NEURAL MANIFOLDS AND GRADIENT-BASED ADAPTATION IN NEURAL-INTERFACE TASKS

ALEXANDRE PAYEUR[1,2], AMY L. ORSBORN[3,4,5] AND GUILLAUME LAJOIE[1,2]

[1] Department of Mathematics and Statistics, Université de Montréal, Montréal, Québec, Canada

[2] Mila - Québec Artificial Intelligence Institute, Montréal, Québec, Canada

[3] Department of Bioengineering, University of Washington, Seattle, Washington, USA

[4] Department of Electrical and Computer Engineering, University of Washington, Seattle, Washington, USA

[5] Washington National Primate Research Center, Seattle, Washington, USA

Abstract. Neural activity tends to reside on manifolds whose dimension is much lower than the dimension of the whole neural state space. Experiments using brain-computer interfaces with microelectrode arrays implanted in the motor cortex of nonhuman primates tested the hypothesis that external perturbations should produce different adaptation strategies depending on how "aligned" the perturbation is with respect to a pre-existing intrinsic manifold. On the one hand, perturbations within the manifold (WM) evoked fast reassociations of existing patterns for rapid adaptation. On the other hand, perturbations outside the manifold (OM) triggered the slow emergence of new neural patterns underlying a much slower—and, without adequate training protocols, inconsistent or virtually impossible—adaptation. This suggests that the time scale and the overall difficulty of the brain to adapt depend fundamentally on the structure of neural activity. Here, we used a simplified static Gaussian model to show that gradient-descent learning could explain the differences between adaptation to WM and OM perturbations. For small learning rates, we found that the adaptation speeds were different but the model eventually adapted to both perturbations. Moreover, sufficiently large learning rates could entirely prohibit adaptation to OM perturbations while preserving adaptation to WM perturbations, in agreement with experiments. Adopting an incremental training protocol, as has been done in experiments, permitted a swift recovery of a full adaptation in the cases where OM perturbations were previously impossible to relearn. Finally, we also found that gradient descent was compatible with the reassociation mechanism on short adaptation time scales. Since gradient descent has many biologically plausible variants, our findings thus establish gradient-based learning as a plausible mechanism for adaptation under network-level constraints, with a central role for the learning rate.

1. Introduction

The firing rates of an ensemble of N neurons naturally live in a N-dimensional neural space, with one dimension per neuron. The set of neural responses produced during the execution of a well-defined task define an intrinsic manifold embedded in this neural space [1, 2]. The manifold's dimensionality depends on task complexity, with simpler tasks leading to lower-dimensional manifolds [3, 4]. A low-dimensional manifold signifies the existence of a comparatively small set of dominant patterns of co-variability among neurons. In general, one could argue that structured neural activity is a reflexion of the structure in the World [5]. This idea manifests itself in the existence of a continuous attractor encoding eye position [6], of object manifolds in deep convolutional networks modeling object recognition [7], of low-dimensional cognitive maps in hippocampal networks [8] and of manifolds for the control of movements [1]. In turn, structured activity is thought to bear on the efficient encoding of sensory information [9], the few-shot learning of concepts [10] and generalization in timing tasks [11]. However, probing the multifaceted implications of neural manifolds can be difficult without proper control over the behavioral degrees of freedom. Brain-computer interfaces (BCIs) provide such control by having a direct, causal link between network activity and the behavioral output [12, 13]. Perturbing this link forces the brain to correct "misalignments" between neural representations and behaviors, providing information about the relationship between manifold structure and adaptation.

In a seminal work, Sadtler and co-workers [14] reasoned that the difficulty with which new behaviors can be learned might depend on the underlying co-variability structure of neural activity. Learning new behaviors that require activity patterns outside the pre-existing intrinsic manifold would be more difficult than learning behaviors that can repurpose the existing co-variability structure. They tested this hypothesis with a BCI that decoded neural activity from this intrinsic manifold to control a computer cursor and solve a center-out reaching task. It was possible to perturb the mapping and force the control space to lie either within the manifold (within-manifold perturbation [WMP]) or outside the manifold (outside-manifold perturbation [OMP]). They found that the subject could adapt rapidly, within a day, to WMPs but that adaptation to OMPs was not possible during the same time frame. Subsequent work [15] has shown that OMPs need days of training using an incremental protocol (wherein the "size" of the perturbation is slowly increased with time) to adapt. Without the incremental protocol, adaptation to OMPs is inconsistent or virtually impossible, where virtually impossible means almost no improvement on the time scale of a multi-day experiment. It has been argued that adaptation to WMPs relies on a fast reassociation of patterns from the intrinsic manifold [16] whereas incremental adaptation to OMPs slowly generates new neural patterns [15], hence the difference of time scales between the two types of perturbation. Therefore, the mechanisms responsible for adaptation to WMPs would not change the co-variability structure significantly, as opposed to the mechanisms responsible for adaptation to OMPs.

A few recent theoretical works have attempted to uncover the specific network adaptation mechanisms underlying the within/outside manifold adaptation dichotomy [17, 18, 19]. Wärnberg and Kumar [17] described how the intrinsic manifold can be implemented in functional spiking neural networks and suggested that adaptation to OMPs require larger synaptic changes than adaptation to WMPs, and thus more learning, a claim that was challenged by Ref. [18]. Feulner and Clopath [18] suggested that the disparity between the adaptation to WMPs and OMPs can be explained by a corruption of the feedback error signal driving the learning process underlying adaptation. Although that work reproduced several aspects of the neural constraints on learning in BCI experiments, the learning rule used in their main text is not biologically plausible, adaptation to OMPs remain impossible even when using incremental mappings and reassociation is not truly a feature of their model. Finally, a recent preprint by Humphreys and co-workers [19], whose thesis is closest to ours, suggested that gradient-descent-based learning—which possesses biologically-plausible counterparts [20, 21, 22, 23] and relies on uncorrupted feedback error—could explain both the slower relearning following OMPs and the reassociation mechanism underlying adaptation to WMPs. However, they ignored the inconsistent and often impossible adaptation to OMPs without incremental training, even over long learning time scales, and used a non-recurrent neural networks for their model which is at odds with motocortical circuit features.

In our case, we used a simplified noisy recurrent network model with gradient-descent learning to study the adaptation to WMPs and OMPs. As in other studies, we observed different adaptation time scales for the two types of perturbation, by comparing the loss curves as a function of the weight updates post-perturbation. Using our simple model, we found that the loss gradient differs significantly in magnitude and dynamics when comparing the two types of adaptation. Other signatures of differential adaptation related to alignments in the neural space (among neural and BCI-decoder-related variables) were also obtained. As in [19], with sufficient training and an adequate value for the learning rate hyperparameter, ultimately the network was able to adapt fully to both perturbations. However, we found that increasing the learning rate could either completely prevent adaptation to OMPs or make the adaptation process noisy and unstable, while adaptation to WMPs was unaffected for the most part and simply rescaled over training duration. This is consistent with the difficulty to adapt to OMPs in experiments not using incremental training [15]. For the random seeds showing near-impossible relearning following an OMP, incremental training facilitated adaptation quite significantly. For WMPs, we further found that gradient-descent learning can underlie the reassociation mechanism when the input is high-dimensional and when input plasticity dominates adaptation. Overall, our results suggest that gradient-based learning can explain adaptation under network-level constraints, with the learning rate directly affecting the propensity to adapt to perturbations away from the intrinsic manifold.

2. Results

We designed our model (Fig. 1A, top) to capture salient aspects of typical center-out reaching tasks with BCIs. In these, neural activity is recorded and fed to a decoder in order to drive a cursor on a 2D screen, from a center point to K equidistant targets on a circle. The experimental subject must learn to produce the correct dynamics given the decoder to accomplish the task and receive rewards. Here, we simply trained our network to produce K uniformly distributed unit vectors representing the targets and denoted $\mathbf{d}_k$, with $k = 0, \ldots, K-1$. To show the adaptation

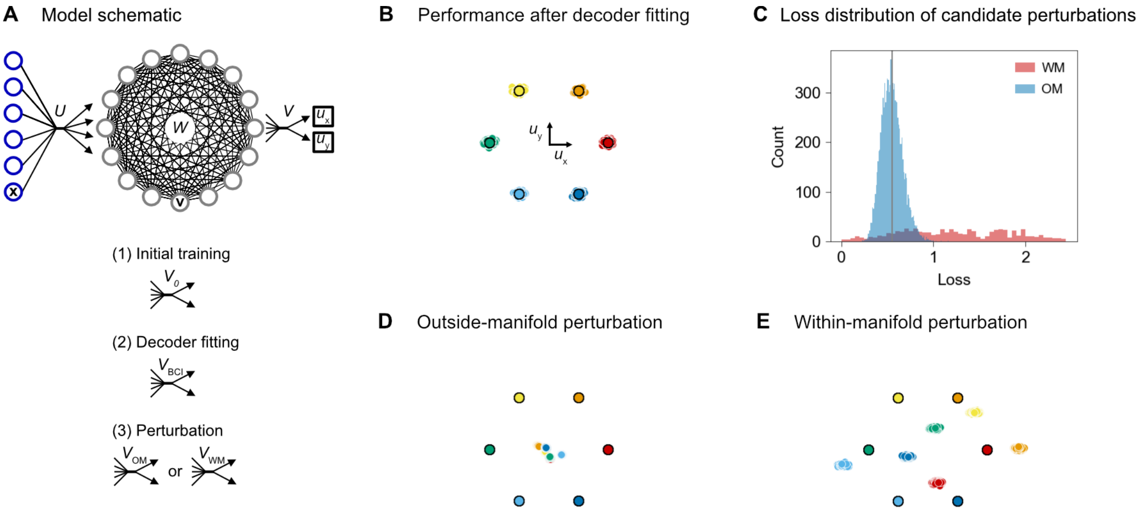

FIGURE 1. *Impact of the perturbations on the model's output before adaptation.* **A** Schematic of the model (top) and the output matrices at different stages of an experiment (bottom). See text for details. **B** Samples drawn from the model's output, $\mathbf{u} = (u_x, u_y)$, after fitting the decoder. Large circles with black edges represent the targets. **C** Distribution of candidate perturbations' losses. Vertical grey line indicates the median of the combined distributions. **D** Sampled outputs of the model before adaptation to the selected outside-manifold perturbation. **E** Same as C for the selected within-manifold perturbation.

phenomena in the most straightforward way, we studied a simple class of network models, which can be construed as the limit of a noisy recurrent neural network close to its fixed point (see Methods) [24]. The recurrent activity $\mathbf{v}$, a vector of size N, was given by

$$\mathbf{v} = (I - W)^{-1}(U\mathbf{x} + \boldsymbol{\xi}), \tag{1}$$

where U and W are the input and recurrent weight matrices, I is the identity matrix, $\boldsymbol{\xi}$ is a private Gaussian noise applied to all recurrent units and $\mathbf{x}$ is an input vector encoding information about the targets, to be described below. As in Ref. [18], the output of the network was determined by a linear readout of the network activity, i.e. $\mathbf{u} = V\mathbf{v}$; we might refer to $\mathbf{u} = (u_x, u_y)$ as a reach or cursor "trajectory" in the Cartesian plane. The only learnable network parameters, U and W, were learned *via* gradient descent on the loss function

$$L \propto \sum_{k=0}^{K-1} \mathbb{E}\left[\|\mathbf{u} - \mathbf{d}_k\|^2 | k\right] \tag{2}$$

which is the sum of the expected mean squared errors between the network's output and each target $\mathbf{d}_k$. The output matrix V adopted one of four possible matrix values (Fig. 1A, bottom). The initial matrix $V = V_0$ was a random Gaussian matrix meant to represent the readout during the experimental calibration blocks [14], prior to determining the *intuitive mapping*. The intuitive mapping, denoted V_{BCI}, was designed by projecting the network activity with $V = V_0$ onto its first M (typically 6–10) principal eigenvectors and then by fitting a linear readout connecting these projections to the output. Finally, the WMP and OMP were applied to V_{BCI}, as described in more detail in the Methods, and gradient descent again adapted the learnable parameters. We stress that V was never learned through gradient descent but rather changed externally. In the following sections, V shall refer to the output matrix for either type of perturbations (Eqs. 20-21).

