Feedback control of recurrent dynamics constrains learning timescales during motor adaptation

Harsha GURNANI[1,2*] Weixuan LIU [1,3,4] Bingni W. BRUNTON [1,2,5†]

{[1]Department of Biology, [2]eScience Institute,
[3]Paul G. Allen School of Computer Science & Engineering,
[4] Department of Applied Mathematics, [5]Computational Neuroscience Center}
University of Washington, Seattle, USA
*harsha84@uw.edu †bbrunton@uw.edu

Abstract

Latent dynamical models of the primary motor cortex (M1) have revealed fundamental neural computations underlying motor control; however, such models often overlook the impact of sensory feedback, which can continually update cortical dynamics and correct for external perturbations. This suggests a critical need to model the interaction between sensory feedback and intrinsic dynamics. Such models would also benefit the design of brain-computer interfaces (BCIs) that decode neural activity in real time, where both user learning and proficient control require feedback. Here we investigate the flexible feedback modulation of cortical dynamics and demonstrate its impact on BCI task performance and short-term learning. By training recurrent network models with real-time sensory feedback on a simple 2D reaching task, analogous to BCI cursor control, we show how previously reported M1 activity patterns can be reinterpreted as arising from feedback-driven dynamics. Next, by incorporating adaptive controllers upstream of M1, we make a testable prediction that short-term learning for a new BCI decoder is facilitated by plasticity of inputs to M1, including remapping of sensory feedback, beyond the plasticity of recurrent connections within M1. This input-driven dynamical structure also determines the speed of adaptation and learning outcomes, and explains a continuous form of learning variability. Thus, our work highlights the need to model input-dependent latent dynamics for motor control and clarifies how constraints on learning arise from both the statistical characteristics and the underlying dynamical structure of neural activity.

1 Introduction

Performing skilled movements requires the integration of sensory information with task goals to generate coordinated muscle activation. The primary motor cortex (M1) receives these inputs from areas such as sensory, premotor, and parietal cortices, and is one of the regions implicated in the generation of motor commands. Low-dimensional patterns of M1 activity has been linked to control of limb kinematics [1–7]. How does M1 and related neural circuitry produce a useful sequence of motor commands to translate spatial or task goals into smooth movements? Theoretical frameworks of motor control propose that these motor commands are not entirely pre-specified before movement initiation. Instead, they are continually optimized for task performance during movement execution, and feedback laws generate appropriate commands from each evolving state [8–10]. While these theories have a rich history of interpreting and predicting behavioral dynamics, their application to understanding neural data has been more limited. Bridging this gap requires new network models incorporating feedback [11–13] as well as predictions about how neuronal interactions shape M1 activity structure for feedback control [14].

Over the last decade, there has been tremendous progress in using dynamical systems techniques to explain the structure of M1 activity during movements. During reaching tasks in primates, M1 activity has been viewed as a (semi)-autonomous dynamical system. In other words, the prevailing view has been that simple rules dictate the evolution of M1 activity based on its current state, and the resultant trajectories generate appropriate motor commands via downstream partners such as the spinal cord [7, 15–17]). Moreover, recurrent neural network models (RNNs) trained to perform similar tasks

and receiving relatively simple inputs also show activity structure similar to M1 recordings [18, 19]. These results have strengthened the notion that M1 can act as a pattern generator to produce complex spatiotemporal dynamics independently from other regions. Despite the success of these models, they fail to fully capture the observations that M1 dynamics are altered by sensory feedback, and are dependent on input from other regions such as premotor cortex and motor thalamus [11, 20–27]. Motivated by the crucial role of feedback in shaping neural dynamics, including in M1, here we built computational models of recurrent cortical dynamics in the context of feedback control and examined how motor cortical activity is driven by a combination of time-varying feedback inputs and internal states.

Insights into neural computations for motor control have enabled the use of cortical activity to successfully drive rehabilitative brain-computer interfaces (BCIs) [28, 29]. BCIs use an experimenter-defined mapping to translate recorded neural activity into motor outcomes, typically moving effectors such as a robotic limb or a cursor; BCIs are also dependent on the use and design of sensory feedback [30–33]. Thus, beyond their clinical utility, BCIs are valuable research tools for investigating the mechanisms - activity structure and plasticity—underlying learning and control of new effectors [34, 35]. One pivotal study examined how primates trained on well-calibrated BCI decoders could learn to use new decoders over the course of a few hundred trials [36]. Animals were first trained to control a computer cursor by modulating activity in their primary motor cortex, during which activity was observed to lie in a low-dimensional space (termed the "intrinsic manifold"). Subsequently, when the BCI mapping was changed such that the required control patterns were either within or outside the intrinsic manifold, the animals adapted to the new within-manifold decoders much more easily. Further analysis revealed that the statistical structure of neural activity was largely conserved across short-term adaptation to new BCI decoders, which is to say, the new activity patterns remained within the intrinsic manifold [37]. Instead, learning was accompanied by "neural reassociation", where the network produced useful activity patterns from within its repertoire "at the right time", both across targets and within single-trial trajectories.

What synaptic or neural mechanisms underlie this neural reassociation during adaption to BCI decoder perturbations? Other studies on adaptation to force-field perturbations and visumomotor rotations, where learning occurs on similarly short timescales, have provided evidence for changes upstream of M1 [38–42], such as changes in preparatory activity [43, 44] and dependence on sensory processing [38, 45–47]. The presence and structure of sensory feedback is also critical for successful BCI learning [31–33, 48–50]. This motivated us to explore the interaction between input plasticity and preserved recurrent connectivity in generating adapted trajectories during BCI learning. Critically, this allowed us to study the role of feedback interactions not only during the execution of learnt movements but when the motor cortex needs to produce new activity patterns, such as during sensorimotor adaptation and learning to use new BCI mappings.

In this work, we investigated the role of flexible feedback modulation of cortical dynamics in the context of both BCI task performance and short-term learning. We developed a computational framework that incorporated sensory feedback inputs to recurrent neural networks, and trained these networks on a *control* task (unlike a fixed pattern generation or an input association task), where the inclusion of feedback was essential for the network to adjust its output in response to perturbations. We first examined the task-relevant dynamics that emerged over training and showed that many experimentally observed features of M1 activity could be recapitulated in feedback-driven networks. Secondly, we observed a geometric misalignment of the intrinsic manifold, the task-dynamics subspace and the decoder (output) weights, which has implications for the initialization of BCI decoders. Next, we suggest that short-term adaptation, including to new BCI decoders, can be facilitated by plasticity of inputs from upstream controllers such as a remapping of sensory feedback, instead of plasticity of recurrent connections within M1. Crucially, we show that beyond the alignment of decoders with the intrinsic manifold, the pre-existing structured dynamics affect the speed of adaptation to different decoders. This offers an explanation for the variability of learning outcomes across different within-manifold decoders, that has been missing from related computational studies [51–55]. Lastly, we show adaptation using a biologically-plausible learning rule that modifies input weights is consistent with experimentally-observed variability of BCI learning outcomes. Together, we present a framework to study feedback-driven dynamics in BCIs, to analyze the activity structure, and to dissect the learning mechanisms in the context of feedback control, with broad implications for studying the role of feedback—sensory or internal—in flexibly modulating diverse cortical dynamics.

2 Structured dynamics in feedback-driven networks

2.1 Feedback enables robustness against perturbations

We developed a computational model to study feedback modulation of motor cortical dynamics during a BCI cursor manipulation task, in order to relate the resultant activity structure to experimentally-characterized properties of primary motor cortex (M1). In particular, we trained rate-based recurrent neural network models (RNNs) on a 2D center-out delayed reaching task, similar to a common BCI paradigm (Figure 1A). The network was cued on each trial to move a cursor to one of 8 targets. A set of readout weights ("decoder") specified the mapping from the activity of this RNN to the cursor's velocity. Unlike previous studies that used position decoders [51–55], here we chose a velocity decoder to be more similar to experimental studies [31, 32, 36, 56–58]. The network received trial-specific inputs, namely the Target-cue that specified the 2D position of the trial target, a binary Go-cue. Initially, the Go-cue was set to 0 during a preparatory period when the network needed to hold the cursor still, after which the Go-cue turned 1 instructing the network to move the cursor to the target. We refer to these as "feedforward inputs", and they were fixed within the relevant trial epochs. Another set of inputs were dependent on the position of the cursor (sensory feedback), and potentially network state itself; therefore, we refer to these as "feedback inputs." The mapping between sensory state and these inputs could be pre-specified as just the position error signal (vector-to-target), or learnt via another neural network F ("feedback controller").

For initial training, a random decoder W_{out} was fixed while both recurrent weights W_{rec} and input weights W_{in}, W_{fbk} (or output layer of the feedback controller F) were trained via gradient descent. To encourage networks to use feedback input for corrective movements, we trained them in the presence of small unintended cursor movements (see Methods). After training, we tested the networks' robustness to larger, sudden shifts in the cursor position (Figure 1A,C). The trained networks were able to correct for these perturbations and reach the target on all trials. Secondly, while we trained networks without any activity noise, we observed perfect task performance even when tested with small additive noise to the RNN activity (Figure 1B). This suggests that the trained networks were robust against both internal and external perturbations, including those not observed during training. We also observed that slow recurrent processing was important to produce a velocity output that corrects errors over time, rather than simply mapping the feedback input into the output direction (Figure S1, Supplementary Note). In the following sections, we dissect the mechanisms behind these observations and examine the structure of the network dynamics in more detail.

2.2 Intrinsic manifold of task-trained RNNs

Network activity in task-trained RNNs was low-dimensional for all tested architectures ($\sim$8 PCs captured 95% of the variance) (Figure 1D,E), similar to motor cortical activity during both forelimb- and BCI-based reaching tasks. This observation allowed us to define an "intrinsic manifold" as the subspace spanned by the top 8 activity PCs. Next, we separately evaluated activity during the preparatory and movement epochs, defining their respective subspaces. As observed in M1 data from reaching tasks, the requirement to hold the cursor at the centre position during the preparatory period led to orthogonality between the two subspaces (Figure 2A). Specifically, the preparatory period PCs were orthogonal to the readout weights W_{out}, ensuring that target-specific preparatory period activity developed without triggering cursor movement. After the Go-cue, activity collapsed onto the movement subspace and showed rotational structure (Figure 2B), as observed in primate M1 during reaching tasks [15, 59].

2.3 Low-dimensional latent dynamics

To better understand the qualitative dynamics of these high-dimensional networks, we wanted to describe a lower-dimensional latent dynamical system that could capture the network response to inputs and evolution of neural activity over time. We thus fit a latent linear dynamical system that defines a lower-dimensional shared latent subspace to describe the network dynamics. The RNN units' activity is a high-dimensional observation of these latent dynamics, obtained via an affine transformation of the latent factors (Figure 2C, see Methods). The latent dynamical system also received inputs in the form of the target cue and the position error. By fitting these models on reach condition-averaged activity of trained

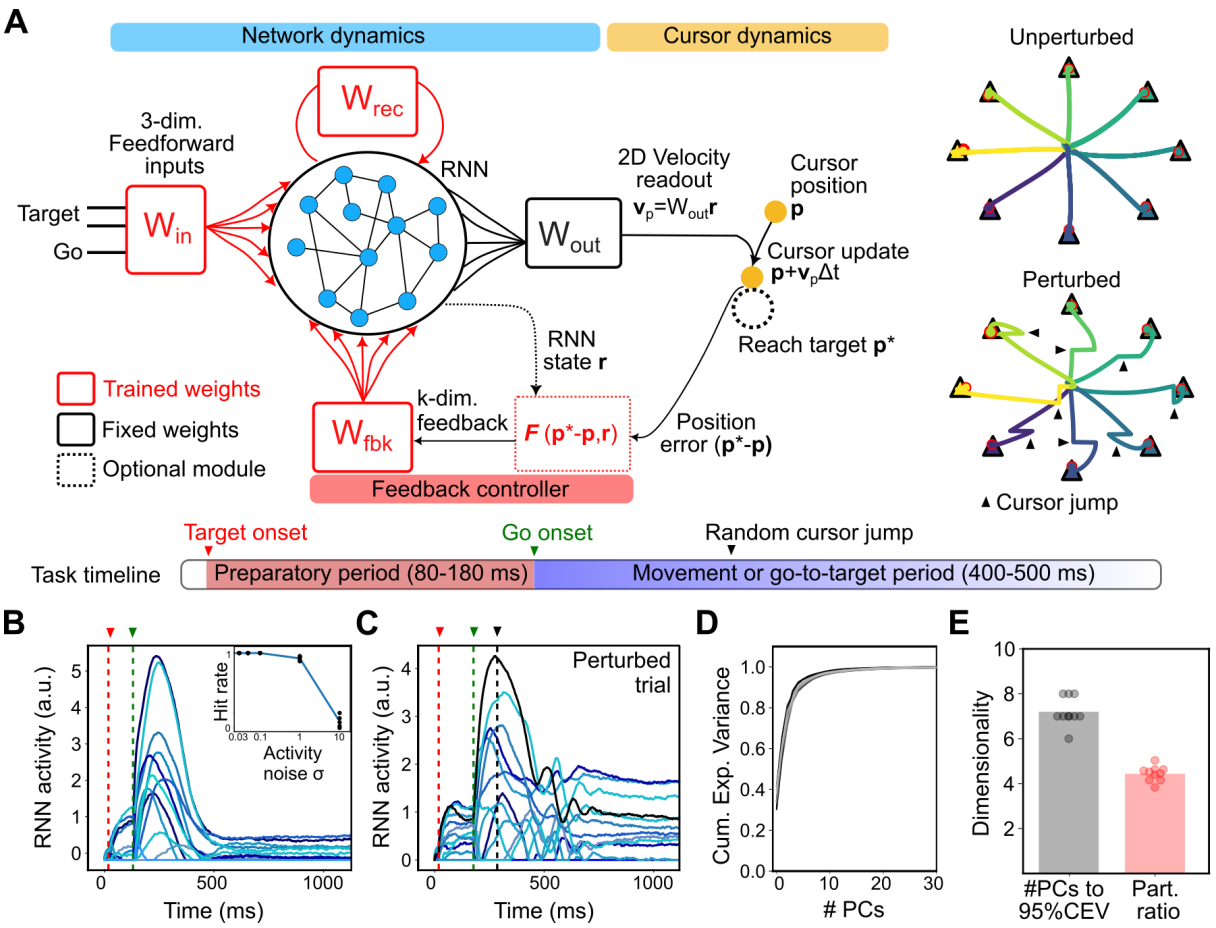

Figure 1: **Training recurrent networks in combination with feedback enables robustness to perturbations. (A)** (Left) Model architecture of recurrent neural network (RNN) controlling a cursor and receiving sensory feedback. The network activity $\mathbf{r}$ depends on recurrent interactions as well as two sets of inputs: feedforward and feedback. A linear readout of the network activity controls the velocity $\mathbf{v}_p$ of an external cursor that is to be moved to a target location $\mathbf{p}^*$. (Right) Example cursor trajectories showing reaches to each of 8 radial targets without an external perturbation (top) or with a cursor shift perturbation (bottom; example perturbations indicated by black triangles). Reaches to different targets are shown in different colors. Target locations are shown as triangles, endpoints of example reach trajectories are indicated by red circles. (Bottom) Task timeline. Target input is turned on at 20 ms, though the network must maintain the cursor at the centre position during the variable preparatory period. Once the Go-cue is turned on, the network should move the cursor to the target and hold it there until the end of the trial. On some trials, the cursor could be shifted unpredicably in the x- or y direction (cursor jump) at either 100, 300 or 500 ms. **(B-C)** Example neural activity (13 example neurons in different colors) in task-trained networks, during an unperturbed trial in (B) with stimulus and Go-cue onset times indicated by dashed lines, and during a trial with cursor perturbation in (C) with the perturbation time indicated by dashed line. The two examples are reaches to different targets, with small additive activity noise ($\sigma = 0.03$). Inset in (B): Average hit rate for different levels of activity noise (n=5 networks). **(D-E)** Low dimensional activity in trained RNNs. **(D)** Cumulative explained variance (CEV) with increasing numbers of principal components, with different curves for individual networks. **(E)** Dimensionality measured as number of PCs to get 95% CEV and as participation ratio using the eigenvalues of the correlation matrix. Each curve/scatter is a separate network of one of 3 architectures: direct feedback (n=5), indirect feedback via a network F with 4 outputs and one hidden layer (n=3), and indirect feedback via network F with 12 outputs (n=2).

