and retrained manifold for within- (W-o) and outside-manifold (O-o) perturbation, as well as the overlap between retrained and target manifold for outside-manifold perturbations (O-p). Lower right panel shows the mean principal angle between the same manifold as in lower left panel. **A.** Result for simulations where inputs are modelled as a pulse (same as Figure 2.S1A). **B.** Results for simulations where inputs are modelled as tonic input. **C-D.** Reconstructed cursor trajectories for pulsed input simulations (C) and tonic input simulations (D).

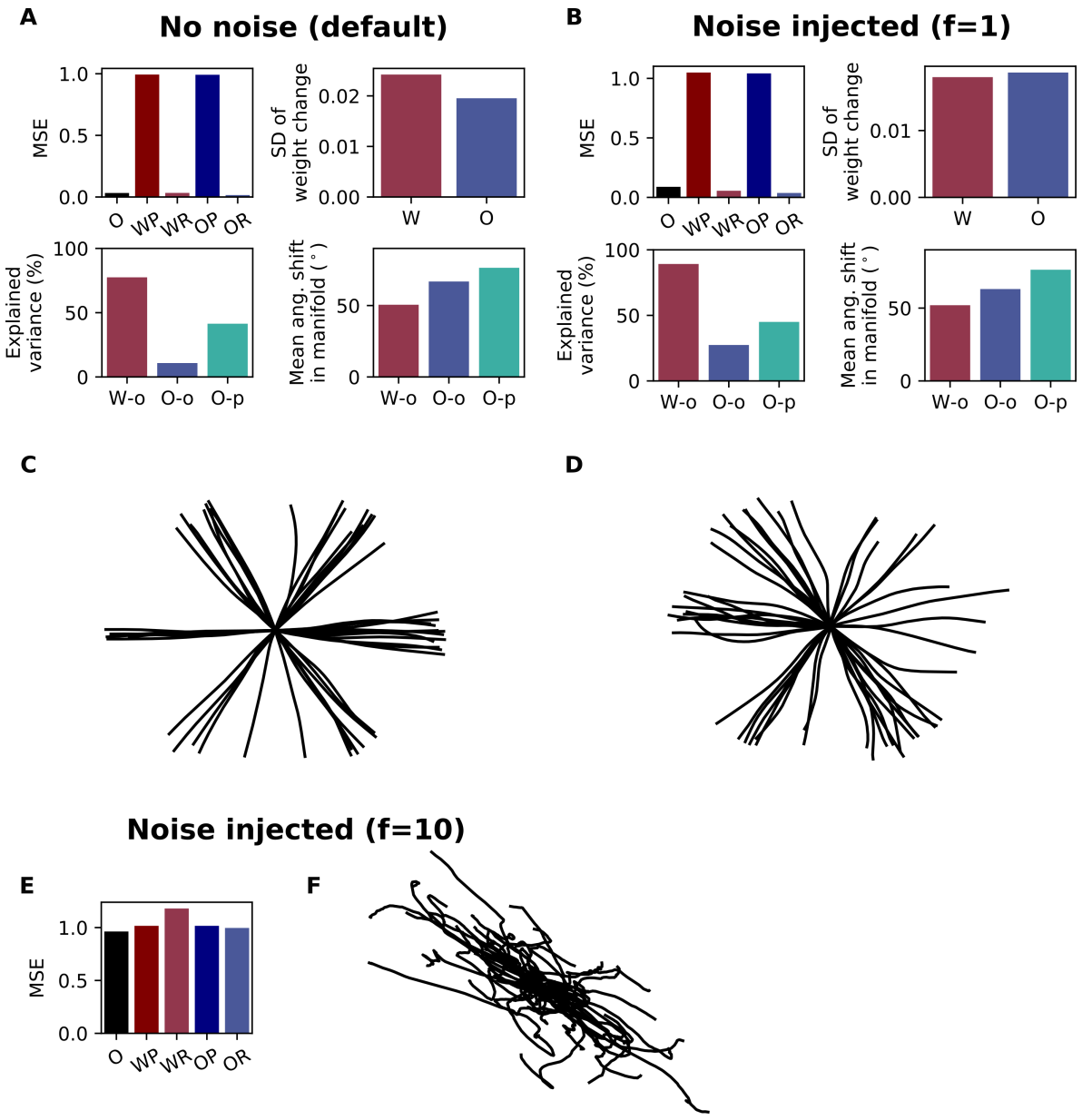


Figure 2.S20: Simulation results are robust to noise. A-B. Upper left panel shows task performance, measured as mean squared error (MSE), after original training (O), after within-manifold perturbation (WP), after within-manifold retraining (WR), after outside-manifold perturbation (OP) and after outside-manifold retraining (OR). Upper right panel shows the standard deviation of the weight change distribution between before and after retraining for within- (W) and outside-manifold (O) retraining. Lower left panel shows the manifold overlap between original and retrained manifold for within- (W-o) and outside-manifold (O-o) perturbation, as well as the overlap between retrained and target manifold for outside-manifold perturbations (O-p). Lower right panel shows the mean principal angle between the same manifold as in lower left panel. **A.** Standard simulations where no noise is injected (same as Figure 2.S1A). **B.** At each time step each neuron receives an extra input, drawn from a normal distribution with zero mean and standard deviation f. The noise for each neuron is independent. **C.** Reconstructed trajectories for simulations without noise. **D.** Reconstructed trajectories when noise (f=1) is injected. **E-F.** Network cannot be trained to perform the task if noise is too high (f=10).

Simplified model to investigate feedback learning

In order to understand why it is possible to learn feedback weights for a within-manifold perturbation, but not for an outside-manifold one, we developed and analysed a simplified setup. We created a synthetic dataset, which has no temporal structure, but is simply drawn from a multivariate Gaussian distribution. The number of variables was matched to the number of neurons in the simulated network, such that there was 800 random variables ξ in total. Without loss of generality, we set the mean to zero, as we wanted to analyse the effect of the covariance structure on the ability to perform feedback learning. To create the covariance matrix Σ of the synthetic data, we firstly created the desired Eigenvalue spectrum which is given by

$$f(x;\gamma) = \exp^{-x/\gamma}$$

where γ parametrizes how fast the spectrum decays and $x = 0$ correspond to the first Eigenvalue, $x = 1$ to the second and so forth. After setting the diagonal values of the covariance matrix, we performed a basis transformation which puts the synthetic data in the same Eigenspace as found in the simulated data.

$$\Sigma = C^T \operatorname{diag}(f(0;\gamma), f(1;\gamma), ..., f(n-1;\gamma))C$$

$$\xi \sim \mathcal{N}((0,0,...,0), \Sigma)$$

To simulate a within- or an outside-manifold BCI transformation, we used the Eigenvector space C of the data and either took the first d Eigenvectors (within-manifold) or the first d Eigenvectors after shuffling the entries of each vector (outside-manifold). For Figure 2.S11 the number of modes d - which are decoded - is set to 10 to match the number of modes used in the main paper.

Adaptation of analysis performed in Golub et al. 2018[132]

To test whether our simulation results for within-manifold learning are in line with further experimental observations (Golub et al. 2018), we performed an additional analysis dissecting the strategy the neural network is using to compensate for a within-manifold perturbation. In the original paper,

Golub et al. focused on three different hypotheses: 1) Realignment, 2) Reassociation and 3) Rescaling. They constructed neural activity patterns according to these three different types of neural adaptation and compared it to their experimental recordings. We followed a similar path, constructing artificial data according to the three hypotheses and comparing it to our simulation data. The methods to create artificial data according to a given hypothesis are adapted from the original paper (finding the target-specific mean under a certain hypothesis) (Golub et al. 2018) and will briefly be explained below.

Realignment

To find the neural activity patterns which would correspond to the realignment hypothesis we searched each target for patterns that maximize the cursor velocity along the specific target direction, yet still obey the initial boundaries of neural activity in our model (-1,1). As a result, we obtained an estimate for the adapted mean neural activity pattern per target under the realignment hypothesis.

Reassociation

Under the reassociation hypothesis, adapted neural activity patterns were found by selecting from the set of initial activity patterns the best pattern for the current target and under the current perturbation. What is optimized here is again the target-specific cursor velocity, as described before. Solving this constrained optimization problem leads to estimates for the target-wise mean neural activity pattern under the reassociation hypothesis.

Rescaling

Finding the estimates for the target-wise mean neural activity pattern under the rescaling hypothesis is similar to the reassociation hypothesis. Here however, initial neural activity patterns are rescaled according to the change in BCI readout, before the constrained optimization is performed. The scale factor for neuron i is given by $f_i = \frac{\|T_{:,i}\|_2}{\|T^{WM}_{:,i}\|_2}$ where T is the BCI mapping before perturbation and T^{WM} is the BCI mapping after the within-manifold perturbation.

CHAPTER 3

Learning location

Understanding learning in the brain is not only hard because of the complexity of single neurons or the intricacies of synaptic plasticity, but also on a more coarse-grained level because of the modularisation into brain areas. Can we say *where* - in which brain region - learning is happening for a given task? Classic work suggests that motor adaptation is implemented in the cerebellum[137–140]. However, there is no definitive answer to whether or not this is the only plastic brain region contributing to adaptation. In contrast, synaptic plasticity in M1 has been traditionally more associated to long-term skill learning, not adaptation[123,141–143]. In this chapter we test whether we can verify or falsify one or both of these intuitions based on neural recordings during a motor adaptation task. Or in other words: whether we can derive if learning has happened locally or upstream by solely analysing local neural activity changes.

The work in this chapter is presented in an article format as published in Nature Communications in 2022[97]. I would like to thank my supervisors Juan Alvaro Gallego and Claudia Clopath for their guidance and inspiration throughout this project. Further, I would like to thank our experimental collaborators Matthew G. Perich, Lee E. Miller and Raeed H. Chowdhury for kindly providing their data and wisdom.

Small, correlated changes in synaptic connectivity may facilitate rapid motor learning

3.1 Abstract

Animals rapidly adapt their movements to external perturbations, a process paralleled by changes in neural activity in the motor cortex. Experimental studies suggest that these changes originate from altered inputs (H_{input}) rather than from changes in local connectivity (H_{local}), as neural covariance is largely preserved during adaptation. Since measuring synaptic changes *in vivo* remains very challenging, we used a modular recurrent neural network to qualitatively test this interpretation. As expected, H_{input} resulted in small activity changes and largely preserved covariance. Surprisingly given the presumed dependence of stable covariance on preserved circuit connectivity, H_{local} led to only slightly larger changes in activity and covariance, still within the range of experimental recordings. This similarity is due to H_{local} only requiring small, correlated connectivity changes for successful adaptation. Simulations of tasks that impose increasingly larger behavioural changes revealed a growing difference between H_{input} and H_{local}, which could be exploited when designing future experiments.

3.2 Introduction

Animals, particularly primates, can perform a great variety of behaviours, which they are able to adapt rapidly in the face of changing conditions. Since behavioural adaptation can happen even after a single failed attempt[144], the neural populations driving this process must be able to adapt equally fast. How this occurs remains unexplained[145]. Rapid motor learning is typically studied

using external perturbations such as a visuomotor rotation (VR), which rotates the coordinates of the visual feedback with respect to those of the movement. Both humans and monkeys can learn to compensate for the resulting error between actual and expected visual feedback in a few tens of trials[70,146]. The behavioural adaptation is accompanied by changes in the activity of neurons in primary motor cortex (M1)[147], and the upstream dorsal premotor cortex (PMd)[146]. It is unclear whether these neural activity changes are mediated by synaptic weight changes within the motor cortices or are driven by altered inputs from even further upstream areas.

When learning a skill over many days, behavioural improvements are paralleled by rewiring between M1 neurons[123,141–143]. This seems not to be the case for rapid learning: throughout VR adaptation, the statistical interactions across neural populations in both M1 and PMd remain largely preserved[107]. These preserved interactions rule out any large synaptic changes within the motor cortices, as they would cause these models to degrade[148,149]. Instead, rapid VR adaptation may be driven by the cerebellum[137–140] and/or posterior parietal cortex[150,151].

A pioneering Brain Computer Interface (BCI) study cast further doubt that significant synaptic changes occurring within M1 are necessary for rapid learning[33,132]. In that study, monkeys controlled a computer cursor linked by a "decoder" to the activity of recorded M1 neurons. After learning to use a decoder that used the natural "intrinsic" mapping of neural activity onto cursor movements, the monkeys were exposed to one of two types of perturbations. When faced with a new decoder that preserved the statistical interactions (i.e., neural covariance) across M1 neurons, the monkeys could master it within minutes. In stark contrast, if the new decoder required changes in the neural covariance (an "out of manifold" perturbation), they could not learn it within one session – in fact, it required a progressive training procedure spanning just over nine days on average[152].

Recording large scale synaptic changes *in vivo* remains challenging and has not been achieved during rapid motor learning. Alternatively, recurrent neural network (RNN) models offer an exciting yet unexplored opportunity to test the effect of synaptic changes (in the model) on simulated activity during motor learning. RNNs trained on motor, cognitive and BCI tasks exhibit many striking similarities with the activity of neural populations recorded in animal studies[42,60,61,65,99,153,154], suggesting a fundamental similarity between the two. Previous work using RNNs to model the BCI experiment described above[33] showed that network covariance can be highly preserved even when learning is happening through weight changes within the network[96]. Thus, contrary to intuition,

functionally relevant synaptic weight changes may not necessarily lead to measurable changes in statistical interactions across neurons[155]. As a consequence, synaptic changes within PMd and M1 during VR adaptation may be very hard to identify through the analysis of neural population recordings.

Here, we used RNN models to test whether VR adaptation might be mediated by synaptic changes within PMd and M1, yet with largely preserved neural covariance within these areas. We addressed this question by asking how adaptation based on connection weight changes within PMd and M1 (H_{local}) alters network activity compared to the corresponding activity changes if VR adaptation is based on altered inputs from upstream areas (H_{input})[107,137–140,150,151] (Figure 3.1A). To validate our modelling results, we compared our simulations to experimental recordings from PMd and M1 populations during the same VR task[107].

Under H_{local}, the changes in covariance following VR adaptation only slightly exceeded those under H_{input} and were comparable to experimental observations. Thus, when using neural population recordings alone, it may be more challenging to disentangle these two hypotheses than previously thought. For both H_{input} and H_{local}, the learned connectivity changes were small and highly coordinated, which made them surprisingly robust to noise. To identify additional differences between H_{input} and H_{local}, we examined learning during paradigms requiring larger behavioural changes. Covariance changes were larger for these paradigms in both PMd and M1 under H_{input}, but only in M1 under H_{local}, thus providing a possible way to distinguish between the two hypotheses in future experiments. Our findings have implications for the interpretation of neural activity changes observed during learning, and suggest that tasks eliciting larger behavioural changes may be necessary to elucidate how neural populations adapt their activity during rapid learning.

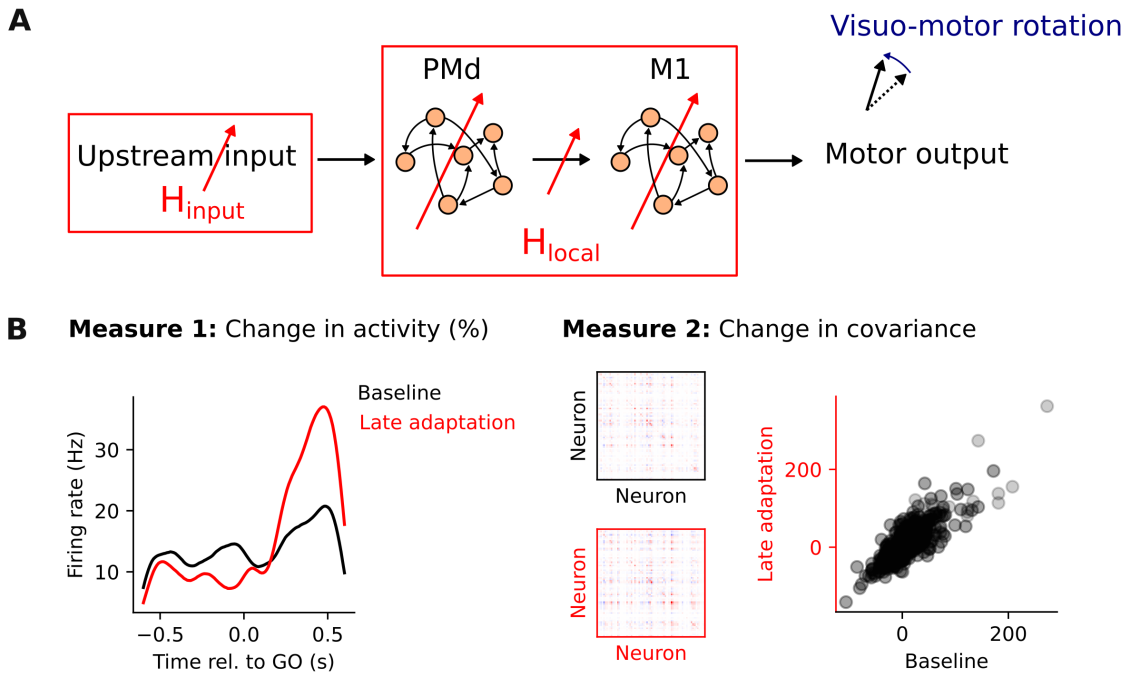


Figure 3.1: Competing hypotheses to explain where learning happens during a visuomotor rotation task. A. To study the processes mediating motor cortical activity changes during adaptation in a visuomotor rotation task, we analyse and model the activity of neural populations within dorsal premotor cortex (PMd) and primary motor cortex (M1). We compare two hypotheses: plasticity upstream of PMd/M1 (H_{input}) and plasticity within PMd/M1 (H_{local}). **B.** Measures to quantify the changes in neural activity following adaptation: 1) relative change in trial-averaged single neuron activity; 2) change in neural covariance. Both measures compare baseline trials to late adaptation trials, after monkeys had adapted to the task.

3.3 Results

To understand whether motor adaptation could be driven by synaptic changes within PMd and M1, we simulated a VR adaptation task using a modular RNN that modelled these two areas, and compared the resulting changes in network activity to those of neural population recordings from PMd and M1 during the same VR task[107]. We quantified neural activity changes both in the experimental data and in the model using two measures (Figure 3.1B): 1) the relative change in trial-averaged single neuron activity, and 2) the change in neural covariance (Methods). Combined, they capture aspects of single neuron as well as population-wide activity changes during adaptation.

Small but measurable changes in neural activity within PMd and M1 during VR adaptation

Monkeys were trained to perform an instructed delay task, in which they reached to one of eight visual targets using a planar manipulandum to receive a reward (Methods). After performing a block of unperturbed reaches (200-243 trials, depending on the session) , visual feedback about the position of the hand was rotated by 30°, either clockwise or counterclockwise, depending on the session. Monkeys adapted rapidly to these perturbations: the curved reaches observed immediately after the perturbation onset became straighter after tens of trials, with the hand trajectories in the second (late) half of the adaptation block becoming more similar to the baseline trajectories (Figure 3.2A). The angular error quantifying the difference between initial reach direction and target location decreased during adaptation (Figure 3.2B). This error curve followed a similar trend for clockwise and counterclockwise perturbations, allowing us to analyse the different perturbations together.

Behavioural adaptation was accompanied by changes in neural activity within both PMd and M1 (Figure 3.2C)[107]. These changes exceeded those during control sessions, where no perturbation was applied (Figure 3.2C black; linear mixed model analysis: t=4.4, P=0.0017). The amount of change was greater within PMd than M1 (t=8.9, $P<0.0001$). We also found small but detectable changes in neural covariance during VR perturbation, suggesting that the statistical interactions among neurons change slightly during adaptation (Figure 3.2D). Again, these changes exceeded those of the control sessions (Figure 3.2D black; t=2.6, P=0.026).

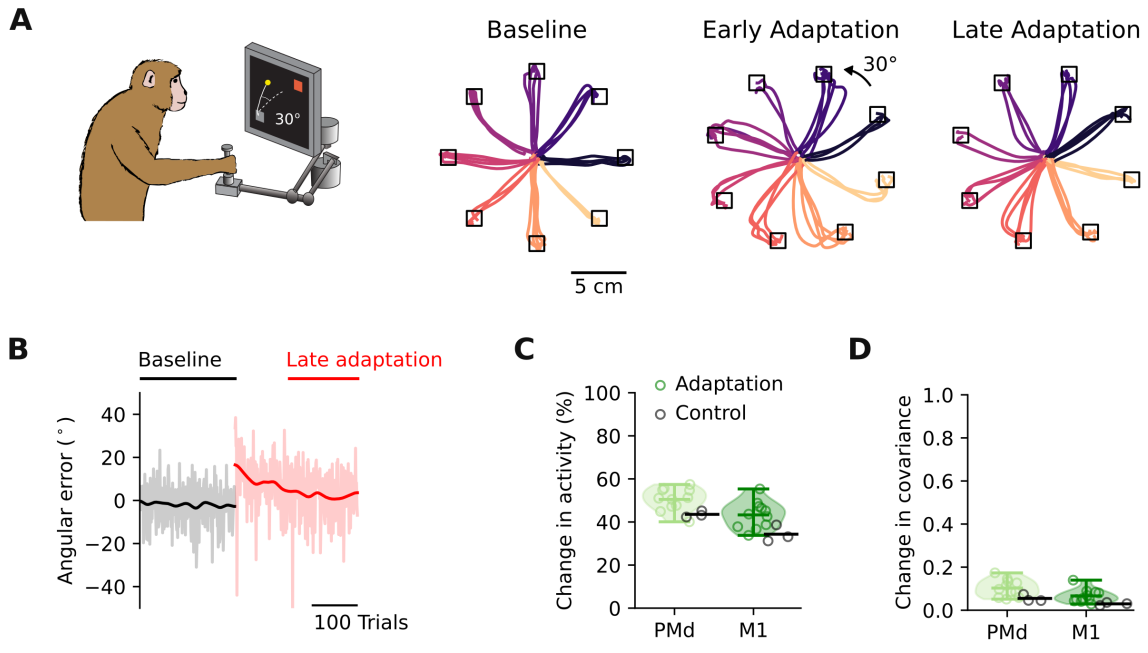


Figure 3.2: Small but measurable changes in neural activity within PMd and M1 during VR adaptation. A. Hand trajectories during the first 30 trials of the baseline, early adaptation (first 150 adaptation trials) and late adaptation epoch (last 150 adaptation trials). Trajectories are color-coded by target. Data from Monkey C. **B.** Angular error of the hand trajectories for the example session in (A) has the typical time course of adaptation. **C.** Change in trial-averaged activity following adaptation. Data pooled across all sessions from the two monkeys for PMd and M1 separately (green markers, 11 sessions, 5 Monkey C, 6 Monkey M). Control sessions during which no perturbation was applied are shown for comparison (black markers, 3 sessions, 1 Monkey C, 2 Monkey M). Shaded area and horizontal bars, data distribution with mean and extrema (n=11 experimental sessions). **D.** Change in covariance following adaptation. Same format as C. The monkey image was created by Carolina Massumoto who gave permission to use it under CC-BY license.

A modular recurrent neural network model to study VR adaptation

To test whether experimentally observed changes in motor cortical activity could be driven by rapid synaptic plasticity[143] within PMd and M1, we trained a modular RNN model[42,61] to perform the centre-out reaching task that we studied experimentally. To mimic broadly the hierarchical architecture of the motor cortical pathways, input signals were sent to the PMd module which then projected to the M1 module to produce the final output signal (Figure 3.3A; Methods). After initial training on the task, the model produced correct reaching trajectories to each of the eight different targets (Figure 3.3B and Figure 3.S1). These RNN-controlled movements had the same dynamics as those of monkeys (Figure 3.3C). Furthermore, Principal Component Analysis (PCA; Methods)

revealed that the population activity of the PMd and M1 network modules was similar to that of the corresponding recorded neural populations (Figure 3.3D-E). We used Procrustes analysis[156] to quantify this apparent similarity between model and experimental population activity (Figure 3.S2). This analysis confirmed that the modular RNN captured the area-specific features in the neural data accurately, as the PMd and M1 modules better explained neural data from the respective brain area compared to a cross-area control (Figure 3.S2).

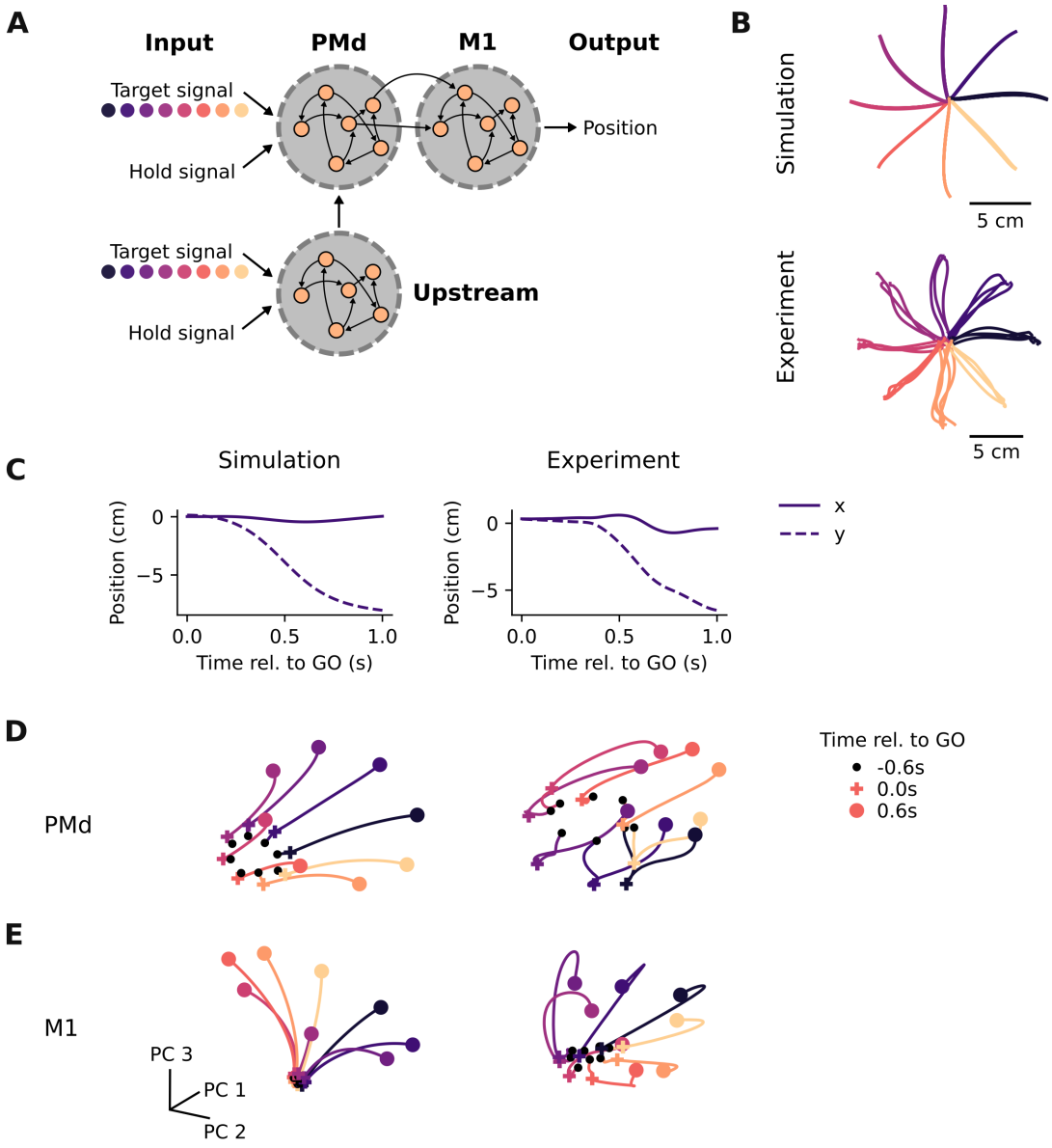


Figure 3.3: A modular recurrent neural network model to study VR adaptation. A. A modular RNN that models key motor cortical areas to study adaptation. **B.** Simulated (top) and actual (bottom) hand trajectories during 30 reaches to each target taken from one session from Monkey C. **C.** Example simulated and actual hand trajectories to one target. Note the similarity in kinematics between the model and the experimental data. **D.** Simulated PMd population activity recapitulates key features of actual PMd population activity. Neural trajectories extend from 600 ms before the go cue (black dots) to 600 ms after the go cue (coloured dots); go cue is indicated with coloured crosses. Reaching targets are colour-coded as in B. **E.** Same as D for M1.

Motor adaptation through altered inputs matches neural recordings

After having verified that our modular RNN recapitulates the key aspects of PMd and M1 population activity during reaching, we simulated the VR adaptation experiment. The model was retrained to produce trajectories rotated by 30°, replicating the perturbation monkeys had to counteract. Having full control of the location of learning-related changes, we first constrained it to happen upstream of PMd (H_{input}). As anticipated from previous modelling[151] and experimental work[107], changes in areas upstream of the motor cortices can lead to successful adaptation: the hand trajectories produced after learning were correctly rotated by 30° to counteract the perturbation (Figure 3.4A).

When examining the activity of each of the PMd and M1 modules, the relative change in network activity was similar in magnitude to the changes observed in the corresponding neural population recordings (Figure 3.4B and Figure 3.S3). PMd activity changed slightly more than M1 activity (Figure 3.4B), indicating a relation between the two modules that was also present in the experimental data (Figure 3.2C). With respect to interactions between neurons, covariance within each module was strongly preserved (Figure 3.4C), as was the case for the experimental data (Figure 3.2D). VR adaptation through altered inputs to the motor cortices thus is very similar to the neural activity changes observed *in vivo*.