2.1. **Impact of the perturbations on reach trajectories and losses.** To facilitate comparison with Ref. [18], we first studied the behavior of our model in a simple setting. The input $\mathbf{x}$ was a noiseless 1–of–K encoding—all elements were 0, except for a unique 1 at position k, also called a one-hot encoding—with $K = 6$ targets, the private noise applied to each recurrent unit was low and only the recurrent weight matrix was plastic (see Methods). Figure 1B shows the training outcome with the intuitive mapping V_{BCI}. The resulting outputs $\mathbf{u}$ were sampled from two-dimensional Gaussians mostly centered on the targets with low variance. Of course, higher private noise generates Gaussians with larger spatial variances (not shown). Candidate WMPs and OMPs were then tested by evaluating their effect on the immediate loss (Fig. 1C). As exemplified by the selected OMP (Fig. 1D), the OMPs produced outputs close to the center, resulting in a loss distribution with a small mean and a small variance. WM losses were much more spread out, indicating that this type of perturbation rearranged the outputs more significantly, as shown for the selected WMP in Fig. 1E. Among all the candidate perturbations, the selected WMP and OMP were those whose effect on the immediate loss was closest to the median of the combined empirical distributions (Fig. 1C, grey line). This was done to make sure that adaptation to the WMP and OMP started from a similar loss. Also, given that the median lies among the easiest WMPs, we actually picked a WMP for which the perturbed outputs were not too far from the targets, as in the experiments [16].

2.2. **Adaptation rate and components of the objective function.** Adaptation to both perturbations resulted in end-of-training performances comparable to that obtained with the initial decoder (Fig. 2A). Importantly, this happened at different rates for the WMPs and OMPs, similarly to Ref. [19] which used gradient-descent algorithms. By contrast, in Ref. [18]—which used a modified recursive least-squares algorithm inspired by FORCE learning [25] and innate learning [26]—there were no difference between the two types of perturbations without corruption of the feedback error signals, either at the end of retraining or during retraining (see Supp. Fig. 1 and Appendix A).

To gain further insight into the different adaptation rates for WM and OM adaptations, we resolved the total loss into components with clear meanings. We focused on the following decomposition of the loss (see Methods, Eq. 15)

$$L = \frac{1}{2} + \frac{1}{2}\mathsf{tr}\left\{V(\mathbb{V}[\mathbf{v}] + \bar{\mathbf{v}}\bar{\mathbf{v}}^{\mathsf{T}})V^{\mathsf{T}}\right\} - \frac{1}{K}\sum_{k=0}^{K-1}\mathbf{d}_k^{\mathsf{T}}V\mathbb{E}[\mathbf{v}|k], \tag{3}$$

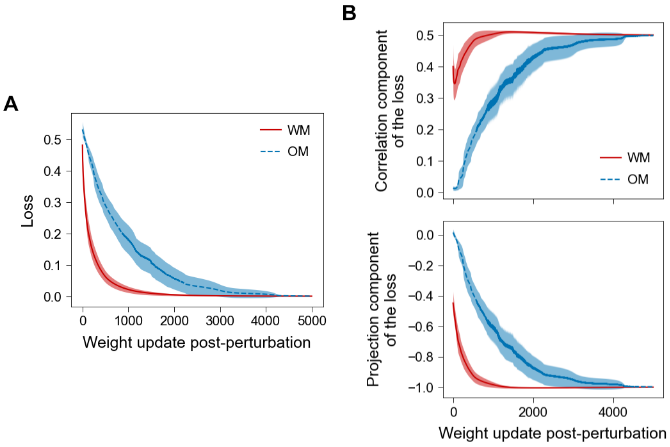

FIGURE 2. *Adaptation to within- and outside-manifold perturbations.* **A** Total loss during adaptation. **B** Correlation (top) and projection (bottom) components of the loss. Note that 1/2 should be added to these components to yield the total loss (Eq. 15). In all panels, we displayed mean $\pm 2 \times$ SEM, for $n = 20$ random seeds.

which resolves the total loss into a contribution due to the total output correlation, denoted L_{corr} (second term), and a negative contribution penalizing any misalignment of the mean output with the target (third term). Right after the perturbation these two components roughly compensated each other for the WMP (Fig. 2B); the only loss remaining comes from the constant first term in Eq. 3. For the OMP, both components started near zero (Fig. 2B), which is expected given the initial effect of the perturbation on the reaches (Fig. 1D). Past the first few hundred updates the correlation component (Fig. 2B, top) increased to reach 1/2 when completely adapted for both types of perturbation, which can be shown to correspond to half the average squared norm of the learned trajectories (see derivation in Appendix C.1). The projection component converged to -1, the size of (minus) the average projection of the trajectories on the targets (Fig. 2B, bottom). The correlation component of the loss can be written as $L_{\text{corr}} \approx L_{\mathbb{V}[\mathbf{v}]} = \frac{1}{2}\sum_{n=1}^{N} \lambda_n \|V\mathbf{e}_n\|^2$, where λ_n and $\mathbf{e}_n$ are the eigenvalues and eigenvectors of the covariance matrix $\mathbb{V}[\mathbf{v}]$ and we neglected the contribution from the global mean activity $\bar{\mathbf{v}}$ (see Appendix C.1 and Supp. Fig. 2). This representation suggests that, to increase the correlation loss, the total covariance can align its eigenvectors to the row space of V and/or adjust its eigenvalues. We thus expect that further insight into the difference in adaptation under WMPs and OMPs would come from analyzing the alignment between $\mathbb{V}[\mathbf{v}]$ and V, which has the advantage of not referring to the targets. This and other signatures of adaptation will be discussed in the next section.

2.3. **Signatures of adaptation to within- and outside-manifold perturbations.** Before adaptation, neural activity mostly resided in the intuitive manifold defined by the leading eigenvectors—put into the rows of a matrix C—of the total covariance matrix. The BCI mapping based on the intrinsic manifold—the intuitive mapping—relies on this matrix C and a linear

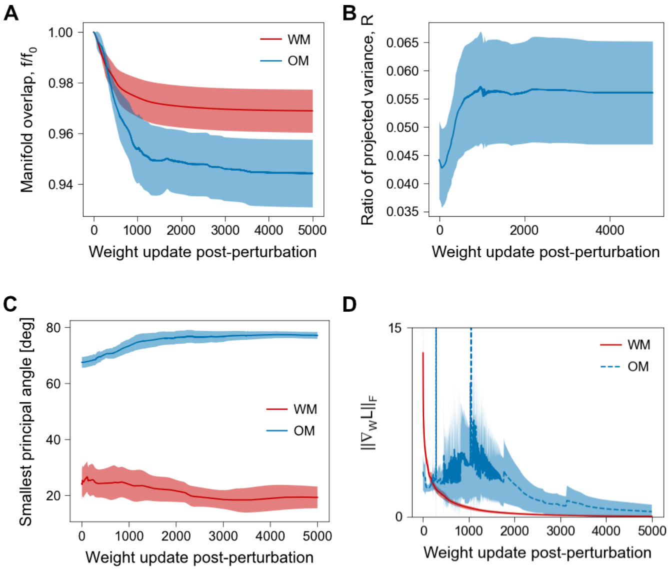

FIGURE 3. *Signatures of adaptation to within- and outside-manifold perturbations.* **A** Manifold overlap. **B** Ratio of projected variance for OM perturbations. **C** Smallest principal angle between the subspaces spanned by $d\mathbb{V}[\mathbf{v}]$ and V^{T}. **D** Frobenius norm of the gradient of the total loss. In all panels, shaded areas represent ±2× SEM, for $n = 20$ random seeds.

decoder D that transforms the projected activity, $C\mathbf{v}$, into cursor trajectories $\mathbf{u} = DC\mathbf{v}$ (see Methods). While relearning the center-out task following the perturbation, the fraction of the total variance within the intuitive manifold (Eq. 26) might change relative to what it was before the perturbation, denoting a learning-induced rearrangement of activity in neural space. The ratio of the fraction of variance explained by the intuitive manifold before and after learning has been called the normalized variance explained [16] or the manifold overlap [18] (Eq. 27). We found that the manifold overlap decreases from its starting value of 1 during adaptation for both types of perturbation, reaching significantly lower values for OMPs compare to WMPs (Fig. 3A). Feulner & Clopath obtained manifold overlaps in the range $\sim 5 - 20\%$ for OMPs and $\sim 40 - 75\%$ for WMPs [18]. In experiments, the WM manifold overlap was found to be at least 90% in 33/48 of experiments (see supplementary figure 3 in Ref. [16]). To our knowledge, manifold overlaps have not been computed for OMPs in experiments yet. But we do know from Ref. [15], which focused on OMPs, that adaptation to OMPs combines WM strategies and OM strategies, the first of which preserving the covariance structure and the second modifying it. In accordance with this view, our results suggest that, under gradient-based learning, a good portion of the adaptation strategy to OMPs relies on the pre-existing activity structure, but less so than WMPs.

We further characterized adaptation to OMPs by computing the ratio of variance projected onto the perturbed manifold CP_{OM} relative to C (Eq. 28). While this ratio did increase during early adaptation, its value stayed small throughout adaptation (Fig. 3B). Again, this suggests that adaptation to the OMP is not mediated by a strong realignment of the variance along the subspace spanned by CP_{OM}. This was further confirmed by calculating the smallest principal angle [27] between the subspace spanned by the columns of $V_{\mathsf{OM}}^{\mathsf{T}}$ and the column space of the change in covariance throughout adaptation. During adaptation to OMPs, the differential covariance subspace was $\sim 70° - 80°$ from the $V_{\mathsf{OM}}^{\mathsf{T}}$ subspace (Fig. 3C). For the WMP, these covariance updates were much more aligned with $V_{\mathsf{WM}}^{\mathsf{T}}$, with angles $\sim 20° - 30°$.

We have shown that the network model adapts more rapidly on average to WMPs compared to OMPs. Since changes to the loss depend on the gradient, we expect the latter to be smaller for OM adaptation. Figure 3D shows that the Frobenius norm of the gradient of the total loss with respect to W, $\|\nabla_W L\|_F \triangleq \sqrt{\mathsf{tr}\left\{\nabla_W L \nabla_W L^{\mathsf{T}}\right\}}$, is smaller during adaptation to the OMP in the first few hundred weight updates. These smaller gradients are conducive to a slower decrease of the loss. The norm of the gradient is also a lot noisier for OM adaptation, an observation that will be addressed in more detail in the next section.

2.4. **Sensitivity to the learning rate.** Given sufficient training time the network did adapt to both types of perturbation (Fig 2A). In experiments, adaptation to OMPs is actually inconsistent and oftentimes virtually impossible without a progressive, incremental training regimen [15]. While investigating the effect of the learning rate hyperparameter η_W on adaptation, we found that relearning following an OMP seemed to be more sensitive to an increase of the learning rate compared to relearning following a WMP (Fig. 4). This sensitivity manifested itself in two different ways: in some cases (8/20 seeds), learning seemed impossible (or at least improbable) on the training interval we used (Fig. 4A). In other cases (10/20 seeds), the loss curve became noisy or oscillatory during learning (Fig. 4B). Two seeds showed little effect (Fig. 4C). While not uncommon in general during gradient-based learning [28], we noted that these instabilities and impossibilities appeared preferentially during adaptation to OMPs. Conversely, WM adaptation was mostly unaffected (13/20 seeds) by the learning rate increase in the tested range, with the loss curves being merely rescaled along the x-axis (Fig. 4A-C). In other cases (7/20), small oscillations appeared during learning. The learning rate had no effect on the distribution of weight changes across the WM adaptation period, but it had a large impact on the weight changes due to the OM adaptation when learning was impossible (Fig. 4D and Supp. Fig. 4). The loss landscape for OM adaptation might be more irregular compared to WM's in general, and large learning rates might produce transient increases of the loss (as in Fig. 4A, right) yielding large weight changes. These results suggest that improper tuning of the learning rate could explain the difficulty to adapt to challenging perturbations.