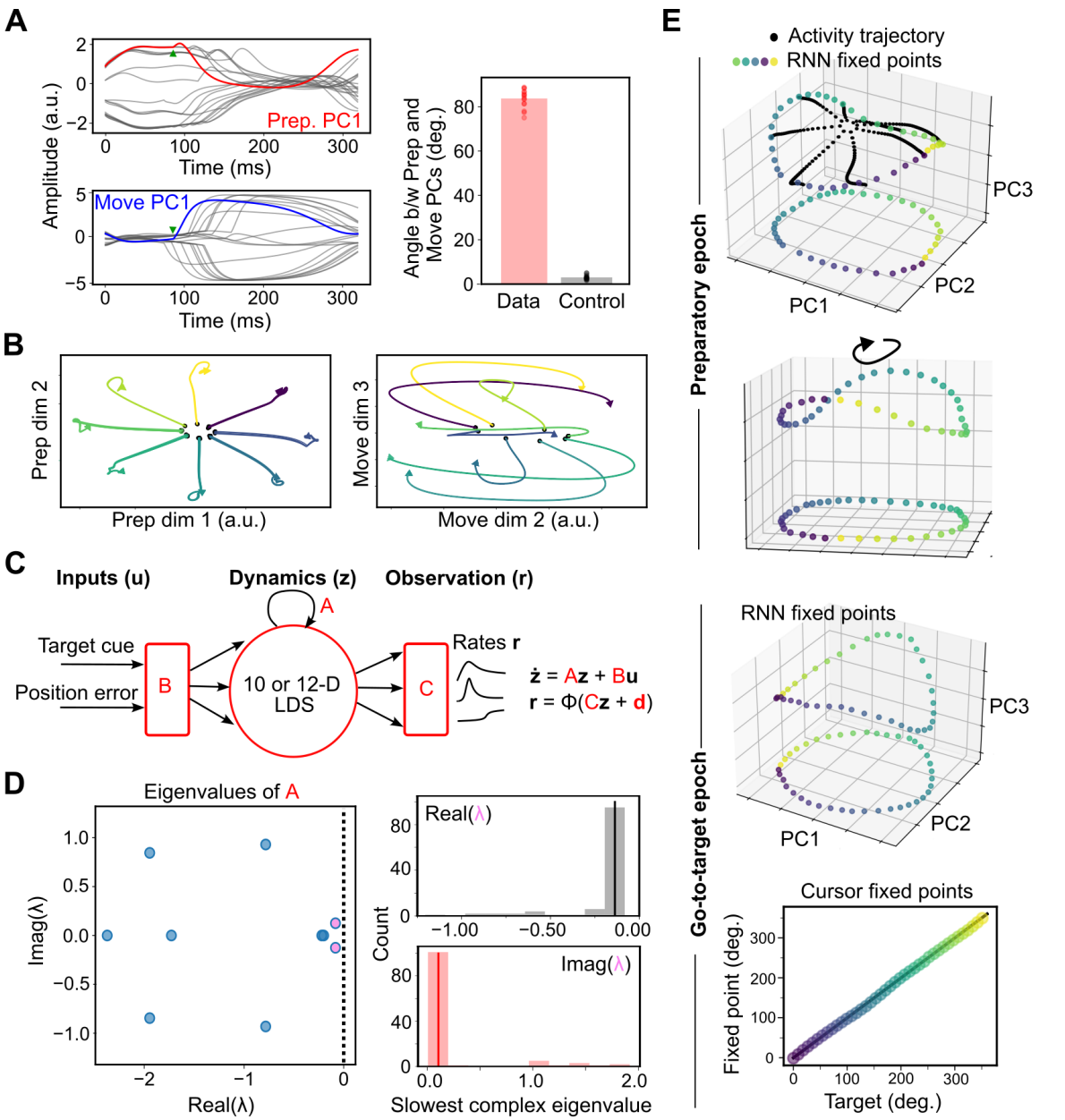

Figure 2: **Emergent structure in trained networks is consistent with experimental data.** **(A)** (Left) Activity along the top preparatory period principal component (PC1) and along the movement period PC1, aligned to stimulus onset. Different trials (lines) received the Go-cue (green triangle) at varying times. An example trial is highlighted as red and blue lines. (Right) Angle between the subspaces spanned by the top 4 preparatory and movement period PCs, for true (red) and time-shuffled control (black) RNN activity. **(B)** Preparatory activity projected onto the preparatory period PCs (left) and movement-related activity projected on the movement period PCs (right), colored by trial target. **(C)** A modified latent linear dynamical system was fit to RNN firing rates as observations. A low-dimensional latent dynamical system was modelled with linear dynamics A, which was fit along with input weights B, observation weights C, and bias terms $\mathbf{d}$. **(D)** (Left) Eigenvalues of the dynamics matrix A from an example fit. The pink scatter denote the slowest complex eigenvalues λ. (Right) Histogram of the real and imaginary parts of the slowest complex eigenvalues across fits. The black and red line indicate the respective median. (N=3 networks; n=40 fits each). **(E)** Fixed points of the joint RNN-cursor system during the preparatory (top) and Go-to-target (bottom) epoch. RNN activity fixed points for different targets (inputs) are plotted along the top 3 PCs along with their projection on the PC1-PC2 plane. Black trajectories show preparatory period activity for 8 trials, which end at one of the fixed points. Similarly, during the movement epoch, target-specific fixed points of RNN activity as well as cursor position are shown. The angular position of the cursor fixed point is plotted against the target angular position. Also see Figure S2.

RNNs, we could explain a large fraction of variance for held-out units (median R^2=0.81), using a model with 10 latent factors.

For each network, we observed that all eigenvalues of the latent recurrent connectivity had negative real parts, indicating stable dynamics. We also robustly identified a rotational plane with slowly decaying dynamics (complex eigenvalues with real value ~ -0.1, imaginary value~ 0.1) (Figure 2D). These $\sim$1Hz rotations are in a similar range as M1 rotational dynamics observed during primate forelimb reaching tasks [15] and during a BCI-based reaching task (Supp Figure S2). By accounting for inputs received by the model RNN, we concluded that the identified rotational modes are not inherited from structured inputs but are a result of recurrent interactions. It is likely that the multiple rotational planes previously identified in neural data using jPCA [15] reflect properties of both the intrinsic and closed-loop (i.e. due to feedback) dynamical system [11]. Overall, these networks displayed features of neural activity and underlying dynamics that are consistent with experimentally-observed M1 activity.

2.4 Mechanism for error-correction

Consistent with the identification of stable, decaying dynamics, we observed that neural activity reached stationary states at the end of the reach, suggesting the presence of stable fixed points. However, unlike for position decoders, fixed points of neural activity with velocity decoders may not carry information about the cursor position. For example, neural activity can arrive at the same fixed point via different trajectories in state space, but the final cursor position given by the integration of velocity readout along those trajectories may be different. Similarly, these fixed points may not maintain the cursor at the target location if they produce a non-zero velocity readout. Thus, we reasoned that the robustness to perturbation arises from the presence of stable fixed points of *cursor dynamics* at the target location.

As the dynamics of the RNN and the cursor are coupled, we identified fixed points of the joint RNN-cursor system (i.e locations where the rate of change of network state and cursor state reached a minimum of zero) and confirmed that they were stable attracting fixed points (see Methods). We identified fixed points during both the preparatory and movement (go-to-target) epochs, and separately examined the coordinates corresponding to the RNN and cursor state. For each target, we found unique stable fixed points, and the fixed points changed continuously as we varied the target input (Figure 2E). As hypothesized, the corresponding fixed points of the cursor position were either near the centre (during preparatory period) or at the target location (during the go-to-target epoch). Correspondingly, the set of RNN fixed points were arranged in a ring-like structure, mirroring the topology of the targets. This structure of preparatory period fixed points is consistent with analyses of data recorded from primary motor (M1) and dorsal premotor (PMd) cortex during forelimb-based reaches [16, 43, 44].

We repeated this fixed point analysis for a BCI dataset [60] by fitting a latent linear dynamical system to M1 activity recorded during a similar cursor control BCI task in primates (Figure S2). Given the stable M1 dynamics (Figure S2E), we could compute target-specific fixed points of the latent dynamics while the cursor is fixed at the target. Interestingly, the neural activity (or rather, the inferred latent states) at the end of different trials clustered around the corresponding fixed point specific to the reach target, suggesting that a similar strategy may be at play in guiding M1 dynamics during this BCI task (Figure S2G,H). This strategy was not the only possible one to solve the task; alternatively, the network could have used amplifying or unstable dynamics to push the neural state in a target-specific direction, which may potentially be the case during ballistic reaches. Building on previous observations that cortical activity is updated by sensory state during BCI use [21, 61], our simplified model enabled us to propose a dynamical mechanism for how feedback guides neural and behavioral states.

While preparatory fixed points can act as initial states for triggering different trajectories in an autonomous system, separating activity into different regions of the state space also allows input-driven systems to use different local dynamics. More generally, when feedback influences network dynamics, jointly analysing the network and plant (e.g. a cursor or arm) dynamics may be more fruitful or appropriate to understand how these interactions enable "computation" to meet task demands. This is especially applicable to more naturalistic conditions where neural activity controls the kinematics (velocity, force, torques) rather than the position of a plant directly, or where physical forces and bio-mechanical coupling influence plant dynamics.

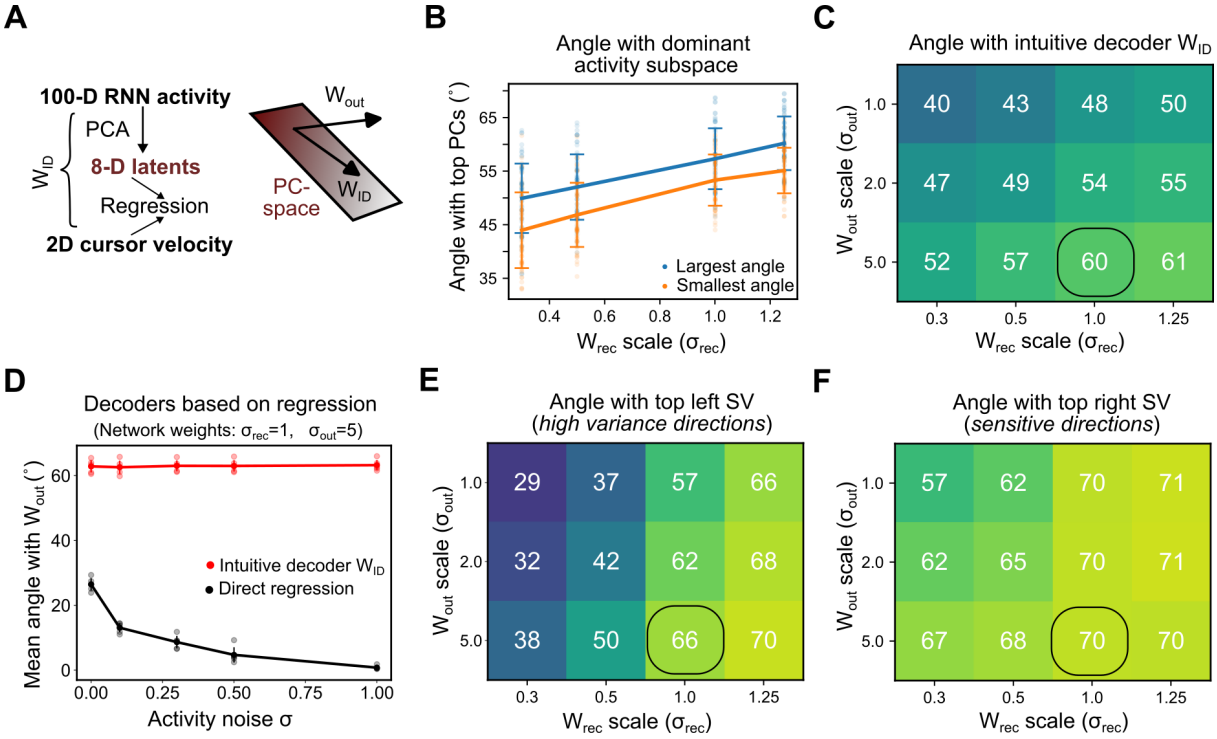

Figure 3: **Misalignment of output weights with network dynamics. (A)** The intuitive decoder W_{ID} was constructed by first projecting RNN activity into a lower dimensional space identified via PCA, and then regressing against 2D cursor velocity. **(B)** Largest and smallest principal angles between the neural activity subspace (n=8 top PCs) and the output weights W_{out} across varying initial scale of recurrent weights (σ_{rec}). **(C)** Smallest principal angle between the readout weights W_{out} and the intuitive decoder constructed from the top PCs (W_{ID}), as a function of initial connectivity scale. Values indicate mean across 3 trained networks. The black rectangle highlights the parameters used in the rest of the paper. **(D)** Angle with true decoder W_{out} of decoders identified by regressing full neural activity (direct regression; black) or by regressing lower-dimensional activity (intuitive decoder; red). Solid line and error bars indicate mean and standard deviation across n=5 trained networks.**(E-F)** Smallest principal angle between the network dynamics subspaces (n=8 singular vectors, SV) and the output weights W_{out}, as a function of initial connectivity scale. Values indicate mean across 3 trained networks.

