Assistive sensory-motor perturbations influence learned neural representations

Pavithra Rajeswaran[1], Alexandre Payeur[2,3], Guillaume Lajoie[2,3], and Amy L. Orsborn[1,4,5,*]

[1]University of Washington, Bioengineering, Seattle, 98115, USA
[2]Université de Montréal, Department of Mathematics and Statistics, Montréal (QC), Canada, H3C 3J7
[3]Mila - Québec Artificial Intelligence Institute, Montréal (QC), Canada, H2S 3H1
[4]University of Washington, Electrical and Computer Engineering, Seattle, 98115, USA
[5]Washington National Primate Research Center, Seattle, Washington, 98115, USA
[*]aorsborn@uw.edu

ABSTRACT

Task errors are used to learn and refine motor skills. We investigated how task assistance influences learned neural representations using Brain-Computer Interfaces (BCIs), which map neural activity into movement via a decoder. We analyzed motor cortex activity as monkeys practiced BCI with a decoder that adapted to improve or maintain performance over days. Population dimensionality remained constant or increased with learning, counter to trends with non-adaptive BCIs. Yet, over time, task information was contained in a smaller subset of neurons or population modes. Moreover, task information was ultimately stored in neural modes that occupied a small fraction of the population variance. An artificial neural network model suggests the adaptive decoders contribute to forming these compact neural representations. Our findings show that assistive decoders manipulate error information used for long-term learning computations, like credit assignment, which informs our understanding of motor learning and has implications for designing real-world BCIs.

Introduction

We learn new motor skills by incrementally minimizing movement errors [1]. This process may require forming entirely new motor control policies [2], and involves computations distributed across cortical and subcortical circuits [3]. While motor computations are distributed, skill-related behavioral improvements have been linked to targeted neurophysiological changes [4, 5, 6]. Such coordinated circuit changes are thought to be mediated by a "credit assignment" process where movement errors lead to adjustment in select neural structures [7]. However, the precise mechanisms underlying the coordination of single neurons and populations are unknown, and may be difficult to detect with standard population analysis methods [8].

Intracortical brain-computer interfaces (BCIs) provide a powerful framework to study the neurophysiological signatures of motor skill learning and how task errors shape neural changes [9, 10]. BCIs define a direct mapping between neural activity and the movement of an external device such as a computer cursor, often referred to as a *decoder* [11, 12, 13]. Similar to other motor skills, practice in BCI produces improvements in performance that result from changes in neural activity [12, 13, 14]. The explicit relationship between neural activity and movement defined by a BCI decoder improves the ability to interpret neural activity changes during skill acquisition and makes it possible to causally interrogate learning. For instance, perturbing well-learned BCI mappings has revealed properties of neural mechanisms supporting adaptation within a single practice session [15, 16]. Alternately, mappings can be manipulated to purposefully improve performance [17, 18, 19, 20, 21, 22], akin to training wheels on a bicycle. These assistive perturbations improve behavioral performance, which actively alters

error feedback. The extent to which this co-opted feedback interacts with learning computations is not well understood. Here, we investigated how decoder manipulations that improve behavioral performance influence the formation of neural activity patterns during learning over many days of BCI practice.

BCI learning over multiple days has primarily been studied with a fixed decoder (Fig. 1A, left). When the decoder is fixed, task improvements must be the result of changes in neural activity. Practice with a fixed decoder over several days leads to performance refinements and the formation of stable neural representations at the single neuron level [14]. At the population level, this learning leads to more coordinated activity across neurons and progressive decreases in the dimensionality of these coordinated patterns [23, 24]. Learning-related changes in BCI are also targeted to select neurons. BCIs define a subset of neurons the decoder uses as the direct "readout" of behavior, in contrast to “nonreadout” neurons that may take part in circuit computations but do not directly influence movement. Extended BCI practice leads to preferential task modulation of readout neurons compared to nonreadouts, consistent with the possibility that credit assignment computations contribute to skill learning [24, 25, 26, 27, 28, 29]. Population-level changes, such as increased coordination among neurons, also appear to be primarily subserved by changes in readout neurons [24]. Together, these studies suggest that acquiring skilled motor BCI control leads to stable neural representations formed through a mixture of targeted and coordinated population-level changes.

Adaptive BCI decoders, in contrast, change in closed-loop with behavior (Fig. 1A, right). This closed-loop decoder adaptation (CLDA) can assist with initial decoder training [17, 30, 31, 32] , maintain performance despite non-stationary neural measurements [18, 20, 33], and has been proposed as way to assist or shape learning [19, 20, 21, 22]. Adaptive decoding has been used in many clinical BCI systems yielding high performance in specific tasks [29, 33, 34, 35, 36]. Adaptive decoders create a co-adaptive system where BCI performance changes result from a combination of brain and decoder adaptation [19, 20, 37]. Decoder adaptation updates the neural-activity–to-movement mapping to reduce movement errors, and can be used to compensate for unstable neural measurements by modifying which units are part of the readout ensemble [20] (Fig. 1B). Despite the dynamic mapping between neural activity and motor outcomes created by adaptive decoders, stable neural representations still form over time [20, 29]. While this suggests that adaptive decoders still allow motor skill acquisition, they could fundamentally alter the error information that is made available to the brain via sensory feedback. We therefore hypothesized that adaptive BCIs may influence learned neural representations, potentially via altering the targeted and population-level neural changes that occur during BCI skill learning.

To understand how decoder perturbations that improve or maintain performance influence skill learning, we analyzed data from our previous co-adaptive BCI learning study [20] and designed an artificial neural network (ANN) that closely models co-adaptive BCI experiments. Motivated by the learning dynamics observed in previous multi-day BCI studies, we used a combination of population- and neuron-level analysis methods. We found that adaptive decoding altered trends in neural population activity, leading to little change or increases in the variance and dimensionality of solutions with learning. Yet, we also found that the learned neural representations were highly targeted to a subset of neurons, or neural modes, within the readout population. To further elucidate this finding, we then used ANN models to compare learning dynamics and neural representations with and without decoder adaptation in a controlled and idealized setting. Doing so, we showed that decoder manipulations actively contribute to establishing the representations observed in our experimental data. Our combination of analyses and simulations revealed seemingly contradictory learning dynamics at the level of individual neurons and the population. Reconciling these perspectives revealed that the learning we observed in adaptive BCIs occurs largely in low-variance components of the neural population, which are often less studied and missed by standard population analysis methods [8]. Together, our findings provide direct evidence that adaptive decoders influence the *encoding* learned by the brain. This sheds new light on how manipulating error feedback shapes neural population activity during learning, which raises new considerations for designing BCIs.

Results

Two rhesus macaques (monkeys J and S) learned to perform point-to-point movements with a computer cursor controlled with neural activity (Fig. 1A), as previously described [20]. We measured multi-unit action potentials from chronically-implanted microelectrode arrays in the arm area of the primary- and pre-motor cortices. The binned firing rate (100ms) of a subset of recorded units, termed readout units, were fed to a Kalman filter decoder that output cursor positions (Fig. 1A-B). The subjects were rewarded for successfully moving from a central position to one of eight targets within an allotted time and briefly holding the final position (Fig. 1C).

Each monkey learned to control the cursor over several consecutive days, which we call a learning series. We focused on learning series lasting four days or longer ($N = 7$ for monkey J; $N = 3$ for monkey S). Learning series used different methods to train the initial decoder, varied in length, and included different numbers of adaptive decoder changes (CLDA events, see Methods). As previously shown [20], performance improved over multiple days, resulting in increased task success rates and decreased reach times (Fig. 1C, a representative series for monkey J; all subsequent single-series example analyses use this series for consistency). The decoder was adapted during learning to adjust the parameters ("weight change only", Fig. 1B) or to replace non-stationary units and update parameters ("readout + weight change", Fig. 1B). To quantify learning within each series, we defined an 'early' and 'late' training day, which correspond to the first and last day with at least 25 successful reaches per target direction, respectively (day 2 and 17 in Fig. 1C, D). To increase the number of trials to analyze on early days, we included any trial wherein the target was reached whether or not they held successfully once reaching the target. Improvements in task success rate and reach time were accompanied by straighter and more accurate reaches (Fig. 1D and [20]).

Adaptive decoders alter the evolution of neural population dimensionality during learning

We first studied the dimensionality of neural activity to assess population-level learning dynamics. Previous fixed-decoder BCI studies reported a decrease in 'shared' dimensionality with learning for both the readouts' activity [23] and nonreadouts [24]. Here, we quantified the overall population dimensionality using the participation ratio (PR), a linear measure derived from the covariance matrix [38, 39]. When $\mathrm{PR} = n$, neural activity occupies all dimensions of neural space equally; when $\mathrm{PR} = 1$, activity occupies a single dimension. To ease comparison across series, animals, neural populations and experiments, we normalized PR using the number of units to yield $\mathrm{PR}_{\mathrm{norm}}$, which varies between 0, when $\mathrm{PR} = 1$, and 1, when $\mathrm{PR} = n$ (see Methods).

We used existing data from [14], reanalyzed in [24], to confirm that BCI learning with a fixed decoder led to a decline in dimensionality for readouts as measured by $\mathrm{PR}_{\mathrm{norm}}$ (Fig. 2A). In contrast, we did not observe a decline with an adaptive decoder (Fig. 2B). Across series and animals, dimensionality was non-decreasing from early day to late day (Fig. 2C). We note that the time axes used to analyze the fixed and adaptive datasets are different. Due to poor performance in early days for the fixed decoder, we adopted the approach of grouping data into epochs, defined as segments with a consistent number of trials [23, 24]. The adaptive decoder led to sufficient trials in the early learning stages to track dimensionality trends across days. The trend of non-decreasing dimensionality for adaptive decoders were robust to changes in analysis, including square-root-transforming spike counts (to "gaussianize" them [40]; Fig. S1A), using other non-linear dimensionality metrics (Two NN [41, 42]; Fig. S1A), or considering only the readout units that were consistently used throughout the entire BCI series (stable readouts only) (Fig. S1B). We did not observe a consistent trend in the dimensionality of nonreadout units during learning, suggesting that learning-related changes were primarily targeted to the readout population (Fig. S1C).

We then aimed to understand potential drivers of dimensionality changes (or lack thereof) in the presence of decoder adaptation within each series. We found that the change in dimensionality was moderately correlated with the duration of training with an adaptive decoder (Fig. 2D). This result suggests that prolonged interactions with the changing decoder contributed to dimensionality changes. To better understand whether decoder adaptation events drove changes in dimensionality, we compared $\mathrm{PR}_{\mathrm{norm}}$ before and after each adaptation event. Importantly, the experiments include two types of decoder changes: those

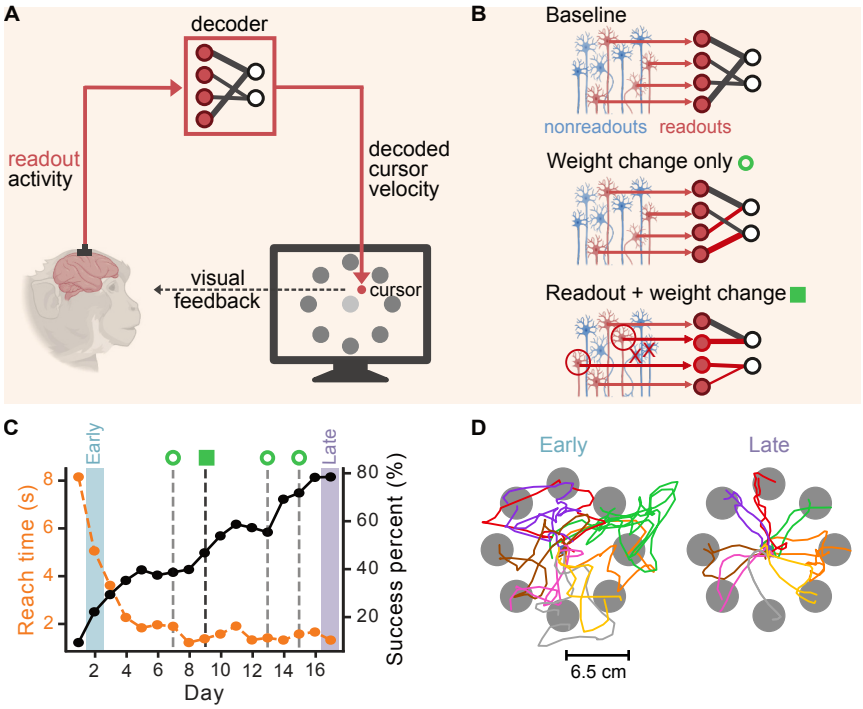

Figure 1. Experimental setup and example behavior (A) Schematic of BCI setup: Monkeys controlled a cursor in a 2D point-to-point reaching task using their neural activity, which was translated into cursor movements by a decoder, with visual feedback. **(B)** Schematic of BCI decoder mapping and adaptation. We recorded multiple units within the motor cortex: readouts (red) and nonreadouts (blue). The initial decoder mapped the activity of readout units into cursor velocities (baseline). During learning, the decoder was adapted in two ways: changing the weights relating readout units to velocity while keeping the identify of readout units constant (Weight change only), and altering both the contributing readout units and their corresponding decoder weights (Readout + weight change). **(C)** Success percentage (solid black line) and mean reach time (dashed orange line) across days for an example learning series for monkey J (jeev070412_072012). Early (blue) and late (purple) training phases are indicated. Vertical dashed lines indicate days where decoder was adapted - gray dashed lines (weight change only) and black dashed lines (readout + weight change). **(D)** Cursor trajectories during early (left) and late (right) training phases for the representative series in panel C. Monkey and neuron icons for schematics A and B are created with BioRender.com