2.5. **Incremental training.** To mitigate the negative effect of high learning rates on adaptation to OMPs, specifically in the cases where adaptation becomes impossible (Fig. 5A), we implemented an incremental training protocol similar to that in Ref. [15]. During relearning, the perturbed readout mapping was "rotated" with respect to the intuitive mapping in steps,

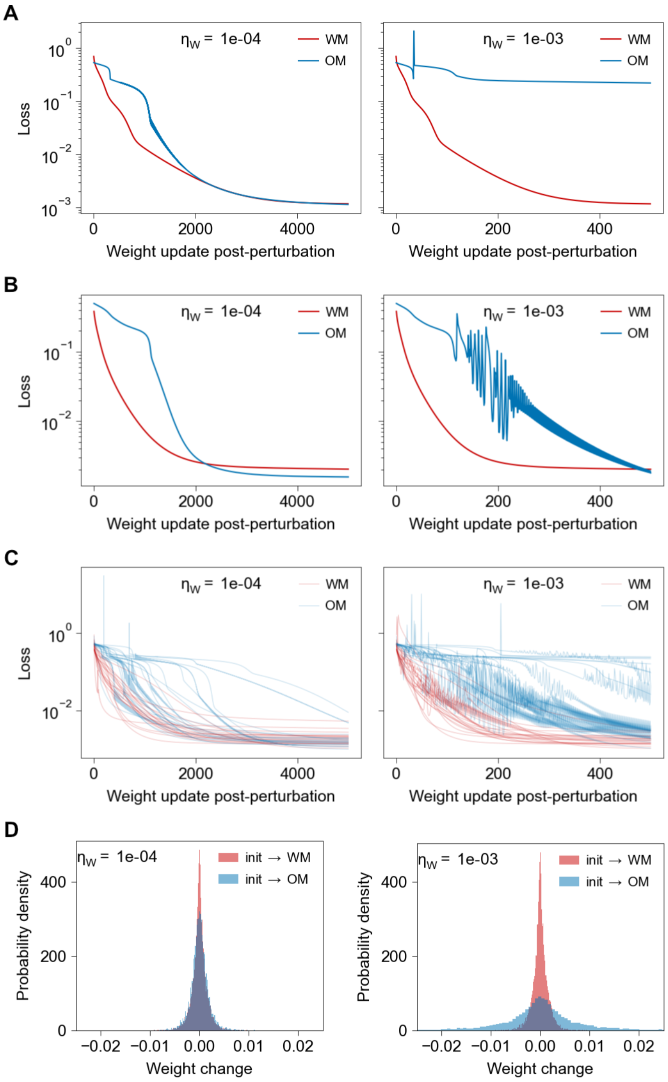

FIGURE 4. *Sensitivity to the learning rate.* **A** Loss during adaptation for two learning rates for an example seed showing impossible OM adaptation (right). Note the different scales for the x-axis for the two panels. **B** Same as A, but for a seed exemplifying noisy/oscillatory learning at high learning rate. **C** Same as A and B, but showing all 20 seeds. **D** Distribution of weight changes across adaptation for the seed of panel A. init → WM signifies the change between W before the perturbation and W after adaptation to the WMP, and similarly for init → OM. The data used for plotting these histograms where the 100^2 weight changes for each condition.

making the perturbed mapping progressively farther from the intuitive mapping and closer to the selected OMP. Although incremental training could produce noisy learning curves (Fig. 5B), its usage systematically helped adaptation significantly (Fig. 5C). In experiments, the incremental

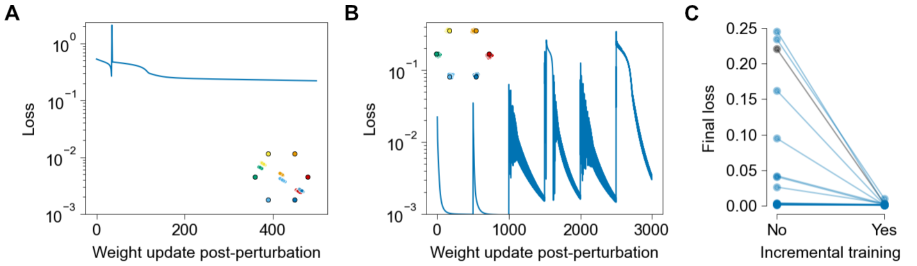

FIGURE 5. *Incremental training.* **A** Loss for a single seed with impossible OM adaptation (same as in Fig. 4A, right). Inset shows network outputs at the end of relearning. **B** Same as A, but with incremental training. **C** Final loss for all seeds, with and without incremental training. Grey line corresponds to the example in A and B.

training protocol did not always lead to successful adaptation [15], but that could be due to the specific selection of the OMP, to varying engagement from the animal and other factors that might affect the learning rate.

2.6. **High-dimensional input encoding and adaptation strategies to within-manifold perturbations.** Thus far, the input has been a static 1–of–6 encoding and plasticity was restricted to the recurrent weights. To understand the impact of the plasticity of the input matrix U on adaptation, we next used a higher-dimensional input and made U learnable. Plasticity of the afferent pathway(s) to the recurrent network has been hypothesized to underlie the reassociation mechanism evoked by WMPs [16] and rapid learning under motor perturbations [29]. The input size was larger than the network size and its dimension (assessed using the participation ratio) was ~ 3 times the dimension of the recurrent activity to mimic the dorsal premotor cortex and other afferent regions being of larger dimension than the primary motor cortex [29]. This was achieved by using a low-rank decomposition of the input covariances $\Sigma_x^{(k)}$; the rank was set to $I/10$ (with $I = 200$ the input size) to embody the observation that sensory encoding has lower dimensionality than the full neural space [9]. The target-conditioned mean inputs $\boldsymbol{\mu}_k$ were random unit vectors (see Methods).

We compared the WM adaptation strategies of this high-dimensional-input network with those used by the network of the preceding sections (Fig. 6). To do so, we followed Ref. [16] and computed the amount of variability projected onto the row space of the decoder D, before and after the WMP. The WMP amounts to permuting the columns of D (see Methods) and we reiterate that the decoder remained fixed once the perturbation had been applied. We considered two cases for the high-dimensional input: one with plasticity only in the input weights U and another with plasticity in both the recurrent and input weights. In both high-dimensional-input cases, the model displayed differences in adaptation to WMPs and OMPs, similar to the previous low-dimensional-input network setup (*cf.* Supp. Fig. 3). However, the effect was most pronounced and easier to achieve when only U was plastic. This observation supports

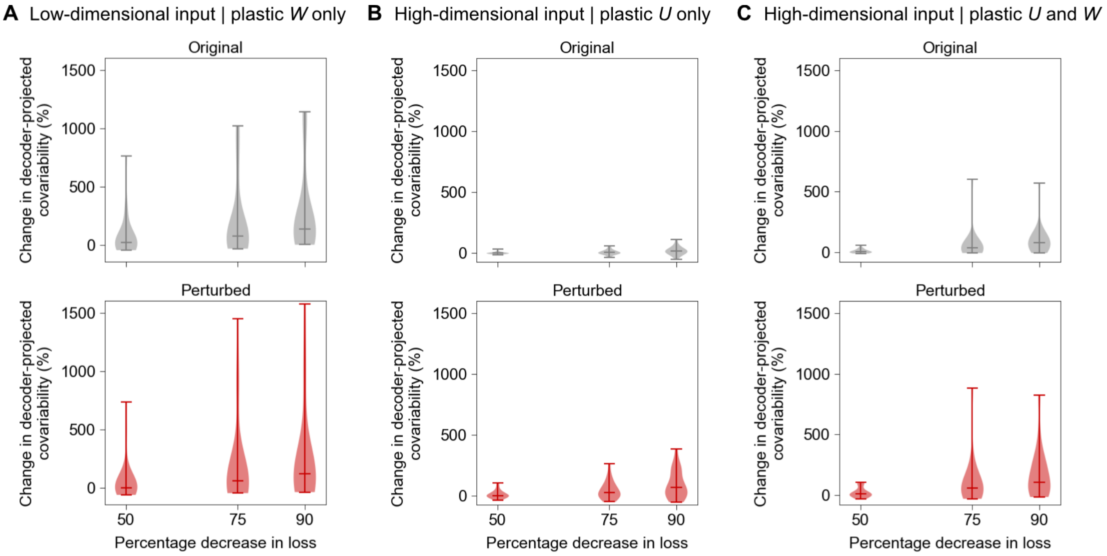

FIGURE 6. *Adaptation strategies to within-manifold perturbations.* The change in decoder-projected co-variability is plotted for the low-dimensional input (**A**) and high-dimensional inputs with plasticity in the input matrix only (**B**) or in both the input and recurrent weight matrices (**C**). The top panels represent the projection of the principal covariance onto the row space of the original (pre-perturbation) decoder, whereas the bottom panels represent the projection onto the perturbed mapping. To allow comparison across setups, the x-axis represents the percentage decrease with respect to the loss right after the WM perturbation, corresponding to different stages of adaptation. Bars in the violin plots are medians over 20 seeds.

the thesis that fast adaptation to WMPs depends on the plasticity of the afferent pathways to primary motor cortex, with gradient-following plasticity playing the lead role here as well. The adaptation strategies soon after the perturbation were different between the low-dimensional- and high-dimensional-input setups, especially concerning the projections onto the original decoder (Fig. 6, top). For the high-dimensional input, the change in projected co-variability onto the original decoder stayed close to zero both with and without a plastic W, whereas there was an increase in both the median and the spread of this projection for the low-dimensional input. For the perturbed mapping, only the variance of the projections were obviously different between the setups (Fig. 6A-C, bottom). According to figure 5c in Ref. [16], adaptation *via* reassociation stipulates that the changes in decoder-projected co-variability would be close to zero for both the original and the perturbed mappings, consistent with a repurposing of the preexisting neural repertoire. In our case, this was mostly observed early after the perturbation, more convincingly so for the high-dimensional input (for example, in Fig. 6B, the median for 75% decrease of the loss was 24.5%). As adaptation progresses, the high-dimensional setup follows an adaptation strategy that becomes more compatible with realignment. For the low-dimensional setup, the adaptation strategy becomes a combination of realignment and rescaling, albeit noisily.

3. Discussion

In this paper, we explored theoretically the hypothesis that a learning algorithm based on gradient descent could explain the experimentally observed relative difficulty to adapt to perturbations which force a control space outside the intrinsic neural manifold. We have shown that gradient descent produces learning curves with different rates of decrease for within- and outside-manifold perturbations (Fig. 2) and studied signatures of this difference, with a focus on the structure of the covariance and its relationship with the control mapping (Fig. 3). We found that increasing the learning rate parameter had a detrimental effect specifically on adaptation to outside-manifold perturbations, rendering the learning process noisy or even impossible (Fig. 4). Incremental training could help alleviate this effect (Fig. 5). Finally, we also studied the role played by the plasticity and dimensionality of the input on the adaptation to within-manifold perturbations, finding support for a gradient-based reassociation mechanism early in training (Fig. 6).

Gradient-based learning offers a parsimonious explanation of in- and out-manifold relearning. It means that the pre-existing neural covariance structure opposes a certain rigidity to new behavioral requirements, even with exact information about the outcome of the task and its associated error. In contrast, in other works not relying on gradient descent, the slower [17] or impossible [18] adaptation to OMPs is due to larger weight changes [17] or to corrupted error information [18]. With gradient descent, weight changes were small for WM adaptation irrespective of the learning rate (Fig. 4D); for OM adaptation, weight changes get significantly larger (i.e. distributionally more spread out) when adaptation is impossible at a high learning rate. We have to differentiate these large weight changes from those in the work of Wärnberg and Kumar [17]. In the latter, the authors quantified the required cumulative changes to undo the perturbation, but afterwards learning can proceed in a stable fashion. In our case, large weight changes were directly caused by the difficulty of relearning the task following an OMP.