3 Misalignment of network dynamics with output directions

Given that task-trained RNNs developed low-dimensional activity, we next asked how this emergent activity subspace relates to the output weights. Dimensionality reduction is sometimes used as a denoising procedure before identifying encoding structure of task variables in a lower-dimensional space (Figure 3A). For example, regression of limb kinematics, EMG signals, or the activity of another brain region may be performed against low-dimensional "latent factors" of the source network activity. The resultant weights are used to define the output-potent directions, also known as communication subspace when decoding the activity of another region. Similar methods can be used to initialize BCI decoders using calibration data, where the inferred mapping from network activity to cursor velocity is referred to as the "intuitive decoder" [36]. Here we evaluate the properties of these inferred decoding weights and their relation to the true decoder the network was trained on (W_{out}).

We first compared the angle between the true decoder and the subspace defined by the top 8 PCs (the intrinsic manifold), and observed that these angles were much greater than zero, increasing with both the magnitude of initial recurrent weights and the output weights (Figure 3B). This is consistent with recent observations of a misalignment between dominant dimensions (here, PC subspace) and decoding or behavior-predicting dimensions (here, W_{out}) [62–64]. However, while this misalignment reflects a *geometric* arrangement, activity within the PC subspace is *correlated* to activity along the decoder W_{out}

and these dimensions are coupled via recurrent dynamics. We obtained an intuitive decoder (W_{ID}) by regressing observed cursor velocity against the low-dimensional activity in the PC subspace (R^2>0.9), using data during the closed-loop performance of the network with the original decoder W_{out}. As expected, since this constructed decoder lies in the top PC subspace, it is also misaligned with the true decoder W_{out} (Figure 3C). This result was consistent across multiple tested architectures (Figure S3). Thus, misalignment of the dominant activity subspace and W_{out} leads to a failure to identify the true output direction (Figure 3D). However, the closed-loop performance of the network remained intact with this new intuitive decoder, albeit with slightly smaller reach velocities, without need for further retraining.

We next identified the subspace in which the intrinsic dynamics are embedded i.e. the directions that define the sensitivity of the network to perturbations and the directions in which initial activity is maximally amplified. We do so by examining the top right and left singular vectors of the recurrent connectivity matrix. We again find that the true decoder is misaligned with these dynamics subspaces (Figure 3E,F). As the variance along W_{out} is inversely related to the scale of these output weights at a fixed velocity, the misalignment of W_{out} with the amplification directions (left singular vectors) increases for large output or recurrent weights. On the other hand, the network's sensitive directions (right singular vectors) were consistently at large angles with W_{out}. This geometry led us to wonder if the observed misalignment would make the network more robust to perturbations along the output directions.

To examine the robustness of neural and behavioral trajectories, we used two types of activity perturbations. First, each RNN unit received additive independent white noise while the network performed the task, and we compared how recurrent interactions reshaped noise variance along different directions in the neural state space (see Methods). We observed that task-irrelevant noise was amplified more along the top 2 PCs as compared to the true decoder W_{out} (Figure S4B), as expected from the misalignment between W_{out} and the amplification directions. However, normalizing this noise by the signal variance surprisingly revealed no difference in robustness (Figure S4C). Next, we examined the impact of a single directed perturbation, which revealed a greater sensitivity to perturbations along the top PCs as compared to W_{out} (Figure S5), consistent with the misalignment of W_{out} with the dynamics. Under both perturbation types, behavioral trajectories were made more robust either by the low-pass filtering of independent noise via the velocity decoder or via feedback-based corrections.

In summary, we observed that the intrinsic manifold is distinct from the output directions, and the degree of misalignment depends on the connectivity scale. Even so, the trained network is capable of controlling the evolution of activity along both this largely "outside-manifold" decoder W_{out} as well as the aligned "intuitive decoder" W_{ID}.

4 Motor adaptation via remapping of inputs

Up until now, we only examined the properties of a pre-trained network while keeping the decoder and other network weights fixed. After establishing these simplified models of feedback-driven dynamics as useful approximations of cortical activity, we use them to understand how networks might adapt to use new decoders. In particular, we next study the role of input plasticity in motor adaptation using the paradigm of adaptation to BCI decoder perturbations in [36]. In this task design, after pretraining, a new decoder was chosen from one of two classes of decoder perturbations—either aligned with the intrinsic manifold (within-manifold perturbation; WMP) or misaligned with it (outside-manifold perturbation; OMP) (Figure 4A; see Methods). Similar to experimental settings, we ensured that selected decoders had sufficient open-loop velocities during the baseline period, so that the network produced enough variance along the new decoder to move the cursor (albeit not necessarily towards the target).

Switching to these perturbed velocity decoders led to a decrease in performance as quantified by the hit rate (i.e. fraction of trials where the cursor came within 10% of the corresponding target). We then retrained both feedforward and feedback input weights (see Methods) for 200 trials to allow the network to recover task performance (Figure 4A,B). While in theory, the input dynamics themselves could be optimized, we chose to modify the input weights instead because such a high-dimensional optimization problem without an explicit dynamical model for the inputs is less biologically plausible. We observed that, similar to the primate experiments [36], the network recovered to different extents from different perturbations, with both within-class and across-class variability in improvements in task

performance, and larger improvements to WMPs (Figure 4C) (median change in hit rate: WMP=0.6, OMP=0.4; normalized improvement in hit rate: WMP=0.9, OMP=0.4).

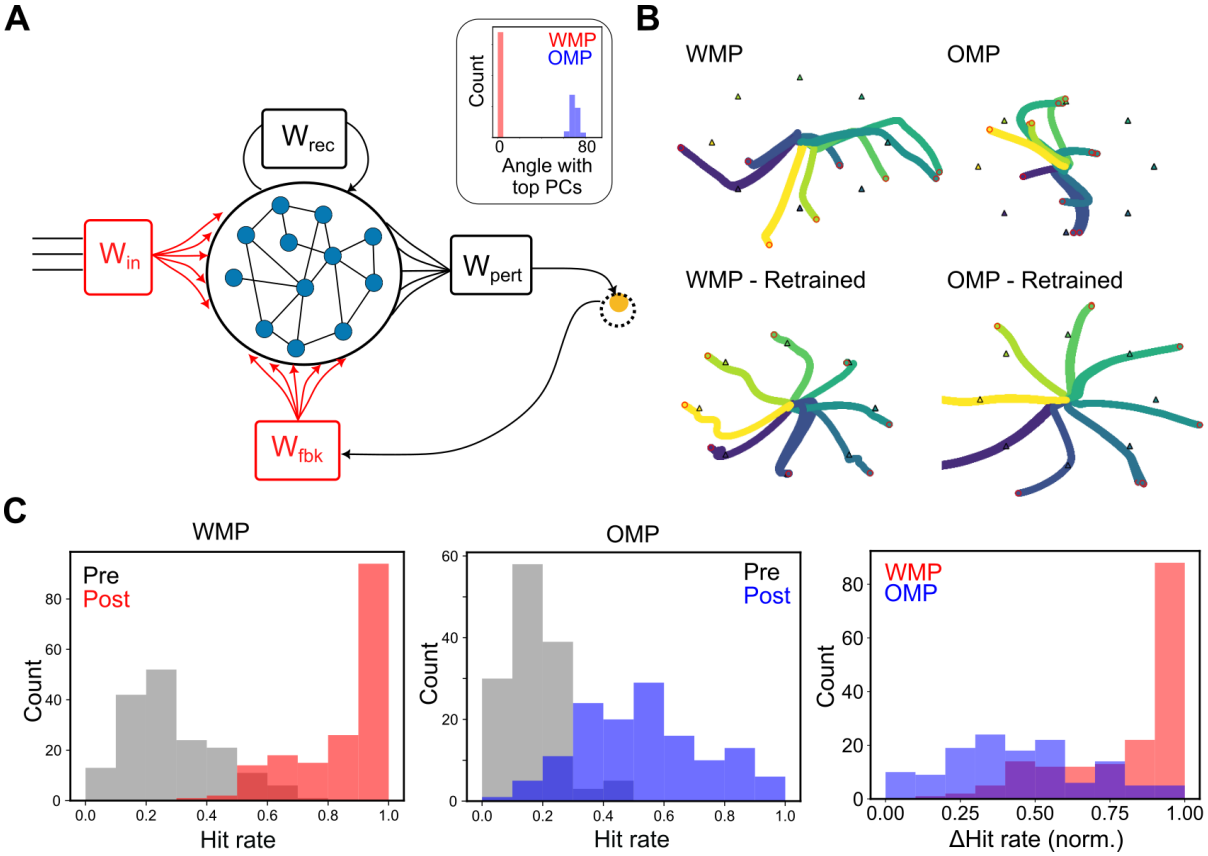

Figure 4: **Plasticity of input weights alone leads to differential learning outcomes during adaptation to decoder perturbations. (A)** Model architecture with weights being retrained during adaptation indicated in red. (Inset) Distribution of angle between new decoder W_{pert} and intrinsic manifold (top PCs) for WMPs (red, n=170) and OMPs (blue, n=135). **(B)** Example cursor trajectories after decoder perturbations (top) and after successful retraining via changes to input weights (bottom) for an example WMP and OMP. **(C)** Histogram of performance (hit rate) after decoder perturbation (grey; Pre-retraining) and after retraining for 200 trials (red/blue) for n=170 WMPs (left) and n=135 OMPs (middle). (Right) Histogram of normalized change in performance (hit rate) after 200 trials.

4.1 Variable adaptation to different decoder perturbations

To further understand how the behavioral outcomes of adaptation varied across perturbations, we examined the post-retraining cursor trajectories in more detail. First, we adopted a previous measure of "progress towards target" as the cursor velocity component in the direction of the target. We measured the average progress on reaching trajectories to different targets and assessed how asymmetric it was across targets, asking if some targets were reached faster than others? While asymmetry was high post-perturbation and reduced after retraining, it remained significantly different from zero for both classes (Progress asymmetry, post-retraining median [90%CI]: WMP=0.46 [0.25,1.19], OMP=0.79 [0.36,1.76])(Figure 5B). Indeed, asymmetric behavioral performance was also observed during the primate BCI experiments, where maximal cursor progress varied across reach targets even after successful adaptation to WMPs [55]. We also observed that the average reach times remained higher for OMPs (Target acquisition time, median [90% CI]: WMP=441 steps [369,526], OMP=554 steps [485,666])(Figure 5C).

Since all perturbations were retrained for a fixed number of iterations, we reasoned that differential improvement was a result of different learning speeds. We observed that for individual perturbations,

the learning curves (change in hit rate across training epochs) were highly consistent across different training runs (different sequence of stimuli), but that they were different across perturbations (Figure 5D). Therefore, we fit a logistic function to the average learning curve for each perturbation. We examined the distribution of both the rate parameter and total learning speed across perturbations, and noted that WMPs on average had higher learning rates (learning speed, median [90% CI]: WMP=24 [8,50], n=170, OMP=12 [6,20], n=135, p=$2e-7$; logistic curve rate parameter, median: WMP=0.030 [0.010, 0.075], OMP=0.020 [0.009, 0.048], p=$4e-7$; Mann-Whitney U test or MWU)(Figure 5E).

We next asked whether this variability depended on our specific feedback architecture or hyperparameters. We compared the use of simple error feedback (via a direct feedback projection) versus feedback inputs being computed via a 2-layer feedforward network F. This perceptron-based controller received both the position error and subsampled RNN activity as inputs, recoded these within a high-dimensional hidden layer (N_h= 100 or 200 units), and finally produced a smaller number of outputs (N_o=4 or 12), which set the dimensionality of the feedback inputs (see Methods). Note that retraining the weights of this network F is more similar to flexibly altering feedback input dynamics by updating an upstream controller. We consistently observed greater improvement and higher learning speeds for WMPs versus OMPs (Figure S7). However, as the feedback dimensionality (and controller expressivity) increased, the rate of learning increased and inter-class differences decreased. Similarly, we confirmed that both within- and across-class variability was observed using a lower rate of gradient-based learning (Figure S8). Altogether, we observed a consistent variability in the amount and speed of task performance recovery, even across WMPs, similar to experimental observations (see [36], Figure 2 and Extended Data Figure 2).