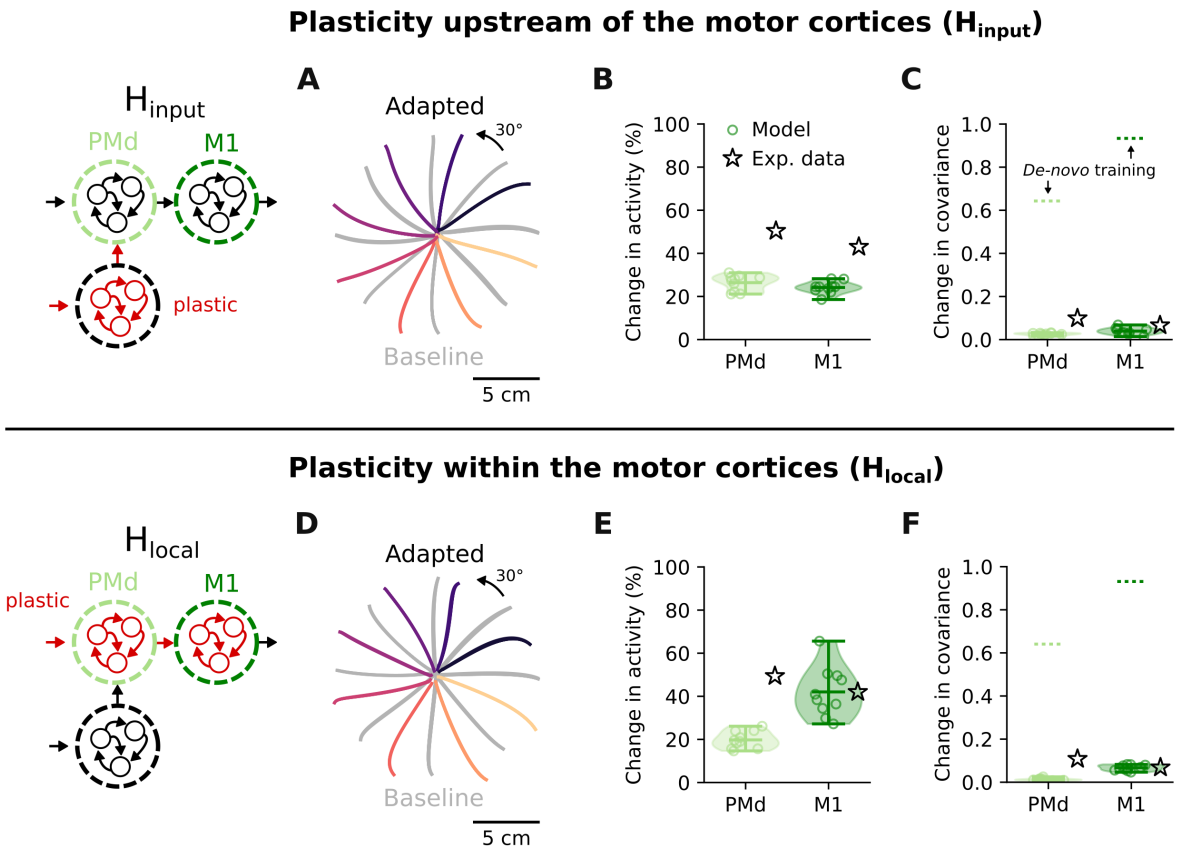


Figure 3.4: Activity changes following learning upstream (H_{input}) and within (H_{local}) the motor cortices. **A.** Hand trajectories after learning under H_{input} (coloured traces; baseline trajectories are shown in grey). **B.** Changes in trial-averaged activity following adaptation under H_{input} (green markers) for PMd and M1, and reference mean experimental values (black stars; same data as presented in Figure 3.2). Shaded area and horizontal bars, data distribution with mean and extrema (n=10 network initialisations). **C.** Change in covariance following adaptation under H_{input}, and reference values for change in covariance following the initial *de-novo* training (dashed lines). Data presented as in B. **D.** Hand trajectories after learning under H_{local}. **E.** Change in trial-averaged activity following adaptation under H_{local}. **F.** Change in covariance following adaptation under H_{local}. Data in D,E,F are presented as in A,B,C.

Learning through plastic changes within PMd and M1 modules occurs despite preserved the covariance

Our simulation results so far are consistent with experimental[107,137–139] and modelling[151] studies proposing that VR adaptation is mediated by regions upstream of the motor cortices. But can our model rule out the alternative that adaptation is instead implemented by recurrent connectivity changes within PMd and M1 (H_{local})?

To test this question, we implemented H_{local} by constraining learning to happen only within PMd

and M1, a process which also led to successful adaptation (Figure 3.4D). Interestingly, the activity changes produced under H_{local} differed both from those of H_{input} and the experimental data: there were larger changes in the M1 module than in the PMd module (Figure 3.4E). However, learning based on recurrent weight changes within PMd and M1 did not lead to large changes in covariance, which was largely preserved (Figure 3.4F), virtually as much as when no local plasticity was allowed (H_{input}) (Figure 3.4C). Thus, the intuition that preserved covariance should be interpreted as a sign of stable underlying connectivity may be misleading.

Small but coordinated connectivity changes enable motor adaptation

We wished to understand how the model can adapt to the VR perturbation by changing the recurrent connectivity within the PMd and M1 modules without altering their covariance. Interestingly, the connectivity changes under both H_{local} and H_{input} (Figure 3.5A,B) were small relative to experimentally observed synaptic changes[157]: an average weight change of 1-2% was sufficient regardless of whether they happened upstream of (Figure 3.5C and Figure 3.S4) or within the motor cortical modules (Figure 3.5E and Figure 3.S5). These changes were smaller than those observed during initial training (4 to 31%), when the model learned to perform the reaching task from random connection weights (Figure 3.S6). Thus, "functional connectivity" within the PMd and M1 modules, as measured here by their covariance, may be largely preserved after VR adaptation under H_{local} because network connection weights change very little (Figure 3.S7).

We next studied how such small changes in connection weights could nevertheless drive effective behavioural adaptation. Recent studies seeking to relate RNN activity and connectivity have highlighted the importance of low-dimensional structures in connectivity, showing their explanatory power for understanding how tasks are solved[54,158–160]. Inspired by this work, we looked for low-dimensional structure in the connectivity changes emerging in the model during adaptation (Methods). Our analysis revealed that the connectivity change patterns of all plastic modules were low-dimensional, independent of where learning happened (Figure 3.5B,D,F). We thus hypothesized that the small changes were effective because they were low-dimensional. To test this, we examined how random changes in the connection weights (noise), which are inherently high-dimensional, would affect the behaviour.

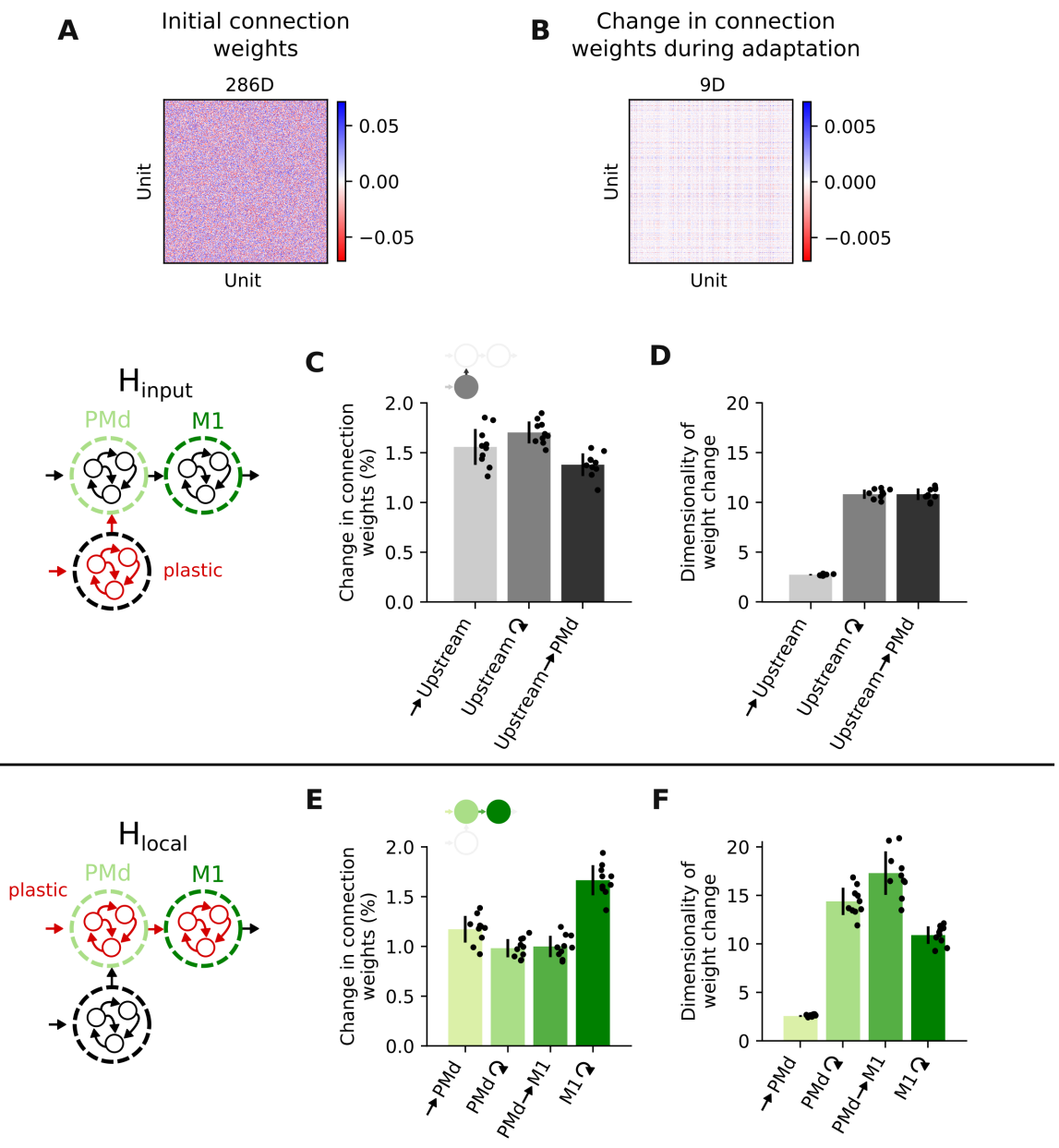


Figure 3.5: Small but coordinated connectivity changes enable motor adaptation. **A.** Example connection weights for the M1 module after initial training. Top: estimated dimensionality. **B.** Example changes in M1 connection weights following VR learning under H_{local}. Same format as A. **C.** Change in connection weights following adaptation under H_{input}. Each bar summarizes results for either one module of the network or a set of cross-module connections; bars and error bars, mean and s.d. (n=10 network initialisations). **D.** Estimated dimensionality of connection weight changes for each network module and cross-module connections; data presented as in C. **E.** Change in connection weights following adaptation under H_{local}. **F.** Dimensionality of connection weight changes under H_{local}. Data in E,F are presented as in C,D, respectively.

Low-dimensional connectivity changes are highly robust to noise

For both H_{local} and H_{input}, the learned connectivity changes in the model were small and low-dimensional. When considering the biological plausibility of our model, this observation raises the question of how such small connectivity changes could compete with ongoing synaptic fluctuation, which is a known challenge for actual brains[161–164]. To test the hypothesis that the low-dimensionality of the learned connectivity changes is what makes them highly effective, we tested how adding synaptic fluctuations, which are inherently high-dimensional, would affect motor output. Simulating synaptic fluctuations by applying random perturbations to the learned connectivity changes increased the dimensionality of the weight changes (Figure 3.6B,G; Methods), but did not lead to any observable change in reaching kinematics (Figure 3.6C) or network activity (Figure 3.6D-E). This was the case even though the applied random perturbations in connectivity were ten times bigger in magnitude than the learned connectivity changes (Figure 3.6F), completely masking them (Figure 3.6A-B). Therefore, our model not only suggests that VR adaptation can be implemented based on coordinated synaptic weight changes within PMd and M1, but also that this type of learning would be highly effective due to its robustness to synaptic fluctuation.

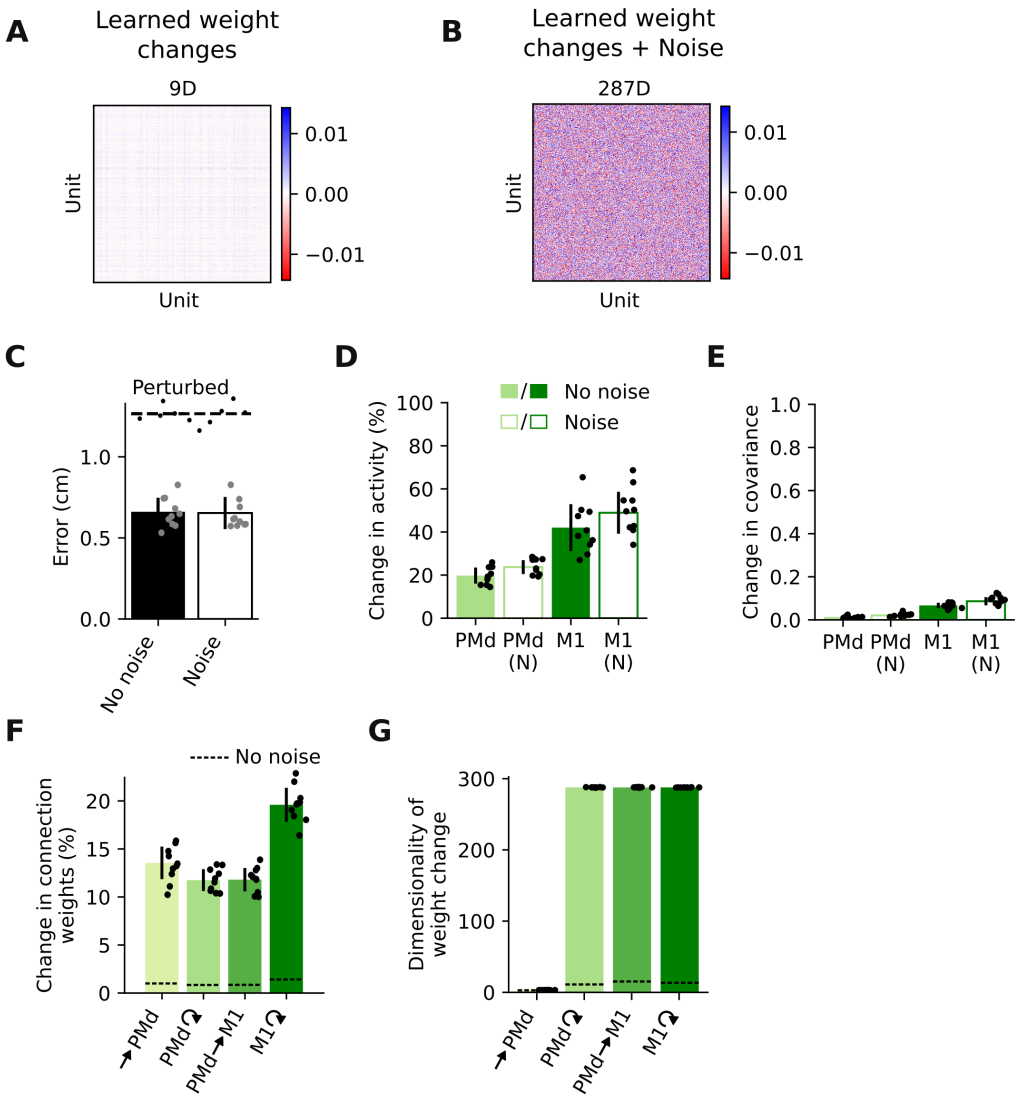


Figure 3.6: Low-dimensional connectivity changes are highly robust to noise. A. Example changes in M1 connection weights following VR adaptation. Same data as in Figure 5B. **B.** Same connection weight changes as in A, but with random connectivity changes added. Note the dramatic increase in the dimensionality of the connection weight changes. **C.** Root mean squared error between target and produced hand trajectories following adaptation in models with and without random weight changes; bars and error bars, mean and s.d. (n=10 network initialisations), as in all the panels in this figure. Dashed line shows error under the VR perturbation without any learning. **D.** Change in trial-averaged activity for PMd and M1 module, without (solid) and with (empty) random weight changes. **E.** Change in covariance. Same format as D. **F.** Change in connection weights for each network module and cross-module connections in models with (green bars) and without (dashed lines) noise in connectivity. **G.** Dimensionality of connection weight changes. Same format as F.

Larger visuomotor rotations allow for a clearer distinction between H_{input} and H_{local}

Although neural activity changes during VR adaptation were better reproduced by a model in which learning happens upstream of the motor cortices (H_{input}), activity changes following learning through weight changes within the motor cortices (H_{local}) also lined up surprisingly well with the experimental data. To verify that the stable covariance (Figure 3.4C,F) is not a general feature of the model but reflects task-specific demands, we modelled tasks for which we would expect larger changes.

We first asked the network to learn larger VRs of 60° and 90° instead of the original 30° rotation (Figure 3.7A). The model was able to compensate for these larger perturbations under both H_{input} and H_{local} (Figure 3.7B,E). As expected, larger perturbations led to changes in network activity and covariance that increased with rotation angle (Figure 3.7B,C,F,G). For the 90° rotation, we found a clear difference between H_{input} and H_{local}: H_{input} produced larger activity changes in PMd compared to M1, opposite that under H_{local} (Figure 3.7C,F). Larger rotation angles also increased the learning-related difference in covariance between H_{input} and H_{local}. Under H_{input}, the increase in covariance was similar for the PMd and M1 modules as the rotation increased (Figure 3.7D). In contrast, under H_{local}, the M1 covariance changed more with increasing rotation angle than did that of PMd (Figure 3.7G). These model predictions could help differentiate between H_{input} and H_{local} in future experiments. In fact, preliminary M1 population recordings obtained during larger VRs (45° and 60°) seemed to match the model predictions for the covariance change under H_{input} (Figure 3.7D stars), but not H_{local} (Figure 3.7G stars).

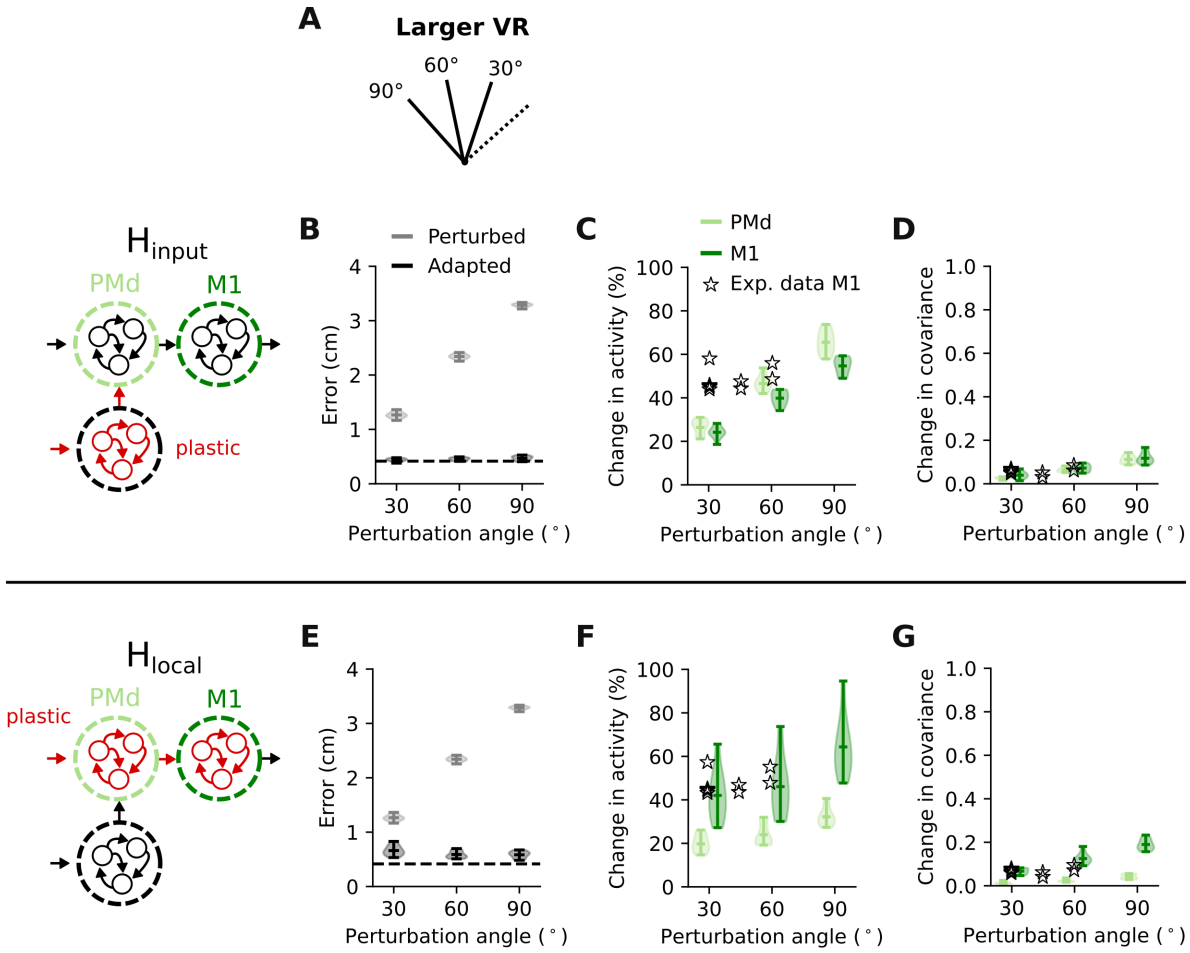


Figure 3.7: Larger visuomotor rotations allow for a better distinction between H_{input} and H_{local}. A. To verify that the modelled perturbation does not always produce small activity changes, we tested adaptation to larger VR perturbations (60° and 90°). **B.** Root mean squared error between target and produced hand trajectories without (grey) and with learning (black) under H_{input}. Dashed line, error after initial training, with no perturbation applied; shaded area and horizontal bars, data distribution with mean and extrema (n=10 network initialisations). Same data presentation in all panels. **C.** Change in trial-averaged activity for PMd and M1 under H_{input}. A few experimental sessions from Monkey C with larger rotations are shown as comparison (stars). **D.** Change in covariance following adaptation. Data are presented as in C. Note the similarity between PMd (light green) and M1 (dark green) across all rotation angles. **E.** Error without (grey) and with learning (black) under H_{local}. Same format as B. **F.** Change in trial-averaged activity for PMd and M1 under H_{local}. **G.** Change in covariance following adaptation. Data in E,F,G are presented as in B,C,D.

A visuomotor reassociation task can differentiate between H_{local} and learning through remapping of input signals

Although larger visuomotor rotations help differentiate between upstream learning and learning within PMd and M1, we sought to identify a task that would lead to an even clearer distinction. To

this end, we implemented a reassociation task where the model had to learn a new, random mapping between cues and reaching directions (Figure 3.8A; Methods). This task allowed us to test a very specific change in the input signal to the motor cortices that could implement adaptation[132,165]: instead of adjusting the connectivity in an upstream network (H_{input}), which allows for highly unconstrained modulation of input signals, the target-related input signals were manually reordered to compensate for the reassociation of cue-reaching direction pairs (Figure 3.8B). This "learning through input reassociation" resulted in large changes in network activity (Figure 3.8C), comparable in magnitude to those under H_{local} (Figure 3.8F). Nevertheless, it did not cause any change in covariance (Figure 3.8D), which clearly distinguished it from H_{local} (Figure 3.8G) and the standard H_{input} (Figure 3.S8). This was the case because, in contrast to the standard H_{input} during VR adaptation, the input signals did not change per se, but were only reassigned to different targets, thereby entirely preserving the network activity patterns.

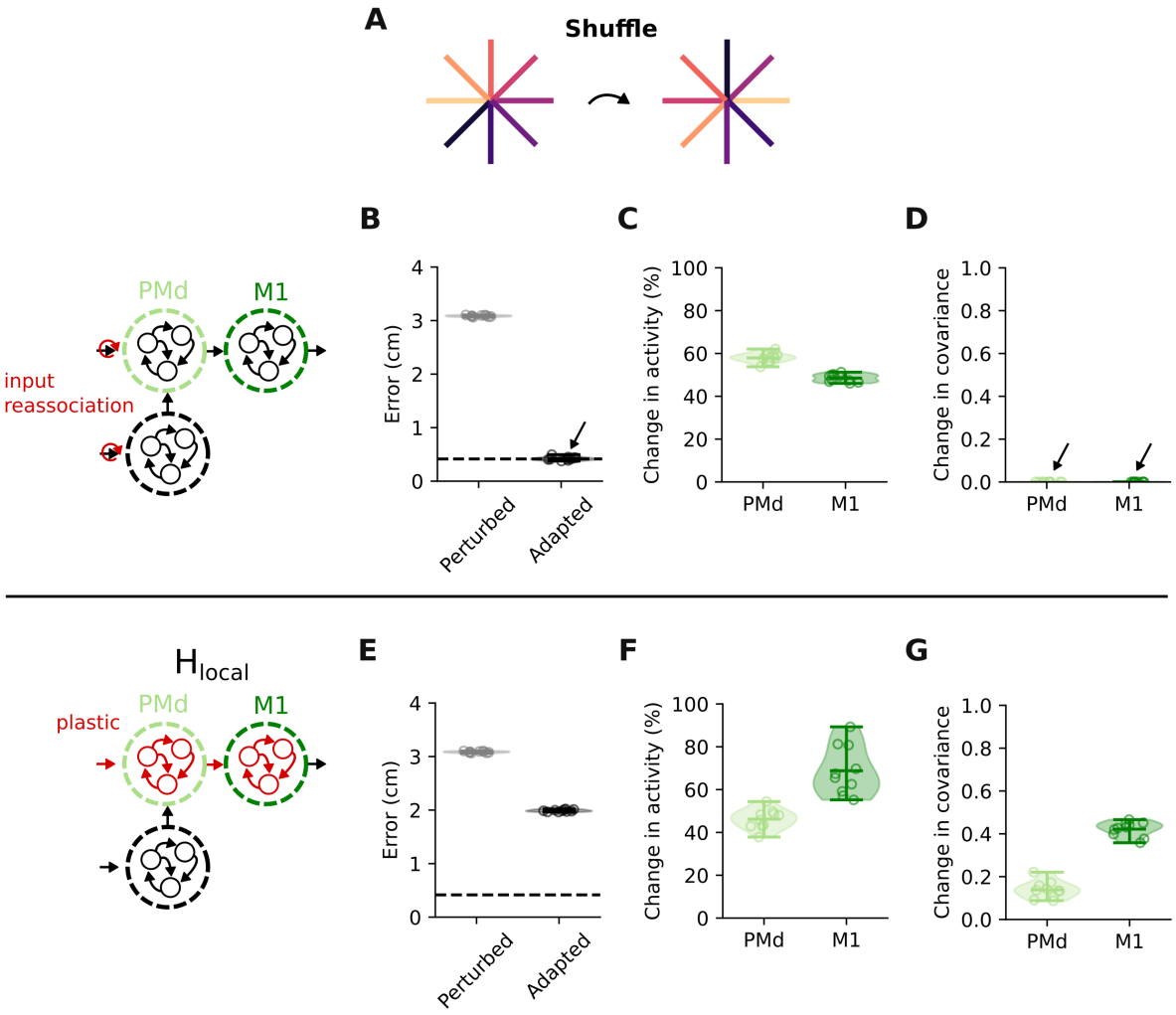


Figure 3.8: A visuomotor reassociation task can clearly differentiate between H_{local} and learning through reassociation of input signals. **A.** We simulated a task in which the network had to learn new associations between target locations and reach directions ("Reassociation"). **B.** Root mean squared error between target and produced hand trajectories without (grey) and with learning (black) through input reassociation. Dashed line indicates error during baseline trial. Shaded area and horizontal bars, data distribution with mean and extrema (n=10 network initialisations). Same data presentation in all panels. **C.** Change in trial-averaged activity for PMd and M1 under input reassociation. **D.** Change in covariance following adaptation. Note that the covariance matrices do not change at all. **E.** Root mean squared error between target and produced hand trajectories without (grey) and with learning (black) under H_{local}. **F.** Change in trial-averaged activity for PMd and M1 under H_{local}. **G.** Change in covariance following adaptation. Data in E,F,G are presented as in B,C,D .

3.4 Discussion

Rapid motor learning is associated with neural activity changes in the motor cortices. The origin of these changes remains elusive, due to the current challenge of measuring synaptic strength *in vivo*. Here, we have used modular RNNs to simulate the motor cortices and to explore whether learning to counteract a visuomotor rotation within tens of minutes could be mediated by local synaptic changes (H_{local}). By comparing the modelled network activity changes under H_{local} to the modelled changes observed during learning upstream of the motor cortices (H_{input}), we have shown how the two hypotheses could be distinguished based on neural population recordings during behaviour. Critically, despite the intuition that learning through plastic changes should lead to detectable changes in neural interactions within and across PMd and M1 populations, both H_{local} and H_{input} (Figure 3.4) largely preserved the covariance within these two regions, closely matching experimental observations (Figure 3.2). This likely happened because adaptation under H_{local} was achieved through small, coordinated weight changes within the PMd and M1 network modules (Figure 3.5). Finally, using our model, we propose tasks for which we anticipate a more dramatic difference between these contrasting hypotheses (Figure 3.7 and Figure 3.8) which can potentially help to interpret experimental data in the future.