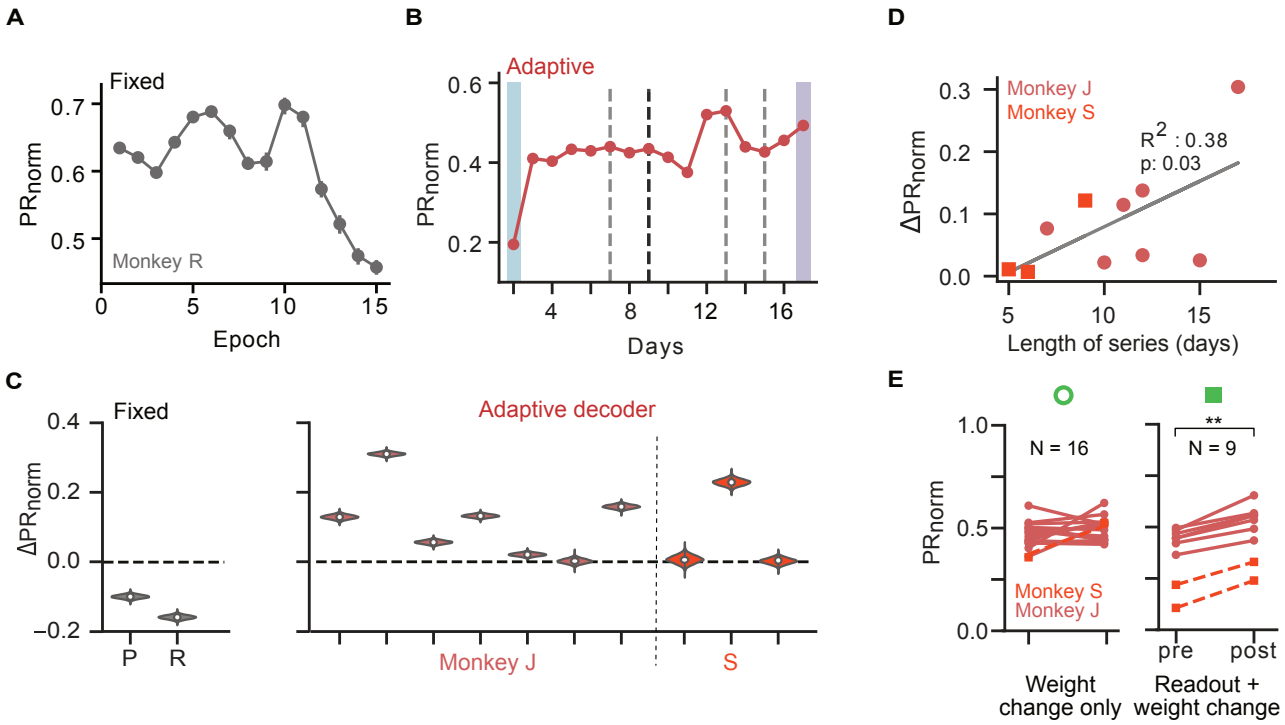

Figure 2. Assistive decoders influence neural population dimensionality (A) Normalized participation ratio (PR_{norm}) for fixed decoders (monkey R; data from [14]). Note: the x-axis denotes epoch, which are groups of constant number of trials, consistent with previous analyses [23]. Error bars are 95% confidence intervals estimated via bootstrapping (see methods). Note that most error bars are hidden by data point markers. **(B)** Same as (A) for an example adaptive decoder learning series (same as in Fig. 1D). **(C)** ΔPR_{norm} distributions for learning series with fixed decoders (N = 1 for monkeys P and R) and adaptive decoders (N = 7 for monkey J and N = 3 for monkey S). **(D)** PR_{norm} as a function of training duration in days. **(E)** PR_{norm} before and after decoder adaptation, grouped by the type of change (with (right) and without (left) readout unit changes) across all series and monkeys (Weight change only: $p = 0.16$ (ns), $N = 16$; readout + weight change: $p = 0.0039$ (**), $N = 9$, PR_{norm} for pre $<$ post, one-sided Wilcoxon signed rank test).

where the units remain the same (weight change only), and those where the readout units were changed along with the decoder weights (Fig. 1B). We found that decoder adaptation events led to significant increases in dimensionality, but only when readout units were changed (Fig. 2E, right). This effect, however, was not observed when considering only the stable readout ensemble, which controls for changes in the readout ensemble membership (Fig. S1D). Together, these results reveal that decoder adaptation influences the evolution of neural population dimensionality during learning and highlight the importance of readout unit identity (and changes thereof) in learning dynamics.

Learning leads to credit assignment to readout units

Assistive decoders altered population variance over time, which suggests that these perturbations may change the computational demands placed on the brain as it acquires the task. Variance alone, however, may not capture the extent of the relationships between task variables and neural activity. Further probing neural representations during skill acquisition, however, is challenging because the behavior also changes (e.g., kinematics of trajectories change dramatically, Fig. 1C,D). We therefore used an offline classification analysis to quantify how target information (which remains constant) is distributed within neural activity during learning. As further outlined below, we wished to derive a quantity that would act as a proxy to the information contribution of neural activity to the task, i.e. it's assigned credit, that is disjoint from population variance and the effective decoder used for the task. To this end, we used multiclass logistic regression to predict target identity on each trial using trial-aligned neural activity (Fig. 3A; Methods). The classifier was trained anew each day and all classification accuracies were

computed using cross-validation (Methods). We note that, while powerful, the offline classification analyses did not extend as readily to fixed decoder datasets, due in part to limited numbers of trials, and we therefore focus on adaptive decoder datasets hereafter. Instead, we probed the fixed decoder paradigm exhaustively using closely-related analyses within our ANN model, leading to important validations and insights.

The logistic regression model weighted the contribution of each unit's firing rate towards classifying target identity. These weights thus represent an encoding of task information, and are partially related to parametric analyses of neuron activity such as direction tuning (Fig. S2B). We found that classifier weights varied day to day, but became more similar as learning progressed (Fig. 3B). The correlation coefficient of the weights of stable readouts on consecutive days increased during learning (Fig. 3C), consistent with a progressive stabilization of task encoding representations that mirrors performance improvements (Fig. 1D). These findings are consistent with previous observations that co-adaptive BCIs lead to stable unit-level activity in readout neurons [20].

We then used offline classification to assess credit assignment by examining how task information was distributed between readout and nonreadout neural ensembles. The classification accuracy of target identity increased during BCI training when using all neurons, mirroring the improvement in BCI performance (Fig. 3D). However, this improvement in classification accuracy was mostly driven by the readout units (Figs. 3D, E). Across series, classification accuracy was higher for the readout units compared to the nonreadout units, both early and late in training. More importantly, only the readouts' accuracy significantly increased from early to late (Figs. 3E; Fig. S2E for individual animal results), showing that learning-related changes in classification accuracy are driven predominantly by changes within the readout population. The distinct difference classification performance between the two neural populations is particularly striking when considering that there the experiments used notably fewer readout neurons than nonreadouts (Monkey J: 15-16 readouts, 29-41 nonreadouts; Monkey S: 11-12 readouts, 66-121 nonreadouts). Similar results were obtained with other classification models, such as Support Vector Machines, Naive Bayes, or logistic regression with different penalties (Fig. S2A). To rule out the possibility that these differences in classification accuracy simply reflected other factors such as neural recording quality, we performed the same analysis on data recorded when Monkey J performed the same task via arm movements, which showed no difference between the readout and nonreadout populations for size-matched neural populations (Fig. S2C, D). Thus, despite altered population variance, assistive decoder perturbations lead to the formation of a stable neural encoding that is primarily instantiated by readout units, similar to BCIs with fixed decoders [14, 25]. This further shows a coarse-grained signature of credit assignment computations relevant to behavior acquisition.

Learning leads to compaction of neural representations

Our classifier analysis revealed a striking pattern: within a given readout ensemble, a small number of readouts carried the majority of weight for target prediction (Fig. 3B, bottom row). This suggested credit assignment processes may also occur within the readout population. To quantify whether specific readouts contained more task information than others, we conducted a rank-ordered neuron adding curve analysis [43]. This technique quantifies the impact of each unit's inclusion to the overall classification performance to identify the most influential neurons. On each day (early, late), we first quantified the classification accuracy of a unit in isolation, and then constructed a rank-ordered neuron adding curve by adding units to our decoding ensemble from most to least predictive (Fig. 4A).

As expected given overall improvements in classification, individual unit classification accuracy improved between early and late learning (Fig. 4B). Beyond these global shifts, the relationship between classification accuracy and the number of ranked readout units used for prediction changed with learning. To visualize this, we normalized each day's neuron-adding curves, which revealed that the classification accuracy reaches plateau more rapidly late in learning compared to early (Fig. 3C, see Fig. S3A for non-normalized curves). We quantified this effect by computing the number of neurons needed to reach 80% of the normalized classification accuracy (N_c) at each learning stage (Fig 4C, D). We found that fewer readout neurons were needed to

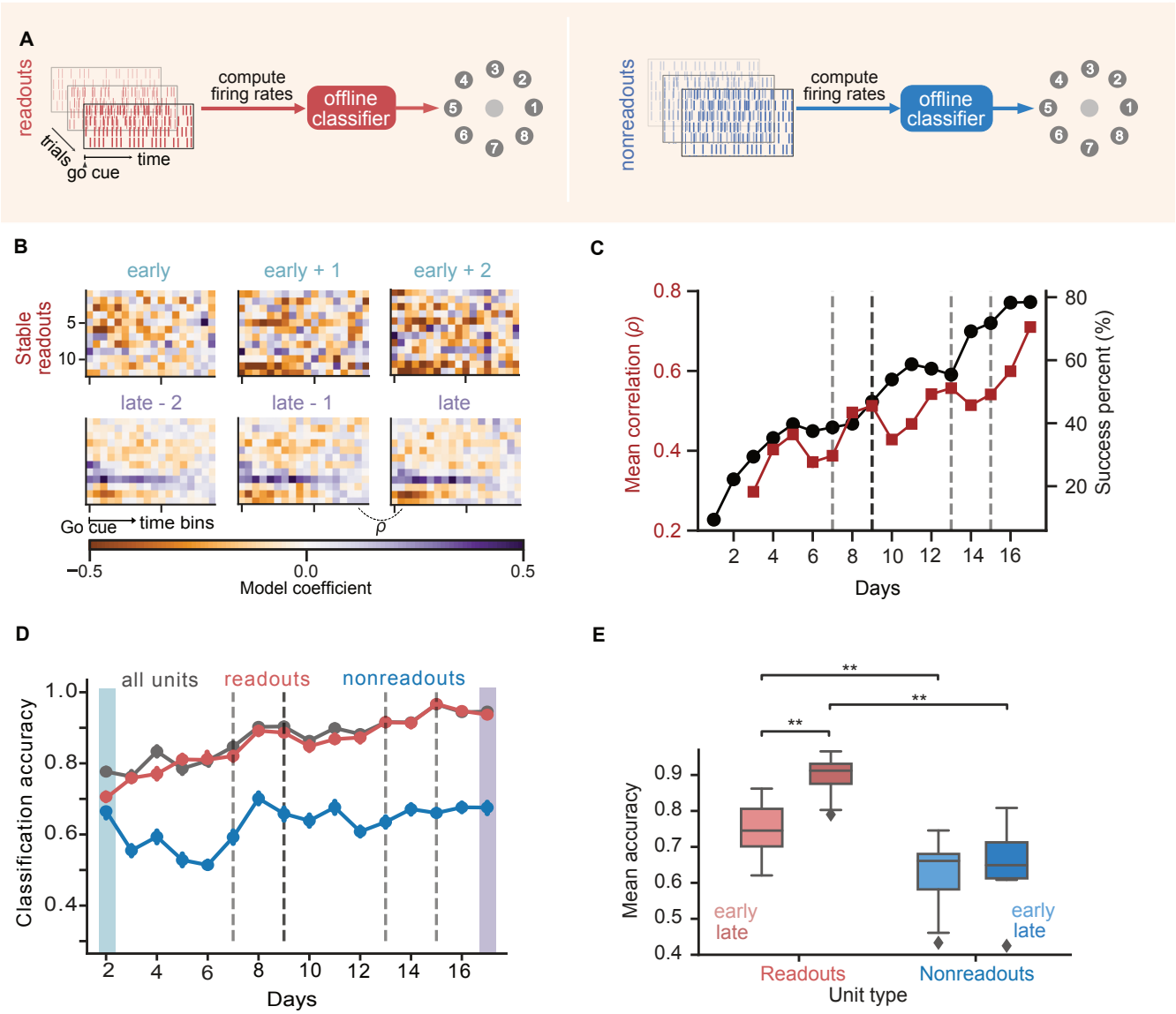

Figure 3. Credit assignment to readout units. (A) Schematic of offline classification model. **(B-D)** Classifier analyses results for the representative series (Fig. 1D). **(B)** Heatmap representation of the classifier weights to predict a single target (2) for the stable readout population on multiple days. **(C)** Success percentage (black; reproduced from Fig. 1D) and mean correlation coefficient (dark red) between classifier weights on consecutive days plotted across days. **(D)** Classification accuracy for all units (gray), readout units (red) and nonreadout units (blue) across training days. Error bars represent 95% confidence intervals on test accuracy (see Methods). **(E)** Early vs late classification accuracy for readout (shades of red) and nonreadout populations (shades of blue) across all series for both monkeys (readout-early vs readout-late : $p = 0.0058$ (**); readout-early vs nonreadout-early: $p = 0.0019$ (**); readout-late vs nonreadout-late: $p = 0.0019$ (**); nonreadout-early vs nonreadout-late: $p = 0.13$ (ns). For all comparisons in this panel, $N = 10$, Two-sided Wilcoxon Signed Rank test)

accurately predict task behavior across all learning series (Fig 4D). This phenomenon was even larger when performing the same analysis on all recorded units (readouts and nonreadouts, Fig. S3B). Shifts in the number of neurons needed to accurately predict behavior were not observed in arm movement tasks, suggesting changes were specific to BCI learning and not related to recording quality or drift (Fig. S3C). Interestingly, beyond this within-series (early vs. late) trend, we noticed a general decrease in N_c across chronologically-ordered series for each animal (Fig. S3D). We term this reduction in the number of neurons needed to accurately decode behavior over time 'compaction', to indicate the brain forming an encoding of task information with a smaller number of neurons.