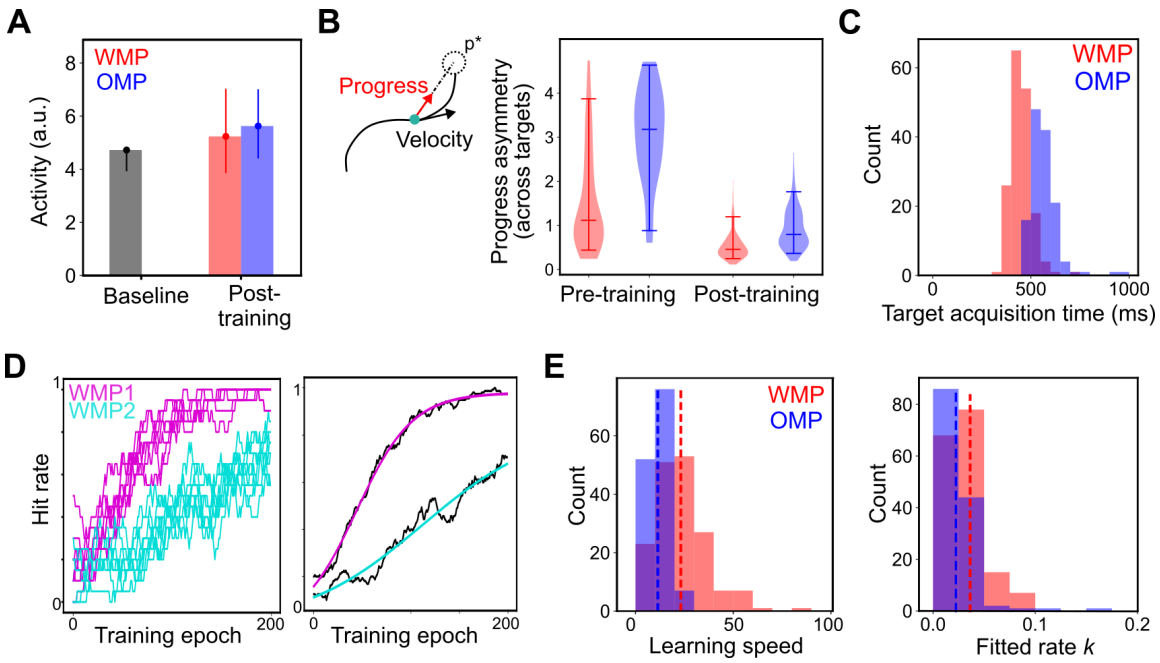

Figure 5: **Learning quality and timescales vary between WMPs and OMPs. (A)** Mean network activity before perturbation (baseline, N=5 networks) and after retraining to WMP (red, n=170) or OMP (blue, n=135) perturbations. **(B)** Distribution of behavioral progress asymmetry for WMPs (red) and OMPs (blue) before and after retraining. Progress is defined as the magnitude of the component of instantaneous velocity readout towards the target. Mean progress along the trajectory to different targets could vary, leading to asymmetric reach speeds. **(C)** Distribution of mean target acquisition time for different OMPs and WMPs after retraining. **(D)** (Left) Hit rate across training for 2 example WMPs (in pink and cyan). Each line is a different training iteration, with hit rate quantified as fraction of successful reaches in a 20-trial period. (Right) Average training curves for the two WMPs (black) are overlaid with a logistic fit, which is used to estimate learning speed. **(E)** Distribution of learning speeds (left) and rate parameter of logistic fit (right) across WMPs and OMPs. Dashed lines indicate median of the respective distributions.

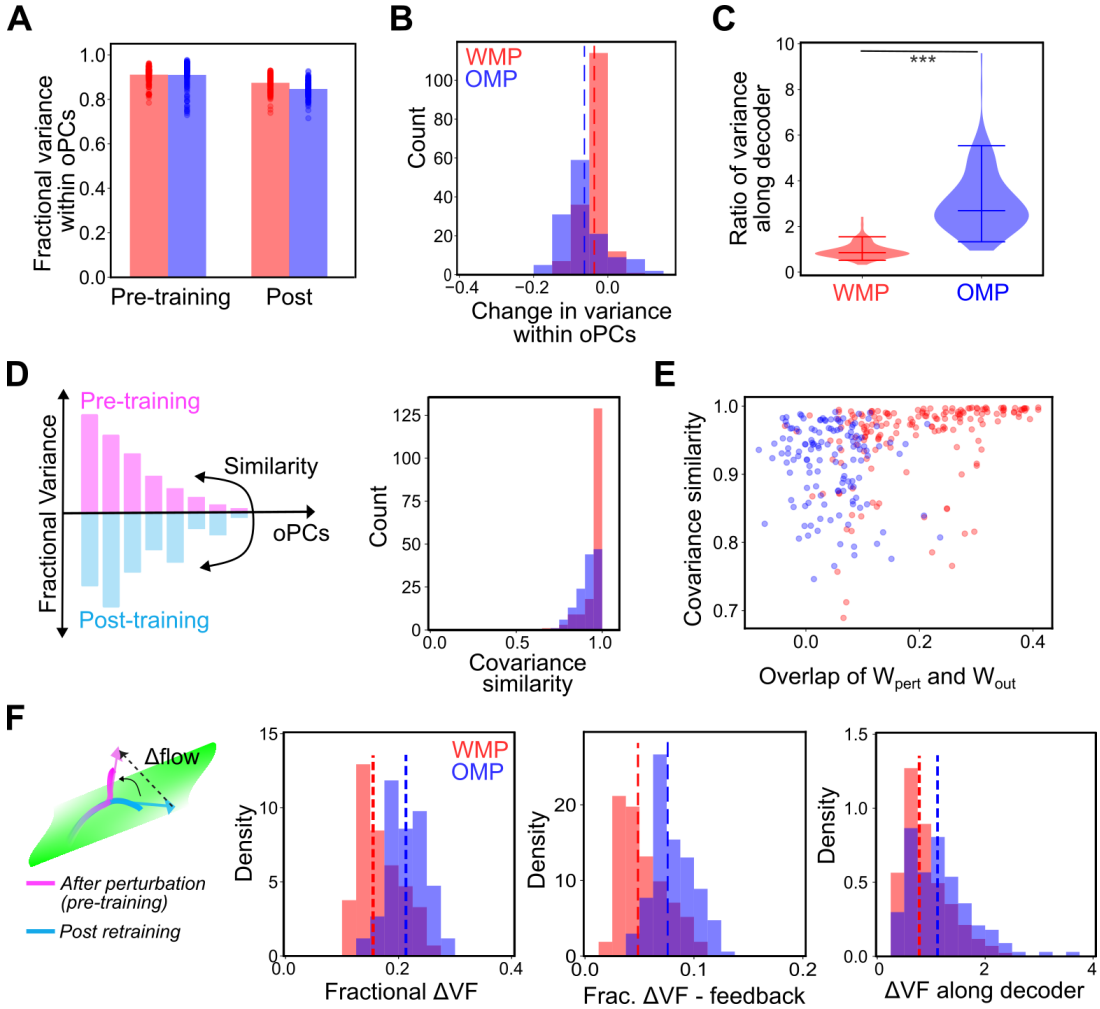

Figure 6: **Small learning-related changes to neural activity structure and underlying dynamics. (A)** Fractional variance within original intrinsic manifold (n=8 top PCS) for different WMPs (red, n=170) and OMPs (blue, n=135), before and after retraining. Scatter denotes individual decoder perturbations, bars indicate median across perturbations. **(B)** Distributions of change in fractional variance within original top PCs. Dashed lines indicate the medians. **(C)** Ratio of post- to pre-retraining activity variance along the perturbed decoder, shown as a distribution. Horizontal lines indicate the 5th, 50th and 95th percentile values. **(D)** Similarity of covariance structure was measured as the overlap between vectors describing the relative variance along different original PCs (oPCs). (Right) Distribution of covariance similarity for different WMPs and OMPs. **(E)** Covariance similarity versus similarity of perturbed and original decoder. **(F)** Retraining leads to a change in flow fields (or vector-field VF), supporting a dynamical reorganization to produce new trajectories. Distribution across WMPs and OMPs, of the mean change in the total flow (left) and the feedback-driven flow component (middle), normalized by the total flow at each point in state space, and averaged across observed neural states during post-retraining task performance. (Right) Normalized change in flow along the perturbed decoder direction.

4.2 Effective reorganization of neural dynamics with input plasticity

What changes in neural activity support this behavioral learning? Previous studies have characterised relatively little change to "neural repertoires" or the distribution of network activity states during within-session WMP learning [37]. To check if input plasticity was a potential mechanism for experimentally observed short-term learning, we first quantified changes in the statistical structure of neural activity. While we observed that mean activity levels were slightly higher post-retraining (WMP=5.3±0.9, n=170; OMP=5.6±0.7, n=135) as compared to baseline performance with the intuitive decoder (baseline=4.5±0.3,

N=5 networks, Figure 5A), activity was still mostly within the original intrinsic manifold (Figure 6A). Indeed, the change in fractional variance along the original top 8 PCs was small for both OMPs and WMPs (ΔFractional variance, median [90%CI]: WMP=-0.03[-0.10,0.01], OMP=-0.07 [-0.13,0.06], Figure 6B). We also compared the relative variance along different PCs, which remained highly similar (covariance structure similarity: WMP=0.97 [0.82, 0.99], OMP=0.94 [0.80,0.98]) (Figure 6D). While the activity manifold remained similar, we reasoned that changes in activity levels would lead to increased variance along the output direction (decoder). Indeed, we observed such an increase for variance along the OMP decoder (Ratio of post-pre training variance along decoder: WMP=0.9, OMP=2.7, p<1e-7, MWU) (Figure 6C). By contrast, the variance along the WMP decoders i.e. cursor speed distributions remained similar, consistent with previous observations of learning via neural reassociation [37].

Despite this small reorganization of activity, the network must produce new trajectories as new sequence of states within the same intrinsic manifold in order to navigate the cursor towards the target via the new decoders. We thus quantified the changes in the temporal evolution of neural activity by comparing the changes in the flow-field at various points in neural state space. While both recurrent interactions and inputs shape the net flow field, changes in input weights meant that the input-driven component was the driver of dynamical reorganisation here. We separately examined both the total change as well as feedforward- and feedback-driven components of flow field changes (Figure 6F). We observed that these changes were small (∼10-30% of original flow-field), and were higher for OMPs, particularly for the feedback-driven component (Frac. ΔVF, mean±SD: WMP=0.15±0.04, OMP=0.21±0.03; Frac. ΔVF-feedback: WMP=0.05±0.02, OMP=0.08±0.02, p<1e-7, MWU). Note that small changes in flow-fields can add up across timesteps to produce more varied trajectories. The proportional reorganization of flow along the output direction was larger, for both WMPs and OMPs (Frac. ΔVF-readout: WMP=0.90±0.42, OMP=1.19±0.66). Thus, while there were small changes in the statistical structure of neural activity during adaptation to most WMPs, changes in the underlying dynamics were more variable. This suggests learning occurred by a re-purposing of activity patterns from the same neural manifold by re-associating them with new reach targets and producing them as a part of new temporal sequences.

4.3 Learning speed depends on feedback-driven dynamics

What are the *statistical* and *dynamical* features of neural activity that determine the learning-related variability across WMPs, as well as between WMPs and OMPs? As we explain below, alignment with the intrinsic manifold is insufficient to explain all constraints on neural trajectories. For instance, Figure 7A,B shows examples of two pairs of trajectories that are generated using a combination of input and recurrent drive. All trajectories are within the same stable subspace and consist of "within-manifold" activity states. Yet, they have different control costs due to their alignment with the recurrent flow-fields and alignment with controllable directions, as determined by the interaction of recurrent and input weights. Moreover, reducing the control costs (e.g., if there are upper bounds or metabolic costs on inputs) requires the input weights to be modified to different degrees (Figure 7C). Similarly, we could define harder WMPs as those where the necessary dynamics to produce behaviorally-appropriate trajectories are quite different from existing flow-fields.

Since we implement adaptation via retraining of input weights, the suitability of pre-existing input-driven dynamics for the required trajectories is likely to influence learning speeds. We measured this suitability in two ways: (i) we looked at the ability of inputs to drive network activity along the new decoder, evaluated as feedback controllability of the new decoder under a linear dynamics assumption, (ii) we quantified the amount of reorganization of input-driven flow-fields. We observed that both the amount of learning and learning speed co-varied with the changes in feedback-driven flow-fields (Figure 7D,E). Indeed, linear regression using these two measures was sufficient to explain a large fraction of the learning variability (Figure 7D-G) (FBK model, Prediction of learning speed, cross-validated R^2=0.44 [0.27,0.54]; prediction of normalized ΔHit rate, R^2=0.51 [0.42,0.66]). Thus, perturbations that required smaller dynamical reorganization were learnt better. Similar results were observed using a mixed linear effects model with a WMP- and OMP-specific intercept (Figure S9).

This predictability was specific to feedback-driven dynamics, as feedforward controllability or changes in feedforward-driven flow did not show a strong relationship to learning measures. This was surprising since learning was poorer when only feedback weights were retrained (normalized ΔHit

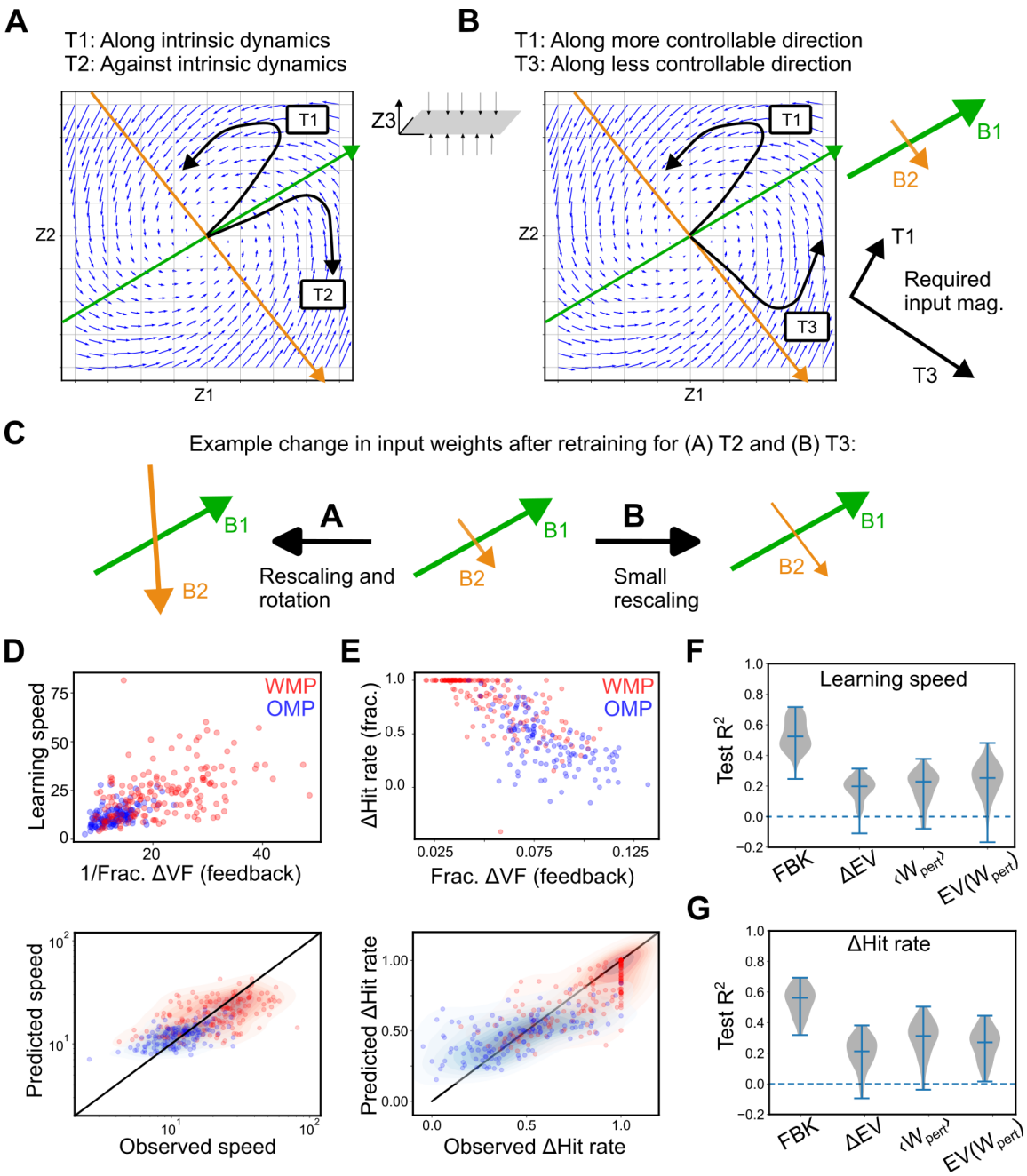

Figure 7: **Input-driven dynamical restructuring constrains learning speeds and outcomes. (A-C)** Illustrative example of how different trajectories within the same subspace can be produced with different ease/magnitude of external input, and how input-driven reorganization of effective dynamics enables new trajectories. **(A)** Recurrent vector field (blue) as well as input weight directions (green, orange) determine how neural trajectories are produced. Trajectory T1 is easier to produce than T2, as T2 requires large inputs to overcome the intrinsic flow field. B1 and B2 are input weight directions i.e. the directions in which each of the inputs pushes the neural state. **(B)** Input weights B1 are larger than B2, creating more and less controllable directions. Trajectory T3 requires larger inputs than T1. **(C)** Example of input weight changes to make trajectories T2 and T3 easier to produce, for the examples in (A) and (B). **(D)** (Top) Learning speed versus change in feedback-driven flow, for WMPs (red, n=170) and OMPs (blue, n=135). (Bottom) Learning speeds predicted using the FBK model, against the true speeds. Scatter denote individual decoder perturbations, and shaded areas show different quantiles. **(E)** Similar to (D), but for normalized change in hit rate. **(F-G)** Cross-validated R^2 (n=40 iterations) for 4 different models, predicting learning speeds in (F) and normalized change in hit rate in (G). Horizontal lines show extrema and median values of the distribution.