Electrophysiological[107,137–139] and modelling studies[151], as well as psychophysical evidence[144,166] suggest that VR adaptation is driven by areas upstream of the motor cortices. Neurophysiological evidence is largely based on the observation that the statistical interactions within PMd and M1 populations remain preserved throughout adaptation[107]. This conclusion is in good agreement with studies showing that learning to generate neural activity patterns that preserve the covariance structure only takes a few tens of minutes[33]. Our direct comparison between H_{input} and H_{local} lends further support to this observation. However, it also paints the intriguing picture that small, globally organized changes in synaptic weights could enable rapid learning without changing the neural covariance, a result that was robust across model initializations (Figure 3.4), parameter settings (Figure 3.S9 and Figure 3.S10), architectural design choices (Figure 3.S11) and learning algorithms (Figure 3.S12). Even implementing the modular RNN as a spiking neural network, bringing it closer to biology, did not change this result (Figure 3.S13). Our simulations thus robustly show that covariance stability is not as directly linked to stable local connectivity as previously thought, as changes in

covariance were comparable between H_{input} and H_{local} for a 30° VR perturbation (Figure 3.4). Instead, the change in neural covariance seemed to be more related to the task itself, as it correlated with the size of the perturbation: the larger the initial error (e.g., caused by larger rotations), the larger the change in covariance (Figure 3.7). However, the relation between initial error and change in covariance differed depending on where the learning happened (H_{input} or H_{local}).

The main difference between the two learning hypotheses we have examined is where in the hierarchical RNN model the connectivity changes occur: within the motor cortices (H_{local}), or upstream of them (H_{input}). Although neural covariance was preserved similarly by H_{local} and H_{input}, we found a key characteristic that distinguished the two. When local connectivity was allowed to be plastic, the largest activity changes happened within the M1 module, with only small changes in the PMd module (Figure 3.4E). In contrast, when learning occurred upstream of the PMd and M1 modules, the activity changes were similar in PMd and M1 (Figure 3.4B), even if some learning was also allowed within PMd and M1 (Figure 3.S10 and Figure 3.S14). The experimental data, with larger activity changes in PMd than M1, better matched the pattern produced by H_{input}. This observation lends further support for the hypothesis that VR adaptation is mediated by plasticity upstream of the motor cortices.

A more arbitrary visuomotor reassociation task allowed us to test an alternate way in which upstream learning could occur, with constraints against input signals changing, but simply being reassigned to different targets (Figure 3.8). Comparing this learning to that mediated by local connectivity changes revealed a clear distinction: learning under H_{local} modified the covariance in both PMd and M1, whereas learning through input reassociation preserved it[132]. Thus, future experiments seeking to disentangle to which extent learning happens within the motor cortices and/or upstream could study this task.

Studies of learning in RNNs have focused on how networks implement *de-novo* training[30,42,58,59,61,63,65,113,160,167,168]. However, our brain does not learn to perform any task from scratch; it has been "trained" over many generations throughout evolution[169]. Here we studied how neural networks adapt a learned behaviour, as opposed to *de-novo* learning. Our work raises the intriguing possibility that rapid learning following a few tens of minutes of practice could be easily achieved through small but specific changes in circuit connectivity. Thus, initial training seems to provide a highly flexible backbone to adapt behaviour as needed[170].

The fact that the connectivity changes during adaptation under both H_{local} and H_{input} were small and low-dimensional (Figure 3.5 and Figure 3.6) suggests that either one could mediate rapid learning. First, as every synaptic change is costly[171], we would expect a constraint on the total amount of connectivity change in the brain. The VR task being solved with only minor weight changes reflects this; in fact, they could be well achieved through long-term potentiation or depression of existing synapses, as experiments have shown that synaptic strength can double within minutes[157]. Second, the low dimensionality of these weight changes is also important with respect to solving "credit assignment", the problem of determining how each synapse should change in order to restore the desired behaviour[86,128,172,173]. Although it is still unclear how this is achieved in the brain, one possibility is that synaptic plasticity is guided by "teacher" signals[174,175]. Since neuromodulatory signals can regulate synaptic plasticity[176,177], they seem ideal candidates to regulate biologically plausible learning[165,178–180]. The finding that the connectivity changes needed to adapt to the VR perturbation are "naturally" low-dimensional is promising, as it suggests that learning could be controlled through relatively few neuromodulatory signals. Such implementation would contrast dramatically with the daunting challenge of learning to regulate every single synapse independently. Lastly, the robustness against synaptic fluctuations conveyed by the low-dimensional connectivity changes makes both H_{local} and H_{input} attractive in terms of ensuring memory stability. Given the fluctuating nature of brain connectivity[161], it remains puzzling how animals remember anything[162–164,181]. That low-dimensional weight changes, much smaller than ongoing synaptic fluctuations, can achieve successful behavioural adaptation provides a potential solution to this problem.

Our model consistently underestimated the changes in trial-averaged activity observed during VR adaptation, despite closely matching the small covariance changes (Figure 3.2 and Figure 3.4). This is to be expected, as the model only captures changes due to the motor adaptation process itself, whereas the actual neural activity contains signals related to other processes such as "impulsivity"[182] or "engagement"[183]. In fact, the experimentally observed neural activity changes between the early and late trials of control reaching sessions with no perturbation were almost as large as the changes during adaptation in our model (Figure 3.2C black dots). How these changes that are not related to learning are combined with the learning-related changes studied here remains unclear. Our modelling predictions for the learning-related changes could help tackle this question in future studies.

A second potential reason why our model consistently underestimated the activity changes during

adaptation could be the fact that we did not include visual or proprioceptive feedback signals in our modelling approach[15,184–186]. As those signals also change during adaptation, they might cause additional changes in trial-averaged neural activity, despite not being directly necessary to solve the motor adaptation task. This could explain why our model could solve the task with smaller changes in neural activity. On the other hand, feedback signals could also actively contribute to the adaptation process. From this view, we may presently overestimate the already small connectivity changes underlying VR adaptation (Figure 3.5), as part of the learning process could have been instead driven by dynamic feedback signals. Thus, when taking feedback into account, rapid learning of a motor perturbation could potentially be realized with even smaller changes in underlying connectivity, or maybe even without any connectivity changes at all[145]. To this point, the concrete role of feedback for rapid motor learning remains unclear and it could be interesting to use our model to further investigate this question.

Our simulations were not designed to study trial-by-trial learning: we were interested in the neural activity changes between the baseline and late adaptation phases when the subjects had largely learned to counteract the perturbation and reached stable behaviour (Figure 3.2B). Given that motor adaptation seems to be mediated by two processes with different timescales[187–189], our model mainly captures the slower of the two. The neural activity changes underlying the early phase adaptation may be driven by different processes[107], which our model currently does not test.

In conclusion, our comparison between the activity changes following VR adaptation through plastic changes within or upstream of the motor cortices shows that local plasticity (H_{local}) leads to neural signatures that are unexpectedly similar to those of upstream learning (H_{input}). Intriguingly, H_{local} not only largely preserved the covariance within PMd and M1 but also resulted in connectivity changes that seem biologically reasonable: they are small, make the network robust against synaptic fluctuations, and can be controlled by relatively few teaching signals. Our simulations thus encourage caution when drawing conclusions from the analysis of neural population recordings during learning, and further suggest potential behavioural tasks that could make it easier to identify where learning is happening.

3.5 Methods

Tasks

We studied motor adaptation using a visuomotor rotation (VR) paradigm, previously described in Perich et al., 2018[107]. Monkeys (*macaca mulatta*) performed an instructed delay center-out-reaching task in which they had to reach to one of eight targets uniformly distributed along a circle. All targets were 2 cm squares. The delay period was variable and ranged between 500 and 1,500 ms. For additional details on the task, see[107]. During the adaptation phase, visual feedback was rotated clockwise or counterclockwise by 30°, 45°, or 60°. All surgical and experimental procedures were approved by the Institutional Animal Care and Use Committee (IACUC) of Northwestern University. Using our modular RNN model, we simulated both this task and a visuomotor reassociation task in which there was no consistent rotation of the visual feedback; instead, each target required reaching to a different direction, uniquely selected from the initial set of eight different targets.

Experimental recordings

We analysed eleven sessions from two monkeys (five for Monkey C, six for Monkey M) that were exposed to a clockwise or counterclockwise 30° rotation (data previously presented in[107]). In addition to these data, we also analysed three control sessions (one for Monkey C, two for Monkey M) in which no perturbation was applied, as well as additional sessions with larger VR angles from Monkey C where only M1 data was collected (30°, nine sessions; 45°, two sessions; 60°, two sessions) (Figure 3.7). The spiking activity of putative single neurons was binned into 10 ms bins and then smoothed using a Gaussian filter (s.d., 50 ms). Only successful trials, where monkeys received a reward at the end, were included in the analysis. We defined the early and late adaptation epochs as the first and last 150 trials of the perturbation phase, when the visuomotor rotation was applied, respectively.

RNN model

Architecture

The neural network contained three recurrent modules, each consisting of 400 neurons, which we refer to as upstream, PMd and M1, respectively (Figure 3.3A). The PMd and the upstream modules

received an identical three-dimensional input signal, with the first two dimensions signalling the x and y target location of that trial, and the third dimension signalling go (1 until the go, and 0 from then on). The upstream module connects to the PMd module and the PMd module connects to the M1 module. The output is calculated as a linear readout of the M1 module activity. Recurrent, as well as feedforward connections were all-to-all. The model dynamics are given by

$$\mathbf{x}_{t+1}^{UP} = \mathbf{x}_t^{UP} + \frac{dt}{\tau}\left(-\mathbf{x}_t^{UP} + \mathbf{W}^{UP}\tanh(\mathbf{x}_t^{UP}) + \mathbf{W}^{in,UP}\mathbf{s}_t\right) \tag{3.1}$$

$$\mathbf{x}_{t+1}^{PMd} = \mathbf{x}_t^{PMd} + \frac{dt}{\tau}\left(-\mathbf{x}_t^{PMd} + \mathbf{W}^{PMd}\tanh(\mathbf{x}_t^{PMd}) + \mathbf{W}^{UP-PMd}\tanh(\mathbf{x}_t^{UP}) + \mathbf{W}^{in,PMd}\mathbf{s}_t\right) \tag{3.2}$$

$$\mathbf{x}_{t+1}^{M1} = \mathbf{x}_t^{M1} + \frac{dt}{\tau}\left(-\mathbf{x}_t^{M1} + \mathbf{W}^{M1}\tanh(\mathbf{x}_t^{M1}) + \mathbf{W}^{PMd-M1}\tanh(\mathbf{x}_t^{PMd})\right) \tag{3.3}$$

$$\mathbf{x}_t^{out} = \mathbf{W}^{out}\tanh(\mathbf{x}_t^{M1}) + \mathbf{b}^{out} \tag{3.4}$$

where $\mathbf{x}^{UP}$ describes the network activity in the upstream module, and $\mathbf{x}^{PMd}$ and $\mathbf{x}^{M1}$ the network activity in the PMd and M1 module respectively. $\mathbf{W}^{UP}$, $\mathbf{W}^{PMd}$ and $\mathbf{W}^{M1}$ define the recurrent connectivity matrix within the upstream module, the PMd module and the M1 module, respectively. $\mathbf{W}^{UP-PMd}$ defines the connectivity matrix from the upstream module to the PMd module, and $\mathbf{W}^{PMd-M1}$ defines the connectivity matrix from the PMd module to the M1 module. The input connectivity matrices for the upstream and the PMd module are given by $\mathbf{W}^{in,UP}$ and $\mathbf{W}^{in,PMd}$, respectively; $\mathbf{s}_t$ represents the three dimensional input signal described above. The two-dimensional output $\mathbf{x}^{out}$ is decoded from the M1 module activity via the output connectivity matrix $\mathbf{W}^{out}$ and the bias term $\mathbf{b}^{out}$. The time constant is $\tau = 0.05$s and the integration time step is $dt = 0.01$s.

Training

Each network was initially trained to produce planar reaching trajectories, mirroring the experimental hand trajectories. The training and testing data set were constructed by pooling the hand trajectories $\mathbf{x}^{target}$ for successful trials during the baseline epochs from all experimental sessions, which resulted in 2238 trials of length 4s (90%/10% randomly split into training/testing). The held out testing data was used to validate that the model had been trained successfully during the initial training period. Model simulations were implemented using PyTorch[190] and training was performed using the Adam optimizer[135] with a learning rate of 0.0001 ($\beta_1 = 0.9$, $\beta_2 = 0.999$). The initial training consisted of 500

training trials. The loss function was defined as

$$L = \frac{1}{B(T-50)2} \sum_{b}^{B} \sum_{t=50}^{T} \sum_{d=x,y} (\mathbf{x}_{t,b}^{out,d} - \mathbf{x}_{t,b}^{target,d})^2 + E^{weights} + E^{rates} \tag{3.5}$$

where the regularization term on the weights is given by ($||\ \ ||$ indicates L2 norm)

$$E^{weights} = \alpha(||\mathbf{W}^{in}|| + ||\mathbf{W}^{out}|| + ||\mathbf{W}^{PMd}|| + ||\mathbf{W}^{M1}|| + ||\mathbf{W}^{PMd-M1}|| + ||\mathbf{W}^{UP}|| + ||\mathbf{W}^{UP-PMd}||) \tag{3.6}$$

the regularization term on the rates is given by

$$E^{rates} = \beta \frac{1}{BTN} \sum_{b}^{B} \sum_{t}^{T} \sum_{n}^{N} \left((\tanh(\mathbf{x}_{t,b}^{PMd,n}))^2 + (\tanh(\mathbf{x}_{t,b}^{M1,n}))^2 + (\tanh(\mathbf{x}_{t,b}^{UP,n}))^2 \right) \tag{3.7}$$

with batch size $B = 80$, time steps $T = 400$ and neurons $N = 400$. The regularization parameters were set to $\alpha = 0.001$, $\beta = 0.8$. We clipped the gradient norm at 0.2 before we applied the optimization step. For the VR adaptation, we trained the initial network for another 100 trials with the target trajectory rotated 30° (or 60° or 90° for the case of the larger VRs).

Data analysis

We quantified the changes in actual and simulated neural activity following adaptation using two measures: changes in trial-averaged activity (or peristimulus time histogram, $PSTH$), and changes in covariance. We calculated both metrics within a window that started 600 ms before the go signal and ended 600 ms after it. The change in activity was calculated by

$$\frac{|PSTH^{Late\ adaptation} - PSTH^{Baseline}|}{\sigma^{Baseline}} \tag{3.8}$$

where $PSTH^{Baseline}$ is the trial-averaged activity in the baseline epoch (experimental data: all baseline trials; simulated data: on a trained model, 100 trials with similar go signal timing), $PSTH^{Late\ adaptation}$ is the trial-averaged activity in the late adaptation epoch (experimental data: last 150 trials of the adaptation epoch; simulation data: on a model trained to counteract the perturbation, 100 trials with similar go signal timing), and $\sigma^{Baseline}$ is the neuron-specific standard deviation across time and targets during the baseline epoch. To summarize the change in trial averaged activity across all neurons, time points, and targets, we calculated their median; this provided one single value for each

experimental session or simulation run. The change in covariance was calculated using the same trial-averaged data from the baseline and the late adaptation epoch. We calculated the covariance in each of these two epochs and then quantified the similarity by calculating the Pearson correlation coefficient between the corresponding entries of the two matrices. The change in covariance is then defined by 1 minus the correlation coefficient. For the experimental sessions, we computed a lower bound for each measure using the control sessions in which monkeys were not exposed to a perturbation. To account for the fact that there could be activity changes unrelated to motor adaptation[182,183], we compared the activity during 150 consecutive trials from the first half of the control session with 150 consecutive trials from the second half of the control session.

To compute the magnitude of the weight changes after networks learned to counteract the perturbation, we computed the average absolute weight change as

$$\mathbf{dW} = \frac{|\mathbf{W}^{Late\ adaptation} - \mathbf{W}^{Baseline}|}{\mathbf{W}^{Baseline}} \tag{3.9}$$

where $|\ \ |$ indicates the element wise absolute value, $W^{Baseline}$ is defined as the model parameter (either $\mathbf{W}^{in}$, $\mathbf{W}^{UP}$, $\mathbf{W}^{UP-PMd}$, $\mathbf{W}^{PMd}$, $\mathbf{W}^{PMd-M1}$, or $\mathbf{W}^{M1}$) after the initial training phase but before training on the VR perturbation, and $\mathbf{W}^{Late\ adaptation}$ is defined as the same model parameter after training on the VR perturbation. To obtain one summary value for each simulation run, we calculated the median of all weight entries for a given parameter. To measure dimensionality of weight change we calculated the singular values k_i of $\mathbf{W}^{Late\ adaptation} - \mathbf{W}^{Baseline}$ and defined the dimensionality, using the participation ratio[109]:

$$(\sum_i k_i)^2 / \sum_i k_i^2 \tag{3.10}$$

Statistics

To statistically compare the change in activity found in the control sessions with the change found in the VR sessions, we performed a linear mixed model analysis using R (lmer package). The brain area (PMd or M1) and whether the experimental session included a perturbation phase or not were included as fixed effects, whereas monkey and session identity were included as random effects. A significance threshold of P=0.05 was used.

Simulation of synaptic fluctuation (Figure 3.6)

To simulate synaptic fluctuation we added random values to the learned connectivity changes during adaptation. Those random values were obtained by taking the learned connectivity changes and randomly shuffling them across connection sites. With that, we created synaptic noise with the same statistics as the learned connectivity changes, yet completely unstructured across connection sites. We did not add or delete any synapses in the model.

3.6 Supporting information

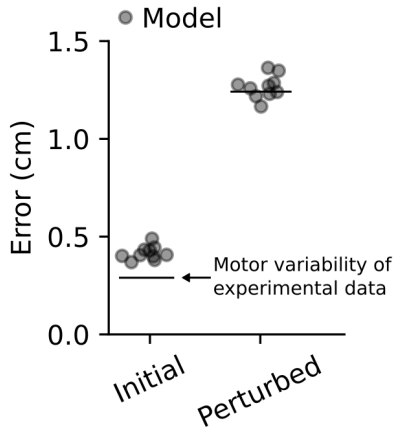


Figure 3.S1: Model kinematics closely match mean target trajectories. Root mean squared error (RMSE) between target and simulated hand trajectories (circles) after initial training (left) and after a VR perturbation is introduced (right). To test where the remaining error after initial training is coming from, we calculated the error relating to the motor variability inherent in the experimental data. For this, we calculated the average RMSE between individual and trial-averaged monkey hand trajectories (lines), resulting in an error close to our model's performance (circles).

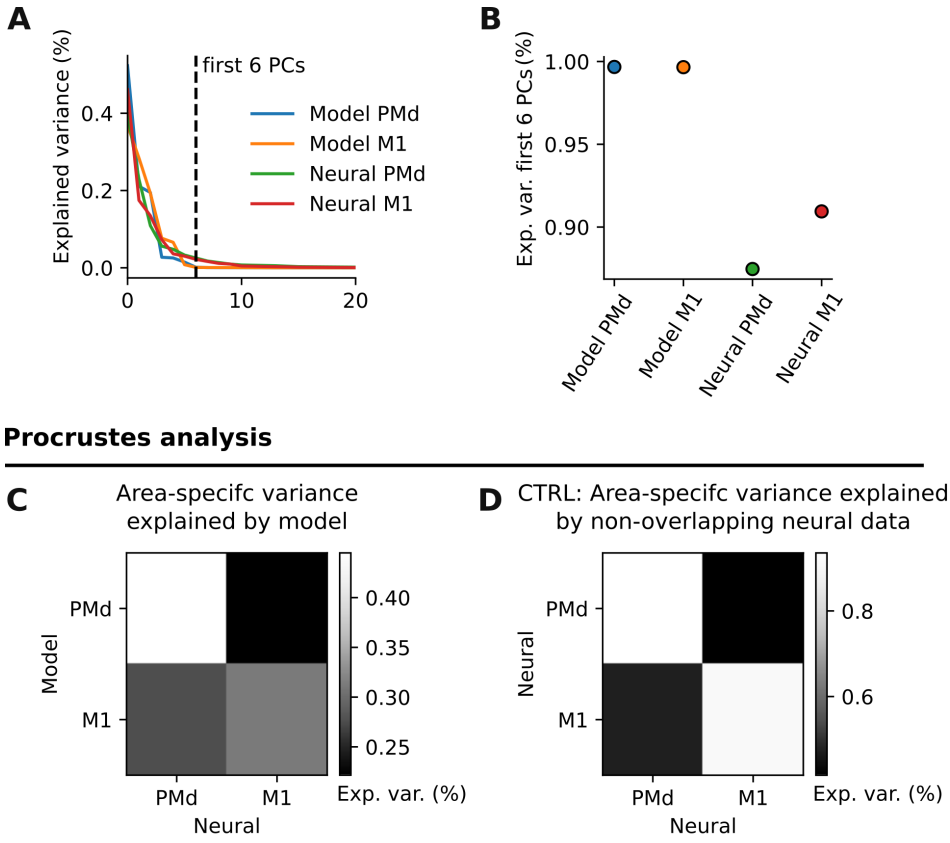


Figure 3.S2: Model activity resembles neural data in an area-specific fashion. A. Percentage of total variance explained for example model PMd (blue), model M1 (orange), monkey PMd (green) and monkey M1 (red). Principal component analysis was performed on trial-averaged activity. **B.** Total percentage of variance explained by the first six principal components for the data in A. **C.** Investigation of the area-specificity of model PMd and M1. Procrustes analysis performed on trial-averaged activity, projected onto the first six principal components. Model PMd best explained monkey PMd, and model M1 best explained monkey M1. **D.** Validation of Procrustes analysis as a measure of similarity of neural activity across different time windows: a comparison of two non-overlapping subsets of trials from monkey PMd and M1 shows that each area most resembles itself.

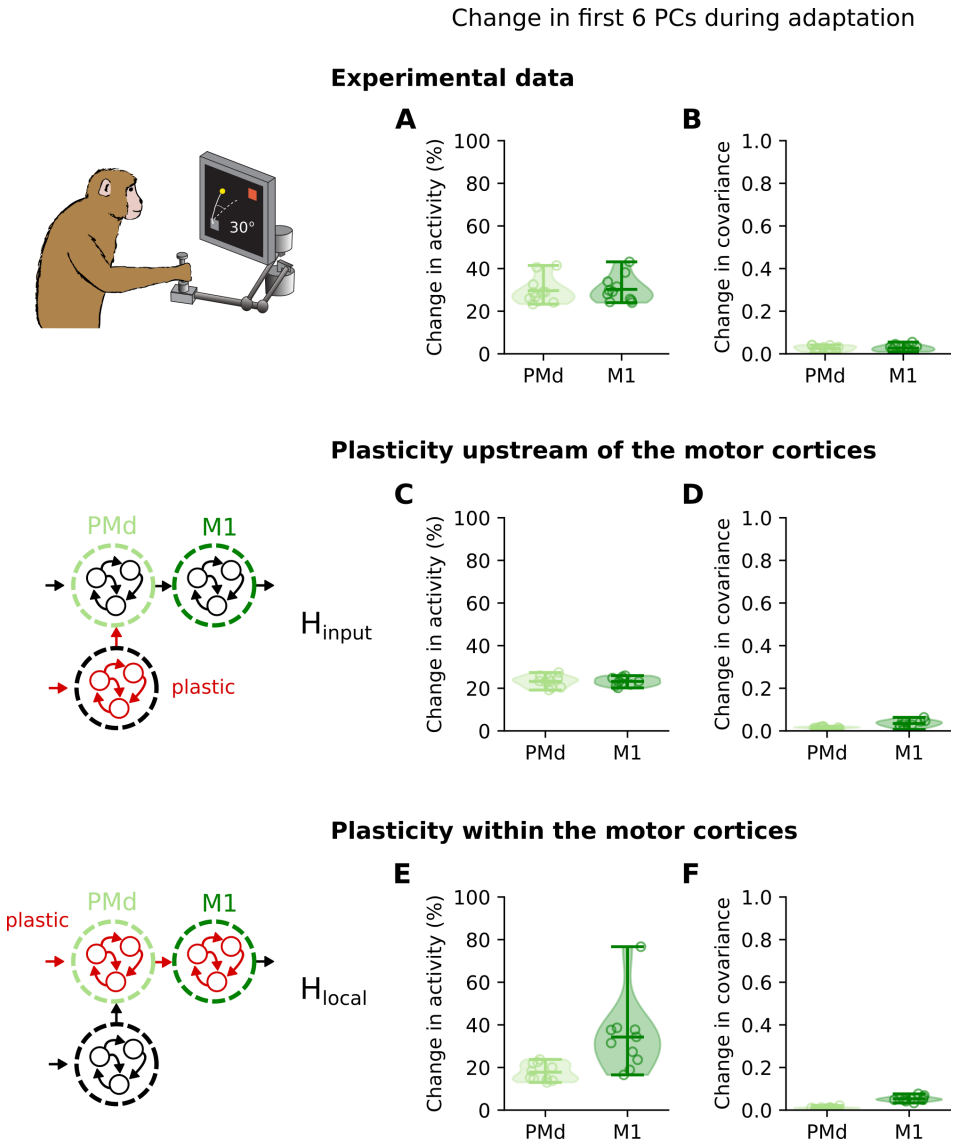


Figure 3.S3: Actual and simulated activity changes within the neural manifold. Here, we focused on the population-wide activity changes, rather than the single neuron/unit changes. We used Principal Component Analysis to identify a six-dimensional neural manifold that captured >80% of the trial-averaged activity (during the baseline epoch). **A.** Changes in trial-averaged activity for the PMd and M1 experimental data. Horizontal lines represent minimum, mean, maximum, empty circles individual data points (n=10 network initialisations). **B.** Change in neural covariance for each of these two regions; data presented as in A. **C.** Change in trial-averaged activity following adaptation under H_{input}. **D.** Change in covariance following adaptation under H_{input}. **E.** Change in trial-averaged activity following adaptation under H_{local}. **F.** Change in covariance following adaptation under H_{local}. Data in C,D and E,F are presented as in A and B, respectively. The monkey image was created by Carolina Massumoto who gave permission to use it under CC-BY license.

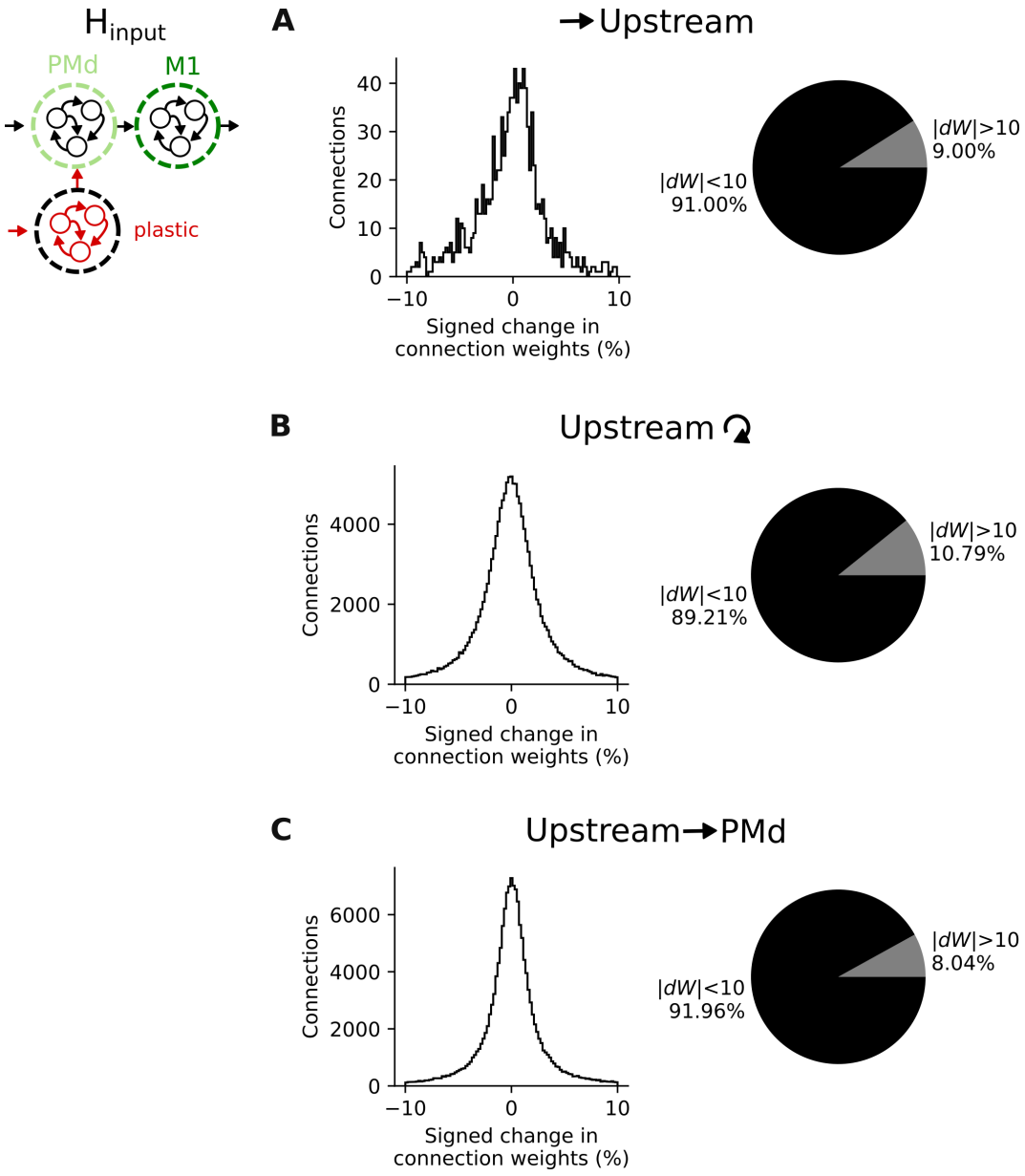


Figure 3.S4: Analysis of weight changes following VR adaptation under H_{input}. A. Distribution of signed weight changes (left) and proportion of connections showing a large absolute weight change ($|\mathbf{dW}| > 10$) (right) for the input connections of the upstream network module. **B.** Weight changes for the recurrent connections of the upstream network module. **C.** Weight changes for the upstream to PMd connections. Data in B,C is presented as in A. Data of one representative network simulation is shown.