We then asked whether the compaction of neural representations reflected a situation where task information was encoded within a specific subset of readout neurons, or a more general increase in encoding efficiency. We performed a combinatorial variant of the neuron-adding curve analysis, in which we computed classification accuracy for all combinations of units as we vary the number of units used to decode (within readout units only). On each day, we identified the top N_c readout units and labeled combinations that contained all top units (purple), did not contain any of the top units (orange), combination that contained some of the top units (gray) (Fig 4E,F). Early and late in learning, combinations with top units held the most classification power (Fig 4E, F). Comparing the distributions between early and late, however, reveals that the overall increase in classification power was driven solely by the increase in classification power from top units. We quantified the performance gap between combinations with or without the top N_c^{late} units using a discriminability index (d'), which quantifies the impact of the most influential units on overall accuracy. This performance gap increased with learning across series (Fig 4G), showing that learning-related changes in target encoding are targeted to a specific subset of readout neurons.

Assistive decoder adaptation contributes to learning compact representations: modeling analysis

Our analysis of neural data revealed that BCI training with assistive decoder perturbations led to compact neural representations over multiple days. However, it is unclear whether these representations reflect a general property of sensorimotor learning, or if the adaptive decoder played a causal role in shaping learned representations. Furthermore, current BCI experimental data may miss potentially confounding population-level mechanisms due to practical limitations like sparsely sampled neural activity and an inability to measure *in vivo* synaptic strengths. To explore these possibilities and causally test the influence of assistive decoders, we simulated BCI task acquisition using an artificial neural network (Fig. 5A). With this model, we could directly compare the BCI learning strategies for identically initialized neural networks when training with a fixed or an adaptive decoders. The precise question was whether compact representations still emerge without decoder adaptation.

The model was composed of a recurrent neural network (RNN) representing a motor cortical circuit and receiving sensory and task information (Fig. 5A). Our goal was to study the simplest model consistent with biology and that reproduces basic learning dynamics of BCI experiments. We first trained the network on an arm movement center-out task (Fig. 5B, left) to establish a plausible inductive bias for the subsequent BCI training and to mirror the experimental approach which consisted in training the subject on arm-based tasks prior to BCI training [20]. Context information about the nature of the task (arm vs BCI) was provided to the network together with information about the position of the effector and of the target. After the initial arm movement training, the context was switched to 'BCI', a random subset of units were selected to be readouts, and a velocity Kalman filter (see Methods) was trained on manual reach trajectories (Fig. 5B, middle). Finally, the network was trained on the BCI task (Fig. 5B, right). In all contexts the REINFORCE algorithm [44] was used to update the parameters of the recurrent neural network and of the input layer (matrix U in Fig. 5A); all other parameters (encoding matrix F and output matrix V) were randomly initialized and remained fixed. REINFORCE does not rely on the backpropagation of task errors through the BCI decoder or the arm model, thus making it more biologically plausible.

Assistive decoding in the model followed CLDA algorithms used in experiments [20]. We virtually rotated actual cursor velocities towards the target position (under the assumption that this corresponds to the user's intention), refit the decoder, and then updated decoder parameters (see Methods). We did not change the readout ensemble during BCI training (Fig. 1B,

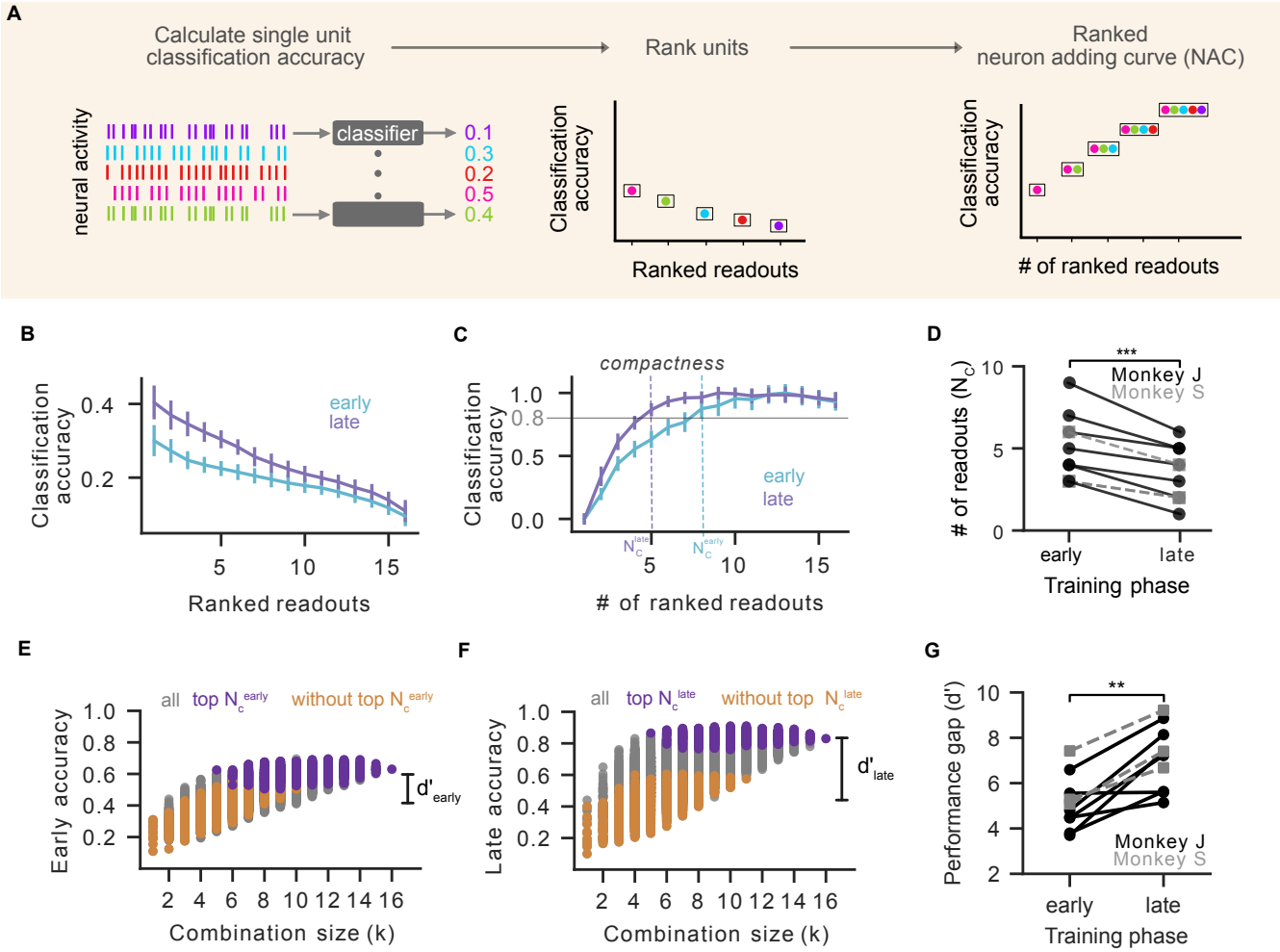

Figure 4. Learning leads to compaction of neural representations: **(A)** Schematic of the ranked Neuron Adding Curve (NAC) analysis. We first estimated classification accuracies of individual units (left), ranked units based on their classification power (middle) and then constructed a neuron adding curve by starting with the best unit, adding the second best unit and so on (right) (see Methods). **(B)** Ranked classification accuracy of individual units on early (cyan) and late (purple) day for the representative series. **(C)** Ranked NAC on early (cyan) vs late (purple) day for the representative series. Each day's classification accuracy was normalized to the day's peak performance. Vertical dashed lines indicate number of units required to reach 80% of peak accuracy (N_c). **(D)** Comparison of N_c^{late} and N_c^{early} across series ($N_c^{\text{late}} < N_c^{\text{early}}$, $p = 9.8 \times 10^{-4} (***)$, $N = 10$, one sided Wilcoxon signed-rank test). Monkey J data shown in black circles; monkey S shown in grey squares. **(E)** Classification accuracy as a function of readout population size (combinatorial NAC) for early day. Combinations with top N_c^{late} units (deep purple) and without top N_c^{late} units (orange). Distance between distributions defined as the discriminability index d'. **(F)** Same as (E) for late day. **(G)** Comparison of distance between distributions between early and late learning (d'_{early} vs d'_{late}) across series ($d'_{\text{late}} > d'_{\text{early}}$, $p = 1.9 \times 10^{-3} (**)$, $N = 10$, one sided Wilcoxon signed-rank test). Formatting as in panel D.

middle) to focus on the impact of decoder weight changes on learned representations. CLDA was performed on each "day", with a day corresponding to seeing each target 100 times. Irrespective of the CLDA intensity, an adaptive decoder sped up performance improvements in the first days compared to a fixed decoder (Fig. 5C). However, CLDA did not necessarily improve the final BCI performance (Fig. 5D). A possible explanation is that CLDA's objective, "reach towards the target", was conflicting with the task objective, "reach the target with vanishing velocity in a specified time" (Methods, Eqs. 4-5). Towards the end of training, the task objective was nearly completed and therefore large decoder alterations negatively impacted performance. We reasoned that stopping CLDA when a given performance threshold has been attained could help recover good end-of-training performances. Using such protocol, we obtained equal or better end-of-training performances for all CLDA intensities (Fig. S4A). Overall, decoder-weight adaptation in the model facilitated acquisition of the BCI task, as in experiments, and underscored potential shortcomings of its undiscerned long-term application.

Next, we explored whether representations became more compact with learning in the model. While this property of neural representations had to be assessed offline in the data (Fig. 3), the model allowed for an online computation using the loss (Methods, Eqs. 4-5) as the performance metric. Each individual readout unit was used to move the cursor in turn and reach performance was evaluated both 'early' (at the end of the first day) and 'late' (at the end of the last day) during training. Readout units were then ranked according to their reach performances. Differences between the early and late individual unit normalized performances (Methods, Eq. 8) appeared only for adaptive decoders (Fig. 5E and Fig. S4C,E), with the ranked contribution of each unit decreasing faster late in training compared to early (Fig. 5E, pink lines). Thus, at the individual unit level, the dominant units became relatively more dominant on average under co-adaptation of the network and decoder parameters. We then used these individual unit rankings to compute a ranked NAC (Methods, Eq. 9). Consistent with our findings based on experimental data, adaptive decoders contributed to evoking more compact representations of task performance (Fig. 5F and Fig. S4D). With a fixed decoder, the ranked NAC for the early and late stages had a large degree of variability across seeds, and there were no signs of increased compactness late in training as 11 ranked readout units were required to reach 80% of the maximal performance (Fig. 5F, left). Our model thus suggests decoder-weight changes single-out certain units for BCI control and progressively shape more compact task representations.

As mentioned above, stopping CLDA early in training helped recover good performances. Stopping CLDA early also tended to prevent the emergence of compact representations (Fig. S4B). Using the online loss metric, we also obtained combinatorial unit-adding curves (Fig. 5G, left) and computed the corresponding interquartile ranges (Fig. 5G, right). The dispersion of normalized performances (Methods, Eq. 10) was significantly greater for a fixed decoder, and was caused by few units with very low contributions to the overall performance (Fig. 5E and Fig. S4C). Therefore, while decoder adaptation contributed to compact representations characterized by only a few dominant units (with potential impacts on the long-term robustness of BCIs; see Discussion), it also seemed to protect against these unreliable units.

Both in the model and in the data, BCI learning with an adaptive decoder evoked compact representations of task information. Moreover, the model suggests that neural plasticity alone (i.e., learning the network parameters with a fixed decoder) does not produce compact representations on average. This suggests that decoder adaptation must interact with brain plasticity to shape this feature of neural representations. To show this, we calculated the total changes in recurrent weights across learning, ΔW, with and without decoder adaptation, and computed their coefficients of determination (Fig. 5H). All connections—among and across readout and nonreadout units—were included, because all connections participate in the network dynamics (Methods, Eq. 3). Since our simulations were precisely matched in terms of random number generation across CLDA intensities, comparing these weight changes one-to-one was meaningful. The coefficients of determination decreased when the CLDA intensity increased, meaning that the total weight changes with the adaptive decoder became increasingly uncorrelated from the total weight changes with a fixed decoder. Overall, these results suggest that brain-decoder co-adaptation elicits specific plastic changes in neural connections which underlie the emergence of more compact representations.