rate, WMP = 0.22 [-0.06, 0.66], OMP=0.16 [-0.13,0.42], n=55, p=0.09). Further, measures of statistical reorganization such as changes in variance within the intrinsic manifold (ΔEV model), angle between initial and perturbed decoders ($\langle W_{pert} \rangle$ model), or initial variance along the new decoder (EV(W_{pert}) model) were all poorer at predicting learning speed (median R^2=0.17, 0.20, 0.23 respectively) as well as the change in hit rate (R^2=0.21, 0.32, 0.28 respectively). Importantly, including these additional variables in the FBK model did not increase the amount of variance explained.

We also tested network adaptation to random decoder perturbations (RMPs) instead of structured WMPs and OMPs. We observed intermediate levels of adaptation, and learning variability was still best predicted by changes in feedback-driven dynamics (learning speed, FBK model R^2 = 0.33, normalized change in hit rate, R^2 = 0.47) (Figure S10). This supports a rather continuous form of learning variability than just discrete inter-class differences.

Lastly, we examined the gradients observed over the course of learning, as previous studies have implicated larger gradients during WMPs for faster learning [53, 54]. By contrast, we observed larger gradients for OMPs (Figure S11A), and thus larger weight updates, yet with slower adaptation speeds. One important difference from previous studies was our selection of OMPs with sufficient open-loop velocities. Overall, the magnitude of the gradient was not predictive of learning speed. When we looked at the consistency of these gradients by computing the similarity between successive weight updates, we observed that the trial-to-trial similarity was higher for WMPs (Figure S11B), which may either indicate a lower curvature of the loss landscape or greater alignment with the true gradient. Overall, rather than a difference in gradient or weight update magnitudes, it is more likely that OMP solutions are harder or take longer to find using noisy gradient descent. Moreover, the initial configuration of feedback weights and changes in feedback-driven flow predicted learning variability across different decoder perturbations and constitute *dynamical* constraints on learning.

4.4 Recurrent plasticity does not lead to variable learning outcomes

While we have focused on input plasticity as the mechanism underlying short-term adaptation, we did investigate the plausibility of recurrent plasticity as an alternative. Previous studies with updates of recurrent weights observed that all perturbed decoders were learnable, but differences between WMPs and OMPs could be attributed to (i) imperfect error feedback [51], (ii) larger synaptic changes for OMPs [52], or (iii) smaller gradients for OMPS [53, 54]. In this study, we examined the consequence of learning via recurrent plasticity using the noisy gradient estimates in trial-to-trial learning (Figure S12A). Similar to previous modelling work, we observed nearly perfect recovery of task performance within the tested duration (200 trials) (Figure S12C,D), even for slower learning rates (Figure S13), which may be attributable to the larger number of modifiable parameters and a greater expressivity under recurrent plasticity. We also observed that the trajectories were straighter after relearning, with little behavioral asymmetry (Figure S12E), and that the cursor came to rest at the targets, indicative of re-emergence of stable fixed points at the targets. Adaptation speeds were greater compared to learning via input plasticity, for both OMPs and WMPs (Figure S12F,G), although they were slightly higher for WMPs, consistent with previous results. Further, we observed that recurrent plasticity led to greater reorganization of neural activity structure, with reduction in variance along the original PCs and realignment of activity to increase variance along the new decoder, particularly for OMPs (Figure S12H,I). Moreover, the underlying dynamics were altered to a greater extent (Figure S12J). Altogether, this suggests that recurrent plasticity is unlikely to be the main driver of within-session adaptation observed in [36].

4.5 Experimental findings are consistent with input plasticity using a local learning rule

So far, we have used a non-local learning rule to estimate gradients and update network weights. To ask whether our conclusions could be extended to biologically-plausible settings, we next explored the weight perturbation learning rule that updates parameters using only local information and a global error signal. In order to make comparisons with experimentally-observed learning, we combined (i) a local learning rule (ii) with trial-to-trial learning, (iii) over a large number of training epochs (1500 epochs) (Figure 8A,B). The increase in task performance was lower compared to gradient descent, but still with a greater amount of learning with WMPs (Normalized ΔHit rate, median [90% CI]: WMP=0.25

[-0.16,0.53], n=75; OMP=0.08 [-0.20,0.38], n=75)(Figure 8C). Moreover, despite the higher-dimensional weight updates on individual trials, consolidation across the epochs was still accompanied by only small changes to fractional variance within the original intrinsic manifold (ΔFrac. explained variance by original top 8 PCs: WMP=-0.01 [-0.16,0.01], OMP=-0.02 [-0.12,0.04])(Figure 8D). There were smaller changes to the total variance along the decoder for WMPs (ratio of post-pre variance along decoder, WMP=0.85, OMP=2.69)(Figure 8E), suggesting a largely conserved neural repertoire for WMP learning. Lastly, we observed a slightly larger reorganization of feedback-driven dynamics for OMPs (Figure 8F), with a (non-significant) trend for smaller reorganization associated with greater learning (Figure 8G). Lastly, we confirmed these observations using a two-layer network as the feedback controller, where the weights from the hidden to output layer were adapted (Figure S14). Thus, the variable learning outcomes and imperfect recovery of task performance can be explained by the plasticity of input weights, using a biologically-plausible learning rule.

5 Discussion

Sensory feedback is a key component of skilled motor control and can modulate the activity of neurons in the primary motor cortex (M1) via direct and indirect pathways [21, 23–25, 65, 66]. Yet, many previous models of M1 have studied cortical dynamics in the absence of this feedback [15, 18, 67]. In this work, we reinterpret cortical trajectories as arising from an interaction between inputs and recurrent connectivity, recapitulating several previously reported features of M1 activity in feedback-driven networks (also see [12, 13]). Moreover, this feedback signal may be flexibly updated to reshape effective cortical dynamics over short-term motor adaptation. We show how input plasticity, and feedback remapping in particular, is more consistent as a mechanism for fast adaptation to new BCI decoders, with pre-existing input-driven dynamical structure serving as a major constraint for learning. Future work that identifies these dynamics from neural recordings during BCI calibration can test whether these features predict subsequent learning outcomes. Interestingly, we find that statistical features of neural activity, such as the low-dimensional task-specific intrinsic manifold, are insufficient to explain the large variability of learning outcomes and to construct robust BCI decoders. Designing more controllable and learnable BCIs requires a better understanding of feedback control of cortical dynamics, with more attention to not just the distribution of neural states but the underlying dynamics that govern the time-evolution of those neural states.

There are two key features of our framework that distinguish us from related computational studies of motor adaptation [51–55] and that enabled these new insights. First, our use of velocity decoders and inclusion of feedback allowed us to take into account the dynamical nature of neural activity, making the network's task from open-loop decoding into a closed-loop *control* task. This choice is not only more aligned with the experimental design of BCIs, but allowed us to quantify the ease of producing neural trajectories, beyond just neural states. Moreover, the presence of feedback led us to consider the asymptotic properties of the joint RNN-cursor system, rather than the network dynamics alone. Future work can include additional dynamics of the controlled plant itself, whether that of a BCI effector or the musculo-skeletal system [12, 13]. Secondly, we examined the variability between different within-manifold decoder perturbations (WMPs), where all the new decoders are aligned to the intrinsic manifold. Neither the alignment with the original decoders nor the original variance along the decoder (which would be low if the WMP is aligned with trailing PCs) could strongly predict which WMPs would be more learnable. Thus, these statistical features of neural activity were insufficient predictors of learning. Rather, we found that constraints on learning arise from the structure of input-driven dynamics that shape both the intrinsic manifold and neural trajectories.

Disentangling different notions of relevant neural dynamics

Echoing recent findings [62, 64], one of our key findings was the presence of oblique dynamics in trained recurrent networks, where the output weights are misaligned to the top activity PCs and the dynamics subspace.This suggests that maximal variance-based reconstructions or dimensionality reduction of neural data may fail to capture task-relevant dimensions that can impact downstream behavior, including what are referred to as output-potent or communication subspaces. Importantly, this oblique regime was more consistent with experimentally observed cortical activity during forelimb-based reaching and cycling tasks [62], as well as during a BCI cursor control task [64].

In another paper where the authors used optogenetic methods to perturb motor cortical activity during reaching tasks [67], strong stimulation had a relatively small effect on successive behavioral performance despite larger perturbations along the top PCs. While robustness can also arise from other factors such as corrective inputs from other regions [68], they concluded that the optogenetic perturbation dimensions and the dominant PCs were misaligned with the dynamically-relevant subspace. Similarly in [69], by studying anterolateral motor cortex (ALM) in a memory-guided licking task, the authors found a paradoxical misalignment between high-variance choice selectivity dimensions (as determined from imaging experiments) and behaviorally impactful dimensions (as identified from optogenetic perturbation experiments). They suggested that the circuit operates in an amplifying-feedforward dynamical regime that can explain the importance of small variance dimensions in shaping both future neural dynamics and behavioral choices. Overall, these results highlight a need for distinguishing between highly-controllable and highly-observable modes of neural dynamics [14, 70], and for cautious interpretation of the causal impact of the dominant low-dimensional activity identified in neural data. This is also relevant for the design of robust BCIs, where performance may depend on identifying feedback-controllable dimensions and not just the low-dimensional intrinsic manifold from calibration data.

There has been a lot of debate around the origins and validity of rotational dynamics of M1 activity [11, 15, 17, 71–74]. A previous study suggested that proprioceptive signaling can give rise to rotational "dynamics" in both sensory feedback and in the activity of somatosensory cortex and M1 [11]. However, in the context of BCI use, it is unclear whether visual feedback will have a similar effect, and it will likely depend on which aspects of cursor motion (e.g.position, velocity) are encoded along this feedback pathway. In our work, we attempted to dissociate the effect of inputs from intrinsic dynamical structure by fitting an input-driven latent dynamical system, rather than performing jPCA on neural activity alone. While we found that rotational modes could arise from the recurrent connectivity—both in our computational models and in an example BCI dataset—the relative contribution of feedback and intrinsic structure towards the effective rotational dynamics observed in M1 activity during reaching tasks needs further investigation. Moreover, the utility of these rotational modes remains an open question, although they have been suggested as a dynamical mechanism to control the kinematic profile [15, 17] and linked to feedback controllability of cortical dynamics [14].

Modelling feedback for learning in brain-computer interfaces

The use of sensory feedback during BCI use is by now well-established [21, 31, 50, 61, 75, 76]. Even simple operant conditioning BCI paradigms generally provide some form of sensory feedback or direct neural stimulation that relates to the network state or task performance (such as a tone modulated by the neuron's activity), ostensibly to aid task learning [49, 77, 78]. This feedback remains essential for the continued volitional modulation of cortical activity. A recent study using barrel cortex photostimulation as task feedback further found that the structure of the photostimulation pattern determined the rate of learning [48]. Thus, understanding the mechanisms behind the learning dynamics and representational changes during BCI use requires a deeper consideration of the role of feedback. We observed that altering the feedback mapping reshaped the effective dynamics of the networks, and the extent of this dynamical reorganization determined the learning rate. We similarly expect that learning to use non-intuitive photostimulation patterns requires greater modification of the feedback pathway, and learning may thus be too slow or limited.

As we have currently not included sensory delays, the networks could use immediately available feedback for updating their activity and consequently, the motor output. However, the relatively large delays along the visual pathway ($\sim$100 ms) necessitates the use of forward internal models that integrate efferent and afferent information to infer the state of the controlled plant, which can guide corrective feedback control. For example, a previous study [61] showed that BCI commands could be best interpreted through the lens of an internal model that combines the last visually-observed cursor state and the intervening network activity states to predict the current cursor state, adjusting the network output accordingly. Extending our model with delays and explicit forward models will help dissect the network mechanisms that support this computation. Moreover, changes to these internal models may underlie and constrain BCI adaptation.

Distributed mechanisms for sensorimotor adaptation

What is the neural and synaptic basis of the proposed input plasticity underlying short-term adaptation? Cerebellar circuitry has a well-established role in sensorimotor adaptation [38, 40, 42, 79–85], which likely extends to BCI learning. Sensory prediction errors that arise due to a mismatch between new BCI mappings and previously learnt internal models of the BCI [61] can drive plasticity within the cerebellar cortex, putatively at the granule cell-Purkinje cell synapse, and recalibrate the internal model for both feedforward and feedback control of cortical dynamics [86–88]. In fact, our results using weight perturbation for retraining inputs were inspired by previous work on slow, noisy cerebellar-dependent adaptation [89, 90]. Alternative computational theories also suggest that cerebellar feedback may guide subsequent local plasticity within M1, either by acting as a teaching signal [91, 92] or by estimating surrogate gradients for weight updates [93]. Nonetheless, better characterisation of cortico-cerebellar interactions during BCI control as well as learning is a promising next step.