In the literature, the real benefits of incremental training for OM adaptation are not completely clear. In Ref. [15], it is stated that “Multiday exposure to an OMP with no incremental training led to inconsistent learning.” In that paper, the amount of learning was quantified as the reward rate (the number of successful trials per unit time) during adaptation relative to the reward rate with the intuitive mapping. The amount of learning across days with incremental training was larger than without incremental training, with $p = 0.052$ (close to the typical significance level), but the number of adaptation days without incremental training was smaller than the number of adaptation days with incremental training (*cf.* their supplementary figure S1G). Would the amount of learning still be marginally larger for adaptation under an incremental training protocol when the number of adaptation days are identical (or not too different)? Also, incremental training was not systematically beneficial, sometimes leading to a decrease in performance or to very small improvements. This could be due to multiple factors not directly connected to this protocol, like the inherent difficulty of the chosen OMP. In our case, incremental training was helpful for all seeds (Fig. 5), irrespective of whether adaptation was possible without it. But this was true only when the last increment was broken down into two smaller increments (see Methods, Ref. [15] and Supp. Fig. 5). Our model suggests that increasing the number of increments could help with learning difficult BCI mappings.

Learning by reassociation in the context of in-manifold perturbations [16] is probably related to learning by reaiming in the context of visuomotor rotations [30]. Visuomotor rotations (VRs), as external perturbations probing motor learning mechanisms, are used both in BCI experiments and in experiments involving the native arm [29]. In the latter case, rapid learning of VRs involved afferent pathways connected to the premotor and primary motor networks [29]. Functional changes to these pathways are unlikely to affect the covariance structure of motor networks, which is reminiscent of the reassociation mechanism. However, small correlated weight changes in local motor circuits have also been implicated in the fast relearning of VRs in theoretical models, since these small correlated changes also have little effect on motocortical co-variability [31]. Our simplified model supports the view that upstream plasticity mediated by gradient-based learning underlies a reassociation mechanism. It also corroborates the observation [31] that local plasticity—in W—tends to affect covariance more significantly when task difficulty increases (assuming that WMPs, by scrambling the targets, are more difficulty to relearn than uniform rotations, as in VRs) and when learning is allowed to occur on longer time periods [16]. Although high-dimensional inputs seemed necessary for reassociation (Fig. 6), we did not explore the necessity of output-null dimensions [32]; we shall leave such study for future work.

We used a simple model to show the adaptation phenomena in the most straightforward way, but using such a simplistic model comes with a number of limitations. Trivial ones are the absence of dynamics, linearity and the Gaussian noise. More important is the fact that our learning rule uses weight transport, which is not biologically plausible [33] (although algorithms exist to circumvent this problem [34, 35, 36]), and includes nonlocal interactions, which again can be partially alleviated in recurrent networks [22, 21]. Alternatively, one can utilize gradient-following algorithms which approximate gradient descent without computing the gradient, as in the REINFORCE algorithm [20]. The latter approach has been used in [19] and compared with the error backpropagation algorithm, yielding similar results. Other limitations are related to the comparison of adaptation metrics between models and experiments. It is difficult to compare the time scale of learning through gradient descent, which is measured in abstract weight updates, to that observed in experiments, which is measured in trials. What corresponds to a day of training, for gradient descent? In Refs. [14, 16], which focused on a single day, adaptation was studied across $\sim 400 - 900$ trials (depending on the monkey and experiment). During that time, performance improved for WMPs but it did not recover its pre-perturbation level, especially in terms of the target acquisition time (a full second of difference). How one should relate both this performance metric and the success rate to the loss in the context of gradient descent? These limitations related to performance metrics are not a prerogative of our model as most models and learning algorithms face similar issues. Future neurotheoretical works should try to adopt the trial and reward structure of center-out BCI tasks to facilitate model-experiment comparisons. Finally, although the learning rate played an important role in adaptation (especially the lack thereof), this parameter possesses no clear-cut biological basis. Neuromodulatory centers innervating cortical circuits are routinely ascribed the role of modulating and controlling different aspects of cortical learning [37]. However, figuring out whether an improperly tuned learning rate can embody some of these neuromodulatory effects will require further research.

4. Conclusion

To conclude, we have shown that gradient-descent learning in a simple BCI model concurs with experimental works suggesting that there could be an inherent difficulty in learning sensorimotor mappings that lies outside the existing neural repertoire. Whether similar difficulties could also underlie the so-called BCI illiteracy [38] remains to be explored. The work from Oby *et al.* [15] suggested that incremental training—a form of curriculum learning [39]—might be a necessary ingredient in a protocol aiming at facilitating the emergence of new neural patterns. In the BCI literature, a number of closed-loop decoder adaptation methods, which consist in adapting the decoder to the emerging neural representations, have been utilized to accelerate BCI skill acquisition [40, 41]. If indeed gradient-descent-based learning is involved in motor learning, then a straightforward extension of our work could also shed light on the mechanisms whereby decoder adaptation facilitates BCI learning, and even helps design new methods to optimize BCI control.

5. Methods

5.1. **Network model.** The starting point of our model was the following discrete-time recurrent network dynamics

$$\begin{aligned}\mathbf{v}_{t+1} &= W\boldsymbol{\varphi}(\mathbf{v}_t) + U\mathbf{x} + \boldsymbol{\xi}_t \\ \mathbf{u}_t &= V\mathbf{v}_t,\end{aligned} \tag{4}$$

where $\mathbf{v}_t \in \mathbb{R}^N$ is the recurrent activity and $\mathbf{u}_t \in \mathbb{R}^2$ is the readout at time t. Matrices W, U and V are the recurrent, input and output matrices, respectively. The noise $\boldsymbol{\xi}_t$ was Gaussian and uncorrelated across units and across time, with diagonal covariance matrix $\Sigma_\xi = \sigma_\xi^2 I$; it modeled neuronal noise sources. The input $\mathbf{x} \in \mathbb{R}^I$ was modeled as a time-independent mixture of Gaussians with means $\boldsymbol{\mu}_k$ and covariance matrices $\Sigma_x^{(k)}$, mimicking afferent inputs encoding task requirements:

$$p(\mathbf{x}) = \sum_{k=0}^{K-1} \pi_k p(\mathbf{x}|k) = \sum_{k=0}^{K-1} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_x^{(k)}). \tag{5}$$

The random vectors $\boldsymbol{\xi}_t$ and $\mathbf{x}$ were independent. The mixing coefficients obeyed $\sum_k \pi_k = 1$, here with $\pi_k = 1/K$, for all k. To simulate this model, one would select a target k at random, draw a Gaussian sample from $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_x^{(k)})$ and then proceed with the numerical integration. (Note that the input could also be made time-dependent by drawing a different sample from the selected Gaussian component independently at each time step; time independence only serves to simplify the treatment in the next paragraph.)

To circumvent the complex nonlinear dynamics of this model and streamline the modeling of the BCI phenomena under study, we assumed that the activation function $\varphi(\cdot)$ was linear and we focused on the (noisy) fixed-point dynamics produced by Eq. 4. The solution to the linear dynamics with initial condition $\mathbf{v}_0$ is

$$\mathbf{v}_t = W^t\mathbf{v}_0 + (I - W)^{-1}(I - W^t)U\mathbf{x} + \sum_{k=0}^{t-1} W^{t-1-k}\boldsymbol{\xi}_k.$$

The last term has zero mean and covariance $(I - W)^{-1}(I - W^t)\Sigma_\xi(I - W^t)^\mathsf{T}(I - W^\mathsf{T})^{-1}$. If the maximum absolute eigenvalue of W is less than 1, then $W^t \to 0$ as $t \to \infty$. We get that $\mathbf{v}_t$'s law is asymptotically Gaussian and can be described by

$$\mathbf{v} = (I - W)^{-1}(U\mathbf{x} + \boldsymbol{\xi}), \tag{6}$$

assuming of course that $I - W$ is invertible. The private noise $\boldsymbol{\xi}$ is the static counterpart of $\boldsymbol{\xi}_t$. Since every variable is Gaussian, we have

$$p(\mathbf{v}) = \sum_k \pi_k p(\mathbf{v}|k) \tag{7}$$

where $p(\mathbf{v}|k)$ is Gaussian with mean $\mathbb{E}[\mathbf{v}|k]$ and covariance $\mathbb{V}[\mathbf{v}|k]$ given by

$$\mathbb{E}[\mathbf{v}|k] = (I - W)^{-1}U\boldsymbol{\mu}_k \tag{8}$$

$$\mathbb{V}[\mathbf{v}|k] = (I - W)^{-1}\left[U\Sigma_x^{(k)}U^\mathsf{T} + \Sigma_\xi\right](I - W^\mathsf{T})^{-1}. \tag{9}$$

We assumed that the dimensions of the output matrix V were such that $2 \ll N$, which is typically the case in BCI experiments using microelectrode arrays ($\sim 10 - 100$ electrodes). The adaptable parameters were W and U. Elements of the output matrix V were fixed to random values before the network was trained for the first time (*cf.* section 5.3). After this initial training, the output matrix was replaced by a (perturbable) BCI decoder to be described below. We stress that V was not continuously adaptable like U and W: it represented a rigid mapping between the recurrent activity $\mathbf{v}$ and the readout $\mathbf{u}$ that can be changed externally (say, by the experimenter).

5.2. **Task.** The objective was to produce unit vectors of the form

$$\mathbf{d}_k = [\cos(2\pi k/K), \sin(2\pi k/K)]^\mathsf{T}, \qquad k = 0, \ldots, K - 1.$$

The objective function L to minimize can be written

$$L = \frac{1}{2}\sum_k \pi_k L_k$$

with the target-specific loss

$$L_k = \mathbb{E}\left[\|\mathbf{u} - \mathbf{d}_k\|^2 | k\right].$$

We can rewrite the objective function as (*cf.* appendix C.2)

$$L = \frac{1}{2}\sum_k \pi_k \mathsf{tr}\left\{V\mathbb{V}[\mathbf{v}|k]V^\mathsf{T}\right\} + \frac{1}{2}\sum_k \pi_k \|V\mathbb{E}[\mathbf{v}|k] - \mathbf{d}_k\|^2 \tag{10}$$

to make explicit the intuition that the network's goal was to reduce the average target-conditioned output variance and the distance between the target-conditioned mean output and $\mathbf{d}_k$.

Another expression for the objective function makes use of the total covariance matrix $\mathbb{V}[\mathbf{v}]$. Using the law of total variance, we have

$$\mathbb{V}[\mathbf{v}] = \mathbb{E}\{\mathbb{V}[\mathbf{v}|k]\} + \mathbb{V}\{\mathbb{E}[\mathbf{v}|k]\} = \sum_k \pi_k \mathbb{V}[\mathbf{v}|k] + \sum_k \pi_k (\mathbb{E}[\mathbf{v}|k] - \bar{\mathbf{v}})(\mathbb{E}[\mathbf{v}|k] - \bar{\mathbf{v}})^\mathsf{T} \tag{11}$$

where the global mean activity $\bar{\mathbf{v}}$ is

$$\bar{\mathbf{v}} \triangleq \mathbb{E}[\mathbf{v}] = \mathbb{E}[\mathbb{E}[\mathbf{v}|k]] = (I - W)^{-1}U\bar{\mathbf{x}} \tag{12}$$

with the global mean input

$$\bar{\mathbf{x}} \triangleq \mathbb{E}[\mathbf{x}] = \sum_k \pi_k \boldsymbol{\mu}_k. \tag{13}$$

Multiplying Eq. 11 to the left by V, to the right by V^T and taking the trace yields (*cf.* appendix C.3)

$$\mathsf{tr}\left\{V\mathbb{V}[\mathbf{v}]V^\mathsf{T}\right\} = 2L - \mathsf{tr}\left\{V\bar{\mathbf{v}}\bar{\mathbf{v}}^\mathsf{T}V\right\} - 1 + 2\sum_k \pi_k \mathbf{d}_k^\mathsf{T} V\mathbb{E}[\mathbf{v}|k] \tag{14}$$

and therefore

$$L = \frac{1}{2} + \frac{1}{2}\mathsf{tr}\left\{V(\mathbb{V}[\mathbf{v}] + \bar{\mathbf{v}}\bar{\mathbf{v}}^\mathsf{T})V^\mathsf{T}\right\} - \sum_k \pi_k \mathbf{d}_k^\mathsf{T} V\mathbb{E}[\mathbf{v}|k]. \tag{15}$$

This expression means that the network will try to minimize the total correlation matrix $\mathbb{V}[\mathbf{v}] + \bar{\mathbf{v}}\bar{\mathbf{v}}^\mathsf{T}$ (as "seen" from the output) and maximize the average overlap between the target and the target-conditioned expected output.