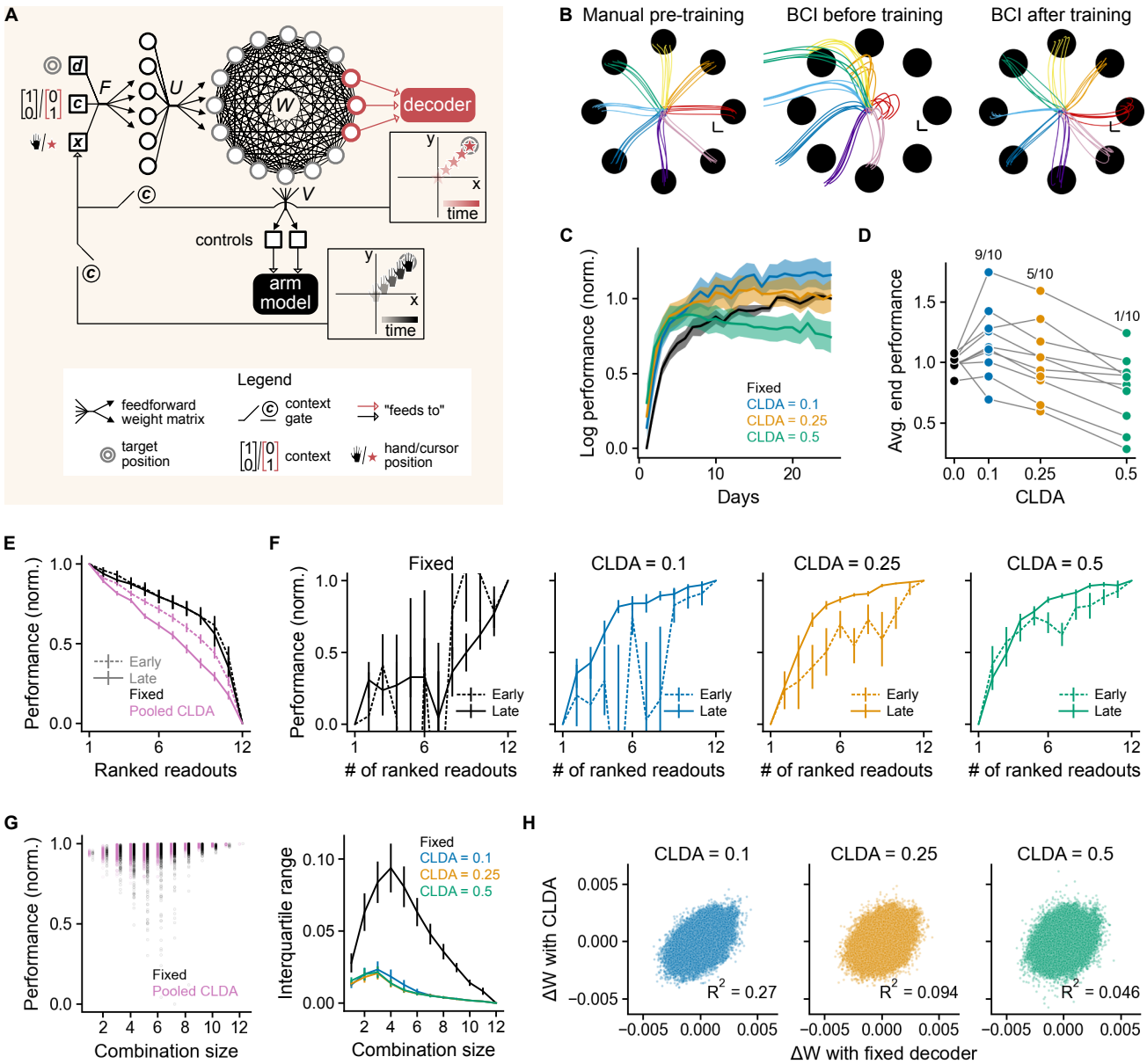

Figure 5. Decoder adaptation contributes to learning compact representations: modeling analysis **(A)** Model schematic. Input (target position **d**, context information **c** and visual feedback **x** about the cursor/arm state) was encoded using a random linear projection *via* matrix F. Outputs of the recurrent neural network (with input, recurrent and output weight matrices U, W and V, respectively) were controls that drove a two-link planar arm model. Sensory information came from either the BCI cursor or the model arm, depending on the state of the context gates. **(B)** Learning stages. The network (here with CLDA intensity = 0.25) was first trained on the arm movement task (left), then decoder parameters were fitted using arm trajectories (middle) and, finally, the context was set to 'BCI' and the network was trained on the BCI task (right). Scale bars are 1 cm. **(C)** Normalized log performance (Methods, Eq. 7) for different CLDA intensities. **(D)** Normalized log performance averaged over last 5 days. Ratios at the top are the fraction of seeds for which the performance with the adaptive decoder was greater than with the fixed decoder. **(E)** Ranked individual-unit normalized performance. Results for adaptive decoders were pooled together (pink lines). **(F)** Unit-adding curves analogous to NAC but evaluates online performance in model as units are added. **(G)** Combinatorial unit-adding curve for a given random initialization of the network parameters (left) and interquartile ranges of the distributions across realizations (right). **(H)** Change in recurrent weights (ΔW) with CLDA as a function of the corresponding change with a fixed decoder. In panels C, E, F and G (right), we plotted mean $\pm$ SEM, for $n = 10$ seeds, i.e. random initializations of the network.

Reconciling population and unit-level perspectives: task information emerges in low-variance modes

Our model results showed that plasticity was altered by the presence of assistive decoders, leading to the formation of neural representations that become compact at the unit-level. However, our data analysis also revealed that the assistive decoder influenced learning phenomena at the level of neural populations, eliminating (or even reversing) reductions in neural population dimensionality that are often observed in sensorimotor learning (Fig. 2). How is it possible for task representations to become contained within a small number of neurons without decreasing the population dimensionality? To reconcile these two seemingly-conflicting descriptions of learning trends, we developed and performed a series of analyses to link population-level descriptions of neural activity to task information and neurons.

We first asked whether the phenomenon of compacting representations was specific to unit-level representations, or would also be observed within population-level descriptions of neural activity. We performed a variation of our neuron-adding curve analysis where we used population-level "modes" as the relevant neural features for classification instead of individual units. We estimated population-level modes for each day (early, late) using principal components analysis (PCs) (see Methods) and then used neural activity projected onto these modes for classification (Fig. 6A). We ranked PCs based on their classification accuracy (Fig. 6B) and performed a rank ordered PC adding curve (Fig. 6C). Similar to the compaction observed at unit level, We found that the number of PCs required for 80% classification accuracy decreased with learning (Fig. 6C,D). These results show that learning leads to the formation of task representations that are compact in terms of individual units *and* population modes, an observation that is not trivially true in general (see Supplementary Discussion).

We then aimed to understand how representations can more compactly represent task information without necessarily decreasing in dimensionality. We did this by linking variance captured by population-level modes with task information across learning. As learning progressed, variance was more distributed across modes (Fig. 6E, top), as expected from dimensionality analyses. Yet, if we examine classification accuracy, we see that modes which capture the largest variance of population activity were not necessarily strongly predictive (Fig. 6E, bottom). To quantify the relationship between variance-explained and task information for population modes, we divided PCs into two groups for each learning series: the top 50%, which explained approximately 80-84% of population variance, and the bottom 50%, which explained 16-22% of population variance. Across all learning series, the average classification accuracy on late days was larger in the bottom 50% group compared to the top 50% group (Fig. 6F), showing that task information was more often found in low-variance modes after learning. Moreover, changes in task encoding during learning were not correlated with the variance explained by that mode (Fig. 6G). Thus, adaptive decoders lead to the formation of compacted representations that do not decrease in overall population dimensionality because task-relevant information becomes embedded in the low-variance modes.

Lastly, we examined whether, after learning, task-predictive units contributed preferentially to task-predictive population modes. We ranked readouts based on their single unit predictive power and then quantified their influence on PC modes using the squared PC loadings (see Methods). Comparing PC loadings for an example high-variance PC with those of a strong decoding PC revealed starkly different patterns of unit contributions (Fig. 6H). As expected, strongly decoding PCs receive large contributions from strongly decoding units. In contrast, high variance PCs have contributions from a seemingly random mix of units, both task decoding and not. To quantify this effect across all learning series, we calculated the sum of squared PC loadings from the top N_c task predictive units vs. bottom N_c task predictive units for the strongest decoding PC and highest variance PC in each series (Fig. 6I). Strongly decoding PCs consistently drew more from the top-performing units, with negligible input from the least predictive units, a pattern not observed in high variance PCs (Fig. 6I). This suggests task information is encoded in low variance population modes, and that population activity dimension might not be indicative of coding mechanisms, especially in the presence of significant credit assignment learning phenomena like those found with with assistive decoders.

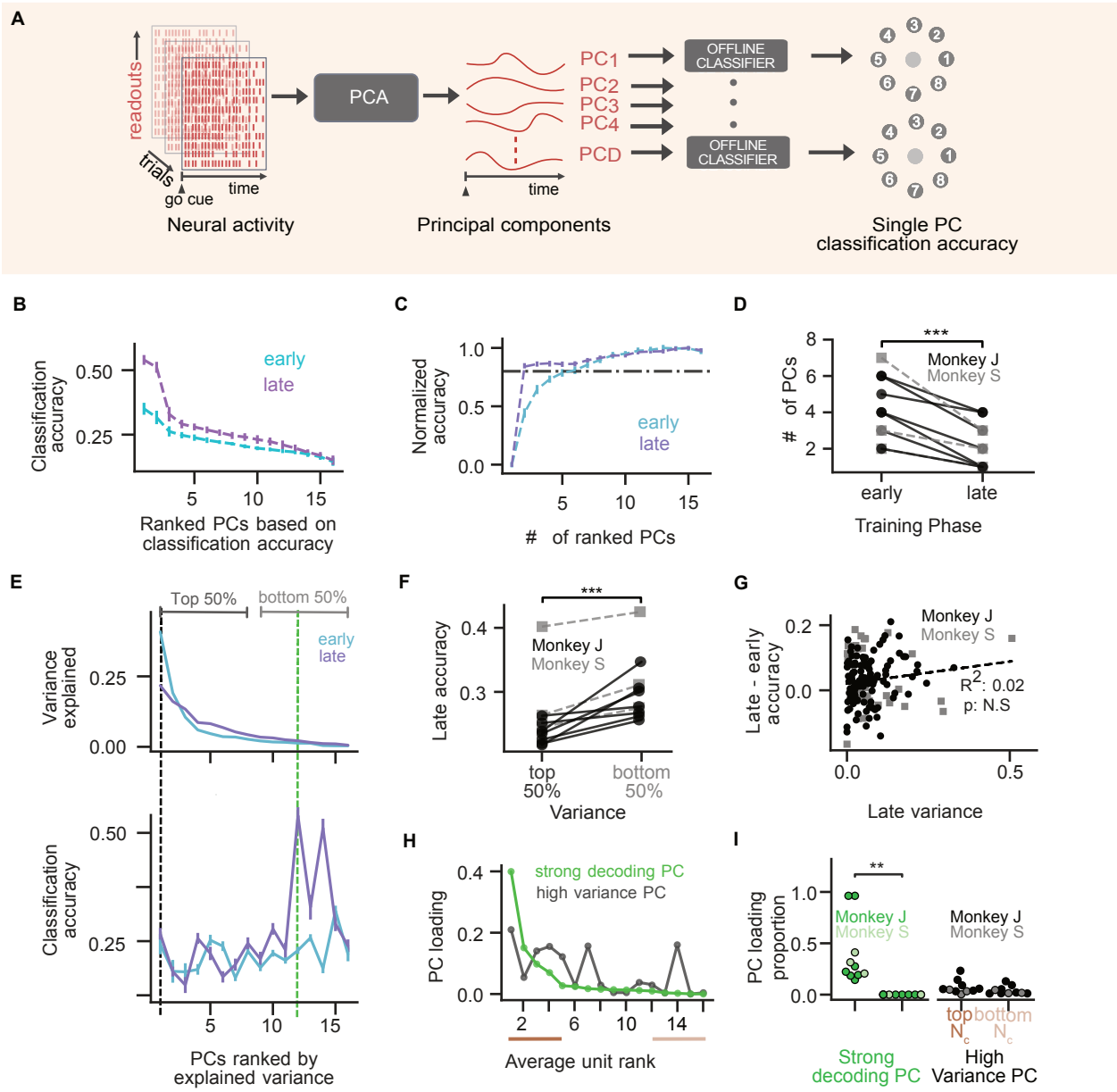

Figure 6. Task information emerges in low-variance modes **(A)** Schematic of decoding analysis with PC modes. **(B)** Individual PC classification accuracy (similar to Fig. 4B) for early (cyan) and late (purple) learning phases from representative series. PCs ranked by classification accuracy. **(C)** Ranked dimension adding curve (similar to ranked NAC in Fig. 4C) for early and late days for the example series. **(D)** Comparison of number of PCs needed for 80% classification accuracy on early and late days across series(late $<$ early, , ($p = 9.7 \times 10^{-4}$, $N = 10$), one sided Wilcoxon signed-rank test). Monkey J data shown in black circles; monkey S data shown in grey squares. **(E)** Top: Variance explained by each PC (ranked by variance explained). Bottom: Classification accuracy per PC, ordered by variance explained, Error bars denote 95% confidence intervals from 10^4 random draws of trials. The strongly decoding PC (green) has the highest accuracy. **(F)** Comparison of mean classification accuracy from top 50% vs bottom 50% PCs during late learning day across series (Top 50% is lower than bottom 50%: $p = 0.001$, $N = 10$, one sided Wilcoxon signed-rank test). Data formatting as in D. **(G)** Change in classification accuracy (late - early) plotted versus late variance per PC. No significant trend ($R^2 = 0.01, p = 0.26, N = 10$). Data formatting as in D. **(H)** Square of PC loadings from each readout unit for the strong decoding PC (green) and high variance PC (black). Units on the x-axis are ranked by their average classification performance. **(I)** Comparison of sum of squares of PC loadings from top 4 ranked readouts and bottom 4 ranked readouts from strong decoding PC and high variance PC across all series. (strong decoding PC: $p = 0.0019(**), N = 10$; high variance PC: $p = 0.3750(n.s), N = 10$, two sided Wilcoxon signed-rank test).

Discussion

Our study reveals that learning with assistive sensory-motor mappings in brain-computer interfaces (BCIs) influences the neural representations learned by the brain. Specifically, we found that assistive decoder perturbations led to neural representations that compact task information into a small number of neurons (or neural modes) that capture a relatively small amount of the overall neural population variance. Our analyses reveal new insights into the neurophysiological changes underlying skill learning with adapting sensory-motor maps, and highlight the complex mixture of targeted neuron-level and distributed population-level phenomena involved. These findings shed light on the neural computations driving skill acquisition. Moreover, our results expose a critical entangling of how neural populations "encode" task information and the algorithms we use to "decode" that information, which has important implications for designing BCI therapies.

Neural variability and the encoding of task information within low-variance neural modes

A hallmark of motor learning is gradual reduction in behavioral variability that correlates with a reduction in neural variability [45]. BCI learning with fixed decoders seems consistent with this characterization [23]. Factor analysis indicates that the amount of neural variance that is “shared” among neurons increases with learning while private, unit-specific, variance decreases [23, 24]. Moreover, both the shared dimensionality—which is based on shared covariance only [23, 24]—and the overall neural dimensionality (Fig. 2C, left) tend to decrease. We found that BCI learning with adaptive decoders leads to departures from these trends in dimensionality (Fig. 2C, right). Preliminary results also suggest that private variance, but not shared variance, might behave differently for fixed- and adaptive-decoder BCI learning [46]. Therefore, it appears that assistive decoders may engage additional learning processes at the level of individual neurons that support the non-decreasing dimensionality.