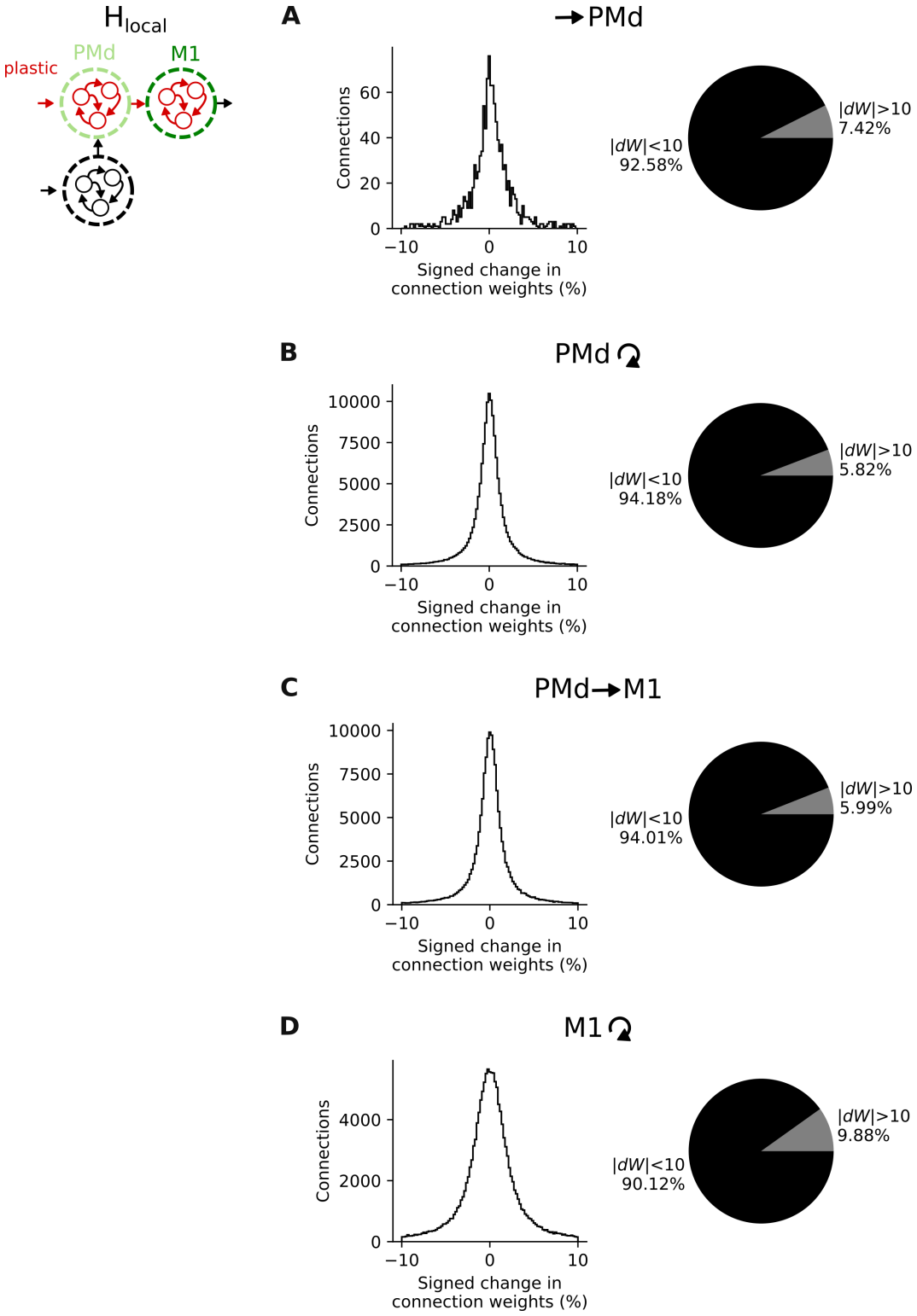


Figure 3.S5: Analysis of weight changes following VR adaptation under H_{local}. **A.** Distribution of signed weight changes (left) and proportion of connections showing a large absolute weight change ($|\mathbf{dW}| > 10$) (right) for the input connections of the PMd network module. **B.** Weight changes for the recurrent connections of the PMd module. **C.** Weight changes for the PMd to M1 connections. D. Weight changes for the recurrent connections of the M1 module. Data in B,C,D is presented as in A. Data of one representative network simulation is shown.

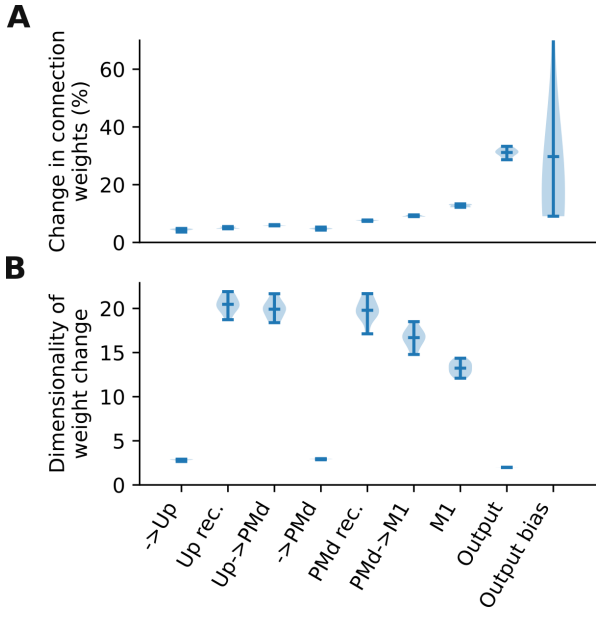


Figure 3.S6: Weight changes after initial training. A. Average magnitude of weight changes following initial training on the reaching task. Shaded area and horizontal bars, data distribution with mean and extrema (n=10 network initialisations). **B.** Dimensionality of weight changes following initial training on the reaching task. Data presented as in A.

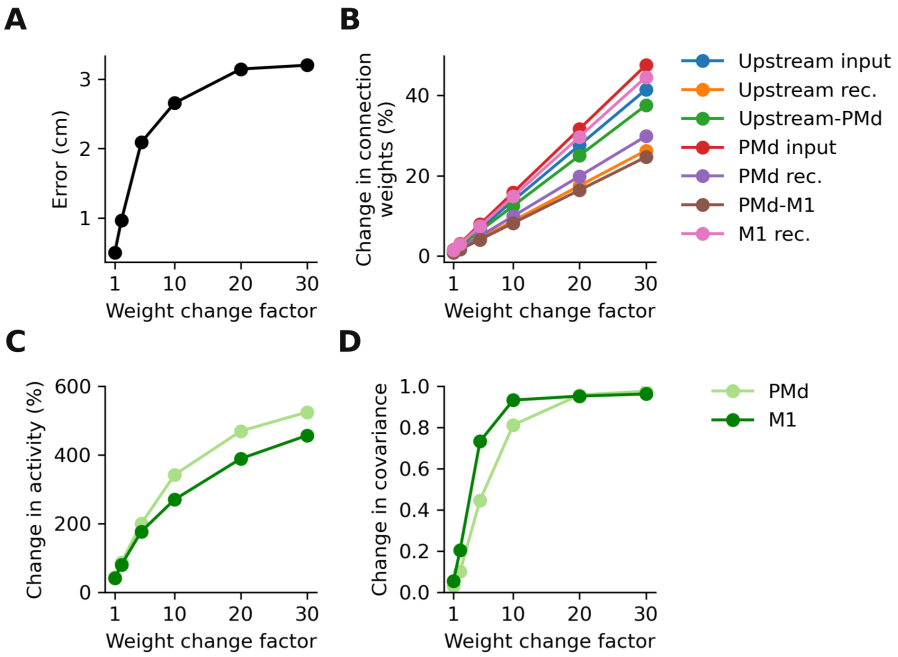


Figure 3.S7: Sensitivity of change in activity and covariance to magnitude of weight change. A. VR adaptation performance, measured as root mean squared error between target and produced hand trajectories. Shown is the performance after adaptation for a model where every parameter is allowed to be plastic. To test the sensitivity of the network activity and the produced output on the magnitude of the underlying connectivity changes, we scaled up the learned weight changes during VR adaptation by a factor (x-axis). A weight change factor of 1 thus corresponds to the true weight changes observed after adaptation. Increasing the learned weight changes deteriorates performance, as error increases for increasing weight change factor. **B.** Measured weight change, shown for all model parameters. **C.** Change in activity in the PMd (light) and M1 (dark) modules, respectively. **D.** Change in covariance. Data shown as in C. C,D show that the change in activity and covariance is highly sensitive to the magnitude of the weight change.

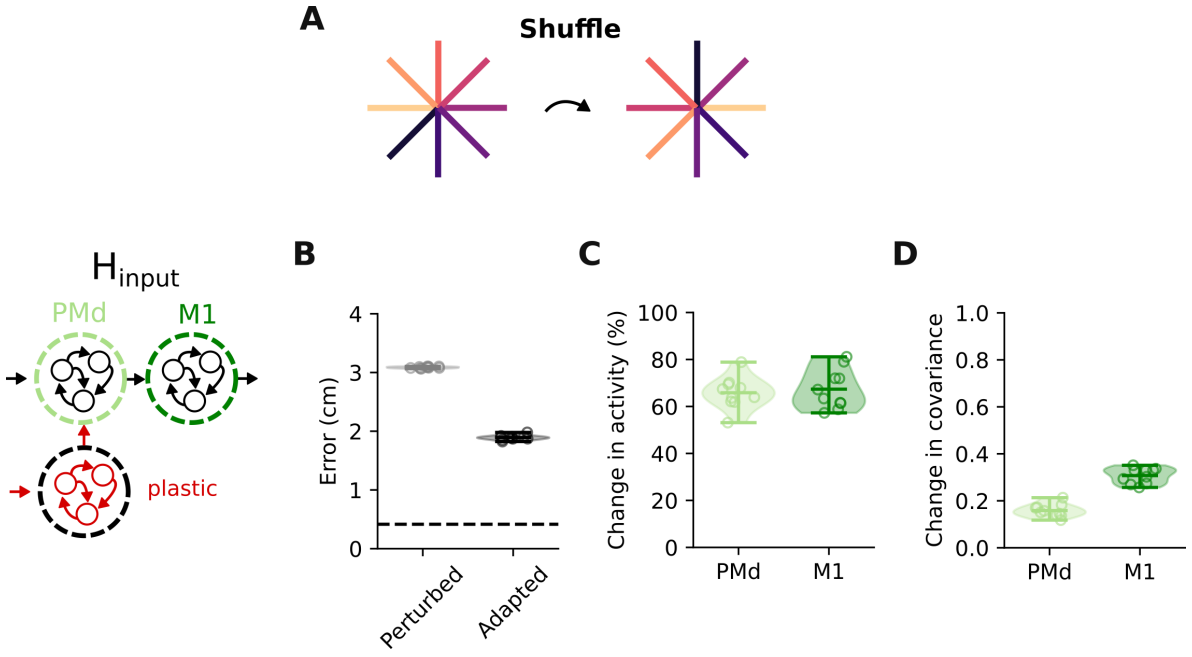


Figure 3.S8: Adaptation to a visuomotor reassociation perturbation under H_{input}. A. We simulated a reassociation task in which the network had to learn new associations between inputs and reach directions. **B.** Root mean squared error between target and produced hand trajectories without (grey) and with learning (black) under H_{input}. Dashed line indicates error during baseline trials. Shaded area and horizontal bars, data distribution with mean and extrema (n=10 network initialisations). Same data presentation in all panels. **C.** Change in trial-averaged activity for PMd and M1. **D.** Change in covariance. Data in C,D are presented as in B.

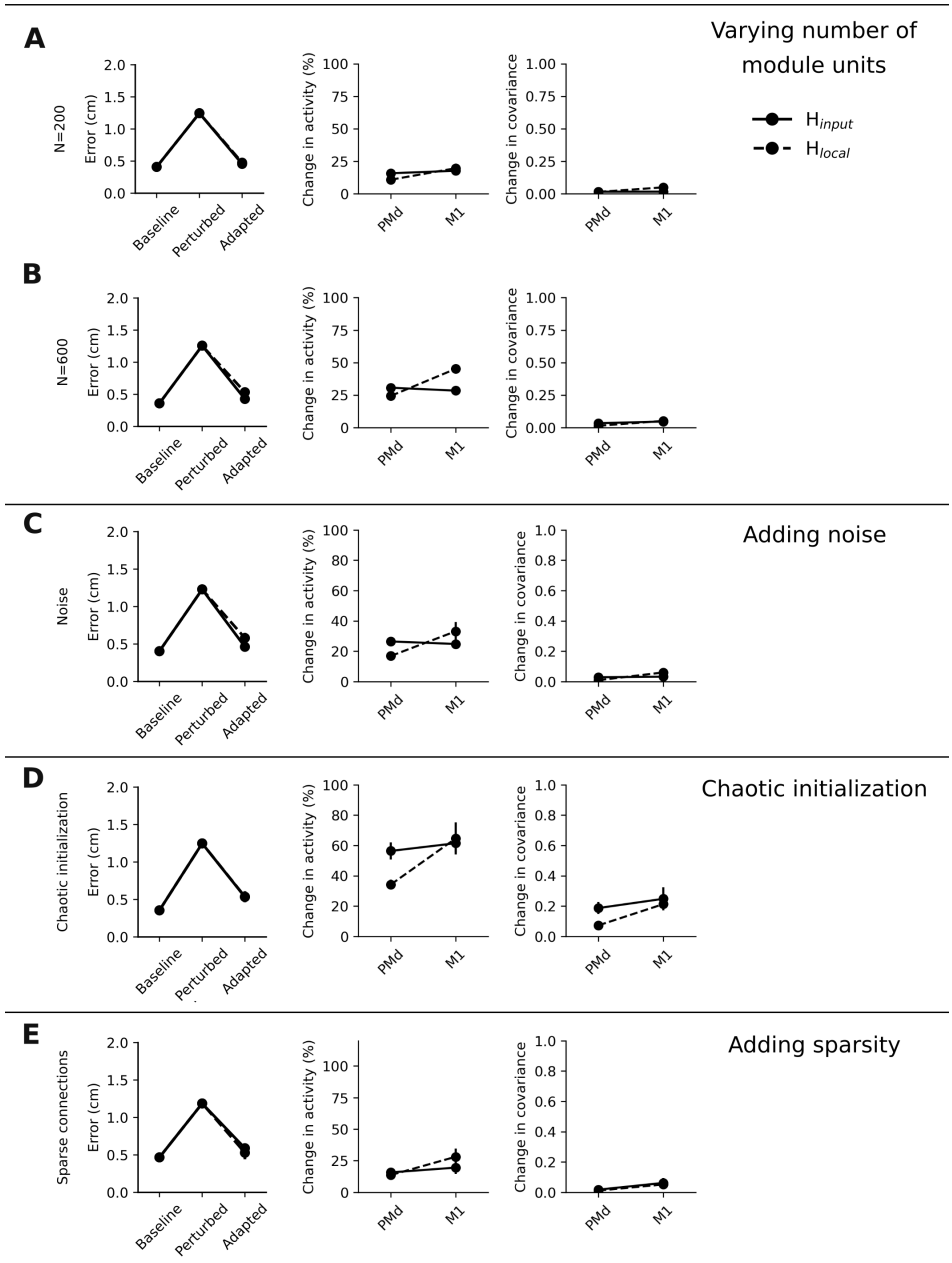


Figure 3.S9: Results are robust to parameter variations. A.,B. Models with different number of units in each network module exhibit qualitatively similar activity changes. VR adaptation performance, measured as root mean squared error between target and produced hand trajectories (left), change in activity following adaptation (middle), and change in covariance following adaptation (right). Data in B,C,D,E is presented as in A. **C.** Adding noise to the network does not fundamentally change the activity changes in the network. Each neuron in the model received an additional, random, independent input at each time step, drawn from a normal distribution with zero mean and s.d. 0.1. Markers and error bars, mean and s.d. (n=10 network initialisations). **D.** Chaotic initialization slightly increases the overall change in activity and covariance following adaptation, yet preserves the fact that H_{input} leads to larger changes in PMd compared to M1. Recurrent and inter module weights were initially drawn from a normal distribution with zero mean and s.d. $1.2/\sqrt{(N)}$. Markers and error bars, mean and s.d. (n=10 network initialisations). **E.** Networks with sparse recurrent and inter module connectivity show similarly low changes in activity and covariance. Only 60% of inter module and 80% of recurrent weights were allowed to be non-zero. Only non-zero connections were plastic during initial training and adaptation. Markers and error bars, mean and s.d. (n=10 network initialisations).

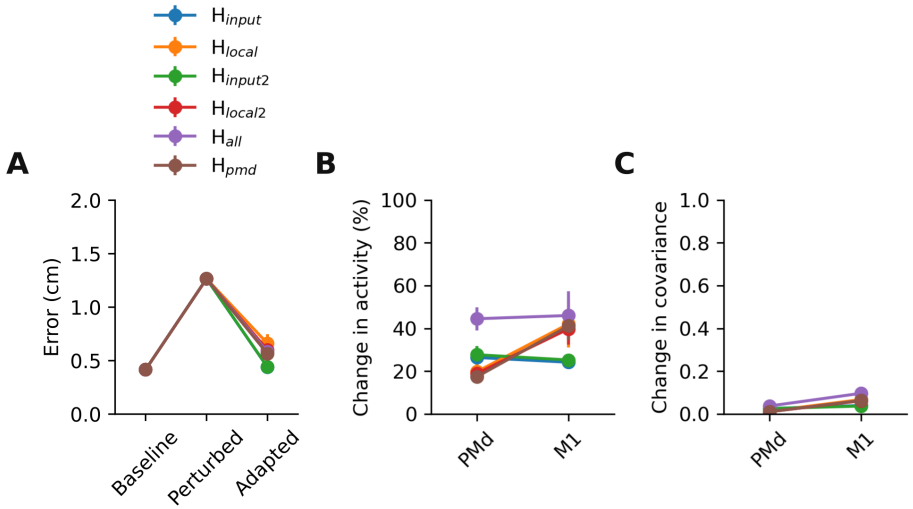


Figure 3.S10: Results are consistent for different plasticity combinations. A. VR adaptation performance, measured as root mean squared error between target and produced hand trajectories. Colours indicate which parameters of the model were allowed to be plastic during adaptation. H_{input2} is similar to H_{input} except that the input weight to PMd ($\mathbf{W}^{in,PMd}$) is also plastic. H_{local2} is similar to H_{local} except that the input weight to PMd ($\mathbf{W}^{in,PMd}$) is not plastic. For H_{all} every parameter is plastic. For H_{pmd} only the recurrent connectivity within PMd is plastic. Markers and error bars, mean and s.d. (n=10 network initialisations), same in all panels. **B.** Change in activity following adaptation. **C.** Change in covariance following adaptation.

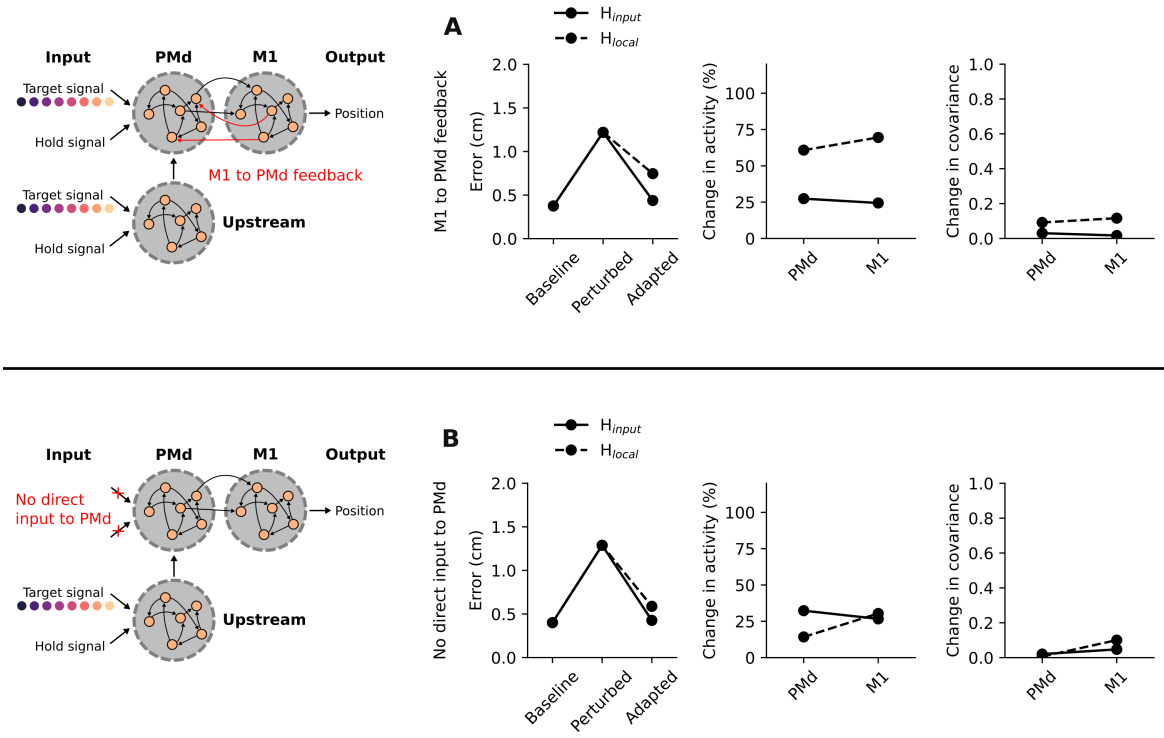


Figure 3.S11: Results are robust to variations in model architecture. **A.** Model with all-to-all feedback connections from M1 to PMd: VR adaptation performance, measured as the root mean squared error between target and produced hand trajectories (left); change in trial-averaged activity following adaptation (middle); and change in covariance following adaptation (right). Markers and error bars, mean and s.d. (n=10 network initialisations). **B.** Model with no direct input to PMd; data presented as in A.

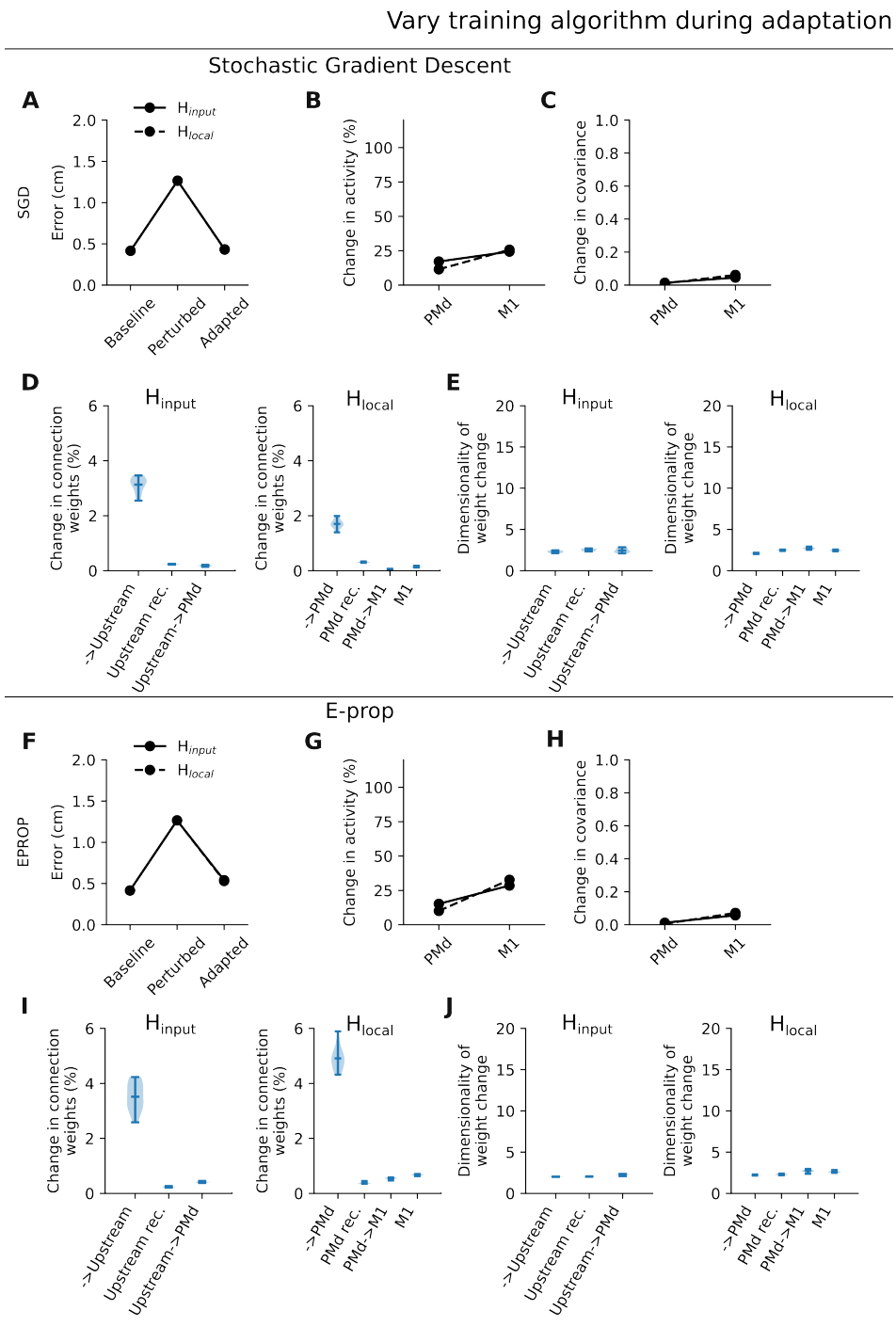


Figure 3.S12: Results are robust to variations in training algorithm. We repeated all the adaptation simulations using either Stochastic Gradient Descent **(A-E)** or a recent biologically plausible training algorithm (E-prop)[173] to retrain the network during adaptation **(F-J)**. **A.** Root mean squared error between target and produced hand trajectories after initial training (Baseline), exposure to the VR perturbation (Perturbed), and after adaptation (Adapted) under H_{input} (solid line) and H_{local} (dashed line). Markers and error bars, mean and s.d. (n=10 network initialisations). **B.** Change in trial-averaged activity. **C.** Change in covariance. Data in B,C are presented as in A. **D.** Change in connection weights following adaptation under H_{input} (left) and H_{local} (right). Shaded area and horizontal bars, data distribution with mean and extrema (n=10 network initialisations). **E.** Estimated dimensionality of connection weight changes. Data presented as in D. Data in F,G,H,I,J are presented as in A,B,C,D,E, respectively.

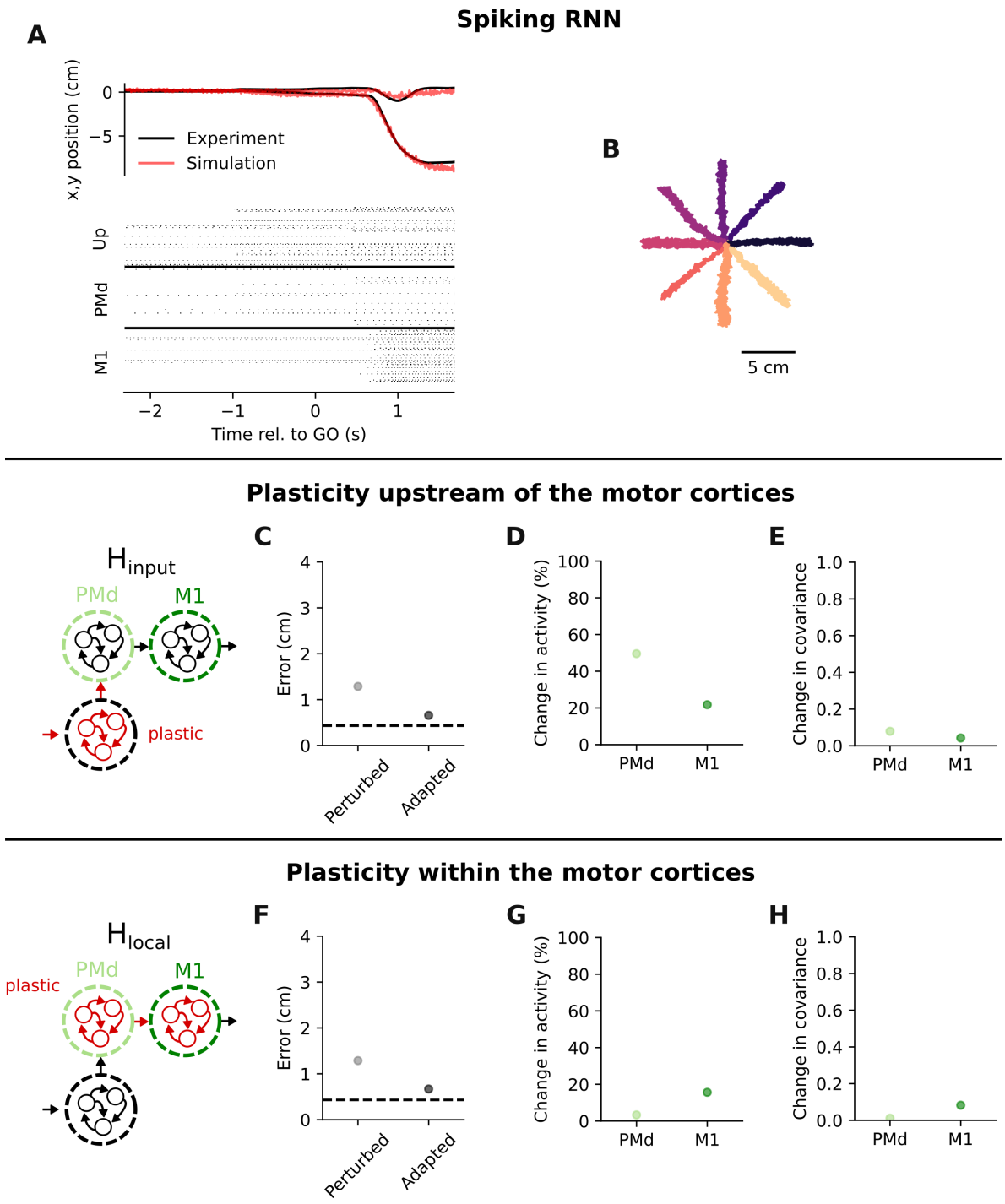


Figure 3.S13: A modular spiking neural network replicates the main results obtained using a rate-based network. A. Modular spiking RNN trained with Backpropagation Through Time using Surrogate Gradients[191,192]. Each module (Upstream, PMd, M1) consisted of 100 neurons, respectively. Top: The produced output matched the experimental data well. Bottom: example neuronal firing for the same example example trial. **B.** Simulated hand trajectories after initial training. **C.** Root mean squared error between target and produced hand trajectories after initial training (dashed line), after the VR perturbation is introduced (grey) and after adaptation to the perturbation under H_{input} (black). **D.** Changes in trial-averaged activity following adaptation under H_{input} for PMd and M1, respectively. **E.** Change in covariance following adaptation under H_{input}; data presented as in D. **F.** Root mean squared error between target and produced hand trajectories under H_{local}. **G.** Changes in trial-averaged activity following adaptation under H_{local} for PMd and M1, respectively. **H.** Change in covariance following adaptation under H_{local}. Data in F,G,H are presented as in C,D,E.