In this paper, we used hit rate rather than trajectory loss as a measure of success, similar to quantification of task performance in experimental data. Indeed, in [36], animals were not required to make perfect reaches or stop at the targets, either during the baseline period or the adaptation period. Moreover, sensorimotor adaptation has been shown to have both a reward-driven policy update and an implicit sensorimotor recalibration component [94–99], which are sensitive to target errors and sensory-prediction errors respectively. These learning processes are not entirely independent and the rate of implicit adaptation can be modulated by the reward rate. Explicitly modeling the interaction between the minimization of sensory errors and reward maximization will yield interesting insights into how different learning pressures shape network connectivity and dynamics. Another recent study implicated a parallel form of input plasticity in BCI learning, where low-dimensional *feedforward* inputs are modified, potentially as a cognitive "re-aiming" strategy [55]. While we exclusively modelled upstream plasticity as the basis of adaptation, it is likely there are concurrent distributed changes, particularly for longer-timescale learning. Several computational studies have suggested error- or reward-based modulation of local connectivity in M1 during motor adaptation [92, 100, 101]. Modelling these parallel learning processes and dissecting the multi-region circuits underlying learning is an exciting avenue for future work.

Acknowledgements

We thank Matt Golub, Stefan Mihalas, Alessandro Barri, Anastasia Bizyaeva, Jacob Sacks, Dhruva Raman, and all members of the Brunton lab for useful discussions and for feedback on our manuscript. HG was supported by a Schmidt Science Fellowship. BWB was supported by Air Force Office of Scientific Research award FA9550-19-1-0386 and the Richard & Joan Komen University Chair.

Ethics declaration

The authors declare no competing interests.

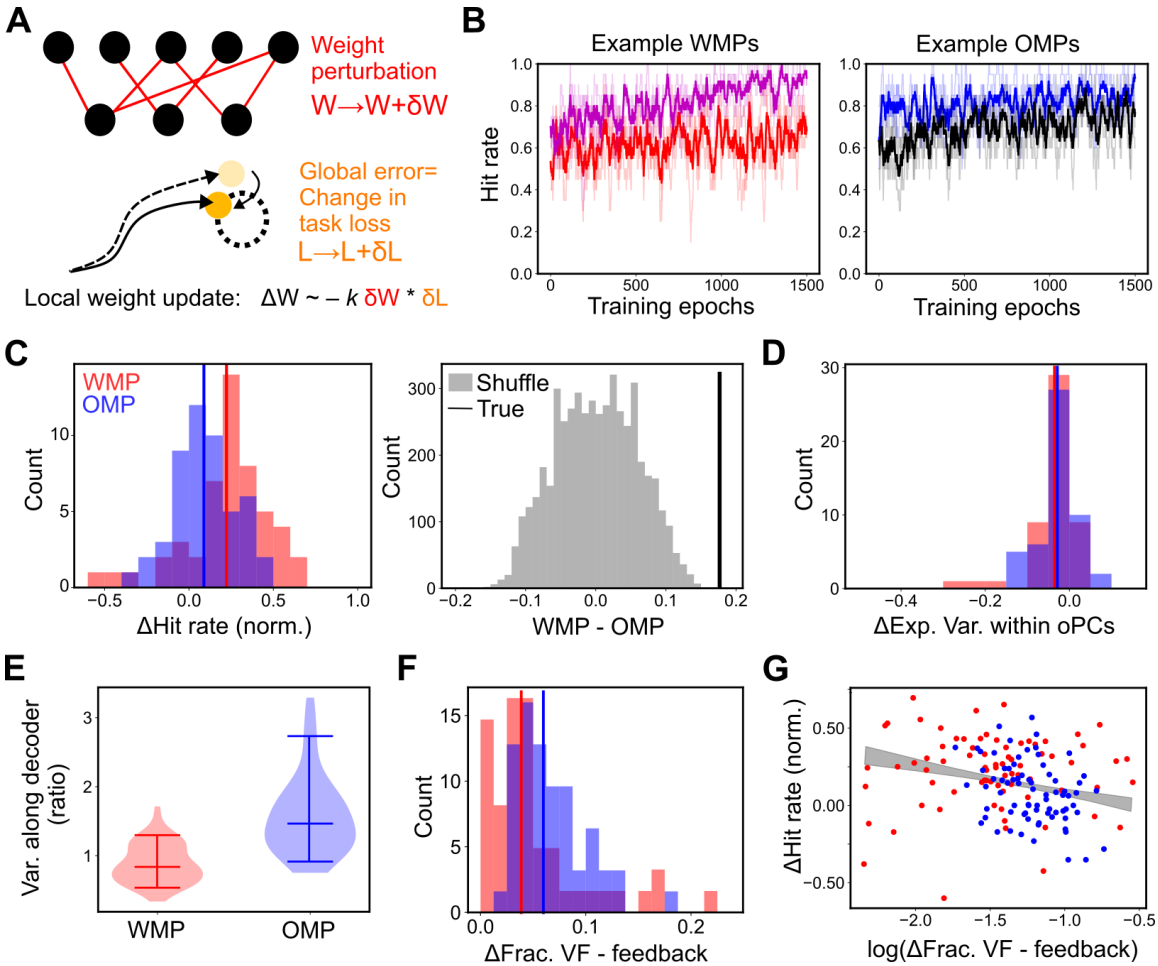

Figure 8: **Weight perturbation training recapitulates learning variability**. **(A)** Schematic of weight perturbation (WP). Each input weight is perturbed randomly (δW), which induces a change in the total loss (δL). This global change in loss is used to consolidate weight changes. k refers to a scaling factor that depends on learning speed and standard deviation of perturbations. **(B)** Learning curves i.e. change in hit rate over training epochs, for (left) 2 example WMPs and (right) 2 example OMPs. **(C-G)** Learning-related outcomes for a network architecture with direct error feedback. **(C)** (Left) Distribution of normalized changes in hit rate (amount of learning) for WMPs (red, n=75) and OMPs (blue, n=75). Vertical lines indicate the medians of the distributions. (Right) Distribution of difference between the average amount of learning with WMPs and OMPs, after shuffling across classes. Solid line shows the true difference in average amount of learning. **(D)** Distribution of change in fractional variance within the original 8 PCs, for WMPs and OMPs. **(E)** Distribution of change in variance along decoder, expressed as a ratio between post-training and pre-training values. Horizontal lines show 5th, 50th and 95th percentiles. **(F)** Distribution of change in feedback-driven vector field (as a fraction of original total flow). **(G)** Relationship between normalized change in hit rate, and the change in feedback-driven flowfields. Scatter shows individual data and line shows confidence interval of linear fit ($R^2 = 0.09, p = 0.03$).

6 Methods

6.1 Modelling framework and task description

Table 1: Network parameters.

Parameter	Value	Description
N	100	No. of RNN units
N_{in}	3	No. of feedforward stimuli (2D target + Go cue)
N_{out}	2	Output dimensionality (2D velocity)
N_{fbk}	2/4/12	Feedback dimensionality
σ_{in}	0.5	Initial scaling factor for SD of input weights W_{in} and W_{fb}
σ_{rec}	1	Initial scale for SD of recurrent weights W_{rec} (except Fig 3)
σ_{out}	5	Scaling factor for SD of readout weights W_{out} (except Fig 3)
σ_b	1	SD of RNN unit bias term
τ	5	Network time constant (dt=1)
Φ	'ReLU'	RNN unit nonlinearity
σ_n	0-1.0	Standard deviation of activity noise

Task. The network was trained to perform a 2D center-out reaching task, similar to 2D cursor-based BCI tasks, where a readout from network activity is mapped onto the velocity of a "cursor". On each trial, the network was given a target from 1 of 8 locations. After a Go cue, the cursor was to be moved towards the target, and held at the target thereafter.

Network dynamics. The network activity ($\mathbf{r} = \Phi(\mathbf{x})$) dynamics were defined as:

$$\tau \dot{\mathbf{x}} = -\mathbf{x} + W_{rec}\, \Phi(\mathbf{x}) + W_{in}\, \mathbf{u}_{in} + W_{fb}\, \mathbf{u}_{fb}, \tag{1}$$

with feedback input set to be the position error ($\mathbf{u}_{fb} = \mathbf{p}^* - \mathbf{p}$). The feedforward input $\mathbf{u}_{in}$ was a 3-dimensional vector that cued the target location $\mathbf{p}^*$ [cos(θ), sin(θ)] and the Hold/Go signal as a binary value. W_{rec}, W_{in}, W_{fb} refer to the recurrent, feedforward input and feedback input weights respectively and the nonlinearity Φ was chosen to be ReLU unless otherwise noted. The cursor position $\mathbf{p}$ evolved according to:

$$\dot{\mathbf{p}} = g\, W_{out}\, \Phi(\mathbf{x}) \tag{2}$$

with W_{out} as the readout weights ("decoder") and g=1e-3 as a velocity scaling factor.

For simulations where another network F was set as the feedback controller, the input $\mathbf{y}$ to the network F was a concatenated vector of sampled RNN state ($\mathbf{r}_s = M_s \Phi(\mathbf{x})$) and the position error ($\mathbf{p}^* - \mathbf{p}$). This input was then passed via the network F (with a chosen architecture) with the final output $\mathbf{x}_{out} = F(\mathbf{y})$. The feedback inputs to the RNN were set to be the output of this network F ($\mathbf{u}_{fb} = \mathbf{x}_{out}$).

Table 2: Feedback controller (F) parameters.

Parameter	Value	Description
N_{mf}	10	Dimensionality of RNN sampling weights M_s
N_{hid}	100/200	No. of hidden units in F
N_{out}	4/12	No. of output units in F
Φ_{hid}	'ReLU'	Nonlinearity at hidden layer
Φ_{out}	None	Nonlinearity at output layer

Training. Weights were initialized from a normal distribution with standard deviation equal to $\sigma_{in}/\sqrt{N_{in}}$ (input weights W_{in}), $\sigma_{in}/\sqrt{N_{fbk}}$ (feedback weights W_{fbk}), $\sigma_{rec}/\sqrt{N}$ (recurrent weights W_{rec}) and $\sigma_{out}/\sqrt{N}$ (readout weights W_{out}) respectively. For initial training, the random readout weight matrix W_{out} was fixed while both recurrent weights (W_{rec}) and input weights (W_{in}, W_{fbk}) were trained

via gradient descent (Adam optimizer) with a batch size of 32. For networks with a feedback controller network F, the output layer $W^F_{hid\to out}$ of the feedback controller was trained instead of W_{fbk}. For all experiments, the readout weights W_{out} were always kept fixed, either initialized randomly or as decoder perturbations. The minimized loss function was the average squared difference between the desired and actual cursor trajectory, plus weight and activity regularization terms (using the L_2 norm), where the desired trajectory was a sigmoidal function (in time) along the direct vector to the target. During training, we introduced small external perturbations in the form of Gaussian bumps added to the current cursor position ($\sigma = 10dt$, amplitude$= 0.02$). For testing, we perturbed the cursor in the form of a discrete jump, by moving either the x- or y- coordinate by 0.1. We tested the networks by introducing these perturbations at timesteps 150, 300, and 500.

For experiments with adaptation to decoder perturbations (see later), weights were trained either using Adam optimizer with batch size 1 (Figures 4–7) or using weight perturbation (Figure 8). For networks with direct error feedback, either both W_{in} and W_{fbk}, or only W_{rec} was trained. For network simulations with controller network F, the weights from the hidden to output layer ($W^F_{hid\to out}$) were trained along with W_{in}. Further, the trajectory loss was just computed during the hold-at-target period (from step no. 300 to 800) such that the network was neither constrained to follow a particular trajectory nor required to hold at target until the end of the trial (1500 steps).

Table 3: Initial training parameters.

Parameter	Value	Description
lr	0.001	Learning rate
T_{max}	500	No. of timesteps in a trial
T_{stim}	20	Stimulus onset timestep
T_{go}	[100,200]	Go cue timestep (range)
Ne_{tr}	400	Total number of training epochs (trials)
Ne_{pert}	250	No. of training epochs with cursor perturbations
B_{tr}	32	Batch size
α	0.0003	Regularization factor - input weight norm
γ	0.0003	Regularization factor - recurrent weight norm
β	0.01	Regularization factor - Activity norm

Weight perturbation. In each training epoch, a pre-perturbation loss (defined above) was computed using the current weights W before adding a small perturbation δW to each of the trainable weights (independently drawn from a Gaussian distribution). The change in the loss due to this perturbation (δL) was used as the error signal, and the weight changes were consolidated as $W \leftarrow W - \frac{\eta}{\sigma_W^2} * \delta L * \delta W$.

Table 4: Altered training parameters for decoder adaptation.

Parameter	Value	Description
lr	0.001/0.0002/0.0001	Learning rate
T_{max}	1500	No. of timesteps in a trial
Ne_{tr}	200	Total number of training epochs (trials)
Ne_{pert}	0	No. of training epochs with cursor perturbations
B_{tr}	1	Batch size
η	0.001	Learning rate for weight perturbation
σ^{WP}_{in}	0.01	SD of feedforward weight perturbation
σ^{WP}_{fbk}	0.01	SD of feedback weight perturbation

6.2 Decoder manipulations

Intrinsic manifold and intuitive decoders. The trained network exhibited low-dimensional activity, allowing us to define an intrinsic manifold using the top 8 principal components (PCs) that captured most

of the activity variance. We then performed linear regression between the cursor velocity and activity along these 8 PCs to identify the intuitive decoder W_{ID} as:

$$W_{ID} = K\, U_{IM}, \tag{3}$$

where K are the regression coefficients, and the rows of U_{IM} are the loadings for the top 8 PCs.

Within-manifold, outside-manifold and random decoder perturbations. Following [36], we constructed new perturbed readouts either by permuting the regression coefficients i.e. columns of K to construct WMPs, or by permuting columns of U_{IM} to construct OMPs. To constrain the maximum number of OMPs to be the same as WMPs, we followed the steps in [36], dividing neurons into 8 groups, and permuting grouped columns. (For dividing neurons into groups, we first sorted the neurons by their activity variance, then distributing the top 8 neurons randomly into 8 groups, then the next 8 neurons randomly into those 8 groups, and so on). For random decoder perturbations (RMPs), we independently sampled 2 random vectors from the full state space for the 2D velocity decoder and rescaled them to have the same norm as the original decoder.