Why does the variability of individual neurons become important in BCI skill acquisition with an adaptive decoder? The participation ratio (our dimensionality metric) and other variance-based analyses provide limited insight because they do not take task information explicitly into account. In contrast, the offline classification of target identity with neural data (Fig. 4) and the online assessment of neuronal contributions to control in the model (Fig. 5) integrate such information. These revealed that adaptive decoders strongly influence which neurons participate in the task, leading to a more exaggerated credit assignment to fewer neurons. Moreover, offline classification using population activity modes suggested why dimensionality does not decrease: late in learning, task-relevant information was preferentially contained within modes that capture relatively low variance in the neural population (Fig. 6). While variance-based analyses have uncovered important aspects of neural computation, they may overlook key mechanisms that operate locally and in low-variance modes [8].

We cannot yet identify the mechanisms leading the brain to learn this type of representation, but recent studies offer possible hypotheses. Adapting to altered dynamic environments in arm movements produces “indexing” of stored motor memories within motor cortex activity in non-task-coding dimensions [47]. Do adaptive BCIs drive the brain to store motor memories in spaces that are distinct from other tasks? This possibility is consistent with our finding that dimensionality tends to increase with longer series, where more decoder perturbations occur (Fig. 2D). Previous studies also propose that sparse, high-dimensional neural representations enhance computational flexibility [48, 49, 50]. Our paired observations of increasingly compact representations that do not decrease in dimensionality suggests that adaptive BCIs may increase pressure on the brain to learn efficient and flexible solutions. A testable prediction of this hypothesis could be that the neural activity of natural motor control interferes less with learned BCI representations emerging from adaptive mappings compared to fixed mappings.

Signatures of learning at multiple levels

We investigated how task information is represented at the level of neural ensembles, population modes and multi-unit activity, and we examined relationships between these representations. As in prior work [24, 25, 26, 27, 28], we observed learning processes specifically targeted the readout ensemble, which we interpret as a form of credit assignment (Fig. 3D,E). Within the readout ensemble, credit was assigned more strongly to specific units (Figs. 4, 5). Representations also became more compact with learning when we considered a population-level description of readout activity involving principal components

rather than units (Fig. 6B-D). Importantly, population-mode compactness does not trivially follow from unit-level compactness in general (see Supplementary Discussion), despite the dominant units contributing to the strongly-decoding low-variance modes (Fig. 6H-I). These results suggest that the learning and credit-assignment computations we observe during extended BCI practice operate at multiple levels, from ensembles to units and to population modes. How these different mechanisms interact remains an open question.

BCI perturbations that disrupt performance have revealed related observations of neural changes at multiple levels. In some experiments, the preferred decoding directions of a subset of the readout units were artificially rotated, effectively implementing a visuomotor rotation [15, 28]. Adaptation to the perturbation was first dominated by global changes to neural activity which affect both the rotated and non-rotated units [15], while more local changes appeared over time on top of these global changes [28]. In other experiments, the BCI was designed to control the cursor via shared population factors and the perturbation could be either aligned or not with this mapping [16]. Aligned perturbations are learned on short time scales (within a day) because neural correlations can be repurposed rapidly [51]; unaligned perturbations can be learned only over many days because some degree of neural exploration outside the correlation structure is required [52]. These experiments suggest a strong link between the training duration after a perturbation and the neural representation level at which learning occurs.

The analyses used in our study cannot precisely resolve timescales of learning computations, in part because of the inherent difficulty to quantify changes in neural activity during skill acquisition. Indeed, existing analysis methods are often designed to leverage assumptions about common structure in neural dynamics invariant over time [53]. In experiments involving disruptive perturbations of well-practiced skills, the consequences of the perturbation can be directly compared to representations acquired before the perturbation. Such shared structures stem in part from consistency in behavior [54, 55], which is not necessarily true when a new ability is being acquired (e.g., Fig. 1C,D) and the differences we seek to measure are themselves changes over time. To circumvent some of these difficulties, we have focused on the encoding of target identity, which does not vary from day to day (in contrast to cursor trajectories) and thus facilitates a fine-grained comparison of early and late representations (Figs. 4 and 6). However, we did not test whether this method could yield significant results at finer temporal scales. Ultimately, a better understanding of the neural computations underlying skill acquisition, with or without assistive devices, will likely require developing new analytical approaches.

Adaptive decoders shape learned representations via error manipulation

Our findings add to our understanding of how the *decoders* we use in a BCI shape the *encoding* learned by the brain. Past work highlights how decoders can influence the learning demands placed on the brain. For example, the brain may only reorganize neural activity patterns when the decoder is changed in such a way that violates existing neural population correlations [16, 52]. We find that assistive decoder changes also influence the learned neural activity patterns. Yet, these two decoder manipulations influence task error differently, suggesting they may impact learning in distinct ways. Most studies perturb decoders to introduce new errors [15, 16, 28, 51, 52]. The form of these perturbations influences the error available to the brain, which modeling suggests may contribute to differences in how these perturbations are learned [56]. In contrast, our experiments incorporate decoder perturbations that either maintain or improve task performance in closed loop [20]. Our results highlight that BCI decoder changes, even if they do not introduce *task* errors, represent external error manipulations that interact with the brain's innate error-driven learning computations. This makes a prediction that the brain uses internally-generated error signals, generated through mechanisms like an "internal model" [57], to shape neural representations.

Our model revealed that adaptive decoder error manipulations influence synaptic-level changes within the network (Fig. 5H). While we cannot yet experimentally validate this prediction, motor learning studies provide evidence consistent with this perspective. Acquiring motor and BCI skills is generally thought to involve multiple learning processes, one that is "fast" (within a training session) and one that is "slow" (occurring over many days of training) [1, 3, 28, 58]. Evidence also suggest that these processes have different error sensitivities [59]. The neural mechanisms associated with fast and slow learning process

are not fully understood, but recent work suggests that fast adaptation may not require significant modification to synaptic connections [51, 60]. One possibility is that initial performance enhancements from adaptive decoders effectively supplement or even supplant the brain's rapid learning mechanisms, while slow mechanisms that shape synaptic changes remain and are impacted by the algorithm. BCI learning targeted to specific neural populations are most often observed after multiple training sessions [25, 26, 28], potentially as part of "slow" learning mediated via synaptic plasticity [61]. We observe that adaptive decoders clearly alter these slower credit-assignment processes. BCI experiments with systematic decoder perturbations will provide new avenues to fully dissect the neural computations involved in learning, their error sensitivities, and the potential sources of these error signals.

Encoder-decoder interactions and implications for BCI technology

We found that assistive BCI decoders lead the brain to learn compact neural representations, whether measured at the level of single neurons or population-level modes (Figs. 4, 5, 6). This suggests that the brain and BCI decoder together rely on a smaller set of neural features to communicate task information. These representation features may be driven by computational demands, but reveal potential challenges for maintaining BCI performance over time. The brain's encoding of information presents a fundamental bottleneck on the performance of any BCI — we cannot decode information that is not present in neural activity. Compact representations could therefore lead to significant drops in BCI performance if, for instance, measurement shifts leads to the loss of the small number of highly informative neural features. On the other hand, assistive decoders and the compact representations they produce may lead to BCI task encoding that is separated from other behaviors, which could reduce interference from other task requirements. Understanding the mechanisms by which assistive perturbations shape task encoding and the functional implications will improve our ability to design BCIs that provide high performance for someone's lifetime.

Our model, validated by experimental observations, provides a valuable testbed to study how compact representations form. For example, simulations revealed that stopping adaptive decoding early (before reaching maximum task performance) can provide faster performance improvements while avoiding overly compact representations, a form of "overfitting". CLDA aims to fit decoder parameters to match the brain's representation at a point in time. If the brain has a biased representation, with some features more strongly contributing to the task, CLDA will increase the contribution of these features to the decoded predictions. Assuming feedback about how a feature contributes to movement influences learning and credit assignment computations (e.g., [56, 61]), this would further enhance biased encoding in the brain. Adaptive decoding procedures like those in our experiment repeat this process over time, which could lead to accelerated "winner-take-all" dynamics. Similar phenomena have been observed in deep networks where the dominance of a few strong but informative features can prevent more subtle information from influencing learning ("gradient starvation"), which can be mitigated with network training methods that better regularize weights [62]. Deeper insights into the neural mechanisms of learning in BCI, paired with simulation testbeds to explore a wide range of algorithms, will allow us to design machine learning algorithms that robustly interact with a learning brain.

Our dataset includes nearly daily BCI training spanning over a year and a half, which reveals that the brain adapts over extended timescales potentially independent of task errors. For instance, beyond the compactness changes within a 1-2 week learning series (Fig. 4), we also noticed that representations became more compact across learning series (Fig. S3D). This is consistent with the possibility that assistive decoder perturbations strongly influence slow learning phenomena like credit assignment, as discussed above. This raises the possibility that adaptive BCIs may influence other phenomena thought to contribute to long-term learning, like representational drift [63]. Most efforts to maintain BCI performance over time focus on technical challenges like non-stationary neural measurements and do not explicitly consider learning-related changes to neural representations. Indeed, many recent approaches for long-term BCI decoder training aim to leverage similarity in neural population structure over time [55, 64, 65, 66, 67], which can be sensitive to large shifts in neural activity [68]. Emerging alternatives instead use similarity in task structure for decoder re-training [68, 69], which may be more compatible with long-term learning. Understanding how shifts in neural representations over time influence BCI control will likely be important

for real-world BCI applications. Critically, our findings highlight that representations of task information will be influenced by the decoder algorithms used for BCI. This opens the possibility to develop new generations of adaptive decoders that not only aim to improve task performance, but also contain explicit objectives to shape and guide underlying neural representations for robust long-term performance.

Acknowledgements

The authors thank Jose M. Carmena, who shared data collected in his laboratory for this study. The authors' research was supported in part by an IVADO postdoctoral fellowship, Canada First Research Excellence Fund/Apogée (AP), an NSF Accelnet INBIC fellowship (PR), a Simons Collaboration for the Global Brain Pilot award (898220, GL and ALO), a Google Faculty Award (ALO and GL) and an NSERC Discovery Grant (RGPIN-2018-04821, GL), the Eunice Kennedy Shriver National Institute of Child Health and Human Development (NIH K12HD073945, ALO), and NIH grant R01 NS134634 (ALO). 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 funding agencies.

References

[1] John W. Krakauer et al. "Motor Learning". en. In: *Comprehensive Physiology*. John Wiley & Sons, Ltd, 2019, pp. 613–663. ISBN: 978-0-470-65071-4.

[2] Sebastian Telgen, Darius Parvin, and Jörn Diedrichsen. "Mirror Reversal and Visual Rotation Are Learned and Consolidated via Separate Mechanisms: Recalibrating or Learning De Novo?" en. In: *Journal of Neuroscience* 34.41 (Oct. 2014). Publisher: Society for Neuroscience Section: Articles, pp. 13768–13779. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.5306-13.2014.

[3] Eran Dayan and Leonardo G. Cohen. "Neuroplasticity Subserving Motor Skill Learning". en. In: *Neuron* 72.3 (Nov. 2011), pp. 443–454. ISSN: 0896-6273. DOI: 10.1016/j.neuron.2011.10.008.

[4] Ling Wang et al. "Structural plasticity within highly specific neuronal populations identifies a unique parcellation of motor learning in the adult brain". In: *Proceedings of the National Academy of Sciences* 108.6 (Feb. 2011). Publisher: Proceedings of the National Academy of Sciences, pp. 2545–2550. DOI: 10.1073/pnas.1014335108.

[5] Tonghui Xu et al. "Rapid formation and selective stabilization of synapses for enduring motor memories". en. In: *Nature* 462.7275 (Dec. 2009). Number: 7275 Publisher: Nature Publishing Group, pp. 915–919. ISSN: 1476-4687. DOI: 10.1038/nature08389.

[6] Min Fu et al. "Repetitive motor learning induces coordinated formation of clustered dendritic spines in vivo". en. In: *Nature* 483.7387 (Mar. 2012). Number: 7387 Publisher: Nature Publishing Group, pp. 92–95. ISSN: 1476-4687. DOI: 10.1038/nature10844.

[7] Blake A Richards and Timothy P Lillicrap. "Dendritic solutions to the credit assignment problem". en. In: *Current Opinion in Neurobiology*. Neurobiology of Learning and Plasticity 54 (Feb. 2019), pp. 28–36. ISSN: 0959-4388. DOI: 10.1016/j.conb.2018.08.003.

[8] Eva L. Dyer and Konrad Kording. "Why the simplest explanation isn't always the best". In: *Proceedings of the National Academy of Sciences* 120.52 (Dec. 2023). Publisher: Proceedings of the National Academy of Sciences, e2319169120. DOI: 10.1073/pnas.2319169120.

[9] Matthew D Golub et al. "Brain–computer interfaces for dissecting cognitive processes underlying sensorimotor control". In: *Current Opinion in Neurobiology*. Neurobiology of cognitive behavior 37 (Apr. 2016), pp. 53–58. ISSN: 0959-4388. DOI: 10.1016/j.conb.2015.12.005.

[10] Amy L. Orsborn and Bijan Pesaran. "Parsing learning in networks using brain-machine interfaces". eng. In: *Current Opinion in Neurobiology* 46 (Oct. 2017), pp. 76–83. ISSN: 1873-6882. DOI: 10.1016/j.conb.2017.08.002.

[11] John K. Chapin et al. “Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex”. en. In: *Nature Neuroscience* 2.7 (July 1999). Number: 7 Publisher: Nature Publishing Group, pp. 664–670. ISSN: 1546-1726. DOI: 10.1038/10223.