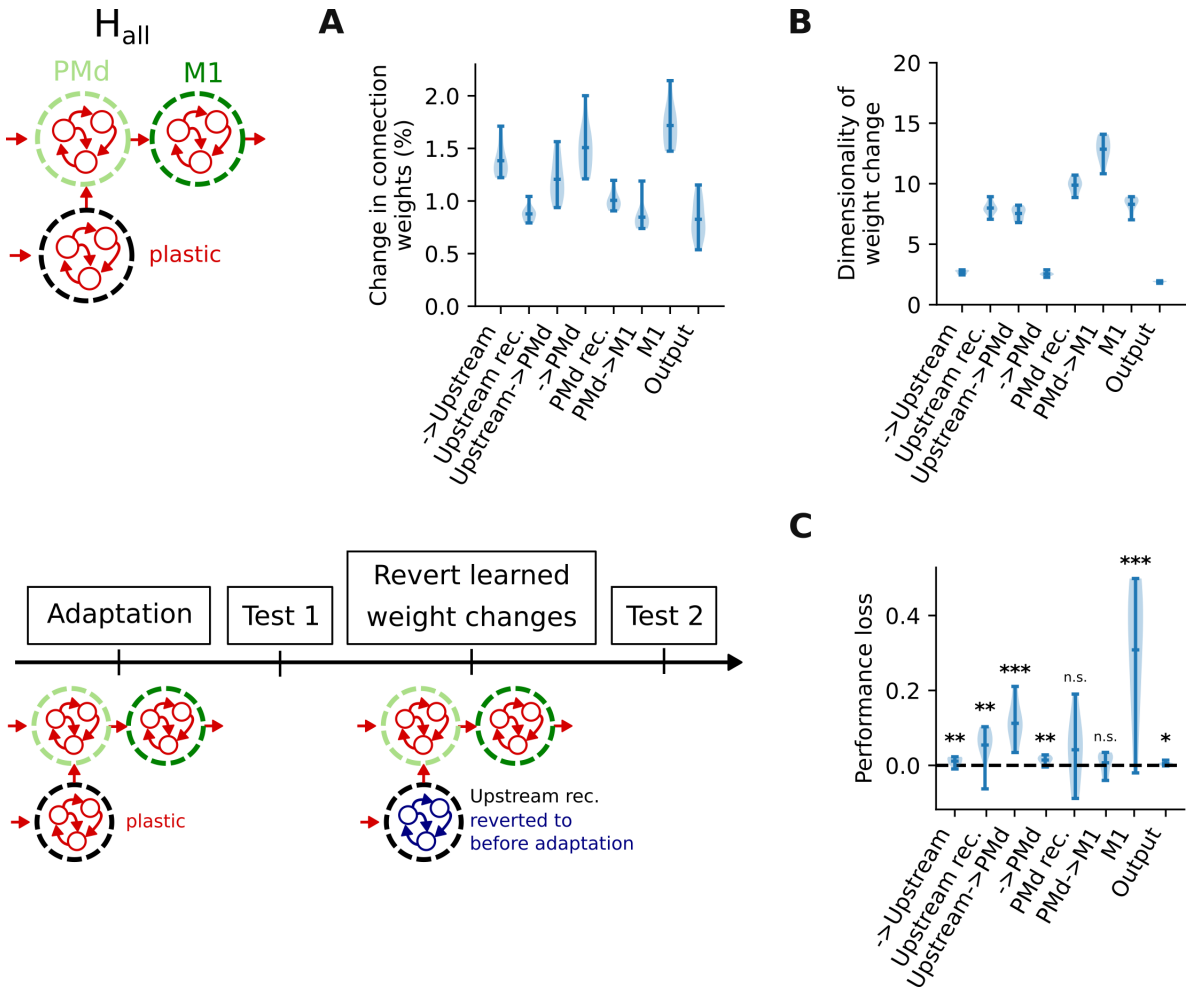


Figure 3.S14: Analysis of weight changes for H_{all}. A. Change in connection weights following adaptation under H_{all} (all parameters are allowed to be plastic). Shaded area and horizontal bars, data distribution with mean and extrema (n=10 network initialisations). Same data presentation in all panels. **B.** Estimated dimensionality of connection weight changes for each network module and cross-module connections. **C.** Contribution of the learned within-module or cross-module weight changes to adaptation. Following VR adaptation, we manually set the connectivity values of one module or set of cross-module connections to its corresponding connectivity values before adaptation, and tested how much this impaired motor output. Performance loss is given by the difference in mean squared error between before and after we reverted the learned weight changes (Test 1 versus Test 2 in the schematic). Two-sided t-test was used to test the null hypothesis of zero population mean (scipy package ttest_1samp function) (t=3.32 p=0.0089; t=3.61 p=0.0056; t=6.59 p=0.0001; t=4.12 p=0.0026; t=1.53 p=0.1599; t=0.84 p=0.4236; t=5.77 p=0.0003; t=3.18 p=0.0111; ordered as presented in C).

E-prop learning algorithm (Figure 3.S12)

Instead of using the Adam optimizer during the adaptation phase, we also tested a more biologically plausible, online learning rule[93,173]. Each neuron received a weighted error signal about the current difference between produced and target hand position. The weighted error was given by projecting the position error back through the network hierarchy, using the transpose of the feedforward matrices.

$$\epsilon = \mathbf{x}^{out} - \mathbf{x}^{target} \tag{3.11}$$

$$\epsilon^{M1} = (\mathbf{W}^{out})^T \ (\epsilon - \mathbf{b}^{out}) \tag{3.12}$$

$$\epsilon^{PMd} = (\mathbf{W}^{PMd-M1})^T \ \epsilon^{M1} \tag{3.13}$$

$$\epsilon^{UP} = (\mathbf{W}^{UP-PMd})^T \ \epsilon^{PMd} \tag{3.14}$$

The weight update for a single connection from neuron i to neuron j was then given by multiplying the error signal of the postsynaptic neuron j with a term dependent on the activity of the postsynaptic and the presynaptic neuron.

$$dw_t^{ji} = -\eta \ \epsilon_t^j \left(1 - \tanh(x_t^j)^2\right) \sum_{t'<t} \tanh(x_{t'}^i) \tag{3.15}$$

where $\eta = 0.000001$ and a batch size of one was used. The weight update described above was applied every 0.2s, starting from 0.5s in the trial. Note that the sum above is bounded as we reset the time for every trial. An example weight update for $\mathbf{W}^{UP-PMd}$ would then be

$$dw_t^{PMd,j;\ UP,i} = -\eta \ \epsilon_t^{PMd,j} \left(1 - \tanh(x_t^{PMd,j})^2\right) \sum_{t'<t} \tanh(x_{t'}^{UP,i}) \tag{3.16}$$

Defining the learning location was then similar to what has been done in the main manuscript.

- $\mathrm{H}_{\mathrm{input}}$: $\mathbf{W}^{in,UP}$, $\mathbf{W}^{UP}$, $\mathbf{W}^{UP-PMd}$ are plastic
- $\mathrm{H}_{\mathrm{local}}$: $\mathbf{W}^{in,PMd}$, $\mathbf{W}^{PMd}$, $\mathbf{W}^{PMd-M1}$, $\mathbf{W}^{M1}$ are plastic

Spiking neural network (Figure 3.S13)

To transform our model from a rate-based to a spiking neural network we closely followed a recently published tutorial[193,194], based on this paper[192]. We mapped the RNN dynamics to a network of leaky-integrate-and-fire neurons, preserving the model architecture described above. We used a membrane time constant of 50ms and a synaptic time constant of 25ms. The readout neurons were modelled as simple integrators with a longer timescale, potentially mimicking slower processes related to muscle activation. For the readout neurons a membrane time constant of 500ms and a synaptic time constant of 200ms. To train the spiking network using the Adam optimizer we used the method of surrogate gradients[192], where the spiking nonlinearity is replaced by a sigmoid for the backward pass of gradient descent. The steepness of that sigmoid was set to 100 (following the tutorial). The stimulus given to the network was defined as before, as a non-spiking, tonic input. The parameters for these simulations were: number of neurons per module = 100, batch size=20, $dt = 0.01$, learning rate=0.001, regularization $\alpha = 0.001$ (here, we used regularization only on the weights, not the rates), gradient clipping at norm 0.2.

CHAPTER 4

Biologically plausible learning

After looking at learning constraints (Chapter 2) and learning location (Chapter 3), we have left the potentially most difficult question to the end: *how* can the brain know which synapses to tweak in order to learn a new behaviour? The goal in the following chapter is to test one potential solution: can we make use of the fact that the motor system is a feedback controller? We first create a model which is able to flexibly correct its output when there are unforeseen perturbations. We then show that the feedback signals, which evolved during initial training, can drive a local synaptic plasticity rule which acts on the recurrent weights of the network. The proposed local plasticity rule leads to adaptation behaviour similar to those of humans and monkeys, which we test by analysing several aspects previously reported in behavioural experiments. Furthermore, we recover a predicted neural signature of this rule from neural population recordings from monkey M1. In summary, our results suggest that there might be local synaptic plasticity in M1 guided by the feedback signals used to control our body.

The work in this chapter is presented in an article format as published on bioRxiv[98] and currently under review. I would like to thank my supervisors Claudia Clopath and Juan Alvaro Gallego for their guidance and inspiration throughout this project. Further, I would like to thank our experimental collaborators Matthew G. Perich and Lee E. Miller for kindly providing their data and wisdom.

Feedback-based motor control can guide plasticity and drive rapid learning

4.1 Abstract

Animals use afferent feedback to rapidly correct ongoing movements in the presence of a perturbation. Repeated exposure to a predictable perturbation leads to behavioural adaptation that counteracts its effects. Primary motor cortex (M1) is intimately involved in both processes, integrating inputs from various sensorimotor brain regions to update the motor output. Here, we investigate whether feedback-based motor control and motor adaptation may share a common implementation in M1 circuits. We trained a recurrent neural network to control its own output through an error feedback signal, which allowed it to recover rapidly from external perturbations. Implementing a biologically plausible plasticity rule based on this same feedback signal also enabled the network to learn to counteract persistent perturbations through a trial-by-trial process, in a manner that reproduced several key aspects of human adaptation. Moreover, the resultant network activity changes were also present in neural population recordings from monkey M1. Online movement correction and longer-term motor adaptation may thus share a common implementation in neural circuits.

4.2 Introduction

Animals, including humans, have a remarkable ability to rapidly correct their ongoing movements based on perceived errors, even when feedback may be distorted, such as when reaching into a pond to recover an object one has dropped. In the laboratory, these movement corrections and subsequent adaptation can be evoked and studied systematically using the classic visuomotor rotation (VR)

perturbation paradigm[70,146]. In this paradigm, the subject receives distorted visual feedback (typically a rotation about the centre of the workspace) of a reaching movement, thereby creating a perceived error due to the mismatch of expected and observed hand trajectory. Humans can correct their ongoing movements even during the very first trial after perturbation onset[144], a process that is mediated by primary motor cortex (M1) integrating multiple feedback signals arriving from various sensory and motor brain regions[7,8,14,27–29,184,186,195–200].

When repeatedly exposed to a predictable perturbation, animals progressively learn to use their perceived errors to anticipate its effect. For the case of the VR paradigm described above, this leads to a gradual reaiming of the reach, until it starts out in the correct direction[70], thereby eliminating the need for further online corrections. This requires some form of rapid learning along the sensorimotor pathways, likely guided by trial-by-trial error information[144,201,202]. How and where in the brain this "motor adaptation" happens remains inconclusive[15,21,123,132,137–140,146,147,150,151,203–217], and may depend on the characteristics of the perturbation, such as whether it is a rotation of the visual feedback or a force acting on the limb[107,138,218–222].

Feedback-based movement correction and motor adaptation have been mostly studied in isolation and are often assumed to involve different neural substrates[27,70]. What if that were not the case, but instead, they shared a common implementation in neural circuitry? A recent behavioural study proposed that movement correction and adaptation may indeed be tightly linked: fast feedback responses could act as teacher signals that drive trial-by-trial adaptation[203,223,224]. Despite the conceptual beauty of this idea, its feasibility and implementation details remain unexplored.

Here, we hypothesised that the neural circuitry for feedback control may be exploited to drive the "plasticity" that enables successful motor adaptation. To address this hypothesis, we use a recurrent neural network (RNN) model[42,60,61,65,96,97,99,153,154] trained not only to produce a certain output, but to *control* it. The key difference here is that this model should be able to flexibly correct its output in the case of unexpected external perturbations. Having such a model allowed us to test whether feedback signals used for motor control could guide plastic changes within the network that lead to successful trial-by-trial learning.

We first show that an RNN can be trained to perform feedback-based motor control, even in the presence of a relatively long, biologically plausible feedback delay. We then demonstrate how feedback signals, modelled as inputs to the RNN, can guide synaptic plasticity within the network

that drives successful trial-by-trial adaptation. Intriguingly, this form of learning through feedback-driven plasticity led to behavioural adaptation that was similar to that of a human in terms of time course[69,187], generalisation[70], and sensitivity to perturbation variability[225,226]. Moreover, both effective control and learning could be achieved with sparse feedback signals. Finally, comparison with neural population recordings from monkey M1 (data from[107]) supports the plausibility of the proposed plasticity rule: the temporally dissociable activity changes that followed adaptation in our model could also be found in the actual neural activity. This work not only introduces the potential of a combined implementation of motor control and learning in recurrent circuits, but it also relates several aspects of human adaptation behaviour to a single unifying neural process.

4.3 Results

A recurrent neural network that performs feedback-based motor control

We used an RNN model to investigate whether the same feedback signals used to control ongoing behaviour could also mediate motor adaptation. This work was divided into two phases: 1) training the RNN to perform feedback-based motor control (using a gradient-based algorithm), and 2) using this trained RNN to implement trial-by-trial motor adaptation via a local, biologically plausible learning rule acting on the recurrent weights of the network. We validated our model by comparing it to several behavioural and electrophysiological studies.

First, we trained an RNN to control its own output to produce the desired movement (Figure 4.1A,B). Our goal was to have a model that, after training, could dynamically adjust the ongoing movement according to incoming sensory feedback. Feedback control was based on an instantaneous position error signal, ϵ_t, defined as the difference between the produced position, p_t, and the target position, p_t^*, which we computed assuming a straight line between the start and end points (Figure 4.1C). This error signal was fed back as an input to the RNN with a biologically realistic delay of 120 ms[199] (Figure 4.1B).

After the initial training phase (Figure 4.1D, examples in Figure 4.S1; Methods), we tested the RNN on a standard centre-out reaching task with eight equally distributed targets. As expected due to our training procedure, the model was readily able to produce the required straight movement trajectories even without explicit training on this task (Figure 4.1E).

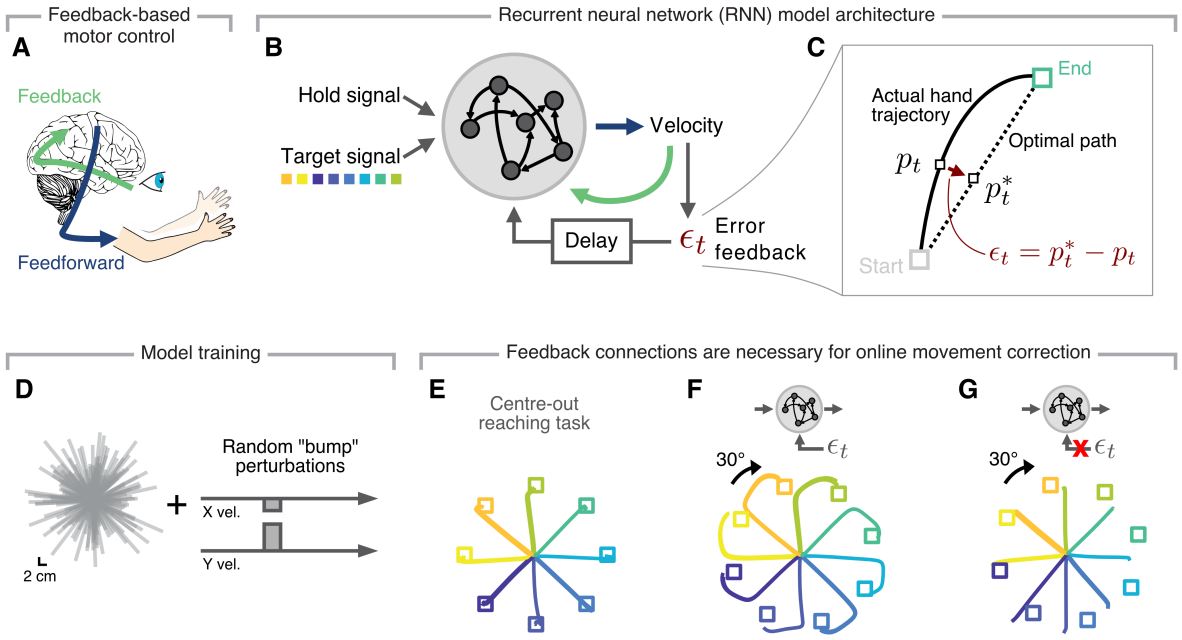


Figure 4.1: Proposed recurrent neural network that controls its output based on feedback. A. Afferent and efferent pathways act together to produce precise movements, and to flexibly correct them in the presence of perturbations. **B.** A recurrent neural network (RNN) model to explore a shared implementation of motor control and adaptation based on a common error feedback loop. **C.** We defined the ongoing error during movement as the difference between the observed and the optimal hand position. **D.** Initial RNN training included reaches of varying lengths and to different locations (grey lines) with occasional random velocity "bump" perturbations (cf. Figure 4.S1). **E.** Hand trajectories produced by a trained RNN required to perform a standard eight-target centre-out reaching task. **F.** Hand trajectories after introducing a 30° rotation of the RNN's output, to mimic a visuomotor rotation perturbation; note that feedback allowed the network to correct its output online and reach to the target. **G.** Hand trajectories for a model without a feedback loop, which could not counteract the same perturbation.

To test the network's ability flexibly correct its output, we replicated the classic VR paradigm[70]. If the RNN could indeed *control* its output, it should still be able to reach to the desired target by correcting the produced movement to counteract the 30° rotations online. Inspecting the movement trajectories after VR onset confirmed that the model could use the error signal to correct its ongoing output (Figure 4.1F; the curved trajectories indicate ongoing correction). Importantly, successful correction relied on the error signal being fed back into the model: trajectory correction only started after the delayed feedback had had enough time to propagate to the model (Figure 4.1F), and an RNN trained without feedback connections (Methods) could not reach to the targets (Figure 4.1G).

An error feedback signal used for feedback-driven motor control can drive trial-by-trial adaptation

We have shown that an RNN that has learnt to use feedback signals to control its output can readily counteract an external perturbation, exhibiting a behaviour upon VR onset that is very similar to that of humans[70] and monkeys (compare the monkey data from Ref.[107] in Figure 4.2A to the model data in Figure 4.2D). However, when repeatedly exposed to a VR, both humans and monkeys learn to adjust their initial "motor plan," which results in their reach take-off angle pointing in the correct direction after dozens of trials[70,107] (Figure 4.2B,C). Can our relatively simple error feedback signal enable persistent learning across trials?

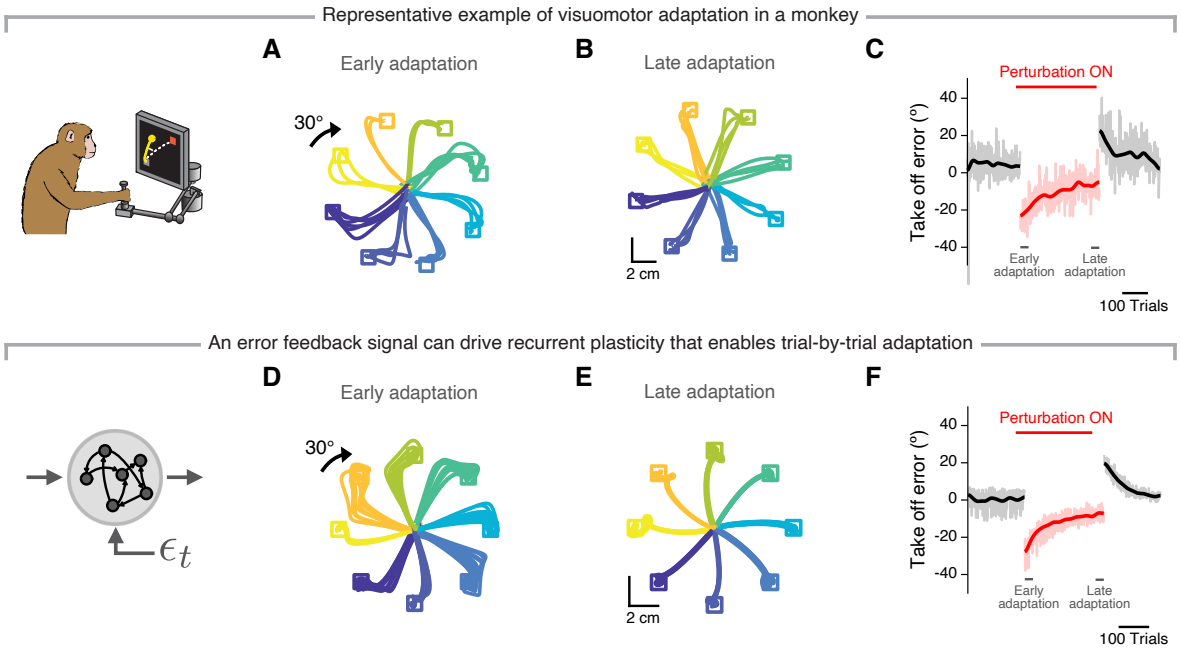


Figure 4.2: Feedback signals can guide local synaptic plasticity that enables successful motor adaptation. **A.** Example hand trajectories as a monkey reached to each of eight targets after a 30° rotation of the visual feedback was introduced (first 30 trials after perturbation onset; data from Perich et al.[107]). **B.** Example hand trajectories after the monkey had adapted to the perturbation by reaiming the reach direction (last 30 trials of the perturbation phase for the same session as in A). **C.** Angular reaching take-off error for the baseline (left, black), perturbation (middle, red), and wash-out phases (right, black). Transparent lines, single trial errors; solid lines, smoothed mean error (Gaussian filter, s.d., 10 trials). **D-F.** Simulation results for an RNN implementing the proposed feedback-driven plasticity rule, by which recurrent weights are modified according to the error signal received by the postsynaptic neuron. The first and last 80 trials of adaptation are shown; otherwise data is presented as in A-C. Note the strong similarities between the behaviour of the network and that of the monkey.

Since the feedback inputs acting on the network correctly modulate each neuron's activity to minimize

the ongoing motor error (Figure 4.1F), we hypothesised that they could also act as a "teacher signal" for local, recurrent synaptic plasticity. To test this, we devised a local biologically-plausible synaptic plasticity rule causing the connection weight from neuron i to neuron j to change in proportion to the feedback signal received by neuron j (Methods). Implementing this plasticity rule led to behaviour that was similar to that of monkeys': the initially large errors in take-off angle became progressively smaller over time, until they reached a plateau close to zero error (note the similarities between Figure 4.2D-F and Figure 4.2A-C). Moreover, when the perturbation was turned off, the model underwent a de-adaptation phase similar to the "wash-out" effect exhibited by monkeys (compare Figure 4.2C and Figure 4.2F) and humans[227]. These results confirm our main hypothesis: an error feedback signal used for online motor control can be leveraged to guide recurrent synaptic plasticity that drives successful trial-by-trial motor adaptation.

Feedback-based learning recapitulates key features of human motor adaptation

The previous simulation results suggest that a relatively simple plasticity rule based on an error feedback signal may mediate motor adaptation. Could this type of learning be implemented in actual brains? To investigate the biological plausibility of our feedback-based plasticity rule, we tested whether our model replicated various key observations from human and monkey motor adaptation studies.

We first addressed the finding that humans learn more from a given trial if they experience a larger error[223]. Indeed, our model reproduced this trend; the measured correlations between movement error and amount of learning in the next trial were comparable in magnitude and sign to those of monkeys performing the same VR task (Figure 4.3A; data from Ref.[107]). Moreover, as in human adaptation studies[225,226], our model's ability to learn was also hindered when the perturbation was inconsistent across trials, with greater perturbation variance leading to progressively less learning (Figure 4.3B). Thus, the amount of trial-by-trial adaptation matched experimental observations well.

When examining the timescale of learning during an experimental session, human motor adaptation seems to be mediated by two simultaneous learning processes: one fast, and another slow[69,187]. Using the same analysis as in Ref.[187], we found that the adaptation time course of our model is also best described by a combination of two learning processes with different time constants (Figure 4.3C). Even the parameters describing these processes were comparable to values reported in a human

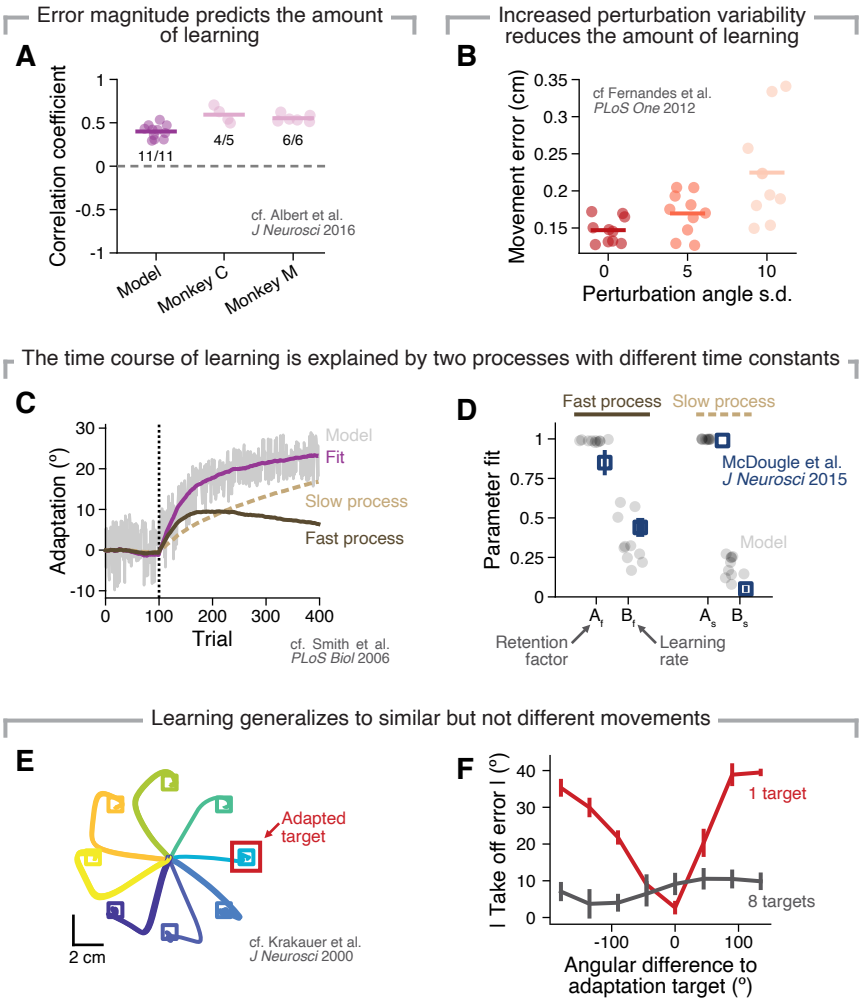


Figure 4.3: Motor adaptation based on feedback-driven plasticity recapitulates key aspects of human and monkey behaviour. A. Correlation between the take-off error in the current trial and the amount of learning from this trial to the next for both simulation (Model) and behavioural data (Monkey C and Monkey M; data from Perich et al.[107]). Individual circles, a different network (Model) or experimental session (Monkey); numbers, proportion of networks or experimental sessions exhibiting a significant correlation ($P<0.05$). **B.** Movement error after adaptation to VR perturbations with different perturbation angle variability (cf. Ref.[225,226] for human behaviour). **C.** As in human experiments, the time course of adaptation in the model is well fitted by a dual-rate model[187]. Grey line, single trial errors of simulated adaptation behaviour in an example simulation. Purple line, model fit; dark brown line, fast process; light brown line, slow process. **D.** Fitted parameters of the dual rate model (black) match those from a visuomotor adaptation study in humans[69] (blue). Individual circles, ten different networks; square and error bars, mean and 95 % confidence interval. **E.** Hand trajectories produced by the model after visuomotor adaptation to a single target ("Adaptation target", in red). **F.** Take-off error after adaptation to visuomotor perturbations applied to a single target, as shown in E (red), or to eight targets (dark grey) (cf. Ref.[70] for human behaviour). Lines and error bars, mean and s.d. across ten networks.