5.3. **Output matrix and BCI decoder.** The output matrix V adopted one of four possible matrix values. As mentioned above, V was a random matrix initially [18]. For clarity, this matrix will be denoted as V_0. For instance, in section 2.1, we define

$$[V_0]_{ij} \sim \mathcal{N}(0; 1) \tag{16}$$

and then multiply it by $0.07/\|V_0\|_F$, similarly to Ref. [18]. The symbol $\|\cdot\|_F$ denotes the Frobenius norm. The network was then trained to solve the task with the learning algorithm described in section 5.5 below. At the end of this initial training, neural activity had variance $\mathbb{V}[\mathbf{v}]_0$ and global mean $\bar{\mathbf{v}}_0$, where the subscript $_0$ refers to using V_0 as the output.

Next, following Ref. [18], we designed a BCI decoder which consisted in a composition of two linear transformations D and C so that the new output matrix was

$$V_{\mathsf{BCI}} = DC. \tag{17}$$

The $M \times N$ matrix C, where M is the dimension of the linear manifold, projected the network activity onto the first M principal eigenvectors of $\mathbb{V}[\mathbf{v}]_0$; the rows of C are the transpose of these eigenvectors. The $2 \times M$ matrix D was a linear decoder obtained by solving

$$D = \mathrm{argmin}_{D'} \mathbb{E}\left[\|(D'C - V_0)\mathbf{v}\|^2\right] \tag{18}$$

with solution (*cf.* appendix C.4)

$$D = V_0(\mathbb{V}[\mathbf{v}]_0 + \bar{\mathbf{v}}_0\bar{\mathbf{v}}_0^{\mathsf{T}})C^{\mathsf{T}}[C(\mathbb{V}[\mathbf{v}]_0 + \bar{\mathbf{v}}_0\bar{\mathbf{v}}_0^{\mathsf{T}})C^{\mathsf{T}}]^{-1}. \tag{19}$$

Optionally, we would then briefly retrain the model with $V = V_{\mathsf{BCI}}$ to make sure the network was able to use the decoder, as in the experiments [14].

Then, one of two perturbations was applied. The within-manifold (WM) perturbation [14, 18] yielded

$$V_{\mathsf{WM}} = DP_{\mathsf{WM}}C \tag{20}$$

where P_{WM} is a $M \times M$ permutation matrix. The outside-manifold (OM) perturbation used

$$V_{\mathsf{OM}} = DCP_{\mathsf{OM}} \tag{21}$$

where P_{OM} is a $N \times N$ permutation matrix. Permutation matrices contain only one entry equal to 1 in each row and column. Therefore, the WMP shuffles the rows of C and the perturbation thus scrambles how the projections are fed to D. The OMP shuffles the *columns* of C, thus altering how neural activity contributes to each pattern.

5.4. **Manifold's dimension and permutation selection.** The assigned dimensionality of the intrinsic manifold, M, should be large enough to include the main neural modes [14], but small enough so that the WMP does not become as difficult as the OMP to relearn. The rule of thumb was that the chosen M should represent roughly 99% of the total variance. For the parameters used here, setting $M = 6$ allowed to represent most of the variance and kept the search for the WMPs computationally fast (see below). For different random seeds (different initializations of the network model), the number of principal components to reach 99% might differ (5 ± 0 for the low-dimensional setup of Figs. 1-5 and 6.2 ± 0.3 for the high-dimensional setup of Fig. 6 [mean ± SEM for $n = 20$ seeds]), but M itself had the same value across seeds nevertheless.

There were $M!$ possible WMPs and $N!$ OMPs. Permutations were evaluated by computing their initial impact on the loss (with the pre-adaptation parameter values). Following [18, 14], for each type of perturbation we randomly generated a large number ($\mathsf{min}(M!, 10^4)$ for WMPs and 10^4 for OMPs) of candidate permutations, and then selected among these the permutation generating the loss closest to the median of the combined distributions, to make sure that adaptation to the WMP and OMP started from a similar loss.

5.5. **Learning.** For each training episode (with $V = V_0$ and then with either $V = V_{\mathsf{WM}}$ or $V = V_{\mathsf{OM}}$), we needed to solve

$$W^*, U^* = \text{argmin}_{W,U} L.$$

Gradient descent means that an adaptable parameter X is learned according to

$$\Delta X = -\eta_X \nabla_X L, \tag{22}$$

where η_X is the corresponding learning rate. The gradients of L with respect to W and U are derived in Appendix B. They are

$$\nabla_W L = (I - W^\mathsf{T})^{-1} V^\mathsf{T} \sum_k \pi_k \left[V \mathbb{V}[\mathbf{v}|k] + (V\mathbb{E}[\mathbf{v}|k] - \mathbf{d}_k)\mathbb{E}[\mathbf{v}|k]^\mathsf{T} \right] \tag{23}$$

$$\nabla_U L = (I - W^\mathsf{T})^{-1} V^\mathsf{T} \sum_k \pi_k \left[V(I-W)^{-1} U \Sigma_x^{(k)} + (V\mathbb{E}[\mathbf{v}|k] - \mathbf{d}_k)\boldsymbol{\mu}_k^\mathsf{T} \right] \tag{24}$$

These gradients all share the same basic structure: each sum over k combines a first term that relates to the output noise structure of the associated parameter and a second term that involves the expected output error conditioned on the input component, $V\mathbb{E}[\mathbf{v}|k] - \mathbf{d}_k$. The sums are then pre-multiplied by $(I - W^\mathsf{T})^{-1} V^\mathsf{T}$ to recast their results into the neural space.

It is also possible to obtain an alternative expression for $\nabla_W L$ connected with Eq. 15, namely (*cf.* appendix B)

$$\nabla_W L = (I - W^\mathsf{T})^{-1} V^\mathsf{T} \left[V \left(\mathbb{V}[\mathbf{v}] + \bar{\mathbf{v}}\bar{\mathbf{v}}^\mathsf{T} \right) - \sum_k \pi_k \mathbf{d}_k \mathbb{E}[\mathbf{v}|k]^\mathsf{T} \right], \tag{25}$$

which emphasizes the impact of the total covariance matrix $\mathbb{V}[\mathbf{v}]$ on the gradient.

5.6. **Parameters.**

5.6.1. *Figures 1-5.* The network had $N = 100$ recurrent units. The recurrent weights were initialized with $W_{ij} \sim \mathcal{N}(0, 1/N)$. The input weights U were fixed to $U_{ij} \sim \mathcal{U}(-1, 1)$. There were six targets. The input was a noiseless 1–of–6 encoding, i.e. $\boldsymbol{\mu}_0 = [1,0,0,0,0,0]^\mathsf{T}$, $\boldsymbol{\mu}_1 = [0,1,0,0,0,0]^\mathsf{T}$, …, and $\Sigma_x^{(k)} = 0$, for $k = 0, \ldots, 5$. The private noise variance was low, fixed to 10^{-3}. For the initial output matrix $V = V_0$, we first drew $V_{ij} \sim \mathcal{N}(0, 1)$ and then reassigned $V \leftarrow 0.07\|V\|_F V$ [18]. Initial training of the network (with $V = V_0$) was with learning rate $\eta_W = 10^{-3}$. In Figs. 1-3, subsequent training with the perturbed readout matrices V_{WM} and V_{OM} was done with $\eta_W = 6.7 \times 10^{-5}$ to diminish the likelihood of very noisy OM adaptations.

5.6.2. *Figure 6.* For the high-dimensional input setup, we had $N = 100$ and the number of input units was $I = 200$. The initial input weight matrix, recurrent weight matrix, and output weight matrix were drawn from $\mathcal{N}(0, 1/I)$, $\mathcal{N}(0, 1/N)$ and $\mathcal{N}(0, 1/N)$, respectively. The target-conditioned mean inputs $\boldsymbol{\mu}_k$ were Gaussian random vectors, normalized to have unit norm. The target-conditioned input covariance matrices were set to $\Sigma_x^{(k)} = \sigma_x^2 QQ^\mathsf{T}$, where Q was an $I \times I/10$ matrix with elements drawn from a standard normal distribution, and $\sigma_x^2 = 2.5 \times 10^{-5}$ set the relative intensity of the input compared to the private noise. Note that $\Sigma_x^{(k)}$ had rank $I/10$ to take account of the fact that sensory encoding has lower dimensionality than the full neural space [9]. Private noise intensity was set to 5×10^{-4}. The latter and the input noise intensity were chosen so that the number of principal vectors to represent 99% of the total variance for the initial network trained with $V = V_0$ was close to that of the low-dimensional input network (*cf.* subsection 5.4).

5.7. **Normalized explained variance and manifold overlap.** The matrix C effectively projects the activity $\mathbf{v}$ onto the M principal orthogonal eigenvectors of the initial (i.e., before the perturbation) covariance matrix, denoted $\mathbb{V}[\mathbf{v}]_0$. Before adaptation, the fraction of explained variance by these projections is

$$f_0 = \frac{\mathbb{E}[\|C(\mathbf{v} - \bar{\mathbf{v}}_0)\|^2]}{\mathbb{E}[\|\mathbf{v} - \bar{\mathbf{v}}_0\|^2]} = \frac{\mathsf{tr}\left\{C\mathbb{V}[\mathbf{v}]_0 C^\mathsf{T}\right\}}{\mathsf{tr}\left\{\mathbb{V}[\mathbf{v}]_0\right\}}.$$

During adaptation, the covariance might change and with it the fraction of variance explained by C, denoted f:

$$f = \frac{\mathsf{tr}\left\{C\mathbb{V}[\mathbf{v}]C^\mathsf{T}\right\}}{\mathsf{tr}\left\{\mathbb{V}[\mathbf{v}]\right\}}. \tag{26}$$

Intuitively, if the activity aligns more with C relative to the total variance through the adaptation process, f increases. The fraction of explained variance will tend to decrease if misalignments become proportionally more prevalent. Therefore, the normalized variance explained [16] given by

$$\mathsf{NVE} = \frac{f}{f_0} \tag{27}$$

represents the relative proportion of the total variance that can be explained by the projection during adaptation. Feulner and Clopath [18] refer to this quantity as the manifold overlap because this ratio represents the relative fraction of variance that lives in the row space of C. A manifold overlap of 100% thus means that the same proportion of variance (as compared to the pre-perturbation activity) lives in that space at that point during adaptation.

5.8. **Ratio of projected variances.** Outside-manifold perturbations yield a new C matrix, $C_\mathsf{OM} = CP_\mathsf{OM}$. Like the original C, the rows of C_OM are orthonormal, because $P_\mathsf{OM}P_\mathsf{OM}^\mathsf{T} = I \Rightarrow C_\mathsf{OM}C_\mathsf{OM}^\mathsf{T} = I$. Thus, as above, we can measure the intensity of the projected covariance within this subspace as $\mathsf{tr}\left\{C_\mathsf{OM}\mathbb{V}[\mathbf{v}]C_\mathsf{OM}^\mathsf{T}\right\}$. At each stage of the adaptation process, we can compute the ratio of projected variances

$$R = \frac{\mathsf{tr}\left\{C_\mathsf{OM}\mathbb{V}[\mathbf{v}]C_\mathsf{OM}^\mathsf{T}\right\}}{\mathsf{tr}\left\{C\mathbb{V}[\mathbf{v}]C^\mathsf{T}\right\}} \tag{28}$$

to quantify how adaptation to the OMP redistributes the covariance between the old (C) and new (C_OM) projections.