[12] Jose M. Carmena et al. “Learning to Control a Brain–Machine Interface for Reaching and Grasping by Primates”. en. In: *PLOS Biology* 1.2 (Oct. 2003). Publisher: Public Library of Science, e42. ISSN: 1545-7885. DOI: 10.1371/journal.pbio.0000042.

[13] Dawn M. Taylor, Stephen I. Helms Tillery, and Andrew B. Schwartz. “Direct Cortical Control of 3D Neuroprosthetic Devices”. In: *Science* 296.5574 (June 2002). Publisher: American Association for the Advancement of Science, pp. 1829–1832. DOI: 10.1126/science.1070291.

[14] Karunesh Ganguly and Jose M. Carmena. “Emergence of a Stable Cortical Map for Neuroprosthetic Control”. en. In: *PLOS Biology* 7.7 (July 2009). Publisher: Public Library of Science, e1000153. ISSN: 1545-7885. DOI: 10.1371/journal.pbio.1000153.

[15] Beata Jarosiewicz et al. “Functional network reorganization during learning in a brain-computer interface paradigm”. In: *Proceedings of the National Academy of Sciences of the United States of America* 105.49 (Dec. 2008), pp. 19486–19491. ISSN: 0027-8424. DOI: 10.1073/pnas.0808113105.

[16] Patrick T. Sadtler et al. “Neural constraints on learning”. en. In: *Nature* 512.7515 (Aug. 2014). Number: 7515 Publisher: Nature Publishing Group, pp. 423–426. ISSN: 1476-4687. DOI: 10.1038/nature13665.

[17] Amy L. Orsborn et al. “Closed-loop decoder adaptation on intermediate time-scales facilitates rapid BMI performance improvements independent of decoder initialization conditions”. eng. In: *IEEE transactions on neural systems and rehabilitation engineering: a publication of the IEEE Engineering in Medicine and Biology Society* 20.4 (July 2012), pp. 468–477. ISSN: 1558-0210. DOI: 10.1109/TNSRE.2012.2185066.

[18] Zheng Li et al. “Adaptive Decoding for Brain-Machine Interfaces Through Bayesian Parameter Updates”. In: *Neural Computation* 23.12 (Dec. 2011), pp. 3162–3204. ISSN: 0899-7667. DOI: 10.1162/NECO_a_00207.

[19] Krishna V. Shenoy and Jose M. Carmena. “Combining Decoder Design and Neural Adaptation in Brain-Machine Interfaces”. en. In: *Neuron* 84.4 (Nov. 2014), pp. 665–680. ISSN: 0896-6273. DOI: 10.1016/j.neuron.2014.08.038.

[20] Amy L. Orsborn et al. “Closed-Loop Decoder Adaptation Shapes Neural Plasticity for Skillful Neuroprosthetic Control”. en. In: *Neuron* 82.6 (June 2014), pp. 1380–1393. ISSN: 0896-6273. DOI: 10.1016/j.neuron.2014.04.048.

[21] Dalia De Santis. “A Framework for Optimizing Co-adaptation in Body-Machine Interfaces”. English. In: *Frontiers in Neurorobotics* 0 (2021). Publisher: Frontiers. ISSN: 1662-5218. DOI: 10.3389/fnbot.2021.662181.

[22] Maneeshika M. Madduri, Samuel A. Burden, and Amy L. Orsborn. “Biosignal-based co-adaptive user-machine interfaces for motor control”. In: *Current Opinion in Biomedical Engineering* 27 (Sept. 2023), p. 100462. ISSN: 2468-4511. DOI: 10.1016/j.cobme.2023.100462.

[23] Vivek R. Athalye et al. “Emergence of Coordinated Neural Dynamics Underlies Neuroprosthetic Learning and Skillful Control”. eng. In: *Neuron* 93.4 (Feb. 2017), 955–970.e5. ISSN: 1097-4199. DOI: 10.1016/j.neuron.2017.01.016.

[24] Ellen L. Zippi et al. *Selective modulation of population dynamics during neuroprosthetic skill learning*. en. Pages: 2021.01.08.425917 Section: New Results. Jan. 2021. DOI: 10.1101/2021.01.08.425917.

[25] Karunesh Ganguly et al. “Reversible large-scale modification of cortical networks during neuroprosthetic control”. en. In: *Nature Neuroscience* 14.5 (May 2011). Number: 5 Publisher: Nature Publishing Group, pp. 662–667. ISSN: 1546-1726. DOI: 10.1038/nn.2797.

[26] Tanuj Gulati et al. “Neural reactivations during sleep determine network credit assignment”. en. In: *Nature Neuroscience* 20.9 (Sept. 2017). Number: 9 Publisher: Nature Publishing Group, pp. 1277–1284. ISSN: 1546-1726. DOI: 10.1038/nn.4601.

[27] Pierre-Jean Arduin et al. ""Master" Neurons Induced by Operant Conditioning in Rat Motor Cortex during a Brain-Machine Interface Task". en. In: *Journal of Neuroscience* 33.19 (May 2013). Publisher: Society for Neuroscience Section: Articles, pp. 8308–8320. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.2744-12.2013.

[28] Xiao Zhou et al. "Distinct types of neural reorganization during long-term learning". In: *Journal of Neurophysiology* 121.4 (Apr. 2019). Publisher: American Physiological Society, pp. 1329–1341. ISSN: 0022-3077. DOI: 10.1152/jn.00466.2018.

[29] Daniel B. Silversmith et al. "Plug-and-play control of a brain–computer interface through neural map stabilization". en. In: *Nature Biotechnology* 39.3 (Mar. 2021). Number: 3 Publisher: Nature Publishing Group, pp. 326–335. ISSN: 1546-1696. DOI: 10.1038/s41587-020-0662-5.

[30] Siddharth Dangi et al. "Design and Analysis of Closed-Loop Decoder Adaptation Algorithms for Brain-Machine Interfaces". In: *Neural Computation* 25.7 (July 2013), pp. 1693–1731. ISSN: 0899-7667. DOI: 10.1162/NECO_a_00460.

[31] Siddharth Dangi et al. "Continuous Closed-Loop Decoder Adaptation with a Recursive Maximum Likelihood Algorithm Allows for Rapid Performance Acquisition in Brain-Machine Interfaces". en. In: *Neural Computation* 26.9 (Sept. 2014), pp. 1811–1839. ISSN: 0899-7667, 1530-888X. DOI: 10.1162/NECO_a_00632.

[32] Vikash Gilja et al. "A high-performance neural prosthesis enabled by control algorithm design". en. In: *Nature Neuroscience* 15.12 (Dec. 2012), pp. 1752–1757. ISSN: 1546-1726. DOI: 10.1038/nn.3265.

[33] Beata Jarosiewicz et al. "Virtual typing by people with tetraplegia using a self-calibrating intracortical brain-computer interface". In: *Science Translational Medicine* 7.313 (Nov. 2015). Publisher: American Association for the Advancement of Science, 313ra179–313ra179. DOI: 10.1126/scitranslmed.aac7328.

[34] Leigh R. Hochberg et al. "Neuronal ensemble control of prosthetic devices by a human with tetraplegia". In: *Nature* 442.7099 (July 13, 2006), pp. 164–171. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature04970.

[35] Jennifer L Collinger et al. "High-performance neuroprosthetic control by an individual with tetraplegia". In: *The Lancet* 381.9866 (Feb. 2013), pp. 557–564. ISSN: 01406736. DOI: 10.1016/S0140-6736(12)61816-9.

[36] Chethan Pandarinath et al. "High performance communication by people with paralysis using an intracortical brain-computer interface". In: *eLife* 6 (Feb. 2017). Ed. by Sabine Kastner. Publisher: eLife Sciences Publications, Ltd, e18554. ISSN: 2050-084X. DOI: 10.7554/eLife.18554.

[37] Maneeshika M. Madduri, Samuel A. Burden, and Amy L. Orsborn. "A Game-Theoretic Model for Co-Adaptive Brain-Machine Interfaces". In: *2021 10th International IEEE/EMBS Conference on Neural Engineering (NER)*. ISSN: 1948-3554. May 2021, pp. 327–330. DOI: 10.1109/NER49283.2021.9441081.

[38] Luca Mazzucato, Alfredo Fontanini, and Giancarlo La Camera. "Stimuli reduce the dimensionality of cortical activity". In: *Frontiers in systems neuroscience* 10 (2016). Publisher: Frontiers Media SA, p. 11.

[39] Stefano Recanatesi et al. "Dimensionality in recurrent spiking networks: Global trends in activity and local origins in connectivity". In: *PLoS Computational Biology* 15.7 (July 2019), e1006446. ISSN: 1553-734X. DOI: 10.1371/journal.pcbi.1006446.

[40] J. K. Kihlberg, J. H. Herson, and W. E. Schotz. "Square Root Transformation Revisited". In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 21.1 (1972). Publisher: [Wiley, Royal Statistical Society], pp. 76–81. ISSN: 0035-9254. DOI: 10.2307/2346609.

[41] Elena Facco et al. "Estimating the intrinsic dimension of datasets by a minimal neighborhood information". en. In: *Scientific Reports* 7.1 (Sept. 2017). Number: 1 Publisher: Nature Publishing Group, p. 12140. ISSN: 2045-2322. DOI: 10.1038/s41598-017-11873-y.

[42] Ege Altan et al. "Estimating the dimensionality of the manifold underlying multi-electrode neural recordings". en. In: *PLOS Computational Biology* 17.11 (Nov. 2021). Publisher: Public Library of Science, e1008591. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1008591.

[43] Mikhail A. Lebedev. "How to read neuron-dropping curves?" In: *Frontiers in Systems Neuroscience* 8 (May 2014). ISSN: 1662-5137. DOI: 10.3389/fnsys.2014.00102.

[44] Ronald J. Williams. "Simple statistical gradient-following algorithms for connectionist reinforcement learning". en. In: *Machine Learning* 8.3 (May 1992), pp. 229–256. ISSN: 1573-0565. DOI: 10.1007/BF00992696.

[45] Ashesh K. Dhawale, Maurice A. Smith, and Bence P. Ölveczky. "The Role of Variability in Motor Learning". In: *Annual Review of Neuroscience* 40.1 (2017). _eprint: https://doi.org/10.1146/annurev-neuro-072116-031548, pp. 479–498. DOI: 10.1146/annurev-neuro-072116-031548.

[46] Albert K. You et al. *Flexible Modulation of Neural Variance Facilitates Neuroprosthetic Skill Learning*. en. Pages: 817346 Section: New Results. Mar. 2020. DOI: 10.1101/817346.

[47] Xulu Sun et al. "Cortical preparatory activity indexes learned motor memories". en. In: *Nature* 602.7896 (Feb. 2022). Number: 7896 Publisher: Nature Publishing Group, pp. 274–279. ISSN: 1476-4687. DOI: 10.1038/s41586-021-04329-x.

[48] N. Alex Cayco-Gajic, Claudia Clopath, and R. Angus Silver. "Sparse synaptic connectivity is required for decorrelation and pattern separation in feedforward networks". In: *Nature Communications* 8 (Oct. 2017), p. 1116. ISSN: 2041-1723. DOI: 10.1038/s41467-017-01109-y.

[49] Anton Spanne and Henrik Jörntell. "Questioning the role of sparse coding in the brain". English. In: *Trends in Neurosciences* 38.7 (July 2015). Publisher: Elsevier, pp. 417–427. ISSN: 0166-2236, 1878-108X. DOI: 10.1016/j.tins.2015.05.005.

[50] Carsen Stringer et al. "High-dimensional geometry of population responses in visual cortex". en. In: *Nature* 571.7765 (July 2019). Number: 7765 Publisher: Nature Publishing Group, pp. 361–365. ISSN: 1476-4687. DOI: 10.1038/s41586-019-1346-5.

[51] Matthew D. Golub et al. "Learning by neural reassociation". en. In: *Nature Neuroscience* 21.4 (Apr. 2018), pp. 607–616. ISSN: 1097-6256, 1546-1726. DOI: 10.1038/s41593-018-0095-3.

[52] Emily R. Oby et al. "New neural activity patterns emerge with long-term learning". en. In: *Proceedings of the National Academy of Sciences* 116.30 (July 2019), pp. 15210–15215. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1820296116.

[53] John P Cunningham and Byron M Yu. "Dimensionality reduction for large-scale neural recordings". en. In: *Nature Neuroscience* 17.11 (Nov. 2014), pp. 1500–1509. ISSN: 1097-6256, 1546-1726. DOI: 10.1038/nn.3776.

[54] Juan A. Gallego et al. "Cortical population activity within a preserved neural manifold underlies multiple motor behaviors". en. In: *Nature Communications* 9.1 (Oct. 2018). Number: 1 Publisher: Nature Publishing Group, p. 4233. ISSN: 2041-1723. DOI: 10.1038/s41467-018-06560-z.

[55] Juan A. Gallego et al. "Long-term stability of cortical population dynamics underlying consistent behavior". en. In: *Nature Neuroscience* 23.2 (Feb. 2020). Number: 2 Publisher: Nature Publishing Group, pp. 260–270. ISSN: 1546-1726. DOI: 10.1038/s41593-019-0555-4.

[56] Barbara Feulner and Claudia Clopath. "Neural manifold under plasticity in a goal driven learning behaviour". en. In: *PLOS Computational Biology* 17.2 (Feb. 2021). Publisher: Public Library of Science, e1008621. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1008621.

[57] Daniel M. Wolpert and J. Randall Flanagan. "Motor prediction". English. In: *Current Biology* 11.18 (Sept. 2001). Publisher: Elsevier, R729–R732. ISSN: 0960-9822. DOI: 10.1016/S0960-9822(01)00432-8.

[58] Avi Karni et al. "The acquisition of skilled motor performance: Fast and slow experience-driven changes in primary motor cortex". In: *Proceedings of the National Academy of Sciences* 95.3 (Feb. 1998). Publisher: Proceedings of the National Academy of Sciences, pp. 861–868. DOI: 10.1073/pnas.95.3.861.