We screened all possible WMPs and OMPs, and 10^5 RMPs, and only selected those that passed several criteria, adapted from the conditions in [36]. The mean "open loop velocity" under the RMP/WMP/OMP was constrained to be within 0.5x-to-3x of mean velocity observed with using W_{ID} as readout, and the mean angular difference between the two open loop velocities was less than 90 degrees. The mean closed loop velocity was constrained to be within 0.5x-to-2x of velocity observed with using W_{ID} as readout. One of these RMPs, WMPs or OMPs was then set to be the new decoder W_{pert}, and the network weights were retrained to recover task performance.

6.3 Data analysis

6.3.1 Network structure analysis

Preparatory and movement PCs. We separately computed principal components of network activity during the delay period (stimulus onset to Go-cue) and during the movement period (Go-cue to reaching target), and computed the principal angles between the subspaces defined by the top 4 PCs for each (referred to as the preparatory and movement subspace). For visualisation of preparatory and movement-related activity in Figure 2B, we concatenated both sets of top 4 PCs, and performed QR decomposition to get fully orthogonalized subspaces, similar to [59].

Error-correction dynamics. We simulated the network in "open-loop" by clamping the feedback input to be zero (unperturbed baseline) or setting the error to be nonzero for one timestep (perturbed network). First we initialized the network randomly but allowed the network to evolve towards its Go-period fixed point over 100 timesteps while the error was clamped to zero. Throughout, the feedforward inputs were set to an independently sampled target location (θ_{tgt}) and the Go-cue set to 1. Then an error (of magnitude s and direction θ_e) was provided via the feedback weights for one timestep. The velocity correction was calculated based on the difference in network activity between the baseline and perturbed conditions: $\Delta \mathbf{v}_t^p = W_{out}\, \Delta \mathbf{r}_t$. Corrections were averaged across 1000 different initial activity states for each reach direction (θ_{tgt}, 13 targets, between $-\pi$ and π), for each angular error (θ_e, 20 directions) and 5 different error magnitudes ($s = 0.1, 0.3, 0.5, 0.8, 1.0$). We analyzed both the magnitude (norm) of the 2D velocity correction at each timestep, as well as its angular direction (θ_v).

Fixed point analysis. We computed the fixed points of the joint system for different values of feedforward inputs - the target inputs were set to each of the 8 targets, and the Go-cue set to 0 for preparatory epoch and to 1 for the movement/hold-at-target epoch. For example, by freezing the target input to (0,0) and the Go-cue input to 1, we obtain the fixed point during the movement epoch, when the network needs to reach target at (0,0). We obtained the minima of summed neural and cursor speed using gradient descent with an adaptive step size, and confirmed that the minimal speed was less than a small tolerance value. We computed the eigenvalues of the Jacobian at the obtained minima, and confirmed that the real parts were negative, indicative of stable fixed points. For visualization, we concatenated all fixed points during the preparatory (or hold-at-target) epochs, and used the top 3 PCs of this dataset.

LDS fitting to model data. We fit an input-driven linear latent dynamical system to the trial-averaged neural activity (for each target) during the movement epoch. Latent dynamics were defined as:

$$\dot{\mathbf{z}}(t) = A\mathbf{z}(t) + B\mathbf{u}(t); \qquad \mathbf{z}(0) = \mathbf{z}_0 \tag{4}$$

where $\mathbf{z}$ is the latent variable, and input $\mathbf{u}$ is a concatenation of the 2D target location and a 2D error signal i.e. vector-to-target from the current position, computed using the ideal trajectory. The neural activity $\mathbf{r}$ was then defined as a readout of these latent dynamics:

$$\mathbf{r}(t) = \Phi(C\mathbf{z}(t) \ + \ d) \tag{5}$$

with a ReLU nonlinearity (Φ) and a bias current d.

For model fitting, we first selected a random subset of neurons ($\sim 90\%$) as training set and used the rest for testing. Since we used a non-probabilistic model, we fit the parameters $A, B, C_{tr}, d_{tr}, \mathbf{z_0}$ using gradient descent, where C_{tr} and d_{tr} corresponds to the rows of C and d for the training set neurons. The loss was defined as the squared difference between true and fitted neural activity. Next we fixed the latent dynamics ($A, B, \mathbf{z_0}$) and retrained C, d (in practice, C_{test}, d_{test}) on a subset of the targets to identify the observation weights for the test neurons. Model performance with these learnt parameters was then computed as variance explained for the activity of test neurons on the remaining held-out targets. We compared model performance (test R^2) for different numbers of latent variables (k=4 to 14) and found k=8 or 10 to be the best fits.

We computed the eigenvalues of the network dynamics A, and identified the complex pair of eigenvalues with the largest real value as the first rotational mode. We plotted a distribution of real and imaginary values for these rotational modes across different training iterations and across 3 different trained RNN models, by varying parameter initialization and train/test sets.

LDS fitting to BCI dataset. For comparison of network dynamics between experimental recordings and trained computational models, we used an openly available dataset of M1 activity during a center-out BCI reaching task from [60]. This dataset is available at (https://dandiarchive.org/dandiset/000404/0.230605.2024). For analysis we downsampled data from monkey G by 6x (60Hz to 10Hz), and data from monkey J by 5x (200Hz to 40Hz). We used the binned spike counts as neural observations and centred cursor positions as behavioural observations.

For LDS fits, we did not perform trial averaging as trial lengths were quite variable. Instead, we fit an input-driven LDS to trial-concatenated data using the `ssm` package (https://github.com/lindermanlab/ssm) [102]. The latent dynamics were again given by equation (4), with the feedback input computed based on the behavioral trajectory on each trial. The probability of observed neural spike counts were based on an inhomogeneous Poisson process with time-varying firing rates given by

$$\mathbf{r}(t) = exp(C\mathbf{z}(t) \ + \ d). \tag{6}$$

We first selected a subset of the trials (80%) as the training dataset to estimate the model parameters. The variational lower bound for the data log-likelihood (or evidence lower bound, ELBO) was then computed for both the training and test datasets using the estimated parameters. We compared the consistency of the inferred underlying latent dynamics by comparing the eigenvalues of the dynamics matrix A across different training subsets. We confirmed that this consistency was not observed in a shuffle control, where activity of each neuron was independently shifted in time, thus destroying shared dynamical structure. We also identified the rotational frequency and decay timescale of the slowest complex eigenvalue of each fit, and compared it to the inferred rotational modes of the trained models.

For fixed point analysis, we solved the linear equation $A\mathbf{z} + B\mathbf{u} = 0$, by setting the feedforward input to each of the 8 targets and the feedback input to be 0 (corresponding to a cursor held stationary at the target). We then computed the first 2 principal components from the set of 8 fixed points, and visualized the inferred latent state $\hat{\mathbf{z}}_k(T)$ (based on observed neural activity) at the end of each trial k in the same 2D space ($\hat{\mathbf{z}}_k^{FP}$). We also used nonlinear classification techniques (K-nearest neighbours) to decode the trial target from this projected latent state $\hat{\mathbf{z}}_k^{FP}$, as well as from the full latent state $\hat{\mathbf{z}}_k(T)$, the inferred firing rate $\hat{\mathbf{r}}_k(T)$, and the true binned spike count at the end of each trial. A similar result was obtained using other nonlinear classifiers.

Alignment with decoder. We computed the principal angles between the readout weights W_{out} and the task-activity subspace defined by the top 8 activity PCs (intrinsic manifold) or the intuitive decoder W_{ID}. For alignment with the task-dynamics subspace we used the top 8 right singular vectors (sensitive directions) and top 8 left singular vectors (amplified directions), and computed the principal angles between these subspaces and W_{out}.

Robustness to activity noise. First, we simulated networks with additive white noise added independently to each neuron ($\sigma = 0.3$). We next computed the task-irrelevant noise by subtracting the trial-averaged activity (separately for each reach target). We computed the variance of this noise when projected onto different directions - random 2D subspace, true output weights W_{out}, intuitive decoder W_{ID} and top 2 activity PCs. To compute the signal-to-noise (SNR), we normalized this noise variance by the signal variance i.e. variance of the condition-averaged activity. We repeated this analysis for varying levels of injected noise (standard deviation between 0.03 to 10).

Second, we looked at robustness to directed activity perturbations. The activity was perturbed 50 ms after the Go-cue by an additive shift of $\pm a\hat{\mathbf{u}}$, where a is the amplitude of the perturbation (stim. amplitude), and $\hat{\mathbf{u}}$ is a unit vector pointing in the direction of output weights W_{out}, activity PC 1 and 2, intuitive decoder W_{ID}, or a random direction W_{rnd}. For each of the 2D subspaces, $\hat{\mathbf{u}} = \frac{1}{\sqrt{2}}(\hat{\mathbf{u}}_1 + \hat{\mathbf{u}}_2)$, where $\hat{\mathbf{u}}_i$ are the 2 column vectors. The activity was allowed to evolve undisturbed after this single impulse. The net error was computed as the area under the curve (AUC) for the difference between perturbed and unperturbed trajectories between 0-150 ms after perturbation (i.e. during the movement period). The immediate impact of the perturbation was quantified as the AUC for the difference in trajectory between 0-10 ms after perturbation. For net error in full activity space, the AUC was computed for each single neuron and averaged across neurons. For error along the top 2 activity PCs or of cursor position, the AUC was averaged across each of the 2 dimensions. We repeated this analysis for varying perturbation amplitude a.

6.3.2 Adaptation analysis

Task performance. To facilitate comparison with measures of task performance and adaptation defined on experimental data, we computed the hit rate as the fraction of trials where the cursor came within 10% (of total distance) of the target in the defined time window, and used this instead of using the trajectory loss. Similar to experiments in [36], this meant that for successful adaptation, the network was not required to keep the cursor still at the target but just to "hit" the target within a defined time window. The normalized improvement in hit rate was computed as $(h_{post} - h_{pre})/(1 - h_{pre})$, where hit rate h was computed using performance on 200 trials. This quantified the "amount of learning" and was used as a measure of success of adaptation. We defined the acquisition time for successful (hit) trials as the first timepoint where the cursor crossed this 10% threshold. Progress was defined at each timepoint as the component of the cursor velocity along the current vector-to-target. Progress asymmetry was calculated by first computing mean progress (P_i) for each of the targets i and then computing the index $\frac{max(P_i)-min(P_i)}{mean(P_i)}$.

Adaptation learning rate. Individual learning curves were calculated by getting smoothed hit rates across the training trials using a uniform window (of size 20 trials). The smoothing was necessary as hit rates on individual trials took the value of 0 or 1 during this training period. These learning curves were similar across different runs of adapting to the same decoder (different sequence of stimuli). Hence we averaged the learning curves across 3 to 5 adaptation runs to the same decoder and fit a logistic function to these learning curves:

$$h(t) = b + \frac{a}{1 + exp(-k(t - t_0))}, \tag{7}$$

where the rate of learning was quantified as the parameter k or the total learning speed (defined as $a * k$).

Statistical reorganization. To compare similarity of covariance structure between pre- and post-adaptation activity, we took the vectors $\mathbf{s}_i$ of relative standard deviation along the top 8 PCs, and computed the similarity as the dot product between the normalized vectors: $\hat{\mathbf{s}}_{pre} \cdot \hat{\mathbf{s}}_{post}$. Similarity of

decoders was computed as either the angle between the decoders or the average overlap (inner product) between the corresponding decoder dimensions i.e. for a 2-dimensional decoder, the overlap is $\frac{1}{2}\sum_{j=1}^{2} W_{out}^{j} \cdot W_{pert}^{j}$.

Controllability of readout. For controllability analysis, we assumed a linear approximation of network dynamics by ignoring the ReLU nonlinearity. We first computed the controllability gramians W_c separately for feedforward weights W_{in} and feedback weights W_{fbk} by solving the Lyapunov equation:

$$AW_c + W_c A^T + BB^T = 0, \tag{8}$$

where $A = (W_{rec} - I)$ and B are the corresponding input weights. We then computed the squared controllability along the readout as $Tr(W_{out}^T W_c W_{out})/2$, where Tr denotes the trace operation.

Changes in flow-fields. The flow i.e. the direction in which the network state would move at the next timestep could be computed at any state along a trajectory using the network dynamics equation in (1). We then computed the change in this flow (or vector-field) along the final observed trajectory, by using pre-training and post-training weights:

$$\Delta VF_{total} = \Delta W_{rec}\Phi(\mathbf{x}) + \Delta W_{in}\mathbf{u}_{ff} + \Delta W_{fbk}\mathbf{u}_{fb}. \tag{9}$$

The three terms in Eq. 9 separately give the contribution of recurrent-driven, feedforward-driven, and feedback-driven changes respectively. We normalized the $L2$ norm of the individual contributions by the norm of the total pre-training vector field, and computed its mean along all the trajectories to different targets.

Regression analysis. We created a linear regression model for both the log learning speed and the normalized improvement in hit rate, using different sets of predictors:

- FBK model: Log feedback controllability of new decoder W_{pert}, and log fractional change in feedback-driven vector field.
- ΔEV model: Angle between the perturbed decoder and the intrinsic manifold, and change in fractional variance within the original PCs over training.
- $< W_{pert} >$ model: Principal angle between perturbed decoder and the original decoder W_{out}, as well as angle with the intuitive decoder W_{ID}
- $EV(W_{pert})$: Principal angle between perturbed decoder and the original decoder, and initial activity variance along the perturbed decoder.

Regression performance was evaluated as the cross-validated explained variance captured on a test set, and recomputed for multiple random partitions (K=40) of the data into training and test sets (80/20).

For the linear mixed effects model, we included a group-dependent (WMP/OMP) intercept as the random effect. We computed the predictions of learning speed and amount of learning for held-out data (20%) and quantified the explained variance R^2 on this test set, for multiple random partitions of the data (K=40).

Statistical analysis. Measures are generally reported as median values, along with the 5th-95th percentile values as the confidence interval, unless otherwise noted. For most analyses, we used non-parametric tests to calculate the relevant test statistic and p-values. For comparison of various measures between WMP and OMP adaptation, we used Mann-Whitney U test. For linear regression, model performance is reported as the median cross-validated variance explained (R^2), and p-values reported based on the F-statistic. Significance of individual predictors was based on the t-statistic, and a posthoc Wald test.

Code and Data availability

All code to generate the main simulations in the manuscript and to perform data analysis will be released on Github: https://github.com/harshagurnani/FeedbackControlledRNN.