5.9. **Realignment, rescaling and reassociation.** Golub and co-workers [16] delineated three strategies whereby the motor cortex could adapt to WMPs. The initial decoder matrix D maps the projection of the activity $C\mathbf{v}$ onto "cursor positions". The WMP amounts to either leaving D unchanged and permuting the rows of C, or permuting the columns of D and leaving C unchanged. From the latter perspective, the WMP alters the two-dimensional row space $\subset \mathbb{R}^M$ of the decoder mapping. To define this row space, we can compute the (reduced) singular value decomposition (SVD) of the D matrix, $D = X_D S_D Y_D^{\mathsf{T}}$, where Y_D^{T} has two orthonormal rows spanning the row space of D. The amount of co-variability projected along this row space can be computed as [16]

$$A(D) = \mathsf{tr}\left\{ Y_D^{\mathsf{T}} C \mathbb{V}[\mathbf{v}] C^{\mathsf{T}} Y_D \right\}.$$

This quantity may vary throughout learning for both the original and perturbed mappings, which we denote by $A(D)$ and $A(DP_{\mathsf{WM}})$. *Realignment* means that the magnitude of the projections along the original row space decreases while those along the perturbed mapping increases, suggesting the activity changes to aligns itself to the new mapping. *Rescaling* consists in an increase of $A(D)$ during adaptation while $A(DP_{\mathsf{WM}})$ mostly does not change, suggesting that the network "stretches" its activity along the original mapping to compensate for the perturbation. Golub and co-workers showed that it is a third strategy, *reassociation*, that best explained the experimental data. The name suggests that the network relies on a fixed repertoire of neural patterns, which are repurposed following the perturbation. This reassociation means that both $A(D)$ and $A(DP_{\mathsf{WM}})$ barely change during learning. Following Ref. [16], we computed the relative change in decoder-projected co-variability (in %) as

$$\delta_{\mathsf{rel.}} A(\mathcal{D}) = 100 \frac{A(\mathcal{D}) - A_0(\mathcal{D})}{A_0(\mathcal{D})} \tag{29}$$

where $\mathcal{D}$ is either D for the covariance projected onto the original decoder or DP_{WM} for the covariance projected onto the perturbed decoder. The change is measured relative to the projected covariance right before the perturbation was applied, denoted $A_0(\mathcal{D})$. For realignment, rescaling and reassociation, the pairs $(\delta_{\mathsf{rel.}} A(D), \delta_{\mathsf{rel.}} A(DP_{\mathsf{WM}}))$ should be (small negative, large positive), (large positive, ≈ 0) and (≈ 0, ≈ 0), respectively.

5.10. **Incremental training.** Following Ref. [15], incremental training for OMPs was performed by changing the output mapping *via*

$$V = (1 - a)V_{\mathsf{BCI}} + aV_{\mathsf{OM}}$$

with $a = 0.2, 0.4, 0.6, 0.8, 0.9, 1$ in turn.

Acknowledgments

The authors' research was supported in part by an IVADO postdoctoral fellowship, Canada First Research Excellence Fund/Apogée (AP), the Simons Foundation (GL and ALO), a Google Faculty Award (ALO and GL) and an NSERC Discovery Grant (GL). GL further acknowledges support from the Canada CIFAR AI Chair Program and the Canada Research Chair in Neural Computations and Interfacing (CIHR, tier 2). The content of the present paper is solely the responsibility of the authors and does not necessarily represent the official views of the founding agencies.

References

[1] J. A. Gallego, M. G. Perich, L. E. Miller, and S. A. Solla, "Neural Manifolds for the Control of Movement," *Neuron*, vol. 94, pp. 978–984, June 2017.

[2] M. Jazayeri and A. Afraz, "Navigating the Neural Space in Search of the Neural Code," *Neuron*, vol. 93, pp. 1003–1014, Mar. 2017.

[3] P. Gao and S. Ganguli, "On simplicity and complexity in the brave new world of large-scale neuroscience," *Current Opinion in Neurobiology*, vol. 32, pp. 148–155, June 2015.

[4] P. Gao, E. Trautmann, B. Yu, G. Santhanam, S. Ryu, K. Shenoy, and S. Ganguli, "A theory of multineuronal dimensionality, dynamics and measurement," tech. rep., Nov. 2017. Company: Cold Spring Harbor Laboratory Distributor: Cold Spring Harbor Laboratory Label: Cold Spring Harbor Laboratory Section: New Results Type: article.

[5] H. S. Seung and D. D. Lee, "The Manifold Ways of Perception," *Science*, vol. 290, pp. 2268–2269, Dec. 2000. Publisher: American Association for the Advancement of Science.

[6] H. S. Seung, "How the brain keeps the eyes still," *Proceedings of the National Academy of Sciences*, vol. 93, no. 23, pp. 13339–13344, 1996.

[7] U. Cohen, S. Chung, D. D. Lee, and H. Sompolinsky, "Separability and geometry of object manifolds in deep neural networks," *Nat Commun*, vol. 11, p. 746, Feb. 2020. Number: 1 Publisher: Nature Publishing Group.

[8] E. H. Nieh, M. Schottdorf, N. W. Freeman, R. J. Low, S. Lewallen, S. A. Koay, L. Pinto, J. L. Gauthier, C. D. Brody, and D. W. Tank, "Geometry of abstract learned knowledge in the hippocampus," *Nature*, vol. 595, pp. 80–84, July 2021. Number: 7865 Publisher: Nature Publishing Group.

[9] C. Stringer, M. Pachitariu, N. Steinmetz, M. Carandini, and K. D. Harris, "High-dimensional geometry of population responses in visual cortex," *Nature*, vol. 571, pp. 361–365, July 2019. Number: 7765 Publisher: Nature Publishing Group.

[10] B. Sorscher, S. Ganguli, and H. Sompolinsky, "Neural representational geometry underlies few-shot concept learning," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 119, p. e2200800119, Oct. 2022.

[11] M. Beiran, N. Meirhaeghe, H. Sohn, M. Jazayeri, and S. Ostojic, "Parametric control of flexible timing through low-dimensional neural manifolds," *Neuron*, Jan. 2023.

[12] M. D. Golub, S. M. Chase, A. P. Batista, and B. M. Yu, "Brain–computer interfaces for dissecting cognitive processes underlying sensorimotor control," *Current Opinion in Neurobiology*, vol. 37, pp. 53–58, Apr. 2016.

[13] A. L. Orsborn and B. Pesaran, "Parsing learning in networks using brain–machine interfaces," *Current Opinion in Neurobiology*, vol. 46, pp. 76–83, Oct. 2017.

[14] P. T. Sadtler, K. M. Quick, M. D. Golub, S. M. Chase, S. I. Ryu, E. C. Tyler-Kabara, B. M. Yu, and A. P. Batista, "Neural constraints on learning," *Nature*, vol. 512, pp. 423–426, Aug. 2014.

[15] E. R. Oby, M. D. Golub, J. A. Hennig, A. D. Degenhart, E. C. Tyler-Kabara, B. M. Yu, S. M. Chase, and A. P. Batista, "New neural activity patterns emerge with long-term learning," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 116, pp. 15210–15215, July 2019.

[16] M. D. Golub, P. T. Sadtler, E. R. Oby, K. M. Quick, S. I. Ryu, E. C. Tyler-Kabara, A. P. Batista, S. M. Chase, and B. M. Yu, "Learning by neural reassociation," *Nature Neuroscience*, vol. 21, pp. 607–616, Apr. 2018.

[17] E. Wärnberg and A. Kumar, "Perturbing low dimensional activity manifolds in spiking neuronal networks," *PLOS Computational Biology*, vol. 15, p. e1007074, May 2019. Publisher: Public Library of Science.

[18] B. Feulner and C. Clopath, "Neural manifold under plasticity in a goal driven learning behaviour," *PLOS Computational Biology*, vol. 17, p. e1008621, Feb. 2021. Publisher: Public Library of Science.

[19] P. C. Humphreys, K. Daie, K. Svoboda, M. Botvinick, and T. P. Lillicrap, "BCI learning phenomena can be explained by gradient-based optimization," Dec. 2022. Pages: 2022.12.08.519453 Section: New Results.

[20] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach Learn*, vol. 8, pp. 229–256, May 1992.

[21] J. M. Murray, "Local online learning in recurrent networks with random feedback," *eLife*, vol. 8, p. e43299, May 2019. Publisher: eLife Sciences Publications, Ltd.

[22] G. Bellec, F. Scherr, A. Subramoney, E. Hajek, D. Salaj, R. Legenstein, and W. Maass, "A solution to the learning dilemma for recurrent networks of spiking neurons," *Nature Communications*, vol. 11, p. 3625, July 2020. Number: 1 Publisher: Nature Publishing Group.

[23] A. Payeur, J. Guerguiev, F. Zenke, B. A. Richards, and R. Naud, "Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits," *Nat Neurosci*, vol. 24, pp. 1010–1019, July 2021. Number: 7 Publisher: Nature Publishing Group.

[24] G. Hennequin, Y. Ahmadian, D. B. Rubin, M. Lengyel, and K. D. Miller, "The Dynamical Regime of Sensory Cortex: Stable Dynamics around a Single Stimulus-Tuned Attractor Account for Patterns of Noise Variability," *Neuron*, vol. 98, pp. 846–860.e5, May 2018.

[25] D. Sussillo and L. Abbott, "Generating Coherent Patterns of Activity from Chaotic Neural Networks," *Neuron*, vol. 63, pp. 544–557, Aug. 2009.

[26] R. Laje and D. V. Buonomano, "Robust timing and motor patterns by taming chaos in recurrent neural networks," *Nature Neuroscience*, vol. 16, pp. 925–933, July 2013. Number: 7 Publisher: Nature Publishing Group.

[27] Ȧ. Björck and G. H. Golub, "Numerical methods for computing angles between linear subspaces," *Math. Comp.*, vol. 27, no. 123, pp. 579–594, 1973.

[28] C. Ma, L. Wu, and W. E, "A Qualitative Study of the Dynamic Behavior for Adaptive Gradient Algorithms," in *Proceedings of the 2nd Mathematical and Scientific Machine Learning Conference*, pp. 671–692, PMLR, Apr. 2022. ISSN: 2640-3498.

[29] M. G. Perich, J. A. Gallego, and L. E. Miller, "A Neural Population Mechanism for Rapid Learning," *Neuron*, vol. 100, pp. 964–976.e7, Nov. 2018.

[30] B. Jarosiewicz, S. M. Chase, G. W. Fraser, M. Velliste, R. E. Kass, and A. B. Schwartz, "Functional network reorganization during learning in a brain-computer interface paradigm," *Proc Natl Acad Sci U S A*, vol. 105, pp. 19486–19491, Dec. 2008.

[31] B. Feulner, M. G. Perich, R. H. Chowdhury, L. E. Miller, J. A. Gallego, and C. Clopath, "Small, correlated changes in synaptic connectivity may facilitate rapid motor learning," *Nat Commun*, vol. 13, p. 5163, Sept. 2022. Number: 1 Publisher: Nature Publishing Group.

[32] M. T. Kaufman, M. M. Churchland, S. I. Ryu, and K. V. Shenoy, "Cortical activity in the null space: permitting preparation without movement," *Nature Neuroscience*, vol. 17, pp. 440–448, Mar. 2014.

[33] S. Grossberg, "Competitive learning: From interactive activation to adaptive resonance," *Cognitive Science*, vol. 11, pp. 23–63, Jan. 1987.