[59] Maurice A. Smith, Ali Ghazizadeh, and Reza Shadmehr. "Interacting Adaptive Processes with Different Timescales Underlie Short-Term Motor Learning". en. In: *PLOS Biology* 4.6 (May 2006). Publisher: Public Library of Science, e179. ISSN: 1545-7885. DOI: 10.1371/journal.pbio.0040179.

[60] Matthew G. Perich, Juan A. Gallego, and Lee E. Miller. "A Neural Population Mechanism for Rapid Learning". en. In: *Neuron* 100.4 (Nov. 2018), 964–976.e7. ISSN: 0896-6273. DOI: 10.1016/j.neuron.2018.09.030.

[61] Vivek R Athalye, Jose M Carmena, and Rui M Costa. "Neural reinforcement: re-entering and refining neural dynamics leading to desirable outcomes". en. In: *Current Opinion in Neurobiology*. Neurobiology of Behavior 60 (Feb. 2020), pp. 145–154. ISSN: 0959-4388. DOI: 10.1016/j.conb.2019.11.023.

[62] Mohammad Pezeshki et al. "Gradient Starvation: A Learning Proclivity in Neural Networks". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 1256–1272.

[63] Laura N. Driscoll, Lea Duncker, and Christopher D. Harvey. "Representational drift: Emerging theories for continual learning and experimental future directions". In: *Current Opinion in Neurobiology* 76 (Oct. 2022), p. 102609. ISSN: 0959-4388. DOI: 10.1016/j.conb.2022.102609.

[64] Alan D. Degenhart et al. "Stabilization of a brain-computer interface via the alignment of low-dimensional spaces of neural activity". In: *Nature biomedical engineering* 4.7 (July 2020), pp. 672–685. ISSN: 2157-846X. DOI: 10.1038/s41551-020-0542-9.

[65] David Sussillo et al. "Making brain–machine interfaces robust to future neural variability". en. In: *Nature Communications* 7.1 (Dec. 2016). Number: 1 Publisher: Nature Publishing Group, p. 13749. ISSN: 2041-1723. DOI: 10.1038/ncomms13749.

[66] Brianna M. Karpowicz et al. *Stabilizing brain-computer interfaces through alignment of latent dynamics*. en. Pages: 2022.04.06.487388 Section: New Results. Nov. 2022. DOI: 10.1101/2022.04.06.487388.

[67] Xuan Ma et al. "Using adversarial networks to extend brain computer interface decoding accuracy over time". eng. In: *eLife* 12 (Aug. 2023), e84296. ISSN: 2050-084X. DOI: 10.7554/eLife.84296.

[68] Guy H. Wilson et al. *Long-term unsupervised recalibration of cursor BCIs*. en. Pages: 2023.02.03.527022 Section: New Results. Feb. 2023. DOI: 10.1101/2023.02.03.527022.

[69] Chaofei Fan et al. *Plug-and-Play Stability for Intracortical Brain-Computer Interfaces: A One-Year Demonstration of Seamless Brain-to-Text Communication*. arXiv:2311.03611 [cs, q-bio]. Nov. 2023. DOI: 10.48550/arXiv.2311.03611.

[70] R. E. Kalman. "A New Approach to Linear Filtering and Prediction Problems". In: *Journal of Basic Engineering* 82.1 (Mar. 1960), pp. 35–45. ISSN: 0021-9223. DOI: 10.1115/1.3662552.

[71] Peiran Gao et al. *A theory of multineuronal dimensionality, dynamics and measurement*. en. Tech. rep. Company: Cold Spring Harbor Laboratory Distributor: Cold Spring Harbor Laboratory Label: Cold Spring Harbor Laboratory Section: New Results Type: article. Nov. 2017, p. 214262. DOI: 10.1101/214262.

[72] *sklearn.linear_model.LogisticRegression — scikit-learn 0.24.1 documentation*.

[73] I.T. Jolliffe. *Principal Component Analysis*. Second edition. Springer Series in Statistics. New York, NY: Springer, 2002. ISBN: 978-0-387-95442-4.

[74] Gaël Guennebaud, Benoît Jacob, et al. *Eigen v3*. http://eigen.tuxfamily.org. 2010.

[75] Weiwei Li and Emanuel Todorov. "Iterative linear quadratic regulator design for nonlinear biological movements systems". en. In: *Proceedings of the First International Conference on Informatics in Control, Automation and Robotics*. 2004, pp. 222–229. ISBN: 978-972-8865-12-2. DOI: 10.5220/0001143902220229.

[76] Emanuel Todorov. "Optimal Control Theory". In: *Bayesian Brain: Probabilistic Approaches to Neural Coding*. _eprint: https://direct.mit.edu/book/chapter-pdf/2321908/9780262294188_cal.pdf. The MIT Press, Dec. 2006. ISBN: 978-0-262-29418-8. DOI: 10.7551/mitpress/1535.003.0018.

[77] Ernest J. Cheng and Stephen H. Scott. "Morphometry of Macaca mulatta forelimb. I. Shoulder and elbow muscles and segment inertial parameters". en. In: *Journal of Morphology* 245.3 (2000), pp. 206–224. ISSN: 1097-4687. DOI: https://doi.org/10.1002/1097-4687(200009)245:3<206::AID-JMOR3>3.0.CO;2-U.

[78] Emanuel Todorov. "Direct cortical control of muscle activation in voluntary arm movements: a model". en. In: *Nature Neuroscience* 3.4 (Apr. 2000), pp. 391–398. ISSN: 1546-1726. DOI: 10.1038/73964.

[79] Jiahao Chen and Hong Qiao. "Motor-Cortex-Like Recurrent Neural Network and Multi-Tasks Learning for the Control of Musculoskeletal Systems". In: *IEEE Transactions on Cognitive and Developmental Systems* (2020), pp. 1–1. ISSN: 2379-8939. DOI: 10.1109/TCDS.2020.3045574.

[80] Nicolas Frémaux, Henning Sprekeler, and Wulfram Gerstner. "Functional Requirements for Reward-Modulated Spike-Timing-Dependent Plasticity". en. In: *Journal of Neuroscience* 30.40 (Oct. 2010), pp. 13326–13337. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.6249-09.2010.

[81] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *arXiv:1412.6980 [cs]* (Jan. 2017). arXiv: 1412.6980.

Methods

Experiment

Neural data

We analyzed neural activity as two male rhesus macaques (Macaca Mulatta, monkeys S and J) learned to control a 2D BCI cursor with CLDA intermittently performed across days. Both monkeys were implanted with 128 microwire electrode arrays in motor and premotor cortices to record spiking activity. Spike-sorted multi-unit activity (monkey S) or unsorted threshold crossings defining channel-level multi-unit activity (monkey J) were used for BCI control. We refer to neural activity for both animals as "units". A subset of recorded units was used for the real-time BCI cursor control, which defines the readout neural population. Units that were recorded on the same array, but not used as inputs to the BCI decoder for cursor control, define the nonreadout neural population. Stable readout units were defined as units consistently used throughout an entire BCI series.

BCI control with adaptive decoders

Readout neural activity controlled the 2D cursor using a position-velocity Kalman filter (KF) [17, 20, 32, 70]. The KF is a state-based decoder that includes a linear state transition model, which determines cursor dynamics (e.g., position is the integral of velocity, velocities are related in time), and a linear observation model, which determines the relationship between neural activity and cursor states (position, velocity). KF parameters were trained and intermittently updated using previously-developed algorithms that estimate parameters during closed-loop BCI control (closed-loop decoder adaptation, CLDA) [17]. CLDA re-estimates the parameters of the KF using the subject's neural activity and estimated motor intent during BCI control and updates the decoder used for BCI control in real-time. Updates to the KF BCI used the SmoothBatch algorithm [17] (monkey J) to update only the KF observation model, or the ReFIT algorithm [32] (monkey S) that updates all KF parameters (state and observation models; see below for details).

Monkeys practiced with a given decoder mapping (and subsequently updated versions of that mapping) for multiple days, defining a learning series. CLDA was performed on the first day to initialize decoder parameters, and intermittently as needed on subsequent days to maintain task performance as neural recordings shifted. These intermittent updates included alterations to decoder parameters without changing the readout unit identities (weight change only), and interventions where readout unit identities were altered (to address loss of a unit previously in the readout ensemble) along with decoder weight changes (readout + weight change) (Fig. 1B). Monkey J and monkey S learned 13 and 6 adaptive decoder mappings in total, respectively [20]. This analysis focused on extended learning series (longer than four days) that included intermittent CLDA on the KF decoder (monkey J: 7, monkey S: 3). Each learning series varied in length and continued until performance plateaued. To

quantify changes during a series, we compared "early" and "late" training days. We defined an early training day as the first day with at least 25 trials per target direction. A late training day was the last day in the learning series with at least 25 trials per target direction. Full details on adaptive decoder methods and the dataset are provided in Orsborn *et al.* [20].

Behavioral performance

We assessed behavioral improvements using standard performance metrics such as mean success percentage and reach times (Fig 1C). Success percentage was computed as the fraction of rewarded trials to the total number of trials initiated. Reach time was the time between cursor leaving center target and entering peripheral target.

Data analysis

Neural data preprocessing

Neural activity was binned at 100 ms and trial-aligned to the go cue. We included neural activity from the onset of the go cue to 1800 ms after as inputs to an offline decoding model (see below). Altering the time included was not found to qualitatively change reported findings (not shown). For a given day, we considered all trials where the cursor enters the peripheral target, both rewarded and unrewarded. Unrewarded trials in this case were solely unsuccessful due to a failure to hold at the peripheral target. Because reach times vary over the course of learning, trials with a post-go-cue duration shorter than 1800 ms were zero-padded; this was most common in late phases of learning.

Participation ratio

We computed the neural dimensionality for readouts and nonreadouts using the participation ratio (PR) [39, 71]. PR summarizes the number of collective modes, or degrees of freedom, that the population's activity explores. It provides information about the spread of data and can be directly computed from the pairwise covariances among units C_{units} in a population. It is defined by

$$\text{PR} = \frac{[\text{tr}\{C_{\text{units}}\}]^2}{\text{tr}\{C_{\text{units}}^2\}}.$$

We estimated population dimensionality for readouts and nonreadouts separately each day. $\text{PR} \in [1, n_{\text{units}}]$, but since we had different number of nonreadout units that were recorded each day and different number of readout units across series, we normalized PR using

$$\text{PR}_{\text{norm}} = \frac{\text{PR} - 1}{n_{\text{units}} - 1}.$$

Therefore, $\text{PR}_{\text{norm}} \in [0, 1]$ with $\text{PR}_{\text{norm}} = 0$ when PR = 1 and $\text{PR}_{\text{norm}} = 1$ when $\text{PR} = n_{\text{units}}$. Note that a small difference in PR_{norm} yields a big change in actual dimension (PR of 9.5 normalizes to $\text{PR}_{\text{norm}} = 0.25$ whereas PR of 11.2 normalizes to $\text{PR}_{\text{norm}} = 0.3$ for $n_{\text{units}} = 16$).

We estimated the dimensionality each day (Fig. 2B) using trials with completed reaches by creating a single time-series of concatenated spike counts from all reach segments (from go cue to the cursor entering the peripheral target, $n_{\text{units}} \times (n_{\text{timebins}} \times n_{\text{trials}})$. The readout population dimensionality was estimated using only the readout units used in the decoder that day. The readout populations before/after readout unit changes (Fig. 2E) did not include any units that were swapped to match the populations being compared. The change in dimensionality between early and late learning (Fig 2C) was estimated by $\Delta\text{PR} = \text{PR}_{\text{late}} - \text{PR}_{\text{early}}$. We computed the 95% confidence intervals for PR calculations (Fig. 2B,C) by re-sampling the original trials with replacement, matching target identity distributions (N = 10^4). As noted below, the number of trials to each target varies over days. We found that matching the number of trials for each day (as done for decoding analyses described below) did not qualitatively change results, and including all trials allowed us to better estimate properties of the neural populations.

Dimensionality analysis for BCI learning with a fixed decoder (Fig. 2A,C) was performed on the dataset from [14]. It used the

same method as above except that training epochs were used instead of days [23, 24]. Each consecutive training epoch contains a constant number of trials, which may combine data across training days. Note that our analyses differ from past studies [23, 24], which computed a measure of dimensionality based on the shared covariance extracted using factor analysis, instead of the total covariance as used by the PR metric.

Logistic regression model

We analyzed changes in neural representations with multiclass logistic regression (LR) models that used neural activity (Figs. 3, 4) or transformations thereof (Fig. 6; see below), to predict target identity. The classifier model received the activity across all units and time bins for each trial as input, and generated an output corresponding to the predicted target. To control for variability in the number of trials completed to each target across days, we generated a training set where trials were randomly selected (without replacement) to match the number of trials to each target across days. We selected 25 trials per reach direction (200 trials per day), matching our criteria for defining "early" training phases. The remainder of trials on a day were used as a test set. Unit firing rates were z-scored across trials for each set separately. We drew a 100 training-test splits on each day to determine error bars as the 95% confidence interval on classification accuracy. We used the mean accuracy across the training-test splits to compare classification accuracy across series and animals (Fig. 3E). We implemented our LR models using the Python library Scikit-learn [72] with the L-BGFS solver and L2 regularization. We did not observe any considerable changes in classification accuracy between LR models with or without regularization on late days (Fig. S2A).

Parameter values (model weights) for a single training-test split and for a single target (target 2) were used to illustrate the LR model (Fig. 3B). We quantified the similarity of decoder weights across days by calculating the similarity between the model weight matrices W on consecutive days, namely W^d, W^{d-1} (Fig. 3C), similar to previous analyses [14, 20]. This evaluation was performed by calculating the correlation coefficient between the flattened matrices of model weights (where weights for each target are concatenated) from each day.