VR adaptation study (Figure 4.3D; data from Ref.[69]), indicating that, as in actual experiments, our feedback-driven plasticity rule is dominated by fast learning early in adaptation and slow learning

later on.

Finally, animals, including humans, have a remarkable ability to generalise what they have learned to novel situations, yet, the amount of generalisation seems to depend on the similarity between the current and the past situation. During a VR experiment, participants who have adapted to a perturbation applied on a single reach direction generalize when reaching to neighbouring targets, to an extent that decreases as the angle between the new and adapted direction increases[70]. Repeating this single target adaptation experiment in our model revealed the same kind of generalization: the model readily anticipated perturbations applied on adjacent targets and adaptation decreased as the angle between the probed target and the adapted target increased (Figure 4.3E-F). In summary, our model reproduced key features of primate motor adaptation, supporting our unified view of how rapid motor learning could be implemented in the brain by leveraging signals mediating online feedback correction.

Sparse feedback is sufficient for both motor control and adaptation

The fact that our model recapitulates many aspects of human and monkey adaptation behaviour supports the hypothesis that a similar feedback-based learning process may be implemented in actual neural circuits. Yet, in the model we have examined so far, every unit received a feedback signal, whereas only a relatively large subset of primary motor cortical neurons seem to be modulated by sensory feedback (73% of neurons in M1 respond to either visual or proprioceptive feedback[186]). Thus, to further probe the plausibility of our model, we explored how our results were influenced by feedback connection density.

We quantified model performance in terms of both control accuracy and degree of learning for feedback projection densities ranging from 1% to 100%. Critically, the model could control its output effectively with density as low as 25%; even the extremely sparse 1% projection density decreased endpoint error considerably compared to a model without feedback (compare the solid and dashed lines in Figure 4.4A). Likewise, the model did learn substantially for all feedback projection densities (compare the solid and empty markers in Figure 4.4B), and, although the amount of learning decreased almost linearly as feedback projections became sparser, as few as 1% of the neurons receiving feedback sufficed to drive meaningful learning that reduced take-off errors by 30%. Notably, these results were robust to changes in RNN connectivity, as control accuracy and

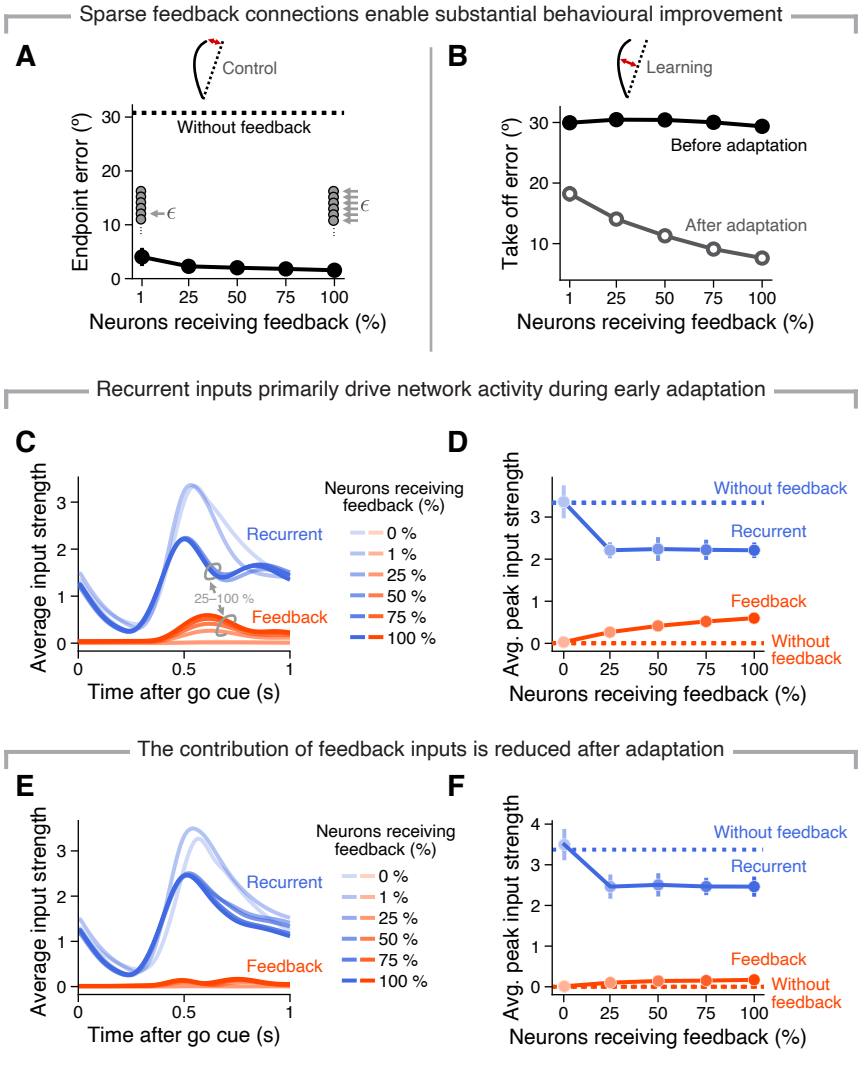


Figure 4.4: Both feedback-based motor control and adaptation can be achieved with sparse feedback signals which are small compared to recurrent signals. A. Angular error between produced and target position at the end of the reach immediately after onset of visuomotor rotation for networks with different percentages of neurons receiving afferent feedback (black markers), including no feedback (dashed line). Lines and error bars, mean and s.d. across ten networks. **B.** Take-off error at visuomotor rotation onset (solid circles), and after adaptation (empty circles) for networks with different percentages of neurons receiving afferent feedback. Lines and error bars, mean and s.d. across ten networks. **C.** Average recurrent (blue) and feedback (red) inputs to an RNN neuron before adaptation. Average input strength is defined as the mean across incoming signals and neurons. Legend, percentage of neurons receiving feedback. **D.** Average magnitude of the peak input strength of the recurrent (blue) and feedback (red) inputs before adaptation. Same colour scheme as in C. Lines and error bars, mean and s.d. across ten networks. **E.** Average recurrent (blue) and feedback (red) inputs to an RNN neuron after adaptation. Data presented as in C. **F.** Average magnitude of the peak input strength of the recurrent (blue) and feedback (red) inputs after adaptation. Data presented as in D.

degree of learning were similar for recurrent connection probabilities of 50% and 80% (Figure 4.S2). Finally, to understand how feedback inputs enabled successful behaviour, we compared their contribution to that of the recurrent inputs, both early in the perturbation block (Figure 4.4C,D), where feedback is crucial to correct the output, and after successful adaptation (Figure 4.4E,F), where we expected feedback to only play a minor role. Even at perturbation onset, the network activity was mostly driven by recurrent inputs, with the feedback making up only a minor part of the overall input (compare the blue and red traces in Figure 4.4C,D, respectively); this was the case for all feedback densities. Thus, feedback signals did not directly drive accurate movements; instead their contribution was amplified through recurrent dynamics —note the different temporal patterns of recurrent input between before and after adaptation (Figure 4.4C and Figure 4.4E, respectively). This likely explains why movement accuracy remained stable across a broad range of feedback projection densities (Figure 4.4A). As expected, after successful adaptation, feedback inputs were much smaller than recurrent inputs, regardless of the feedback projection density (Figure 4.4E,F). Combined, these results suggest that actual brains may perform effective motor control and adaptation even with a relatively small number of feedback connections, and feedback signals may make relatively small contribution to the overall neural activity.

Two temporally dissociable adaptation-related activity changes in the model can be uncovered in monkey primary motor cortex

Our previous simulations suggest that sparse feedback projections may suffice to enable effective motor control and adaptation (Figure 4.4A,B), and that feedback inputs account for only a minor portion of a unit's overall input (Figure 4.4C,D). Combined, these observations indicate that identifying a "signature" of feedback-driven plasticity in neural recordings or even in the RNN activity may be difficult. Our approach to tackle this was to focus on the within-trial timing: we devised an analysis to isolate activity changes related to the two temporally distinct processes that we expected to occur during feedback-driven adaptation: 1) an early feedforward change, reflecting updated pre-planned movement intent, and 2) a change in feedback signals later in the trial, as the error decreased over trial-by-trial learning.

To measure feedback related changes (Figure 4.5A; green) we focused on the activity changes from the baseline epoch (box A in Figure 4.5A) to right after perturbation onset (box B in Figure 4.5A),

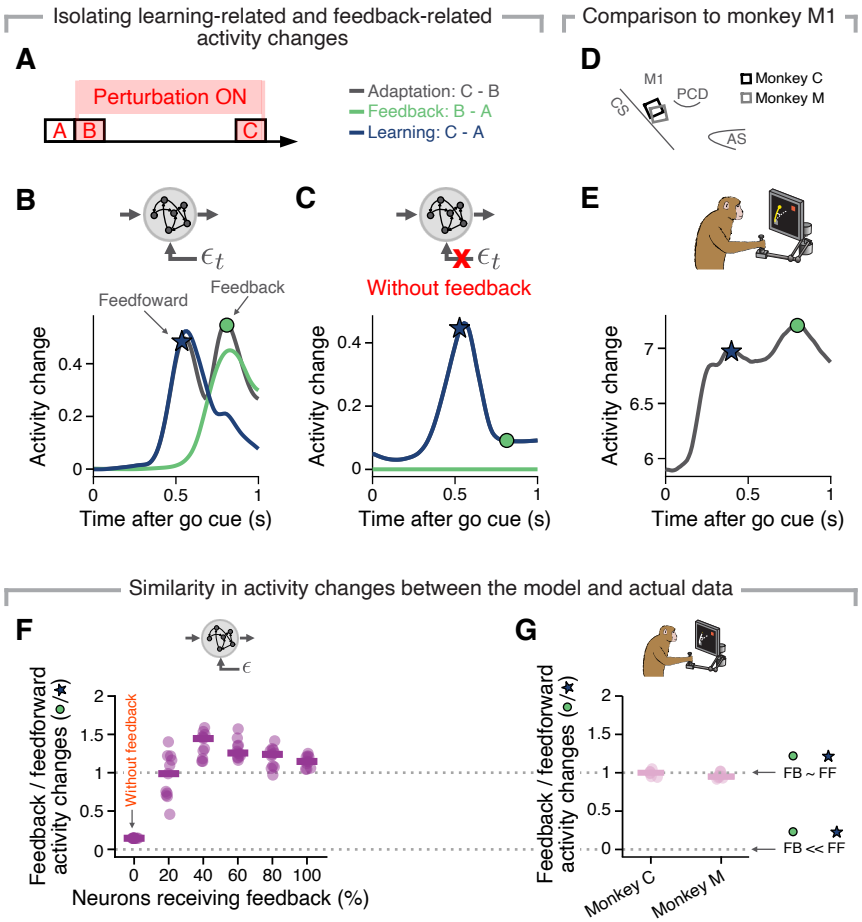


Figure 4.5: Two temporally dissociated activity changes may indicate feedback-driven plasticity in monkey M1 during visuomotor adaptation. A. Epochs used to identify feedback-related (green) and learning-related (blue) activity changes. **B.** The average RNN activity change across pairs of behavioural epochs (as shown in A) reveals putative feedforward-related (blue star) and feedback-related (green circle) activity changes for an example network. Black trace, activity change between perturbation onset and successful adaptation; green, activity change between baseline and perturbation onset; blue, activity change between baseline and successful adaptation. Note that the peak timings are chosen based on ten different network simulations. **C.** An RNN without a feedback loop has only feedforward-related activity changes. Data presented as in A. **D.** Approximate location of the recording array for each of monkey (legend). **E.** The average activity change of monkey M1 neurons between perturbation onset and successful adaptation resembles that of the network (compare with A). Data from one representative session from Monkey C, presented as in B. **F.** Comparison between the ratio of the feedback-related (green circle in B,C,E) to feedforward-related (blue star in B,C,E) activity changes from perturbation onset until successful adaptation for networks with different feedback densities. Individual markers, individual networks. Horizontal line, mean. **G.** Same as F, but for monkey M1 (data shown for each monkey separately). Individual markers, individual sessions. Horizontal line, mean. FB, feedback; FF, feedforward.

by simply taking the difference between the single neuron activities in those two epochs. Similarly, the learning related changes (Figure 4.5A; blue) represent the activity changes between the baseline epoch and the late adaptation epoch (box C in Figure 4.5A), where we expect the feedback component

to have reached baseline levels again as the model no longer needs to correct its movement online. The adaptation related change (Figure 4.5A; dark grey) is defined as the difference between early and late adaptation epochs, respectively.

Inspecting the RNN activity changes during adaptation indeed revealed two distinct peaks in the average activity change during adaptation (Figure 4.5B), assigned to learning (blue) and feedback (green), respectively. To confirm that these early and late "activity change peaks" did reflect feedforward and feedback processes, respectively, we trained a different set of models that used simple gradient descent instead of feedback-driven plasticity to achieve motor adaptation (Methods). As expected, these models also exhibited a change in activity early during the movement (compare the blue traces in Figure 4.5B and Figure 4.5C), confirming that it does represent learning. In contrast, the second peak (green trace in Figure 4.5B) was absent from these models without feedback-driven plasticity, confirming its source.

Having uncovered a signature of feedback-driven adaptation in our model, we sought to identify a similar change in neural population recordings from monkey M1 (Figure 4.5D; data from Ref.[107]; Methods). Figure 4.5E shows the average change in M1 activity during a representative VR adaptation session, qualitatively confirming our prediction of a feedback signal during early adaptation trials (circle in Figure 4.5E). Reassuringly, this feedback signal occurred 300 ms after peak speed (which happened 500 ms after the go cue), as it did in other studies exploring feedback-based movement corrections[186,199]. To further compare the monkey data to our model, we calculated the ratio of the adaptation-related activity change at the "feedback peak" (green circle in Figure 4.5B,C,E) and the learning-related "feedforward peak" (blue star in Figure 4.5B,C,E) (Methods). For both monkeys, there was a substantial change in neural activity at the feedback peak during adaptation (Figure 4.5G), as predicted by our model (Figure 4.5F). Moreover, the value of this ratio was similar across both monkeys and sessions, and best matched, in terms of mean and variability, that of models with high feedback projection densities (Figure 4.5F,G). The fact that models with dense feedback projections best replicated the experimental data is in good agreement with observations suggesting that as many as 73% of M1 neurons are driven by feedback[186]. The similarities in the activity changes during adaptation between our model and monkey M1 suggest that error feedback signals in M1 may indeed guide local plasticity relevant for rapid motor learning.

4.4 Discussion

Both the rapid correction of ongoing movements and progressive adaptation to changing conditions are key landmarks of behaviour that are often studied separately. Here, we have shown that an RNN that can dynamically control its output based on feedback inputs can use those same inputs to achieve motor adaptation through recurrent connectivity changes. Interestingly, this form of feedback-driven plasticity recapitulated key aspects of primate adaptation behaviour[69,70,187,225,226], and led to identifiable activity changes in the model that we subsequently discovered in neural recordings made in the primary motor cortex of two monkeys. These results, which were robust across a broad change of model parameters (Figure 4.4 and Figure 4.S2), support the hypothesis that online movement correction and motor adaptation may share a common implementation in neural circuits.

Implementation of feedback-based motor control in neural circuitry

Most modelling studies on motor control have focused on understanding it at an abstract computational level, mostly ignoring neurons and connections between them[26,187,223]. One potential reason is the challenge of mapping the abstract concepts of optimal feedback control theory into brain regions[7,29]. Our work differs from those previous attempts in that it approaches the problem from a bottom-up, not a top-down perspective: instead of explicitly training neural networks to match the behavioural components predicted by optimal feedback control theory, such as a forward model or a controller[228], we let error minimization guide the emergence of an efficient control strategy, potentially mimicking how brain connectivity developed over evolutionary timescales[169,229,230]. Given that this bottom-up approach led to a feedback-based learning process that recapitulated key aspects of human behavioural adaptation (Figure 4.2 and Figure 4.3), we believe that our model could help map specific functions that have been formalized in optimal feedback control onto neural circuits, adding to existing approaches[28–30,168,231,232].

While in this paper we have compared the network activity to recordings from monkey M1, our model (Figure 5) does not necessarily encompass only M1 function. On the contrary, it is likely that it captures functions mediated in part through multi-region interactions with a variety of cortical and subcortical regions[233,234], such as feedback processing, or sensory gating. Extending our work to

modular, multi-region RNNs whose activity is compared to neural population recordings across the sensorimotor network (such as in Ref.[61,97,154]) may shed light into the distributed implementation across brain circuits of the different computations underlying feedback control.

Early studies proposed the cerebellum as the key structure for motor adaptation[7,137,174,204,207,214,235–239]. Their central premise was that the cerebellum stores both "forward" and "inverse" internal models of the sensorimotor system which are used to generate and update predicted outcomes of motor commands, respectively[22,222,240,241]. The importance of internal models in the cerebellum received additional support from evidence of adaptation deficits in cerebellar patients[242], although this view has recently been challenged by a proposal that adaptation is better described by a direct update of a control policy[224]. Our study does not contradict this work; it illustrates an alternative, perhaps parallel learning process relevant for motor adaptation. Moreover, the cerebellum could readily fit into our model as the brain region responsible for calculating error estimates based on the sensory feedback, which we simply assumed to exist. Interestingly, the fact that cerebellar patients have impairments in motor control as well as learning[137,166] also fits with our model, since both processes are critically dependent on the availability of an accurate error signal. Combining our bottom-up model of feedback-based learning in M1 with previous models of the cerebellum[204,232] might provide a fruitful route to uncover how each of these two brain regions contributes to motor adaptation.

Biologically plausible learning rules

Our proposed feedback-driven learning rule adds to recent efforts to implement biologically plausible learning in RNNs[92–94,173,243–245] —a challenging temporal credit assignment problem. Most of these learning rules share the same basic idea: the weight change is proportional to the error signal arriving at the postsynaptic neuron times the activity history of the presynaptic neuron. What makes those learning rules biologically plausible is that both pieces of information could in principle be locally available at a synapse.

The crucial difference between our model and previous studies is that the error signal is a direct input to the neuron, which allows it to simultaneously guide weight update and affect the ongoing network dynamics. This feature is desirable because it avoids the need to have two distinct pathways for error signals and ongoing network dynamics, respectively[128,246,247]. Moreover, our approach makes weight update dependent on the error signal in two ways: directly, through the error term in the

plasticity rule, and indirectly, through the change in ongoing dynamics, which influences the activity of the presynaptic neuron. This could be beneficial for learning and should be compared to standard learning algorithms like gradient descent.

Finally, our feedback-driven plasticity rule led to an adaptation time course that replicated the simultaneous fast and slow processes found in human behavioural studies (Figure 4.3C;[69,187]). This seems puzzling given that the fast process is often assumed to be explicit[68,69], a component not possible in our RNN implementation. Therefore, the observation of a fast component despite adaptation being mediated purely by an error feedback signal suggests that, in principle, error minimisation alone can give rise to learning at multiple timescales.

Conclusions

We have shown how the same error feedback signal that mediates ongoing motor corrections can drive trial-by-trial motor adaptation by guiding synaptic plasticity using a relatively simple, biologically plausible plasticity rule. Crucially, this bottom-up model of a shared implementation of feedback-based motor control and rapid learning recapitulated key observations from behavioural and neurophysiological adaptation studies in humans and monkeys. Thus, these two landmarks of animal behaviour may be unified in their implementation by the same neural circuitry.

4.5 Methods

Recurrent Neural Network model

Neural activity x was simulated using the following dynamical equations,

$$\begin{aligned} x_j(t+1) &= x_j(t) + \frac{\mathrm{dt}}{\tau}\left(-x_j(t) + \sum_i W_{ji}\Phi(x_i(t)) + \sum_i W^{in}_{ji}s_i(t) + \sum_i F_{ji}\epsilon_i(t-\Delta) + b_j\right) \\ r_j(t) &= \sum_{t'<t}^{T} \Phi(x_j(t')) \\ v_k(t) &= \sum_i W^{out}_{ki}\Phi(x_i(t)) + b^{out}_k \\ p_k(t) &= p_k(t-1) + \mathrm{dt}\cdot v_k(t) \\ \epsilon_k(t) &= p^*_k(t) - p_k(t) \end{aligned}$$

where the network output v represents velocity, and p position, of the simulated planar hand movement (cf. definitions in Table 4.1). The instantaneous error signal ϵ is given by the difference between the target p^* and the produced position p, and is fed back to the network with a time delay Δ. How we constructed the network input s and the target position p^* are described in the "Reaching datasets for model training and testing" section below. Each trial was initialized by setting all x_j to random numbers uniformly distributed between -0.2 and 0.2. All simulations were performed on RNN models consisting of 400 neurons, which were connected all-to-all. Varying the recurrent connection probability did not change the results (Figure 4.S2).

Model training procedure

The first step was to train the RNN to control its own output, that is, to minimize the position error, ϵ. This training was performed using standard gradient descent to find the right set of parameters. The initial training procedure was implemented in Pytorch[190] using the Adam optimizer with learning rate $\alpha = 0.001$ (β_1 = 0.9, β_2 = 0.999)[135]. The weights (W,W^{in},W^{out},F) and biases (b,b^{out}) were initialized by drawing random, uniformly distributed numbers between $-1/\sqrt{l}$ and $1\sqrt{l}$, where l is either the number of neurons in the network (for W,W^{in},F,b) or the dimensionality of the output (for W^{out},b^{out}). The gradient norm was clipped at 0.2 prior to the optimization step. The loss function used for this

Parameter	Definition	Value
dt	Time step	10ms
τ	Time constant	50ms
Δ	Feedback delay	120ms
N	Number of neurons	400
Φ	Nonlinearity	ReLU
$\Phi(x)$	Neural activity	-
s	Stimulus	-
ϵ	Position error	-
r	Eligibility trace	-
W	Recurrent weight matrix	-
b	Recurrent offset	-
W^{in}	Input weight matrix	-
W^{out}	Output weight matrix	-
b^{out}	Output offset	-
F	Feedback weight matrix	-
v	Velocity (2D)	-
p	Position (2D)	-
α	Gradient Descent: Learning rate	0.001
B	Gradient Descent: batch size	20
β	Gradient Descent: weight regularization	0.001
γ	Gradient Descent: activity regularization	0.002
η	Feedback-driven plasticity: learning rate	0.000001
T	Reach trajectories: number of time steps in a trial	300

Table 4.1: Simulation parameters.

initial training phase was defined as

$$L = \frac{1}{2BT}\sum_{b}^{B}\sum_{t}^{T}\sum_{k=x,y}\epsilon_k^2(t,b) + \beta \sum_{M=(W,W^{in},W^{out},F,b,b^{out})} ||M||_2 + \gamma\frac{1}{NBT}\sum_{b}^{B}\sum_{t}^{T}\sum_{j}^{N}\Phi(x_j(t,b))^2$$

where B is the batch size, T the number of time steps, N the number of neurons, β the regularization parameter for the weights and bias terms and γ the regularization parameter for the activity in the network (cf. definitions in Table 4.1). The network was trained for 1100 epochs, divided into three blocks of different lengths (100, 500, 500). For the first 100 epochs, the feedback weights F were kept fixed while the remaining parameters where allowed to change. This ensured that the model learnt to self-generate the appropriate network dynamics to produce a variety of reaching trajectories. In the next 500 epochs, the feedback connections were also allowed to change. In the last 500 epochs, we introduced perturbations on the produced output (see "Reaching datasets for model training and testing"), with all parameters plastic, to make the model learn to use the feedback inputs to compensate for ongoing errors.

A feedback-driven plasticity rule to drive trial-by-trial learning

Having set up the model to control its own output, we next examined how feedback inputs ϵ could guide learning, implemented through synaptic plasticity within the recurrent weights W of the network. To this end, we devised the following *feedback-driven plasticity rule*:

$$\begin{aligned} dW_{ji}(t) &= \mathrm{dt}\ \eta \sum_{k=x,y} F_{jk}\epsilon_k(t) r_i(t) \\ W_{k+1} &= W_k + \sum_t dW(5t) \end{aligned}$$

The weight update dW was calculated online and summed up taking into account every fifth time step until the end of a trial k. After each trial, we applied this accumulated weight change and updated the recurrent weights W accordingly.

Reaching datasets for model training and testing

The network model was trained to produce a broad set of synthetic planar reaching trajectories following an instructed delay phase. The x and y positions of the starting (p^{start}) and ending points (p^{end}) of those trajectories were randomly drawn from a uniform distribution ranging from -6 cm to 6 cm. To simulate natural reaching behaviour, we interpolated between these points using a sigmoid function

$$f(t) = \frac{1}{1 + \exp(-t \cdot \kappa)}$$

where κ=10/s. The manually constructed reach trajectories were thus given by

$$p_t^* = p^{start} + f(t)(p^{end} - p^{start})$$

which resulted in bell-shaped velocity profiles.

Each trial lasted 3 s and included an instructed delay period, randomly drawn from 0 s to 1.5 s.

The network received an input signal consisting of a two-dimensional target signal and a one-dimensional timing signal. The target signal was defined as $(p^{end} - p^{start})$. It was delivered to the network 0.2 s after trial onset, and was fixed until the end of the trial. The timing signal was given in form of a constant, and was switched to zero at the time corresponding to the "go" signal, which varied between 0.2 s and 1.7 s for the random reaching task used during training (that is, when the

network generated reaches of random direction and lengths up to 8.5cm), and between 1.2 s and 1.7 s for the centre-out-reaching task.

As mentioned above, during the last phase of the initial training phase we included brief "bump" perturbations to the output of the network so it had to learn to use the feedback input to correct its output online. In 75 % of the trials, we added a pulse of 0.1 s duration and amplitude 10 cm/s on the velocity output of the model, either in x or y direction. This pulse occurred randomly between 0.2 s and 1.9 s after trial onset, to mimic perturbations at various movement periods.

After training, we tested the model on a centre-out-reaching task with eight targets equally distributed on a circle of 5 cm radius. In this task, the go signal occurred randomly between 1.2 s and 1.7 s, as before.

To probe online feedback correction and motor adaptation, we introduced a visuomotor rotation (VR) perturbation that rotated the output of the model by 30°, similar to previous visuomotor rotation experiments in humans[70] and monkeys[107].

Neural recordings from behaving monkeys

We reanalysed previously published data from two macaque monkeys performing a visuomotor adaptation reaching task with a cursor controlled by movements of a manipulandum. In each session the monkeys performed 154–217 successful trials of an eight target centre-out reaching task. After this baseline period, a 30° rotation (clockwise or counter clockwise, depending on the session) of the cursor position feedback presented on a screen was introduced. Finally, after 219–316 successful adaptation trials, the perturbation was removed in order to study de-adaptation during this "washout" period.

We analysed the activity of populations of putative single neurons recorded using 96-channel microelectrode arrays chronically implanted in the arm area of primary motor cortex (details in Ref.[107]). We quantified trial-by-trial learning by examining the monkey's hand trajectories, which was tracked by recording the position of the handle of the manipulandum. All surgical and behavioural procedures were approved by the Institutional Animal Care and Use Committee at Northwestern University (Chicago, USA).

Data analysis

Movement error metrics to quantify learning

The take-off angle was defined as the initial reach direction, calculated between the go cue and peak velocity. When pooling the angular error across monkeys in Figure 4.2, we smoothed the mean across all sessions from both animals using a Gaussian filter with s.d. of ten trials.

When studying how error magnitude influences learning in the next trial (Figure 4.3A), we computed the Pearson's correlation (*pearsonr* from `scipy.stats` package) between the absolute value of the angular error and the difference in angular error between the current trial and the next trial. To assess whether these correlations were significant, we compared them to a null distribution under the assumption of joint normality. The movement error in Figure 4.3B is defined as the averaged squared position error ϵ.

Analysis of learning timescales

We investigated whether our model's learning time course is composed of two processes with different timescales by implementing the analysis used in earlier studies Ref.[69,187]. We fitted a dual-rate state-space model to the angular error data, defined as below:

$$x_f(n+1) = A_f x_f(n) + B_f e(n) \tag{4.1}$$

$$x_s(n+1) = A_s x_s(n) + B_s e(n) \tag{4.2}$$

$$x(n) = x_f(n) + x_s(n) \tag{4.3}$$

subject to the constraints $A_f < A_s$, $B_f > B_s$. A_f and B_f are the parameters describing the fast process, whereas A_s and B_s are the parameters describing the slow process. The *Adaptation* variable, x (cf. Figure 4.3C), is defined as the amount of change in the take-off reaching direction. The error e is given by the take-off angular error scaled to [-1,1]. The four parameters A_f, A_s, B_f, B_s were obtained by fitting the model to the adaptation time course observed in the simulated data. For this, we used the *Sequential Least Squares Programming* method from the `scipy` optimization package.