[34] T. P. Lillicrap, D. Cownden, D. B. Tweed, and C. J. Akerman, "Random synaptic feedback weights support error backpropagation for deep learning," *Nature Communications*, vol. 7, p. 13276, Nov. 2016. Number: 1 Publisher: Nature Publishing Group.

[35] M. Akrout, C. Wilson, P. Humphreys, T. Lillicrap, and D. B. Tweed, "Deep Learning without Weight Transport," in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d. Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 976–984, Curran Associates, Inc., 2019.

[36] B. Podlaski and C. K. Machens, "Biological credit assignment through dynamic inversion of feedforward networks," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 10065–10076, Curran Associates, Inc., 2020.

[37] K. Doya, "Metalearning and neuromodulation," *Neural Networks*, vol. 15, pp. 495–506, June 2002.

[38] C. Vidaurre and B. Blankertz, "Towards a Cure for BCI Illiteracy," *Brain Topogr*, vol. 23, pp. 194–198, June 2010.

[39] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, (New York, NY, USA), pp. 41–48, Association for Computing Machinery, June 2009.

[40] V. Gilja, P. Nuyujukian, C. A. Chestek, J. P. Cunningham, B. M. Yu, J. M. Fan, M. M. Churchland, M. T. Kaufman, J. C. Kao, S. I. Ryu, and K. V. Shenoy, "A high-performance neural prosthesis enabled by control algorithm design," *Nature Neuroscience*, vol. 15, pp. 1752–1757, Dec. 2012. Number: 12 Publisher: Nature Publishing Group.

[41] A. L. Orsborn, H. G. Moorman, S. A. Overduin, M. M. Shanechi, D. F. Dimitrov, and J. M. Carmena, "Closed-Loop Decoder Adaptation Shapes Neural Plasticity for Skillful Neuroprosthetic Control," *Neuron*, vol. 82, pp. 1380–1393, June 2014.

[42] C. M. Bishop, *Pattern Recognition and Machine Learning*. Information Science and Statistics, Springer, 2006.

[43] S. Haykin, *Adaptive Filter Theory*. Information and System Sciences, Prentice Hall, 1996.

Appendix A. Theoretical comparison of recursive least-squares and gradient descent

To distinguish our work from Ref. [18], it is instructive to first compare the recursive least-squares (RLS) algorithm and gradient descent in a simple case: linear regression. Let the predictor be $\hat{y}(\mathbf{x}, \mathbf{w}) = \mathbf{w}^{\mathsf{T}}\mathbf{x}$, with $\mathbf{x} \in \mathbb{R}^D$ the regressor and $\mathbf{w}$ a weight vector to be learned. The desired output is y and we evaluate the performance of the predictor using the mean-squared error[1]

$$L = \frac{1}{2}\sum_{i=1}^{N}[y_i - \hat{y}(\mathbf{x}_i, \mathbf{w})]^2,$$

where N is here the number of data points. The normal equation giving the optimal weight vector is well-known to be [42]

$$\hat{\mathbf{w}}_N = (X^{\mathsf{T}}X)^{-1}X^{\mathsf{T}}\mathsf{y},$$

where X is a $N \times D$ matrix whose rows are $\mathbf{x}_i^{\mathsf{T}}$ and y (sans serif) is a column vector containing the N observed outputs. This method requires the batch processing of all data. Recursive least-squares [43] is an online version of the normal equation. One first defines

$$P_N \triangleq (X^{\mathsf{T}}X)^{-1},$$

the inverse of the sample correlation matrix. It is possible to write P_N as a function of P_{N-1} as

$$P_N = P_{N-1} - \frac{P_{N-1}\mathbf{x}_N\mathbf{x}_N^{\mathsf{T}}P_{N-1}}{1 + \mathbf{x}_N^{\mathsf{T}}P_{N-1}\mathbf{x}_N}.$$

Then, replacing these in the equation for $\hat{\mathbf{w}}_N$ and replacing the dummy index N by i to stress the online character of the algorithm, one obtains after some algebra

$$\hat{\mathbf{w}}_i = \hat{\mathbf{w}}_{i-1} + P_i\mathbf{x}_i e_i, \tag{30}$$

where $e_i = y_i - \hat{\mathbf{w}}_{i-1}^{\mathsf{T}}\mathbf{x}_i$ is the prediction error.

For the stochastic gradient descent algorithm, the update is simply given by

$$\hat{\mathbf{w}}_i = \hat{\mathbf{w}}_{i-1} - \eta \left.\nabla_{\mathbf{w}}L_i\right|_{\mathbf{w}_{i-1}},$$

where η is the learning rate parameter and $\left.\nabla_{\mathbf{w}}L_i\right|_{\mathbf{w}_{i-1}} = -e_i\mathbf{x}_i$. The stochastic gradient descent algorithm is thus

$$\hat{\mathbf{w}}_i = \hat{\mathbf{w}}_{i-1} + \eta\,\mathbf{x}_i e_i. \tag{31}$$

The obvious difference between Eqs. 30 and 31 is the presence of the matrix P_i, which estimates the inverse correlation matrix of the data at step i. Its initial value P_0—before seeing any data—is typically a multiple of the identity matrix I. The inverse correlation matrix weights the contribution of $\mathbf{x}_i$ according to the distribution of data points at that stage. For instance, if

[1] Sometimes a forgetting factor $\lambda \in (0, 1]$ is included in the loss, so that $L_N = \frac{1}{2}\sum_{i=1}^{N}\lambda^{N-i}[y_i - \hat{y}(\mathbf{x}_i, \mathbf{w})]^2$, to forget older data progressively. Here, we set $\lambda = 1$ for brevity and simplicity.

the data is strongly correlated in a given direction in the D-dimensional regressor space, and $\mathbf{x}_i$ is aligned with that direction, then the impact of that data point in changing the weight vector will be small. The RLS algorithm thus includes a lot more information about the data structure than stochastic gradient descent.

Feulner and Clopath [18] used a modified version of this algorithm—other related modifications appeared in the context of FORCE learning [25] and innate learning [26]—in their paper, modulo some implementational details. The recurrent weight matrix W of their network was changed according to

$$W_{ij}(t) = W_{ij}(t-1) + e_i(t) \sum_k P^i_{jk}(t) r_k(t)$$

where $P^i(t)$ is the inverse correlation matrix of the input rates applied to neuron i, $r_k(t)$ is the firing rate of neuron k at time t and e_i is the prediction error for the activity of neuron i. The latter was obtained from the performance error of the network, $\mathbf{e}^P$, by projecting this error to the network *via* a feedback matrix W^{fb}, so that $\mathbf{e} = W^{\mathsf{fb}}\mathbf{e}^P$. In the ideal-feedback case, the feedback matrix was the exact pseudo-inverse of the output weight matrix of the network. (Note that the error feedback did not affect neural activity directly.) In that case, no differences were seen between in- and out-manifold learning at the end of adaptation [18] and even during adaptation (see Supp. Fig. 1). From the discussion above, this is likely due to the fact that outside-manifold perturbations eventually produce activity in less expected directions in neural space which translates into larger effective learning rates. Within-manifold perturbations would not tend to produce these. Therefore, adaptation to an OMP catches up rapidly with adaptation to a WMP. It is when W^{fb} is corrupted in some way (e.g., making it a noisy version for the ideal-feedback matrix) that the network adapts differently to WMPs and OMPs. It is really the corrupted feedback that does the heavy lifting because they have found that a local learning rule with such feedback also led to different end-of-retraining performances [18] (their supplementary figure S18). Note that they also contributed an example showing that this local learning rule produced WM and OM adaptations at slightly different time scales with uncorrupted feedback (their supplementary figure S6), a result we reproduced a bit more conclusively with our model.

Appendix B. Calculation of the gradients

In this section, we compute the gradient of the objective function with respect to parameters U and W. To do this in the most straightforward way, we will use the following result of matrix calculus. Let $f(M)$ be a scalar function of the matrix M. Its total differential is

$$df = f(M + dM) - f(M) = \mathsf{tr}\left\{(\nabla_M f)^{\mathsf{T}} dM\right\}.$$

Therefore, to get the gradients, we only have to compute the differential, put it in the above form and identify the matrix multiplying the matrix differential as the (transpose of) the gradient.

The differential of the loss is

$$dL = \frac{1}{2}\sum_k \pi_k \mathsf{tr}\left\{V d\mathbb{V}[\mathbf{v}|k]V^\mathsf{T}\right\} + \sum_k \pi_k \mathsf{tr}\left\{V d\mathbb{E}[\mathbf{v}|k](V\mathbb{E}[\mathbf{v}|k] - \mathbf{d}_k)^\mathsf{T}\right\}.$$

First, we have

$$\begin{aligned} d\mathbb{V}[\mathbf{v}|k] &= d\left[(I-W)^{-1}\left[U\Sigma_x^{(k)}U^\mathsf{T} + \Sigma_\xi\right](I-W^\mathsf{T})^{-1}\right] \\ &= (I-W)^{-1}dW\mathbb{V}[\mathbf{v}|k] + (I-W)^{-1}[dU\Sigma_x^{(k)}U^\mathsf{T} + U\Sigma_x^{(k)}dU^\mathsf{T}](I-W^\mathsf{T})^{-1} \\ &\quad + \mathbb{V}[\mathbf{v}|k]dW^\mathsf{T}(I-W^\mathsf{T})^{-1} \end{aligned}$$

where we used $d(I-W)^{-1} = (I-W)^{-1}dW(I-W)^{-1}$. Second,

$$\begin{aligned} d\mathbb{E}[\mathbf{v}|k] &= d[(I-W)^{-1}U\boldsymbol{\mu}_k] \\ &= (I-W)^{-1}dW\mathbb{E}[\mathbf{v}|k] + (I-W)^{-1}dU\boldsymbol{\mu}_k. \end{aligned}$$

By collating all the factors of dW and using the cyclic and transpose property of the trace (i.e., $\mathsf{tr}\{ABC\} = \mathsf{tr}\{CAB\} = \mathsf{tr}\{BCA\}$ and $\mathsf{tr}\left\{A^\mathsf{T}\right\} = \mathsf{tr}\{A\}$), we get

$$(\nabla_W L)^\mathsf{T} = \sum_k \pi_k \mathbb{V}[\mathbf{v}|k]V^\mathsf{T}V(I-W)^{-1} + \sum_k \pi_k \mathbb{E}[\mathbf{v}|k](V\mathbb{E}[\mathbf{v}|k] - \mathbf{d}_k)^\mathsf{T}V(I-W)^{-1},$$

as in the main text (upon transposition). The same process can be carried out for the gradient with respect to U without difficulties.