References

1. Gallego, J. A., Perich, M. G., Miller, L. E. & Solla, S. A. Neural manifolds for the control of movement. *Neuron* **94,** 978–984 (2017).
2. Georgopoulos, A. P., Kalaska, J. F., Caminiti, R. & Massey, J. T. On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *Journal of Neuroscience* **2,** 1527–1537 (1982).
3. Cheney, P. D. & Fetz, E. E. Comparable patterns of muscle facilitation evoked by individual corticomotoneuronal (CM) cells and by single intracortical microstimuli in primates: evidence for functional groups of CM cells. *Journal of neurophysiology* **53,** 786–804 (1985).
4. Morrow, M. M., Jordan, L. R. & Miller, L. E. Direct comparison of the task-dependent discharge of M1 in hand space and muscle space. *Journal of neurophysiology* **97,** 1786–1798 (2007).
5. Hepp-Reymond, M.-C., Kirkpatrick-Tanner, M., Gabernet, L., Qi, H.-X. & Weber, B. Context-dependent force coding in motor and premotor cortical areas. *Experimental brain research* **128,** 123–133 (1999).
6. Hatsopoulos, N. G., Ojakangas, C. L., Paninski, L. & Donoghue, J. P. Information about movement direction obtained from synchronous activity of motor cortical neurons. *Proceedings of the National Academy of Sciences* **95,** 15706–15711 (1998).
7. Shenoy, K. V., Sahani, M. & Churchland, M. M. Cortical control of arm movements: a dynamical systems perspective. *Annual review of neuroscience* **36,** 337–359 (2013).
8. Todorov, E. & Jordan, M. I. Optimal feedback control as a theory of motor coordination. *Nature neuroscience* **5,** 1226–1235 (2002).
9. Scott, S. H., Cluff, T., Lowrey, C. R. & Takei, T. Feedback control during voluntary motor actions. *Current opinion in neurobiology* **33,** 85–94 (2015).
10. Scott, S. Optimal feedback control and the neural basis of volitional motor control. *Nature Reviews Neuroscience* **5,** 532–545 (2004).
11. Kalidindi, H. T. *et al.* Rotational dynamics in motor cortex are consistent with a feedback controller. *Elife* **10,** e67256 (2021).
12. Codol, O., Michaels, J. A., Kashefi, M., Pruszynski, J. A. & Gribble, P. L. MotorNet: a Python toolbox for controlling differentiable biomechanical effectors with artificial neural networks. *bioRxiv,* 2023–02 (2023).
13. Almani, M. N., Lazzari, J., Chacon, A. & Saxena, S. μSim: A goal-driven framework for elucidating the neural control of movement through musculoskeletal modeling. *bioRxiv,* 2024–02 (2024).
14. Kumar, A. & Bouchard, K. Feedback Controllability is a Normative Theory of Neural Population Dynamics. *Preprint at Research Square* (2024).
15. Churchland, M. M. *et al.* Neural population dynamics during reaching. *Nature* **487,** 51–56 (2012).
16. Pandarinath, C. *et al.* Latent factors and dynamics in motor cortex and their application to brain–machine interfaces. *Journal of Neuroscience* **38,** 9390–9401 (2018).
17. 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,** e1005175 (2016).
18. 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).
19. 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).
20. Sauerbrei, B. A. *et al.* Cortical pattern generation during dexterous movement is input-driven. *Nature* **577,** 386–391 (2020).
21. 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).

22. 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).

23. Fang, P.-C., Stepniewska, I. & Kaas, J. H. The thalamic connections of motor, premotor, and prefrontal areas of cortex in a prosimian primate (Otolemur garnetti). *Neuroscience* **143,** 987–1020 (2006).

24. Omrani, M., Pruszynski, J. A., Murnaghan, C. D. & Scott, S. H. Perturbation-evoked responses in primary motor cortex are modulated by behavioral context. *Journal of neurophysiology* **112,** 2985–3000 (2014).

25. Evarts, E. V. & Fromm, C. Sensory responses in motor cortex neurons during precise motor control. *Neuroscience letters* **5,** 267–272 (1977).

26. Cross, K. P., Cook, D. J. & Scott, S. H. Convergence of proprioceptive and visual feedback on neurons in primary motor cortex. *bioRxiv,* 2021–05 (2021).

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

28. Gilja, V. *et al.* A high-performance neural prosthesis enabled by control algorithm design. *Nature neuroscience* **15,** 1752–1757 (2012).

29. Willett, F. *et al.* A high-performance speech neuroprosthesis. *Nature* **620,** 1031–1036 (2023).

30. Bensmaia, S. J. & Miller, L. E. Restoring sensorimotor function through intracortical interfaces: progress and looming challenges. *Nature Reviews Neuroscience* **15,** 313–325 (2014).

31. Shanechi, M. M. *et al.* Rapid control and feedback rates enhance neuroprosthetic control. *Nature communications* **8,** 13825 (2017).

32. Willett, F. R. *et al.* Principled BCI decoder design and parameter selection using a feedback control model. *Scientific reports* **9,** 1–17 (2019).

33. Liu, M. F. *et al.* Volitional control of movement interacts with proprioceptive feedback in motor cortex during brain-computer interface control in humans. *medRxiv,* 2024–02 (2024).

34. Orsborn, A. L. & Pesaran, B. Parsing learning in networks using brain–machine interfaces. *Current opinion in neurobiology* **46,** 76–83 (2017).

35. Golub, M. D., Chase, S. M., Batista, A. P. & Byron, M. Y. Brain–computer interfaces for dissecting cognitive processes underlying sensorimotor control. *Current opinion in neurobiology* **37,** 53–58 (2016).

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

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

38. 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).

39. 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).

40. 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).

41. Perich, M. G., Gallego, J. A. & Miller, L. E. A neural population mechanism for rapid learning. *Neuron* **100,** 964–976 (2018).

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

43. Sun, X. *et al.* Cortical preparatory activity indexes learned motor memories. *Nature,* 274–279 (2022).

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

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

46. Kumar, N., Manning, T. F. & Ostry, D. J. Somatosensory cortex participates in the consolidation of human motor memory. *PLoS biology* **17,** e3000469 (2019).

47. Darainy, M., Manning, T. F. & Ostry, D. J. Disruption of somatosensory cortex impairs motor learning and retention. *Journal of Neurophysiology* **130,** 1521–1528 (2023).

48. Abbasi, A. *et al.* Brain-machine interface learning is facilitated by specific patterning of distributed cortical feedback. *Science Advances* **9,** eadh1328 (2023).

49. Prsa, M., Galiñanes, G. L. & Huber, D. Rapid integration of artificial sensory feedback during operant conditioning of motor cortex neurons. *Neuron* **93,** 929–939 (2017).

50. Fisac, C. F. & Chase, S. M. Sensory constraints on volitional modulation of the motor cortex. *bioRxiv,* 2023–01 (2023).

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

52. Wärnberg, E. & Kumar, A. Perturbing low dimensional activity manifolds in spiking neuronal networks. *PLOS computational biology* **15,** e1007074 (2019).

53. Humphreys, P. C., Daie, K., Svoboda, K., Botvinick, M. & Lillicrap, T. P. BCI learning phenomena can be explained by gradient-based optimization. *bioRxiv,* 2022–12 (2022).

54. Payeur, A., Orsborn, A. & Lajoie, G. Neural manifolds and gradient-based adaptation in neural-interface tasks. *bioRxiv* (2023).

55. Menendez, J. A. *et al.* A theory of brain-computer interface learning via low-dimensional control. *bioRxiv,* 2024–04 (2024).

56. Kim, S.-P., Simeral, J. D., Hochberg, L. R., Donoghue, J. P. & Black, M. J. Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia. *Journal of neural engineering* **5,** 455 (2008).

57. Simeral, J., Kim, S.-P., Black, M., Donoghue, J. & Hochberg, L. Neural control of cursor trajectory and click by a human with tetraplegia 1000 days after implant of an intracortical microelectrode array. *Journal of neural engineering* **8,** 025027 (2011).

58. Gilja, V. *et al.* Clinical translation of a high-performance neural prosthesis. *Nature medicine* **21,** 1142–1145 (2015).

59. 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,** 13239 (2016).

60. Athalye, V. R. *et al.* Invariant neural dynamics drive commands to control different movements. *Current Biology* **33,** 2962–2976 (2023).

61. Golub, M. D., Yu, B. M. & Chase, S. M. Internal models for interpreting neural population activity during sensorimotor control. *Elife* **4,** e10015 (2015).

62. Schuessler, F., Mastrogiuseppe, F., Ostojic, S. & Barak, O. Aligned and oblique dynamics in recurrent neural networks. *arXiv preprint arXiv:2307.07654* (2023).

63. Semedo, J. D., Zandvakili, A., Machens, C. K., Byron, M. Y. & Kohn, A. Cortical areas interact through a communication subspace. *Neuron* **102,** 249–259 (2019).

64. Rajeswaran, P., Payeur, A., Lajoie, G. & Orsborn, A. L. Assistive sensory-motor perturbations influence learned neural representations. *bioRxiv* (2024).

65. Asanuma, H., Larsen, K. & Zarzecki, P. Peripheral input pathways projecting to the motor cortex in the cat. *Brain research* **172,** 197–208 (1979).

66. 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).

67. O'Shea, D. J. *et al.* Direct neural perturbations reveal a dynamical mechanism for robust computation. *bioRxiv,* 2022–12 (2022).

68. Li, N., Daie, K., Svoboda, K. & Druckmann, S. Robust neuronal dynamics in premotor cortex during motor planning. *Nature* **532,** 459–464 (2016).

69. Daie, K., Fontolan, L., Druckmann, S. & Svoboda, K. Feedforward amplification in recurrent networks underlies paradoxical neural coding. *bioRxiv,* 2023–08 (2023).

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

71. Kuzmina, E., Kriukov, D. & Lebedev, M. *Further Observations on the Rotational Structure in Neural Data* in *2022 6th Scientific School Dynamics of Complex Networks and their Applications (DCNA)* (2022), 168–171.

72. Lebedev, M. A., Ninenko, I. & Ossadtchi, A. Rotational dynamics versus sequence-like responses. *bioRxiv,* 2020–09 (2020).

73. Lebedev, M. A. *et al.* Analysis of neuronal ensemble activity reveals the pitfalls and shortcomings of rotation dynamics. *Scientific Reports* **9,** 18978 (2019).

74. Jiang, H., Bu, X., Zheng, Z., Pan, X. & Chen, Y. Internal Dynamics Interact with Proprioceptive Feedback During Movement Execution in an RNN Model of Motor Cortex. *bioRxiv,* 2023–12 (2023).

75. Willett, F. R. *et al.* Feedback control policies employed by people using intracortical brain–computer interfaces. *Journal of Neural Engineering* **14,** 016001 (2016).

76. Sorrell, E., Rule, M. E. & O'Leary, T. Brain–machine interfaces: Closed-loop control in an adaptive system. *Annual Review of Control, Robotics, and Autonomous Systems* **4,** 167–189 (2021).

77. Clancy, K. B., Koralek, A. C., Costa, R. M., Feldman, D. E. & Carmena, J. M. Volitional modulation of optically recorded calcium signals during neuroprosthetic learning. *Nature neuroscience* **17,** 807–809 (2014).

78. Jeon, B. B., Fuchs, T., Chase, S. M. & Kuhlman, S. J. Existing function in primary visual cortex is not perturbed by new skill acquisition of a non-matched sensory task. *Nature communications* **13,** 3638 (2022).

79. Donchin, O. *et al.* Cerebellar regions involved in adaptation to force field and visuomotor perturbation. *Journal of neurophysiology* **107,** 134–147 (2012).

80. Taylor, J. A. & Ivry, R. B. Cerebellar and prefrontal cortex contributions to adaptation, strategies, and reinforcement learning. *Progress in brain research* **210,** 217–253 (2014).

81. 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).

82. Baizer, J. & Glickstein, M. Proceedings: Role of cerebellum in prism adaptation. *The Journal of Physiology* **236,** 34P–35P (1974).

83. 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).

84. Morton, S. M. & Bastian, A. J. Prism adaptation during walking generalizes to reaching and requires the cerebellum. *Journal of neurophysiology* **92,** 2497–2509 (2004).

85. 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).

86. Shadmehr, R., Smith, M. A. & Krakauer, J. W. Error correction, sensory prediction, and adaptation in motor control. *Annual review of neuroscience* **33,** 89–108 (2010).

87. Markov, D. A., Petrucco, L., Kist, A. M. & Portugues, R. A cerebellar internal model calibrates a feedback controller involved in sensorimotor control. *Nature Communications* **12,** 6694 (2021).

88. Desmurget, M. & Grafton, S. Forward modeling allows feedback control for fast reaching movements. *Trends in cognitive sciences* **4,** 423–431 (2000).

89. Bouvier, G. *et al.* Cerebellar learning using perturbations. *Elife* **7** (2018).

90. Albert, S. T. & Shadmehr, R. Estimating properties of the fast and slow adaptive processes during sensorimotor adaptation. *Journal of neurophysiology* **119,** 1367–1393 (2018).

91. Pemberton, J., Chadderton, P. & Costa, R. P. Cerebellar-driven cortical dynamics enable task acquisition, switching and consolidation. *bioRxiv,* 2022–11 (2022).

92. 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,* 2022–10 (2022).

93. Boven, E., Pemberton, J., Chadderton, P., Apps, R. & Costa, R. P. Cerebro-cerebellar networks facilitate learning through feedback decoupling. *Nature Communications* **14,** 51 (2023).

94. 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).

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

96. Tsay, J. S., Haith, A. M., Ivry, R. B. & Kim, H. E. Interactions between sensory prediction error and task error during implicit motor learning. *PLoS computational biology* **18,** e1010005 (2022).

97. Albert, S. T. *et al.* Competition between parallel sensorimotor learning systems. *elife* **11,** e65361 (2022).

98. Kim, H. E., Parvin, D. E. & Ivry, R. B. The influence of task outcome on implicit motor learning. *Elife* **8,** e39882 (2019).

99. Huang, V. S., Haith, A., Mazzoni, P. & Krakauer, J. W. Rethinking motor learning and savings in adaptation paradigms: model-free memory for successful actions combines with internal models. *Neuron* **70,** 787–801 (2011).

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

101. Feulner, B. *et al.* Small, correlated changes in synaptic connectivity may facilitate rapid motor learning. *Nature communications* **13,** 5163 (2022).

102. Linderman, S. *et al. Bayesian learning and inference in recurrent switching linear dynamical systems* in *Artificial Intelligence and Statistics* (2017), 914–922.