Analysis of temporally dissociable adaptation-related activity changes

We sought to identify a "neural signature" of adaptation-related activity changes in the network that could be seen in neural recordings from behaving monkeys. To this end, we probed three different "behavioural epochs" as follows (Figure 4.5A). For the model data, we simulated 200 baseline trials (epoch A in Figure 4.5A), 200 trials beginning immediately after perturbation onset (prior to any learning; epoch B), and 200 trials beginning 300 trials after the onset of learning (epoch C). For the monkey data, we considered the following: 100 baseline trials (epoch A), the first 100 trials after perturbation onset, during which monkeys were beginning to adapt (epoch B), and the last 100 perturbation trials, when monkeys had learned to counteract the perturbation (epoch C). Note that for the monkey data, the feedback epoch B was not as clearly defined as for the simulation data, since the monkeys had already started learning within epoch B.

The activity change in the simulation data was calculated by measuring, for each unit, the activity difference between all pairs of behavioural epochs (A, B, C in Figure 4.5A). For this, we simulated the same trials (using the same random seed) without perturbations (A), with perturbations (B), and with perturbations after the network had adapted (C). To identify the time point within a trial at which the largest activity change happened, we computed the absolute value of activity change, and averaged the respective differences across neurons and trials. This resulted in the time courses shown in Figure 4.5A,B. For the monkey data (Figure 4.5E), since we could not have the exact trials in different epochs, we calculated the difference between trials in different epochs in an all-to-all fashion, then averaged over those trial pairs.

After an initial analysis of the average activity change across all ten RNN models, we could define a "feedforward" time point (0.5 s after the go cue), in which the largest activity change between late adaptation (epoch C) and early adaptation (epoch B) happened, and a "feedback" time point (0.8 s after the go cue), in which the largest activity change between early adaptation (epoch B) and baseline (epoch A) happened. These values were very similar to those identified in the analysis of neural recordings from monkey M1: feedforward time point, 0.4 s after the go cue; feedback time point, 0.8 s after the go cue. For the pooled analysis presented in Figure 4.5F,G, we took the values of the activity change traces at those time points and calculated the ratio between the value at the feedforward time point and the value at the feedback time point.

4.6 Supporting information

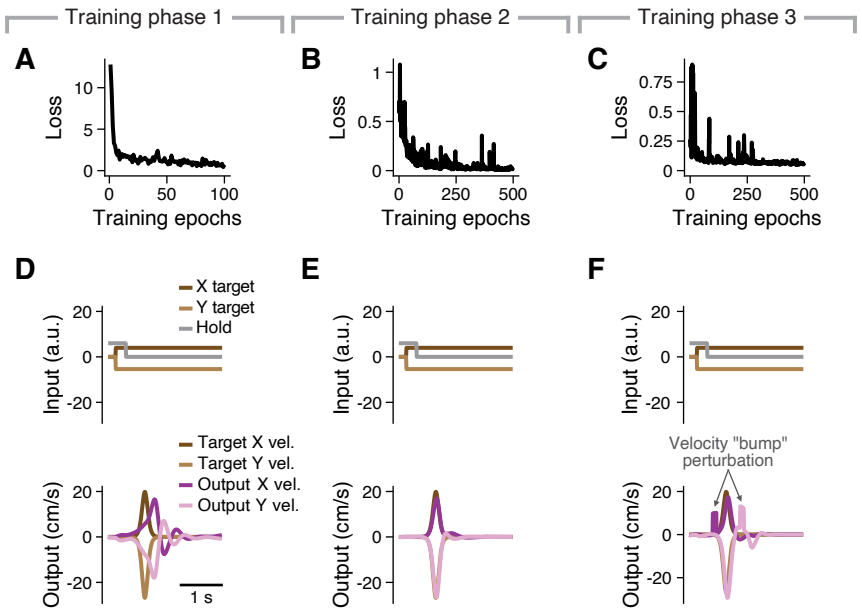


Figure 4.S1: Initial training protocol. The initial training of the RNN is divided into three phases. In the first phase, we kept the feedback weights W^{fb} fixed and at small initial values (A,D). In the second phase, we lifted that constraint, and all model parameters became plastic (B,E). In the third phase, we introduced random velocity perturbations in 75 % of trials (C,F). **A-C.** Training loss for each of the three training phases. **D-F.** Input (top) and output (bottom) for an example test trial for each of the three training phases.

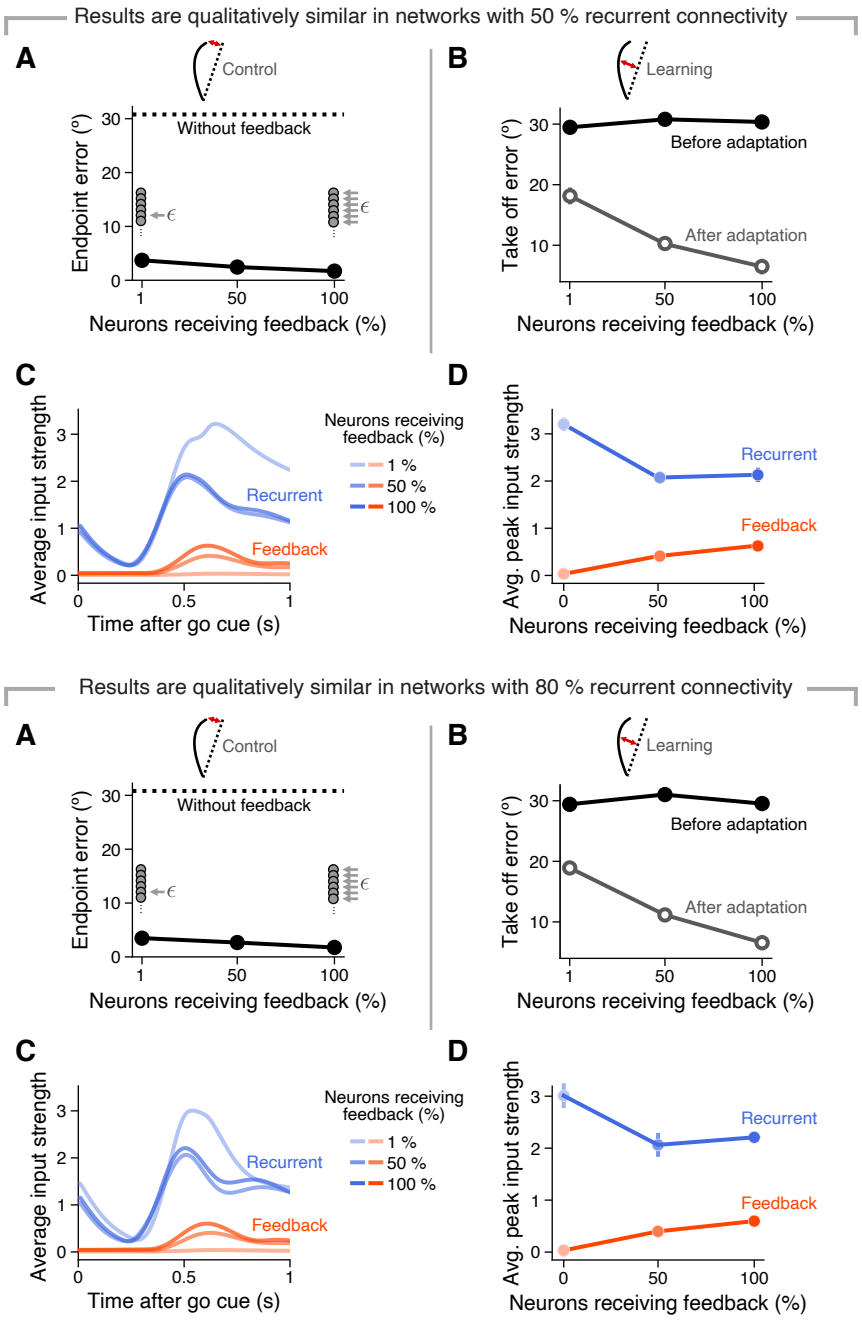


Figure 4.S2: Effective feedback-based motor control and adaptation can both be achieved with sparse feedback for varied degrees of recurrent connectivity. Simulation results for networks with 50% (**A-D**) or 80% (**E-H**) recurrent connection probability. **A.** Angular error between produced and target position at the end of the reach immediately after onset of visuomotor rotation onset for networks with different percentages of neurons receiving afferent feedback (black markers), including no feedback (dashed line). Lines and error bars, mean and s.d. across ten networks. **B.** Take-off error at visuomotor rotation onset (solid circles), and after adaptation (empty circles) for networks with different percentages of neurons receiving afferent feedback. Lines and error bars, mean and s.d. across ten networks. **C.** Average recurrent (blue) and feedback (red) inputs to an RNN neuron before adaptation. Average input strength is defined as the mean across incoming signals and neurons. Legend, percentage of neurons receiving feedback. **D.** Average magnitude of the peak input strength of the recurrent (blue) and feedback (red) inputs before adaptation. Same colour scheme as in C. Lines and error bars, mean and s.d. across ten networks. Data in E,F,G,H are presented as in A,B,C,D.

CHAPTER 5

Conclusion

5.1 Summary

This thesis is concerned with three distinct subtopics, all related to the big question of how learning is implemented in the brain. As it is impossible to directly measure how synapses change over time most of the presented work is based on computational modelling. We believe that such modelling approaches can help to define more precise predictions, which should eventually lead to a better understanding of plasticity in the brain. The main idea throughout this work is to compare characteristics of the neural network model which "learns", either behavioural or neural, to experimental measurements, and thus either verify or falsify certain assumptions about how the brain learns in a more streamlined and quantifiable way.

Firstly, we looked at why it is easier to learn new activity patterns that lie "within" the neural manifold, compared learning "outside" of it (Chapter 2). Studying the effect of different biological constraints on learning in a computational model revealed a potential explanation of the initial finding: Learning how to correctly update the connections between neurons, or neural plasticity, was only feasible in the case of within-manifold learning, as for outside-manifold learning there was no straightforward way to correctly relate the task error with the internal neural firing.

Secondly, we tackled the question of whether we can derive where learning has happened in a hierarchical neural pathway from simply measuring neural activity before and after learning (Chapter 3). Using a similar modelling approach as in chapter two, we found that commonly used motor adaptation paradigms require only minimal changes in connectivity, preserve neural coactivation

patterns, and lead to only small changes in neural activity between before and after learning. This makes it hard to distinguish between learning in the local circuitry and learning in upstream areas. To alleviate this ambiguity in the future we proposed new experiments where the model predicted a more drastic difference between local and upstream learning.

Lastly, we focused on the question of how motor adaptation could be implemented in neural circuitry (Chapter 4). We found that a model trained to actively control its output, such that it can flexibly correct unpredictable perturbations, can use those feedback inputs as a guiding signal for local, recurrent plasticity. The model did not only produce similar characteristics in terms of activity changes during adaptation compared to monkey M1, it also recapitulated several experimental results from human motor adaptation experiments. Those results challenge the current view in motor control that we need multiple "internal models" of the world to be able to accurately control our body in the environment. We argue here that all those computational concepts could in principle be achieved by a single recurrent circuit, posing the question of how such a system would be integrated with other control and learning mechanisms across the brain.

5.2 Impact

Finding direct experimental evidence for learning in the brain is a daunting task as experimental techniques, allowing to track not only neural activity but also synaptic weights remain limited[248–250]. One part of the problem is that the intricate connectivity network is orders of magnitudes more complex than "just" looking at the neurons themselves, as an average neuron could be connected to 7000 other neurons[251]. Therefore, a lot of what we think we know about learning is often based on indirect evidence and intuition. For example, it has been shown that neural connectivity changes are related to long-term learning[123,141–143], but to what extent this is the case on shorter timescales remains unclear. Previous work has argued that rapid motor adaptation might use a different mechanism, mostly based on the idea that connectivity changes would lead to more dramatic changes in neural dynamics than observed experimentally[107,148,149]. Our work challenges this intuition as it shows that local plasticity can lead to similarly small changes in neural activity compared to learning through changes in the input signals (Chapter 3).

We believe that using tools from state-of-the-art machine learning to create artificial agents, which can perform tasks we are interested in and which share desired biological design principles, is an

immensely useful approach to drive experimental research and helps coming up with educated guesses in a more streamlined and quantifiable manner. This work shows how diversely such a method can be used to address questions about neural implementation of learning. RNNs have been used extensively in various topics of neuroscience: motor control[42,58–61], cognition[62,63], sensory processing[64], memory formation[65,66]. However, so far most of this work is concerned about representational similarity between artificial and biological networks. Here we try to go one step further and actually look at learning behaviour in both systems. Although we know that the brain is not using machine-learning-style gradient descent[86], we would argue that we can still use this algorithm to obtain boundaries on the amount of weight change we need to accomplish a certain task. This might very well depend on what kind of biological constraint we insert in our modelling approach, which provides us with even further understanding of how the effect of specific weight changes is potentially limited by distinct biological factors.

5.3 Future work

One future direction of this line of research could be to combine the modelling work presented in chapter 4 with recently presented work on learning in the cerebellum[252,253]. Investigating how those two independent learning processes could coexist, or even support each other, would be highly relevant, as there is growing evidence of multiple, synergistic learning processes in the brain[68]. Another direction could be to ask how learning might depend on the specific biophysical properties of our body[254]. For this, one could start with the model presented in chapter 3, but instead of having a simple output transformation one could use a more complex biophysical arm model. Thanks to recent advancements in translating realistic biophysical models into the framework of modern machine learning, it is only now possible to use such complex output transformations and still be able to sufficiently train the simulated brain networks[255]. Lastly, the model approach presented in chapter 3 could become valuable regarding recent experimental progress in using optogenetic perturbation in monkeys[256–258]. Our models could be used to predict how for example M1 activity should change in response to a perturbation in PMd. Such a combined workflow can be immensely powerful as it allows to directly test causal relationships in simulated and biological neural circuits.

5.4 Conclusion

In summary, we used recurrent neural network models to shed light on previously inaccessible aspects of the neural implementation of learning. We looked at three specific aspects of learning: 1) why can some neural activity patterns be easier learned than others (Chapter 2), 2) can we deduce where learning has happened (Chapter 3), and 3) how can biologically plausible learning be implemented in recurrent networks (Chapter 4)? We believe that this work will guide future research to rely more on model based hypotheses, and that it will eventually lead to a faster turn-around of ideas in the field of neuroscience, which is after all still in its infancy.

CHAPTER 6

Bibliography

1. Chapman, G. Of the movement of worms. *Journal of Experimental Biology* **27,** 29–39 (1950).
2. Biewener, A. A. Biomechanics of mammalian terrestrial locomotion. *Science* **250,** 1097–1103 (1990).
3. Garcia, A., Krummel, G. & Priya, S. Fundamental understanding of millipede morphology and locomotion dynamics. *Bioinspiration & Biomimetics* **16,** 026003 (2020).
4. Klein, C. S., Marsh, G. D., Petrella, R. J. & Rice, C. L. Muscle fiber number in the biceps brachii muscle of young and old men. *Muscle & Nerve: Official Journal of the American Association of Electrodiagnostic Medicine* **28,** 62–68 (2003).
5. Brooks, S. V. Current topics for teaching skeletal muscle physiology. *Advances in physiology education* **27,** 171–182 (2003).
6. Sherrington, C. S. *The integrative action of the nervous system* (Yale University Press, 1906).
7. Shadmehr, R. & Krakauer, J. W. A computational neuroanatomy for motor control. *Experimental brain research* **185,** 359–381 (2008).
8. Scott, S. H. Optimal feedback control and the neural basis of volitional motor control. *Nature Reviews Neuroscience* **5,** 532–545 (2004).
9. Pruszynski, J. A. Primary motor cortex and fast feedback responses to mechanical perturbations: a primer on what we know now and some suggestions on what we should find out next. *Frontiers in Integrative Neuroscience* **8,** 72 (2014).

10. Buschman, T. J. & Miller, E. K. Goal-direction and top-down control. *Philosophical Transactions of the Royal Society B: Biological Sciences* **369,** 20130471 (2014).

11. Duan, L. Y. *et al.* Controlling one's world: Identification of sub-regions of primate PFC underlying goal-directed behavior. *Neuron* **109,** 2485–2498 (2021).

12. Poliak, S. & Peles, E. The local differentiation of myelinated axons at nodes of Ranvier. *Nature Reviews Neuroscience* **4,** 968–980 (2003).

13. Van Acker III, G. M., Luchies, C. W. & Cheney, P. D. Timing of cortico-muscle transmission during active movement. *Cerebral Cortex* **26,** 3335–3344 (2016).

14. Pruszynski, J. A. *et al.* Primary motor cortex underlies multi-joint integration for fast feedback control. *Nature* **478,** 387–390 (2011).

15. Stavisky, S. D., Kao, J. C., Ryu, S. I. & Shenoy, K. V. Motor cortical visuomotor feedback activity is initially isolated from downstream targets in output-null neural state space dimensions. *Neuron* **95,** 195–208 (2017).

16. Cross, K. P., Cluff, T., Takei, T. & Scott, S. H. Visual feedback processing of the limb involves two distinct phases. *Journal of Neuroscience* **39,** 6751–6765 (2019).

17. Morikiyo, Y. & Matsushima, T. Effects of delayed visual feedback on motor control performance. *Perceptual and motor skills* **70,** 111–114 (1990).

18. Kawato, M. Internal models for motor control and trajectory planning. *Current opinion in neurobiology* **9,** 718–727 (1999).

19. Wagner, M. J. & Smith, M. A. Shared internal models for feedforward and feedback control. *Journal of Neuroscience* **28,** 10663–10673 (2008).

20. Volkinshtein, D. & Meir, R. Delayed feedback control requires an internal forward model. *Biological cybernetics* **105,** 41–53 (2011).

21. Kawato, M., Furukawa, K. & Suzuki, R. A hierarchical neural-network model for control and learning of voluntary movement. *Biological cybernetics* **57,** 169–185 (1987).

22. Wolpert, D. M., Miall, R. C. & Kawato, M. Internal models in the cerebellum. *Trends in cognitive sciences* **2,** 338–347 (1998).

23. Shidara, M., Kawano, K., Gomi, H. & Kawato, M. Inverse-dynamics model eye movement control by Purkinje cells in the cerebellum. *Nature* **365,** 50–52 (1993).

24. Kawano, K. *et al.* Visual inputs to cerebellar ventral paraflocculus during ocular following responses. *Progress in brain research* **112,** 415–422 (1996).

25. Todorov, E. Direct cortical control of muscle activation in voluntary arm movements: a model. *Nature neuroscience* **3,** 391–398 (2000).

26. Todorov, E. & Jordan, M. I. Optimal feedback control as a theory of motor coordination. *Nature neuroscience* **5,** 1226–1235 (2002).

27. Scott, S. H., Cluff, T., Lowrey, C. R. & Takei, T. Feedback control during voluntary motor actions. *Current opinion in neurobiology* **33,** 85–94 (2015).

28. Kalidindi, H. T. *et al.* Rotational dynamics in motor cortex are consistent with a feedback controller. *Elife* **10,** e67256 (2021).

29. Takei, T., Lomber, S. G., Cook, D. J. & Scott, S. H. Transient deactivation of dorsal premotor cortex or parietal area 5 impairs feedback control of the limb in macaques. *Current Biology* **31,** 1476–1487 (2021).

30. Kao, T.-C., Sadabadi, M. S. & Hennequin, G. Optimal anticipatory control as a theory of motor preparation: a thalamo-cortical circuit model. *Neuron* **109,** 1567–1581 (2021).

31. Stringer, C. *et al.* Spontaneous behaviors drive multidimensional, brainwide activity. *Science* **364,** eaav7893 (2019).

32. Cunningham, J. P. & Yu, B. M. Dimensionality reduction for large-scale neural recordings. *Nature neuroscience* **17** (2014).

33. Sadtler, P. T. *et al.* Neural constraints on learning. *Nature* **512,** 423–426 (2014).

34. Gao, P. & Ganguli, S. On simplicity and complexity in the brave new world of large-scale neuroscience. *Current opinion in neurobiology* **32,** 148–155 (2015).

35. Gallego, J. A., Perich, M. G., Miller, L. E. & Solla, S. A. Neural Manifolds for the Control of Movement. *Neuron* **94,** 978–984 (2017).

36. Gallego, J. A. *et al.* Cortical population activity within a preserved neural manifold underlies multiple motor behaviors. *Nature communications* **9,** 4233 (2018).

37. Churchland, M. M. *et al.* Neural population dynamics during reaching. *Nature* **487,** 51–56 (2012).

38. Shenoy, K. V., Sahani, M., Churchland, M. M., *et al.* Cortical control of arm movements: a dynamical systems perspective. *Annu Rev Neurosci* **36,** 337–359 (2013).

39. Vyas, S., Golub, M. D., Sussillo, D. & Shenoy, K. V. Computation through neural population dynamics. *Annual Review of Neuroscience* **43,** 249 (2020).

40. Saxena, S. & Cunningham, J. P. Towards the neural population doctrine. *Current opinion in neurobiology* **55,** 103–111 (2019).

41. Kaufman, M. T., Churchland, M. M., Ryu, S. I. & Shenoy, K. V. Cortical activity in the null space: permitting preparation without movement. *Nature neuroscience* **17,** 440–448 (2014).

42. Sussillo, D., Churchland, M. M., Kaufman, M. T. & Shenoy, K. V. A neural network that finds a naturalistic solution for the production of muscle activity. *Nature neuroscience* **18,** 1025–1033 (2015).

43. Remington, E. D., Egger, S. W., Narain, D., Wang, J. & Jazayeri, M. A dynamical systems perspective on flexible motor timing. *Trends in cognitive sciences* **22,** 938–952 (2018).

44. Sauerbrei, B. A. *et al.* Cortical pattern generation during dexterous movement is input-driven. *Nature* **577,** 386–391 (2020).

45. Barak, O., Sussillo, D., Romo, R., Tsodyks, M. & Abbott, L. From fixed points to chaos: three models of delayed discrimination. *Progress in neurobiology* **103,** 214–222 (2013).

46. Sussillo, D. & Barak, O. Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural computation* **25,** 626–649 (2013).

47. Hennequin, G., Ahmadian, Y., Rubin, D. B., Lengyel, M. & Miller, K. D. The dynamical regime of sensory cortex: stable dynamics around a single stimulus-tuned attractor account for patterns of noise variability. *Neuron* **98,** 846–860 (2018).

48. Maheswaranathan, N., Williams, A., Golub, M., Ganguli, S. & Sussillo, D. Reverse engineering recurrent networks for sentiment classification reveals line attractor dynamics. *Advances in neural information processing systems* **32** (2019).

49. Ostojic, S. Two types of asynchronous activity in networks of excitatory and inhibitory spiking neurons. *Nature neuroscience* **17,** 594–600 (2014).

50. Brunel, N. Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. *Journal of computational neuroscience* **8,** 183–208 (2000).

51. Renart, A. *et al.* The asynchronous state in cortical circuits. *science* **327,** 587–590 (2010).

52. Van Vreeswijk, C. & Sompolinsky, H. Chaos in neuronal networks with balanced excitatory and inhibitory activity. *Science* **274,** 1724–1726 (1996).

53. Laje, R. & Buonomano, D. V. Robust timing and motor patterns by taming chaos in recurrent neural networks. *Nature neuroscience* **16,** 925–933 (2013).

54. Aljadeff, J., Renfrew, D., Vegue, M. & Sharpee, T. O. Low dimensional dynamics of structured random networks. *Physical Review E* **93,** 1–16 (2016).

55. Mastrogiuseppe, F. & Ostojic, S. Linking Connectivity, Dynamics, and Computations in Low-Rank Recurrent Neural Networks. *Neuron* **99,** 609–623 (2018).

56. Litwin-Kumar, A. & Doiron, B. Slow dynamics and high variability in balanced cortical networks with clustered connections. *Nature neuroscience* **15,** 1498–1505 (2012).

57. Barak, O. Recurrent neural networks as versatile tools of neuroscience research. *Current opinion in neurobiology* **46,** 1–6 (2017).

58. Hennequin, G., Vogels, T. P. & Gerstner, W. Optimal control of transient dynamics in balanced networks supports generation of complex movements. *Neuron* **82,** 1394–1406 (2014).

59. Stroud, J. P., Porter, M. A., Hennequin, G. & Vogels, T. P. Motor primitives in space and time via targeted gain modulation in cortical networks. *Nature neuroscience* **21,** 1774–1783 (2018).

60. Wang, J., Narain, D., Hosseini, E. A. & Jazayeri, M. Flexible timing by temporal scaling of cortical responses. *Nature neuroscience* **21,** 102–110 (2018).

61. Michaels, J. A., Schaffelhofer, S., Agudelo-Toro, A. & Scherberger, H. A goal-driven modular neural network predicts parietofrontal neural dynamics during grasping. *Proceedings of the national academy of sciences* **117,** 32124–32135 (2020).

62. Sohn, H., Narain, D., Meirhaeghe, N. & Jazayeri, M. Bayesian computation through cortical latent dynamics. *Neuron* **103,** 934–947 (2019).

63. Yang, G. R., Joglekar, M. R., Song, H. F., Newsome, W. T. & Wang, X.-J. Task representations in neural networks trained to perform many cognitive tasks. *Nature neuroscience* **22,** 297–306 (2019).

64. Lindsay, G. W. Convolutional neural networks as a model of the visual system: Past, present, and future. *Journal of cognitive neuroscience* **33,** 2017–2031 (2021).

65. Rajan, K., Harvey, C. D. & Tank, D. W. Recurrent network models of sequence generation and memory. *Neuron* **90,** 128–142 (2016).

66. Masse, N. Y., Yang, G. R., Song, H. F., Wang, X.-J. & Freedman, D. J. Circuit mechanisms for the maintenance and manipulation of information in working memory. *Nature neuroscience* **22,** 1 (2019).

67. Krakauer, J. W., Hadjiosif, A. M., Xu, J., Wong, A. L. & Haith, A. M. Motor learning. *Compr Physiol* **9,** 613–663 (2019).

68. Taylor, J. A., Krakauer, J. W. & Ivry, R. B. Explicit and implicit contributions to learning in a sensorimotor adaptation task. *Journal of Neuroscience* **34,** 3023–3032 (2014).

69. McDougle, S. D., Bond, K. M. & Taylor, J. A. Explicit and implicit processes constitute the fast and slow processes of sensorimotor learning. *Journal of Neuroscience* **35,** 9568–9579 (2015).

70. Krakauer, J. W., Pine, Z. M., Ghilardi, M.-F. & Ghez, C. Learning of visuomotor transformations for vectorial planning of reaching trajectories. *Journal of Neuroscience* **20,** 8916–8924 (2000).

71. Citri, A. & Malenka, R. C. Synaptic plasticity: multiple forms, functions, and mechanisms. *Neuropsychopharmacology* **33,** 18–41 (2008).

72. Abbott, L. F. & Nelson, S. B. Synaptic plasticity: taming the beast. *Nature neuroscience* **3,** 1178–1183 (2000).

73. Zenke, F., Agnes, E. J. & Gerstner, W. Diverse synaptic plasticity mechanisms orchestrated to form and retrieve memories in spiking neural networks. *Nature communications* **6,** 1–13 (2015).

74. Abbott, L. F., Varela, J., Sen, K. & Nelson, S. Synaptic depression and cortical gain control. *Science* **275,** 221–224 (1997).

75. Markram, H., Wang, Y. & Tsodyks, M. Differential signaling via the same axon of neocortical pyramidal neurons. *Proceedings of the National Academy of Sciences* **95,** 5323–5328 (1998).

76. Hennig, M. H. Theoretical models of synaptic short term plasticity. *Frontiers in computational neuroscience* **7,** 45 (2013).

77. Markram, H., Lübke, J., Frotscher, M. & Sakmann, B. Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science* **275,** 213–215 (1997).

78. Bi, G.-q. & Poo, M.-m. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of neuroscience* **18,** 10464–10472 (1998).

79. Sjöström, P. J., Turrigiano, G. G. & Nelson, S. B. Rate, timing, and cooperativity jointly determine cortical synaptic plasticity. *Neuron* **32,** 1149–1164 (2001).

80. Fioravante, D. & Regehr, W. G. Short-term forms of presynaptic plasticity. *Current opinion in neurobiology* **21,** 269–274 (2011).

81. Zenke, F., Gerstner, W. & Ganguli, S. The temporal paradox of Hebbian learning and homeostatic plasticity. *Current opinion in neurobiology* **43,** 166–176 (2017).

82. Gallinaro, J. V. & Rotter, S. Associative properties of structural plasticity based on firing rate homeostasis in recurrent neuronal networks. *Scientific reports* **8,** 1–13 (2018).

83. Trachtenberg, J. T. *et al.* Long-term in vivo imaging of experience-dependent synaptic plasticity in adult cortex. *Nature* **420,** 788–794 (2002).

84. Stepanyants, A., Hof, P. R. & Chklovskii, D. B. Geometry and structural plasticity of synaptic connectivity. *Neuron* **34,** 275–288 (2002).

85. Hebb, D. O. *The organization of behavior: a neuropsychological theory* (Science editions, 1949).

86. Lillicrap, T. P., Santoro, A., Marris, L., Akerman, C. J. & Hinton, G. Backpropagation and the brain. *Nature Reviews Neuroscience* **21,** 335–346 (2020).

87. Kuśmierz, Ł., Isomura, T. & Toyoizumi, T. Learning with three factors: modulating Hebbian plasticity with errors. *Current opinion in neurobiology* **46,** 170–177 (2017).

88. Lukoševičius, M. & Jaeger, H. Reservoir computing approaches to recurrent neural network training. *Computer Science Review* **3,** 127–149 (2009).

89. Sussillo, D. & Abbott, L. F. Generating coherent patterns of activity from chaotic neural networks. *Neuron* **63,** 544–557 (2009).

90. Nicola, W. & Clopath, C. Supervised learning in spiking neural networks with FORCE training. *Nature communications* **8,** 1–15 (2017).

91. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-propagating errors. *nature* **323,** 533–536 (1986).

92. Williams, R. J. & Zipser, D. A learning algorithm for continually running fully recurrent neural networks. *Neural computation* **1,** 270–280 (1989).