Now we obtain the expression for the gradient of the covariance component of the loss, which is denoted by

$$L_{\mathbb{V}[\mathbf{v}]} \triangleq \frac{1}{2}\mathsf{tr}\left\{V\mathbb{V}[\mathbf{v}]V^\mathsf{T}\right\}.$$

It is the second term in Eq. 15, excluding the part related to the global mean $\bar{\mathbf{v}}$. The differential is given by $dL_{\mathbb{V}[\mathbf{v}]} = \frac{1}{2}\mathsf{tr}\left\{Vd\mathbb{V}[\mathbf{v}]V^\mathsf{T}\right\}$ and therefore we have to compute the differential of the total covariance matrix $d\mathbb{V}[\mathbf{v}]$. From Eq. 11, we have

$$d\mathbb{V}[\mathbf{v}] = \sum_k \pi_k \left[d\mathbb{V}[\mathbf{v}|k] + (d\mathbb{E}[\mathbf{v}|k] - d\bar{\mathbf{v}})(\mathbb{E}[\mathbf{v}|k] - \bar{\mathbf{v}})^\mathsf{T} + (\mathbb{E}[\mathbf{v}|k] - \bar{\mathbf{v}})(d\mathbb{E}[\mathbf{v}|k] - d\bar{\mathbf{v}})^\mathsf{T}\right].$$

The only quantity that has not been computed yet is $d\bar{\mathbf{v}}$. From Eq. 12,

$$d\bar{\mathbf{v}} = (I-W)^{-1}dW\bar{\mathbf{v}} + (I-W)^{-1}dU\bar{\mathbf{x}}.$$

To compute the gradient w.r.t. W, we pick all terms involving dW. Focusing on a single target k, we have

$$\begin{aligned} & d\mathbb{V}[\mathbf{v}|k] + (d\mathbb{E}[\mathbf{v}|k] - d\bar{\mathbf{v}})(\mathbb{E}[\mathbf{v}|k] - \bar{\mathbf{v}})^\mathsf{T} + (\mathbb{E}[\mathbf{v}|k] - \bar{\mathbf{v}})(d\mathbb{E}[\mathbf{v}|k] - d\bar{\mathbf{v}})^\mathsf{T} \\ &= (I-W)^{-1}dW\mathbb{V}[\mathbf{v}|k] + \left((I-W)^{-1}dW\mathbb{E}[\mathbf{v}|k] - (I-W)^{-1}dW\bar{\mathbf{v}}\right)(\mathbb{E}[\mathbf{v}|k] - \bar{\mathbf{v}})^\mathsf{T} + h.c. \\ &= (I-W)^{-1}dW\mathbb{V}[\mathbf{v}|k] + (I-W)^{-1}dW\left(\mathbb{E}[\mathbf{v}|k] - \bar{\mathbf{v}}\right)\left(\mathbb{E}[\mathbf{v}|k] - \bar{\mathbf{v}}\right)^\mathsf{T} + h.c. \\ &= (I-W)^{-1}dW\left[\mathbb{V}[\mathbf{v}|k] + \left(\mathbb{E}[\mathbf{v}|k] - \bar{\mathbf{v}}\right)\left(\mathbb{E}[\mathbf{v}|k] - \bar{\mathbf{v}}\right)^\mathsf{T}\right] + h.c. \end{aligned}$$

where $h.c.$ means hermitian conjugate and in the present case means to take the transpose of the previous term. Using Eq. 11 again and spelling out the hermitian conjugate term, we have

$$d\mathbb{V}[\mathbf{v}] = (I - W)^{-1} dW \mathbb{V}[\mathbf{v}] + \mathbb{V}[\mathbf{v}] dW^{\mathsf{T}} (I - W^{\mathsf{T}})^{-1},$$

and thus

$$\begin{aligned} dL_{\mathbb{V}[\mathbf{v}]} &= \frac{1}{2} \mathsf{tr} \left\{ V d\mathbb{V}[\mathbf{v}] V^{\mathsf{T}} \right\} \\ &= \mathsf{tr} \left\{ \mathbb{V}[\mathbf{v}] V^{\mathsf{T}} V (I - W)^{-1} dW \right\}. \end{aligned}$$

The gradient is the transpose of the term inside the trace multiplying dW:

$$\nabla_W L_{\mathbb{V}[\mathbf{v}]} = (I - W^{\mathsf{T}})^{-1} V^{\mathsf{T}} V \mathbb{V}[\mathbf{v}].$$

Appendix C. Other derivations

This section contains other simple derivations, for convenience.

C.1. **Derivation pertaining to the loss components.** We first describe why the correlation component of the loss converges to 1/2. We note that, close to complete adaptation,

$$V\bar{\mathbf{v}} \approx \mathbf{0}, \tag{32}$$

with $\bar{\mathbf{v}}$ the global average activity (Eq. 12) and V is either perturbed mappings, given the symmetry of the task when the targets are equiprobable. Indeed, we have

$$V\bar{\mathbf{v}} = \mathbb{E}[\mathbb{E}[V\mathbf{v}|k]] = \mathbb{E}[\mathbb{E}[\mathbf{u}|k]]$$

and $\mathbb{E}[\mathbf{u}|k] \approx \mathbf{d}_k$ once adaptation is close to completion. Then $\mathbb{E}[\mathbb{E}[\mathbf{u}|k]] \approx \sum_k \pi_k \mathbf{d}_k = \mathbf{0}$ for $\pi_k = 1/K$. Using the expression for the total variance (Eq. 11), the correlation component of the loss becomes

$$\begin{aligned} \frac{1}{2}\mathsf{tr}\left\{V(\mathbb{V}[\mathbf{v}] + \bar{\mathbf{v}}\bar{\mathbf{v}}^{\mathsf{T}})V^{\mathsf{T}}\right\} \approx \frac{1}{2}\mathsf{tr}\left\{V\mathbb{V}[\mathbf{v}]V^{\mathsf{T}}\right\} \\ &= \frac{1}{2}\mathsf{tr}\left\{V\left[\sum_k \pi_k \mathbb{V}[\mathbf{v}|k] + \sum_k \pi_k (\mathbb{E}[\mathbf{v}|k] - \bar{\mathbf{v}})(\mathbb{E}[\mathbf{v}|k] - \bar{\mathbf{v}})^{\mathsf{T}}\right] V^{\mathsf{T}}\right\} \\ &\approx \frac{1}{2}\mathsf{tr}\left\{V\left[\sum_k \pi_k (\mathbb{E}[\mathbf{v}|k] - \bar{\mathbf{v}})(\mathbb{E}[\mathbf{v}|k] - \bar{\mathbf{v}})^{\mathsf{T}}\right] V^{\mathsf{T}}\right\} \\ &\approx \frac{1}{2}\sum_k \pi_k \|V\mathbb{E}[\mathbf{v}|k]\|^2 \end{aligned}$$

after adaptation, with $V\mathbb{E}[\mathbf{v}|k]$ being then close to $\mathbf{d}_k$, the kth unit target vector. We also neglected the contribution of the target-conditioned variance $\mathbb{V}[\mathbf{v}|k]$, which is small for the parameters used in Figs. 1-2. The result follows from the unit norm of the targets.

We also show that the projection component of the loss converges to -1 (*cf.* Eq. 15). Once the network has completely adapted to the perturbation, $-\sum_k \pi_k \mathbf{d}_k^\mathsf{T} V \mathbb{E}[\mathbf{v}|k] \approx -\sum_k \pi_k \|\mathbf{d}_k\|^2 = -1$.

C.2. **Equation 10.**

$$\begin{aligned}
L &= \frac{1}{2}\sum_k \pi_k \mathbb{E}\left[\|V\mathbf{v} - \mathbf{d}_k\|^2 | k\right] \\
&= \frac{1}{2}\sum_k \pi_k \mathbb{E}\left[\|V\mathbf{v} - V\mathbb{E}[\mathbf{v}|k] + V\mathbb{E}[\mathbf{v}|k] - \mathbf{d}_k\|^2 | k\right] \\
&= \frac{1}{2}\sum_k \pi_k \mathbb{E}\left[\|V\mathbf{v} - V\mathbb{E}[\mathbf{v}|k]\|^2 + \|V\mathbb{E}[\mathbf{v}|k] - \mathbf{d}_k\|^2 + 2(V\mathbf{v} - V\mathbb{E}[\mathbf{v}|k])^\mathsf{T}(V\mathbb{E}[\mathbf{v}|k] - \mathbf{d}_k) | k\right].
\end{aligned}$$

Equation 10 follows because the expectation of the last term is zero and because $\|\mathbf{z}\|^2 = \mathsf{tr}\left\{\mathbf{z}\mathbf{z}^\mathsf{T}\right\}$ for any vector $\mathbf{z}$.

C.3. **Equation 14.**

$$\begin{aligned}
\mathsf{tr}\left\{V\mathbb{V}[\mathbf{v}]V^\mathsf{T}\right\} &= \sum_k \pi_k \mathsf{tr}\left\{V\mathbb{V}[\mathbf{v}|k]V^\mathsf{T}\right\} + \sum_k \pi_k \mathsf{tr}\left\{V(\mathbb{E}[\mathbf{v}|k] - \bar{\mathbf{v}})(\mathbb{E}[\mathbf{v}|k] - \bar{\mathbf{v}})^\mathsf{T} V^\mathsf{T}\right\} \\
&= \sum_k \pi_k \mathsf{tr}\left\{V\mathbb{V}[\mathbf{v}|k]V^\mathsf{T}\right\} + \sum_k \pi_k \|V\mathbb{E}[\mathbf{v}|k] - V\bar{\mathbf{v}}\|^2 \\
&= \sum_k \pi_k \mathsf{tr}\left\{V\mathbb{V}[\mathbf{v}|k]V^\mathsf{T}\right\} + \sum_k \pi_k \|V\mathbb{E}[\mathbf{v}|k] - \mathbf{d}_k - (V\bar{\mathbf{v}} - \mathbf{d}_k)\|^2 \\
&= \sum_k \pi_k \mathsf{tr}\left\{V\mathbb{V}[\mathbf{v}|k]V^\mathsf{T}\right\} + \sum_k \pi_k \|V\mathbb{E}[\mathbf{v}|k] - \mathbf{d}_k\|^2 + \sum_k \pi_k \|V\bar{\mathbf{v}} - \mathbf{d}_k\|^2 \\
&\qquad - 2\sum_k \pi_k (V\mathbb{E}[\mathbf{v}|k] - \mathbf{d}_k)^\mathsf{T}(V\bar{\mathbf{v}} - \mathbf{d}_k) \\
&= 2L + \sum_k \pi_k \|V\bar{\mathbf{v}} - \mathbf{d}_k\|^2 - 2\sum_k \pi_k (V\mathbb{E}[\mathbf{v}|k] - \mathbf{d}_k)^\mathsf{T}(V\bar{\mathbf{v}} - \mathbf{d}_k) \\
&= 2L + \mathsf{tr}\left\{V\bar{\mathbf{v}}\bar{\mathbf{v}}^\mathsf{T} V\right\} + 1 - 2\sum_k \pi_k (V\mathbb{E}[\mathbf{v}|k] - \mathbf{d}_k)^\mathsf{T}(V\bar{\mathbf{v}} - \mathbf{d}_k) \\
&= 2L + \mathsf{tr}\left\{V\bar{\mathbf{v}}\bar{\mathbf{v}}^\mathsf{T} V\right\} + 1 - 2(\mathsf{tr}\left\{V\bar{\mathbf{v}}\bar{\mathbf{v}}^\mathsf{T} V\right\} - \sum_k \pi_k \mathbb{E}[\mathbf{v}|k]^\mathsf{T} V^\mathsf{T} \mathbf{d}_k + 1) \\
&= 2L - \mathsf{tr}\left\{V\bar{\mathbf{v}}\bar{\mathbf{v}}^\mathsf{T} V\right\} - 1 + 2\sum_k \pi_k \mathbf{d}_k^\mathsf{T} V\mathbb{E}[\mathbf{v}|k]
\end{aligned}$$

C.4. **Equation 19.** Define $J(D') \triangleq \mathbb{E}\left[\|(D'C - V_0)\mathbf{v}\|^2\right]$. We have

$$J(D') = \mathsf{tr}\left\{(D'C - V_0)(\mathbb{V}[\mathbf{v}] + \bar{\mathbf{v}}\bar{\mathbf{v}}^\mathsf{T})(D'C - V_0)^\mathsf{T}]\right\}.$$

The differential of the right-hand side is

$$dJ(D') = 2\mathsf{tr}\left\{dD'C(\mathbb{V}[\mathbf{v}] + \bar{\mathbf{v}}\bar{\mathbf{v}}^\mathsf{T})(D'C - V_0)^\mathsf{T}\right\},$$

giving the gradient

$$\nabla_{D'} J = 2(D'C - V_0)(\mathbb{V}[\mathbf{v}] + \bar{\mathbf{v}}\bar{\mathbf{v}}^{\mathsf{T}})C^{\mathsf{T}}.$$

This gradient is zero when

$$D = V_0(\mathbb{V}[\mathbf{v}] + \bar{\mathbf{v}}\bar{\mathbf{v}}^{\mathsf{T}})C^{\mathsf{T}} \left[C(\mathbb{V}[\mathbf{v}] + \bar{\mathbf{v}}\bar{\mathbf{v}}^{\mathsf{T}})C^{\mathsf{T}}\right]^{-1}.$$