93. Murray, J. M. Local online learning in recurrent networks with random feedback. *Elife* **8,** e43299 (2019).

94. Marschall, O., Cho, K. & Savin, C. A unified framework of online learning algorithms for training recurrent neural networks. *Journal of machine learning research* (2020).

95. Jaeger, H. & Haas, H. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *science* **304,** 78–80 (2004).

96. Feulner, B. & Clopath, C. Neural manifold under plasticity in a goal driven learning behaviour. *PLoS computational biology* **17,** e1008621 (2021).

97. Feulner, B. *et al.* Small, correlated changes in synaptic connectivity may facilitate rapid motor learning. *Nature communications* **13,** 1–14 (2022).

98. Feulner, B., Perich, M. G., Miller, L. E., Clopath, C. & Gallego, J. A. Feedback-based motor control can guide plasticity and drive rapid learning. *bioRxiv* (2022).

99. Mante, V., Sussillo, D., Shenoy, K. V. & Newsome, W. T. Context-dependent computation by recurrent dynamics in prefrontal cortex. *nature* **503,** 78–84 (2013).

100. Williamson, R. C. *et al.* Scaling Properties of Dimensionality Reduction for Neural Populations and Network Models. *PLoS Computational Biology* **12,** 1–27 (2016).

101. Pang, R., Lansdell, B. J. & Fairhall, A. L. Dimensionality reduction in neuroscience. *Current Biology* **26,** R656–R660 (2016).

102. Mazzucato, L., Fontanini, A. & La Camera, G. Stimuli Reduce the Dimensionality of Cortical Activity. *Frontiers in Systems Neuroscience* **10** (2016).

103. Michaels, J. A., Dann, B. & Scherberger, H. Neural population dynamics during reaching are better explained by a dynamical system than representational tuning. *PLoS computational biology* **12** (2016).

104. Elsayed, G. F. & Cunningham, J. P. Structure in neural population recordings: An expected byproduct of simpler phenomena? *Nature Neuroscience* **20,** 1310–1318 (2017).

105. Williams, A. H. *et al.* Unsupervised discovery of demixed, low-dimensional neural dynamics across multiple timescales through tensor component analysis. *Neuron* **98,** 1099–1115 (2018).

106. Russo, A. A. *et al.* Motor cortex embeds muscle-like commands in an untangled population response. *Neuron* **97,** 953–966 (2018).

107. Perich, M. G., Gallego, J. A. & Miller, L. E. A Neural Population Mechanism for Rapid Learning. *Neuron* **100,** 964–976.e7 (2018).

108. Semedo, J. D., Zandvakili, A., Machens, C. K., Yu, B. M. & Kohn, A. Cortical Areas Interact through a Communication Subspace. *Neuron* **102,** 249–259.e4 (2019).

109. Gao, P. *et al.* A theory of multineuronal dimensionality, dynamics and measurement. *bioRxiv,* 214262 (2017).

110. Sussillo, D. & Abbott, L. Generating Coherent Patterns of Activity from Chaotic Neural Networks. *Neuron* **63,** 544–557 (2009).

111. Hu, Y., Trousdale, J., Josić, K. & Shea-Brown, E. Local paths to global coherence: Cutting networks down to size. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics* **89,** 1–16 (2014).

112. Rivkind, A. & Barak, O. Local dynamics in trained recurrent neural networks. *Physical review letters* **118,** 258101 (2017).

113. DePasquale, B., Cueva, C. J., Rajan, K., Escola, G. S. & Abbott, L. full-FORCE: A target-based method for training recurrent networks. *PloS one* **13,** e0191527 (2018).

114. Recanatesi, S., Ocker, G. K., Buice, M. A. & Shea-Brown, E. Dimensionality in recurrent spiking networks: Global trends in activity and local origins in connectivity. *PLOS computational biology* **15,** e1006446 (2019).

115. Rigotti, M. *et al.* The importance of mixed selectivity in complex cognitive tasks. *Nature* **497,** 585–590 (2013).

116. Rule, M. E., O'Leary, T. & Harvey, C. D. Causes and consequences of representational drift. *Current opinion in neurobiology* **58,** 141–147 (2019).

117. Elsayed, G. F., Lara, A. H., Kaufman, M. T., Churchland, M. M. & Cunningham, J. P. Reorganization between preparatory and movement population responses in motor cortex. *Nature communications* **7,** 1–15 (2016).

118. Gallego, J. A., Perich, M. G., Chowdhury, R. H., Solla, S. A. & Miller, L. E. Long-term stability of cortical population dynamics underlying consistent behavior. *Nature neuroscience,* 1–11 (2020).

119. Golub, M. D., Chase, S. M., Batista, A. P. & Yu, B. M. Brain-computer interfaces for dissecting cognitive processes underlying sensorimotor control. *Current Opinion in Neurobiology* **37,** 53–58 (2016).

120. Oby, E. R. *et al.* New neural activity patterns emerge with long-term learning. *PNAS* (2019).

121. Wärnberg, E. & Kumar, A. Perturbing low dimensional activity manifolds in spiking neuronal networks. *PLOS computational biology* **15,** 1–23 (May 2019).

122. Menendez, J. A. & Latham, P. *Learning low-dimensional inputs for brain-machine interface control* Poster presented at Cosyne 2019. 2019.

123. Xu, T. *et al.* Rapid formation and selective stabilization of synapses for enduring motor memories. *Nature* **462,** 915–919 (2009).

124. Biane, J. S., Takashima, Y., Scanziani, M., Conner, J. M. & Tuszynski, M. H. Reorganization of Recurrent Layer 5 Corticospinal Networks Following Adult Motor Training. *The Journal of Neuroscience* **39,** 4684–4693 (2019).

125. Tavor, I., Botvinik-Nezer, R., Bernstein-Eliav, M., Tsarfaty, G. & Assaf, Y. Short-term plasticity following motor sequence learning revealed by diffusion magnetic resonance imaging. *Human Brain Mapping,* hbm.24814 (2019).

126. Ohbayashi, M. Inhibition of protein synthesis in M1 of monkeys disrupts performance of sequential movements guided by memory. *eLife* **9** (2020).

127. Harvey, M. A., Saal, H. P., Dammann III, J. F. & Bensmaia, S. J. Multiplexing stimulus information through rate and temporal codes in primate somatosensory cortex. *PLoS Biol* **11,** e1001558 (2013).

128. Payeur, A., Guerguiev, J., Zenke, F., Richards, B. A. & Naud, R. Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits. *Nature neuroscience* **24,** 1010–1019 (2021).

129. Michaels, J. A., Schaffelhofer, S., Agudelo-Toro, A. & Scherberger, H. A neural network model of flexible grasp movement generation. *bioRxiv,* 742189 (2019).

130. Bellec, G. *et al.* Biologically inspired alternatives to backpropagation through time for learning in recurrent neural nets. *arXiv* (2019).

131. Kawato, M. Internal models for motor control and trajectory planning. *Current Opinion in Neurobiology* **9,** 718–727 (1999).

132. Golub, M. D. *et al.* Learning by neural reassociation. *Nature neuroscience* **21,** 607–616 (2018).

133. Wolpert, D. M. & Ghahramani, Z. Computational principles of movement neuroscience. *Nature neuroscience* **3,** 1212–1217 (2000).

134. Todorov, E. & Jordan, M. I. Optimal feedback control as a theory of motor coordination (2002).

135. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv* (2014).

136. Roy, O. & Vetterli, M. *The effective rank: A measure of effective dimensionality* in *2007 15th European Signal Processing Conference* (2007), 606–610.

137. Tseng, Y.-w., Diedrichsen, J., Krakauer, J. W., Shadmehr, R. & Bastian, A. J. Sensory prediction errors drive cerebellum-dependent adaptation of reaching. *Journal of neurophysiology* **98,** 54–62 (2007).

138. Rabe, K. *et al.* Adaptation to visuomotor rotation and force field perturbation is correlated to different brain areas in patients with cerebellar degeneration. *Journal of neurophysiology* **101,** 1961–1971 (2009).

139. Schlerf, J. E., Galea, J. M., Bastian, A. J. & Celnik, P. A. Dynamic modulation of cerebellar excitability for abrupt, but not gradual, visuomotor adaptation. *Journal of Neuroscience* **32,** 11610–11617 (2012).

140. Tzvi, E., Koeth, F., Karabanov, A. N., Siebner, H. R. & Krämer, U. M. Cerebellar–Premotor cortex interactions underlying visuomotor adaptation. *NeuroImage* **220,** 117142 (2020).

141. Rioult-Pedotti, M.-S., Friedman, D., Hess, G. & Donoghue, J. P. Strengthening of horizontal cortical connections following skill learning. *Nature neuroscience* **1,** 230–234 (1998).

142. Kleim, J. A. *et al.* Cortical synaptogenesis and motor map reorganization occur during late, but not early, phase of motor skill learning. *Journal of Neuroscience* **24,** 628–633 (2004).

143. Roth, R. H. *et al.* Cortical synaptic AMPA receptor plasticity during motor learning. *Neuron* **105,** 895–908 (2020).

144. Thoroughman, K. A. & Shadmehr, R. Learning of action through adaptive combination of motor primitives. *Nature* **407,** 742–747 (2000).

145. Sohn, H., Meirhaeghe, N., Rajalingham, R. & Jazayeri, M. A network perspective on sensorimotor learning. *Trends in Neurosciences* (2020).

146. Wise, S., Moody, S., Blomstrom, K. & Mitz, A. Changes in motor cortical activity during visuomotor adaptation. *Experimental Brain Research* **121,** 285–299 (1998).

147. Paz, R., Boraud, T., Natan, C., Bergman, H. & Vaadia, E. Preparatory activity in motor cortex reflects learning of local visuomotor skills. *Nature neuroscience* **6,** 882–890 (2003).

148. Gerhard, F. *et al.* Successful reconstruction of a physiological circuit with known connectivity from spiking activity alone. *PLoS computational biology* **9,** e1003138 (2013).

149. Rebesco, J. M., Stevenson, I. H., Koerding, K., Solla, S. A. & Miller, L. E. Rewiring neural interactions by micro-stimulation. *Frontiers in systems neuroscience* **4,** 39 (2010).

150. Diedrichsen, J., Hashambhoy, Y., Rane, T. & Shadmehr, R. Neural correlates of reach errors. *Journal of Neuroscience* **25,** 9919–9931 (2005).

151. Tanaka, H., Sejnowski, T. J. & Krakauer, J. W. Adaptation to visuomotor rotation through interaction between posterior parietal and motor cortical areas. *Journal of neurophysiology* **102,** 2921–2932 (2009).

152. Oby, E. R. *et al.* New neural activity patterns emerge with long-term learning. *Proceedings of the National Academy of Sciences* **116,** 15210–15215 (2019).

153. Song, H. F., Yang, G. R. & Wang, X.-J. Reward-based training of recurrent neural networks for cognitive and value-based tasks. *Elife* **6,** e21492 (2017).

154. Perich, M. G. *et al.* Inferring brain-wide interactions using data-constrained recurrent neural network models. *bioRxiv,* 2020–12 (2021).

155. Das, A. & Fiete, I. R. Systematic errors in connectivity inferred from activity in strongly recurrent networks. *Nature Neuroscience* **23,** 1286–1296 (2020).

156. Schönemann, P. H. A generalized solution of the orthogonal procrustes problem. *Psychometrika* **31,** 1–10 (1966).

157. Froemke, R. C. & Dan, Y. Spike-timing-dependent synaptic modification induced by natural spike trains. *Nature* **416,** 433–438 (2002).

158. Mastrogiuseppe, F. & Ostojic, S. Linking connectivity, dynamics, and computations in low-rank recurrent neural networks. *Neuron* **99,** 609–623 (2018).

159. Schuessler, F., Dubreuil, A., Mastrogiuseppe, F., Ostojic, S. & Barak, O. Dynamics of random recurrent networks with correlated low-rank structure. *Physical Review Research* **2,** 013111 (2020).

160. Schuessler, F., Mastrogiuseppe, F., Dubreuil, A., Ostojic, S. & Barak, O. The interplay between randomness and structure during learning in RNNs. *arXiv preprint arXiv:2006.11036* (2020).

161. Calvin, W. H. & Stevens, C. F. Synaptic noise and other sources of randomness in motoneuron interspike intervals. *Journal of neurophysiology* **31,** 574–587 (1968).

162. Susman, L., Brenner, N. & Barak, O. Stable memory with unstable synapses. *Nature communications* **10,** 1–9 (2019).

163. Fauth, M. J. & van Rossum, M. C. Self-organized reactivation maintains and reinforces memories despite synaptic turnover. *ELife* **8,** e43717 (2019).

164. Rule, M. E. *et al.* Stable task information from an unstable neural population. *Elife* **9,** e51121 (2020).

165. Legenstein, R., Chase, S. M., Schwartz, A. B. & Maass, W. A reward-modulated hebbian learning rule can explain experimentally observed network reorganization in a brain control task. *Journal of Neuroscience* **30,** 8400–8410 (2010).

166. Criscimagna-Hemminger, S. E., Bastian, A. J. & Shadmehr, R. Size of error affects cerebellar contributions to motor learning. *Journal of neurophysiology* **103,** 2275–2284 (2010).

167. Kao, J. C. Considerations in using recurrent neural networks to probe neural dynamics. *Journal of neurophysiology* **122,** 2504–2521 (2019).

168. Logiaco, L., Abbott, L. & Escola, S. Thalamic control of cortical dynamics in a model of flexible motor sequencing. *Cell Reports* **35,** 109090 (2021).

169. Zador, A. M. A critique of pure learning and what artificial neural networks can learn from animal brains. *Nature communications* **10,** 1–7 (2019).

170. Goudar, V., Peysakhovich, B., Freedman, D. J., Buffalo, E. A. & Wang, X.-J. Elucidating the neural mechanisms of Learning-to-Learn. *bioRxiv* (2021).

171. Li, H. L. & Van Rossum, M. C. Energy efficient synaptic plasticity. *Elife* **9,** e50804 (2020).

172. Whittington, J. C. & Bogacz, R. Theories of error back-propagation in the brain. *Trends in cognitive sciences* **23,** 235–250 (2019).

173. Bellec, G. *et al.* A solution to the learning dilemma for recurrent networks of spiking neurons. *Nature communications* **11,** 1–15 (2020).

174. Marr, D. & Thach, W. T. in *From the Retina to the Neocortex* 11–50 (Springer, 1991).

175. Medina, J. F., Nores, W. L. & Mauk, M. D. Inhibition of climbing fibres is a signal for the extinction of conditioned eyelid responses. *Nature* **416,** 330–333 (2002).

176. Bailey, C. H., Giustetto, M., Zhu, H., Chen, M. & Kandel, E. R. A novel function for serotonin-mediated short-term facilitation in Aplysia: Conversion of a transient, cell-wide homosynaptic Hebbian plasticity into a persistent, protein synthesis-independent synapse-specific enhancement. *Proceedings of the National Academy of Sciences* **97,** 11581–11586 (2000).

177. Nitsche, M. A. *et al.* Dopaminergic modulation of long-lasting direct current-induced cortical excitability changes in the human motor cortex. *European Journal of Neuroscience* **23,** 1651–1657 (2006).

178. Legenstein, R., Pecevski, D. & Maass, W. A learning theory for reward-modulated spike-timing-dependent plasticity with application to biofeedback. *PLoS computational biology* **4,** e1000180 (2008).

179. Miconi, T. Biologically plausible learning in recurrent neural networks reproduces neural dynamics observed during cognitive tasks. *Elife* **6,** e20899 (2017).

180. Liu, Y. H., Smith, S., Mihalas, S., Shea-Brown, E. & Sümbül, U. Cell-type–specific neuromodulation guides synaptic credit assignment in a spiking neural network. *Proceedings of the National Academy of Sciences* **118,** e2111821118 (2021).

181. Kossio, F. Y. K., Goedeke, S., Klos, C. & Memmesheimer, R.-M. Drifting assemblies for persistent memory. *bioRxiv* (2020).

182. Cowley, B. R. *et al.* Slow drift of neural activity as a signature of impulsivity in macaque visual and prefrontal cortex. *Neuron* **108,** 551–567 (2020).

183. Hennig, J. A. *et al.* Learning is shaped by abrupt changes in neural engagement. *Nature Neuroscience* **24,** 727–736 (2021).

184. Omrani, M., Murnaghan, C. D., Pruszynski, J. A. & Scott, S. H. Distributed task-specific processing of somatosensory feedback for voluntary motor control. *Elife* **5,** e13141 (2016).

185. Perich, M. G. *et al.* Motor cortical dynamics are shaped by multiple distinct subspaces during naturalistic behavior. *bioRxiv* (2020).

186. Cross, K. P., Cook, D. J. & Scott, S. H. Convergence of proprioceptive and visual feedback on neurons in primary motor cortex. *bioRxiv* (2021).

187. Smith, M. A., Ghazizadeh, A. & Shadmehr, R. Interacting adaptive processes with different timescales underlie short-term motor learning. *PLoS biology* **4,** e179 (2006).

188. Huberdeau, D. M., Krakauer, J. W. & Haith, A. M. Dual-process decomposition in human sensorimotor adaptation. *Current opinion in neurobiology* **33,** 71–77 (2015).

189. Christou, A. I., Miall, R. C., McNab, F. & Galea, J. M. Individual differences in explicit and implicit visuomotor learning and working memory capacity. *Scientific reports* **6,** 1–13 (2016).

190. Paszke, A. *et al.* Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* **32** (2019).

191. Zenke, F. & Ganguli, S. Superspike: Supervised learning in multilayer spiking neural networks. *Neural computation* **30,** 1514–1541 (2018).

192. Neftci, E. O., Mostafa, H. & Zenke, F. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine* **36,** 51–63 (2019).

193. Zenke, F. *Tutorial 1: Training a spiking neural network with surrogate gradients* `https://github.com/fzenke/spytorch/blob/v0.3/notebooks/SpyTorchTutorial1.ipynb/`. 2019.

194. Zenke, F. SpyTorch. Version v0.3. *Zenodo* (Mar. 2019).

195. Hatsopoulos, N. G. & Suminski, A. J. Sensing with the motor cortex. *Neuron* **72,** 477–487 (2011).

196. Scott, S. H. The computational and neural basis of voluntary motor control and planning. *Trends in cognitive sciences* **16,** 541–549 (2012).

197. Nashed, J. Y., Crevecoeur, F. & Scott, S. H. Influence of the behavioral goal and environmental obstacles on rapid feedback responses. *Journal of neurophysiology* **108,** 999–1009 (2012).

198. Pruszynski, J. A., Omrani, M. & Scott, S. H. Goal-dependent modulation of fast feedback responses in primary motor cortex. *Journal of Neuroscience* **34,** 4608–4617 (2014).

199. Scott, S. H. A functional taxonomy of bottom-up sensory feedback processing for motor actions. *Trends in neurosciences* **39,** 512–526 (2016).

200. Inoue, M., Uchimura, M. & Kitazawa, S. Error signals in motor cortices drive adaptation in reaching. *Neuron* **90,** 1114–1126 (2016).

201. Wolpert, D. M., Diedrichsen, J. & Flanagan, J. R. Principles of sensorimotor learning. *Nature reviews neuroscience* **12,** 739–751 (2011).

202. Izawa, J. & Shadmehr, R. Learning from sensory and reward prediction errors during motor adaptation. *PLoS computational biology* **7,** e1002012 (2011).

203. Kawato, M. in *Advanced neural computers* 365–372 (Elsevier, 1990).

204. Kawato, M. & Gomi, H. A computational model of four regions of the cerebellum based on feedback-error learning. *Biological cybernetics* **68,** 95–103 (1992).

205. Hadipour-Niktarash, A., Lee, C. K., Desmond, J. E. & Shadmehr, R. Impairment of retention but not acquisition of a visuomotor skill through time-dependent disruption of primary motor cortex. *Journal of Neuroscience* **27,** 13413–13419 (2007).

206. Saijo, N. & Gomi, H. Multiple motor learning strategies in visuomotor rotation. *PLoS One* **5,** e9399 (2010).

207. Galea, J. M., Vazquez, A., Pasricha, N., Orban de Xivry, J.-J. & Celnik, P. Dissociating the roles of the cerebellum and motor cortex during adaptive learning: the motor cortex retains what the cerebellum learns. *Cerebral cortex* **21,** 1761–1770 (2011).

208. Orban de Xivry, J.-J., Criscimagna-Hemminger, S. E. & Shadmehr, R. Contributions of the motor cortex to adaptive control of reaching depend on the perturbation schedule. *Cerebral Cortex* **21,** 1475–1484 (2011).

209. Krakauer, J. W. & Mazzoni, P. Human sensorimotor learning: adaptation, skill, and beyond. *Current opinion in neurobiology* **21,** 636–644 (2011).

210. Wang, L., Conner, J. M., Rickert, J. & Tuszynski, M. H. Structural plasticity within highly specific neuronal populations identifies a unique parcellation of motor learning in the adult brain. *Proceedings of the National Academy of Sciences* **108,** 2545–2550 (2011).

211. Kawai, R. *et al.* Motor cortex is required for learning but not for executing a motor skill. *Neuron* **86,** 800–812 (2015).

212. Mathis, M. W., Mathis, A. & Uchida, N. Somatosensory cortex plays an essential role in forelimb motor adaptation in mice. *Neuron* **93,** 1493–1503 (2017).

213. Inoue, M. & Kitazawa, S. Motor error in parietal area 5 and target error in area 7 drive distinctive adaptation in reaching. *Current Biology* **28,** 2250–2262 (2018).

214. Herzfeld, D. J., Kojima, Y., Soetedjo, R. & Shadmehr, R. Encoding of error and learning to correct that error by the Purkinje cells of the cerebellum. *Nature neuroscience* **21,** 736–743 (2018).

215. Vyas, S. *et al.* Neural population dynamics underlying motor learning transfer. *Neuron* **97,** 1177–1186 (2018).

216. Vyas, S., O'Shea, D. J., Ryu, S. I. & Shenoy, K. V. Causal role of motor preparation during error-driven learning. *Neuron* **106,** 329–339 (2020).

217. Sohn, H., Meirhaeghe, N., Rajalingham, R. & Jazayeri, M. A network perspective on sensorimotor learning. *Trends in Neurosciences* **44,** 170–181 (2021).

218. Krakauer, J. W., Ghilardi, M.-F. & Ghez, C. Independent learning of internal models for kinematic and dynamic control of reaching. *Nature neuroscience* **2,** 1026–1031 (1999).

219. Shadmehr, R. & Mussa-Ivaldi, F. A. Adaptive representation of dynamics during learning of a motor task. *Journal of neuroscience* **14,** 3208–3224 (1994).

220. Wolpert, D. M., Ghahramani, Z. & Jordan, M. I. Are arm trajectories planned in kinematic or dynamic coordinates? An adaptation study. *Experimental brain research* **103,** 460–470 (1995).

221. Shadmehr, R. & Moussavi, Z. M. Spatial generalization from learning dynamics of reaching movements. *Journal of Neuroscience* **20,** 7807–7815 (2000).

222. Shadmehr, R., Smith, M. A. & Krakauer, J. W. Error correction, sensory prediction, and adaptation in motor control. *Annual review of neuroscience* (2010).

223. Albert, S. T. & Shadmehr, R. The neural feedback response to error as a teaching signal for the motor learning system. *Journal of Neuroscience* **36,** 4832–4845 (2016).

224. Hadjiosif, A. M., Krakauer, J. W. & Haith, A. M. Did we get sensorimotor adaptation wrong? Implicit adaptation as direct policy updating rather than forward-model-based learning. *Journal of Neuroscience* **41,** 2747–2761 (2021).

225. Fernandes, H. L., Stevenson, I. H. & Kording, K. P. Generalization of Stochastic Visuomotor Rotations. *PLOS ONE* **7,** 1–9 (Aug. 2012).

226. Albert, S. T. *et al.* An implicit memory of errors limits human sensorimotor adaptation. *Nature human behaviour* **5,** 920–934 (2021).

227. Kitago, T., Ryan, S. L., Mazzoni, P., Krakauer, J. W. & Haith, A. M. Unlearning versus savings in visuomotor adaptation: comparing effects of washout, passage of time, and removal of errors on motor memory. *Frontiers in human neuroscience* **7,** 307 (2013).

228. Friedrich, J. *et al.* Neural optimal feedback control with local learning rules. *Advances in Neural Information Processing Systems* **34,** 16358–16370 (2021).

229. Cisek, P. Evolution of behavioural control from chordates to primates. *Philosophical Transactions of the Royal Society B* **377,** 20200522 (2022).

230. Cisek, P. & Hayden, B. Y. *Neuroscience needs evolution* 2022.

231. Kao, T.-C. & Hennequin, G. Neuroscience out of control: control-theoretic perspectives on neural circuit dynamics. *Current Opinion in Neurobiology* **58,** 122–129 (2019).

232. Pemberton, J., Boven, E., Apps, R. & Ponte Costa, R. Cortico-cerebellar networks as decoupling neural interfaces. *Advances in Neural Information Processing Systems* **34,** 7745–7759 (2021).

233. Perich, M. G. & Rajan, K. Rethinking brain-wide interactions through multi-region 'network of networks' models. *Current Opinion in Neurobiology* **65,** 146–151 (Dec. 2020).

234. Gallego, J. A., Makin, T. R. & McDougle, S. D. Going beyond primary motor cortex to improve brain–computer interfaces. en. *Trends in Neurosciences* **45,** 176–183 (Mar. 2022).

235. Martin, T., Keating, J., Goodkin, H., Bastian, A. & Thach, W. Throwing while looking through prisms: I. Focal olivocerebellar lesions impair adaptation. *Brain* **119,** 1183–1198 (1996).

236. Ito, M. Mechanisms of motor learning in the cerebellum. *Brain research* **886,** 237–245 (2000).

237. Izawa, J., Criscimagna-Hemminger, S. E. & Shadmehr, R. Cerebellar contributions to reach adaptation and learning sensory consequences of action. *Journal of Neuroscience* **32,** 4230–4239 (2012).

238. Herzfeld, D. J. *et al.* Contributions of the cerebellum and the motor cortex to acquisition and retention of motor memories. *Neuroimage* **98,** 147–158 (2014).

239. Tzvi, E., Loens, S. & Donchin, O. Mini-review: the role of the cerebellum in visuomotor adaptation. *The Cerebellum,* 1–8 (2021).

240. Miall, R. C. & Wolpert, D. M. Forward models for physiological motor control. *Neural networks* **9,** 1265–1279 (1996).

241. Bastian, A. J. Learning to predict the future: the cerebellum adapts feedforward movement control. *Current opinion in neurobiology* **16,** 645–649 (2006).

242. Nawrot, M. & Rizzo, M. Motion perception deficits from midline cerebellar lesions in human. *Vision research* **35,** 723–731 (1995).

243. Mujika, A., Meier, F. & Steger, A. Approximating real-time recurrent learning with random kronecker factors. *Advances in Neural Information Processing Systems* **31** (2018).

244. Gilra, A. & Gerstner, W. Predicting non-linear dynamics by stable local learning in a recurrent spiking neural network. *Elife* **6,** e28295 (2017).

245. Denève, S., Alemi, A. & Bourdoukan, R. The brain as an efficient and robust adaptive learner. *Neuron* **94,** 969–977 (2017).

246. Naud, R. & Sprekeler, H. Sparse bursts optimize information transmission in a multiplexed neural code. *Proceedings of the National Academy of Sciences* **115,** E6329–E6338 (2018).

247. Podlaski, B. & Machens, C. K. Biological credit assignment through dynamic inversion of feedforward networks. *Advances in Neural Information Processing Systems* **33,** 10065–10076 (2020).

248. Pala, A. & Petersen, C. C. In vivo measurement of cell-type-specific synaptic connectivity and synaptic transmission in layer 2/3 mouse barrel cortex. *Neuron* **85,** 68–75 (2015).

249. Graves, A. R. *et al.* Visualizing synaptic plasticity in vivo by large-scale imaging of endogenous AMPA receptors. *Elife* **10,** e66809 (2021).

250. Glasgow, S. D., McPhedrain, R., Madranges, J. F., Kennedy, T. E. & Ruthazer, E. S. Approaches and limitations in the investigation of synaptic transmission and plasticity. *Frontiers in synaptic neuroscience* **11,** 20 (2019).

251. Pakkenberg, B. *et al.* Aging and the human neocortex. *Experimental gerontology* **38,** 95–99 (2003).

252. Boven, E., Pemberton, J., Chadderton, P., Apps, R. & Costa, R. P. Cerebro-cerebellar networks facilitate learning through feedback decoupling. *bioRxiv* (2022).

253. Pemberton, J., Chadderton, P. & Costa, R. P. Cerebellar-driven cortical dynamics enable task acquisition, switching and consolidation. *bioRxiv* (2022).

254. Lillicrap, T. P. & Scott, S. H. Preference distributions of primary motor cortex neurons reflect control solutions optimized for limb biomechanics. *Neuron* **77,** 168–179 (2013).

255. Codol, O., Michaels, J. A., Kashefi, M., Pruszynski, A. J. & Gribble, P. L. *MotorNet: a Python Toolbox for Controlling Biomechanical Effectors with Artifical Neural Networks* 31st meeting of the Society for the Neural Control of Movement. 2022.

256. O'shea, D. J. *et al.* The need for calcium imaging in nonhuman primates: New motor neuroscience and brain-machine interfaces. *Experimental neurology* **287,** 437–451 (2017).

257. Trautmann, E. M. *et al.* Dendritic calcium signals in rhesus macaque motor cortex drive an optical brain-computer interface. *Nature communications* **12,** 1–20 (2021).

258. Tremblay, S. *et al.* An open resource for non-human primate optogenetics. *Neuron* **108,** 1075–1090 (2020).