Rank-ordered neuron adding curve

We estimated the contribution of each individual unit to target identity prediction by running a LR model for each unit separately (Fig. 4A, left). We then ranked units based on their classification accuracy (Fig. 4A, middle and Fig. 4B). A ranked-ordered Neuron Adding Curve (NAC) was produced by calculating the classification accuracy with the highest ranking unit, then with the first highest and second highest ranking units, and so on (Fig.4A, right and Fig.4C) [43]. These analyses used the same training-test split protocol described above. To account for the lower maximum classification accuracy on early days compared to later days, we normalized the NAC classification scores to each day's maximum performance using a linear scaling. We then computed the number of ranked units (N_c) required to achieve 80% normalized prediction accuracy for early and late days. We performed a Wilcoxon signed-rank test using the N_c's across series and monkey with alternative hypothesis $N_c^{\text{late}} < N_c^{\text{early}}$ (Fig. 4D). The "combinatorial" NAC, as described in the main text, tested all possible combinations of units for each ensemble size. We evaluated the separability between the classification accuracy distributions with and without the top N_c units (Fig. 4E-G) using a discriminability index, $d' = |m_1 - m_2|/[(s_1 + s_2)/2]$, where m_i and s_i are the mean and standard deviation of distribution i, respectively. A higher d' indicates that two distributions are more separable.

Classification using neural modes

We performed principal component analysis (PCA) on the readout activity for the early and late day separately (Fig. 6A). All trials for each day were used for PCA, and readout firing rates were z-scored for each day. We constructed PC adding curves and analyzed compaction in the PCs in the same way as in the above rank-ordered NAC, but using PCs instead single-unit activity (Fig. 6B-D). Importantly, we kept all PCs (no dimensionality reduction) and thus the number of PCs was equal to the number of readouts. We compared classification accuracy in low and high-variance PCs by spliting PCs into two groups (top 50% and bottom 50%) (Fig. 6F). These groups were defined by ranking PCs according to variance explained and assigning the first (last) $N/2$ PCs into the top (bottom) group.

PC loadings

The loadings are the components of the vectors produced by PCA, which generate the PCs by projecting a random vector variable onto them [73]. After normalizing the loading vector, we took the square of a readout's loading to represent its relative contribution to the PC. For each learning series, on late days, we identified the PC with the highest variance and the PC with the strongest classification accuracy (Fig. 6E), and the corresponding loading vector. For these PCs, we plotted the squared PC loading for each unit against its average rank based on its classification accuracy (Fig. 6H). We calculated the average rank for a unit from the distribution of their ranks across the training/test splits. Finally, we computed the sum of squared loadings (that we called the PC loading proportion) for the N_c leading and trailing units, according to their average rank (Fig. 6I).

Statistics

To evaluate statistical differences between early and late phases of learning, we applied the Wilcoxon signed-rank test for pairwise comparison. Significance levels are indicated as follows: ****: $p \leq 10^{-4}$, ***: $10^{-4} < p \leq 10^{-3}$, **: $10^{-3} < p \leq 10^{-2}$,*: $10^{-2} < p \leq 0.05$, ns: $p > 0.05$. Unless specified otherwise, the sample size for all such early vs late comparisons was $N = 10$ learning series for the BCI task, consisting of 3 series from monkey S and 7 series from monkey J. Inclusion of all learning series as opposed to only extended ($>$ 4 days) series, did not qualitatively change the trends observed (not shown). In analyses examining the impact of decoder weight adjustments and combined changes in decoder weight and readout membership on dimensionality (Fig. 2E), N represented the total instances of "weight change only" and "readout + weight change" across all series. In the supplementary material, control data from $N = 6$ learning series obtained from monkey J related to the arm movement task were used (Suppl. Figs. S2C,D and S3C).

Model

Simulations were implemented in C++ using the Eigen library [74]. Analyses were performed in Python, using the Numpy and Scipy packages. Values for parameters are included in Table 1. Code will be made publicly available upon publication.

Arm model

Our Recurrent Neural Network (RNN) models were initially trained to control an arm. We used a planar torque-based arm model [75] consisting of a shoulder and an elbow joint; the arm only moved in the horizontal $x-y$ plane. The dynamical variables were the angle of each joint ($\mathbf{q} = [q_1, q_2]^\mathsf{T}$), their velocity ($\dot{\mathbf{q}}$) and the applied torque ($\boldsymbol{\tau}$). These were linked through the dynamics

$$\mathscr{M}(\mathbf{q})\ddot{\mathbf{q}} + \boldsymbol{c}(\mathbf{q}, \dot{\mathbf{q}}) + \mathscr{B}\dot{\mathbf{q}} = \boldsymbol{\tau} \tag{1}$$

where

$$\mathscr{M}(q_2) = \begin{bmatrix} a_1 + 2a_2 \cos q_2 & a_3 + a_2 \cos q_2 \\ a_3 + a_2 \cos q_2 & a_3 \end{bmatrix}$$

is a 2×2 inertia matrix,

$$\boldsymbol{c}(q_2, \dot{q}_1, \dot{q}_2) = a_2 \sin q_2 \begin{bmatrix} -\dot{q}_2(2\dot{q}_1 + \dot{q}_2) \\ \dot{q}_1{}^2 \end{bmatrix}$$

represents the centripetal and Coriolis force vector and

$$\mathscr{B} = \begin{bmatrix} 0.05 & 0.025 \\ 0.025 & 0.05 \end{bmatrix}$$

is the joint friction matrix. Parameters a_1, a_2 and a_3 are functions of the moments of inertia (I), masses (m), lengths (l) and distances from joint center to center of mass (d) of the two links (see Table 1), with $a_1 = I_1 + I_2 + m_2 l_1^2$, $a_2 = m_2 l_1 d_2$ and $a_3 = I_2$. The torques corresponded roughly to muscle activations and were related to the control signals $\mathbf{u}$ generated by the network dynamics by $\tau_f \dot{\boldsymbol{\tau}} = -\boldsymbol{\tau} + \mathbf{u}$ [76]. In simulations, these equations were time-discretized using the standard

Table 1. Parameters of the model. For the arm model, we combined data from Ref. [77] and from the morphology of the subjects in Ref. [20].

Parameters of the arm model			
symbol	*value*	*unit*	*description*
m_1	0.25	kg	mass of upper arm link
m_2	0.20	kg	mass of lower arm link
l_1	0.12	m	length of upper arm link
l_2	0.16	m	length of lower arm link
d_1	0.075	m	distance from center upper arm to center of mass
d_2	0.075	m	distance from center lower arm to center of mass
I_1	0.0018	kg·m^2	moment of inertia of upper arm
I_2	0.0012	kg·m^2	moment of inertia of lower arm
τ_f	40	ms	time constant connecting torques and controls
Parameters of the network dynamics			
N	100	—	number of units
α_v	0.2	—	reciprocal of time constant
σ	0.02	—	noise intensity
Parameters of the objective			
δ_p	0.01	m	scaling factor for position
δ_v	0.02	m/s	scaling factor for velocity
δ_a	0.08	m/s^2	scaling factor for acceleration
γ_v (arm)	0.25	—	velocity cost hyperparameter for arm control
γ_v (BCI)	0.1	—	velocity cost hyperparameter for BCI control
γ_a	0.05	—	acceleration cost hyperparameter for arm control
λ_u	0.005	N^{-2}·m^{-2}	effort penalty hyperparameter
Parameters of the learning algorithm			
α_R	0.3	—	relative weight factor for reward traces
—	2×10^{-5}	—	learning rate

Euler scheme. The RNN defined below generated controls $\mathbf{u}_0,\ldots,\mathbf{u}_{T-1}$ and the arm's dynamics produced a sequence of states starting from an initial condition $[\mathbf{q}_0;\dot{\mathbf{q}}_0;\boldsymbol{\tau}_0]$. The position, velocity and acceleration of the end effector (hand) are denoted by $\mathbf{x}$, $\dot{\mathbf{x}}$ and $\ddot{\mathbf{x}}$, respectively. The initial condition was that the end effector be motionless ($\dot{\mathbf{x}}_0=\ddot{\mathbf{x}}_0=\mathbf{0}$) with its initial position $\mathbf{x}_0$ drawn from a uniform distribution centered on the center of the workspace, with radius 0.5 cm. The end effector position was obtained by a transformation of the joint coordinates, with the length of each link, l_1 and l_2, as parameters: $\mathbf{x}=[l_1\cos q_1+l_2\cos(q_1+q_2), l_1\sin q_1+l_2\sin(q_1+q_2)]^\mathsf{T}$. The corresponding end effector velocity was $\dot{\mathbf{x}}=J(\mathbf{q})\dot{\mathbf{q}}$, where $J(\mathbf{q})$ is the Jacobian of the transformation from $\mathbf{q}$ to $\mathbf{x}$. An expression for the end effector acceleration, needed in the learning objective described below, was obtained by differentiating $\dot{\mathbf{x}}$ with respect to time.

Decoder

The RNN models were then trained to perform BCI control. As in Ref. [20], the BCI decoder was a velocity Kalman filter, using arm trajectories and neural activities to encode the Kalman model parameters, and then inverting the state and observation equations (defined below) to decode newly recorded neural activity. In discrete time, the state and observation equations

read

$$\dot{\mathbf{x}}_t = A\dot{\mathbf{x}}_{t-1} + \mathbf{w}_{t-1}$$
$$\mathbf{y}_t = \mathbf{m} + C\dot{\mathbf{x}}_t + \mathbf{q}_t$$

where $\mathbf{q}_t \sim \mathcal{N}(\mathbf{0}, Q)$ and $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, W)$ are independent Gaussian random vectors, $\mathbf{y}_t$ is the firing activity and $\mathbf{m}$ is the average rate. The dimension of vectors $\mathbf{y}_t$, $\mathbf{m}$ and $\mathbf{q}_t$ was the same as the number of readouts in the model, $N_{\text{readout}} = 12$. Note that only the end effector velocity $\dot{\mathbf{x}}$ appears here, corresponding to a scenario where uncertainty on velocity does not propagate to position [32]. The position was determined by integrating the velocity.

Matrices A, W, C and Q were determined from simulation data during arm control following the algorithm described in Ref. [17]. To decode the cursor velocity while measuring neural activity online, we used the update equations

$$\hat{\dot{\mathbf{x}}}_t = A\hat{\dot{\mathbf{x}}}_{t-1} + K_t(\mathbf{y}_t - \mathbf{m} - CA\hat{\dot{\mathbf{x}}}_{t-1})$$
$$P_t = (I - K_t C)(AP_{t-1}A^\mathsf{T} + W),$$

where $K_t = P_t^- C^\mathsf{T}(CP_t^- C^\mathsf{T} + Q)^{-1}$ is the Kalman gain with $P_t^- = AP_{t-1}A^\mathsf{T} + W$. For simplicity, we set the initial estimate $\hat{\dot{\mathbf{x}}}_0 = \mathbf{0}$ with perfect precision, so that the error covariance matrix was $P_0 = 0$.

Closed-loop decoder adaptation (CLDA) was implemented following SmoothBatch [17, 20] using data recorded under brain control. Recorded velocities were rotated towards the current target, representing intended velocities [32]. This data with the rotated velocities produced new estimates for matrices C and Q and vector $\mathbf{m}$, denoted $\hat{C}$, $\hat{Q}$ and $\hat{\mathbf{m}}$, and the original quantities were replaced according to

$$X \leftarrow \alpha\hat{X} + (1-\alpha)X, \tag{2}$$

where $X \in \{C, Q, \mathbf{m}\}$ and $0 \leq \alpha \leq 1$ is the CLDA intensity; $\alpha = 0$ corresponds to a fixed decoder.

Network model

The RNN contained N units fully connected by a recurrent weight matrix W (see Table 1). The units' output were their firing rates $\mathbf{r}$, which were obtained from their "membrane potential" $\mathbf{v}$ by applying an activation function $\boldsymbol{\varphi}(\cdot)$ elementwise: $\mathbf{r}_t = \boldsymbol{\varphi}(\mathbf{v}_t)$, i.e., $r_{t,i} = \varphi(v_{t,i})$ for each unit i. The transfer function φ was the rectified linear unit (ReLU), producing non-negative firing rates. The membrane potentials obeyed

$$\mathbf{v}_{t+1} = \mathbf{v}_t + \alpha_v[-\mathbf{v}_t + W\boldsymbol{\varphi}(\mathbf{v}_t) + \mathbf{i}_t + \mathbf{b}] + \sigma\sqrt{\alpha_v}\boldsymbol{\xi}_t \tag{3}$$

where $t = 0, 1, \ldots, T-1$. These dynamics integrate with time constant α_v^{-1} a recurrent input from other units in the network ($W\boldsymbol{\varphi}(\mathbf{v})$), an external input representing information about the task ($\mathbf{i}$)—e.g., the position of the target to reach; see below—, a bias ($\mathbf{b}$) and a zero-mean Gaussian white noise ($\sigma\sqrt{\alpha_v}\boldsymbol{\xi}$). The initial condition for the membrane potential was drawn from a uniform distribution $\mathcal{U}(-1, 1)$.

The input $\mathbf{i}$ was given by $\mathbf{i}_t = U\mathbf{r}_t^{\text{in}}$, where U is the input weight matrix and $\mathbf{r}_t^{\text{in}}$ is the activity of the input layer. This input layer activity encoded premotor information. The encoding was a simple random projection of delayed information about target position $\mathbf{d}$, end effector position $\mathbf{x}$ and context $\mathbf{c}$—either $[1, 0]^\mathsf{T}$ for arm control or $[0, 1]^\mathsf{T}$ for BCI control—followed by the ReLU operation. Symbolically, $\mathbf{r}_t^{\text{in}} = \boldsymbol{\varphi}(F[\mathbf{x}_{t-l}; \mathbf{d}; \mathbf{c}])$, where F is a non-learnable random matrix and $l = 10$ is the delay. The output of the network—the controls applied to the arm models—was a linear mapping of the network activity [78]: $\mathbf{u}_t = V\mathbf{r}_t = V\boldsymbol{\varphi}(\mathbf{v}_t)$, where V is the output matrix.

Training objective

Let $\mathbf{d}_k$, $k = 0, \ldots, K-1$, represent the position of the $K = 8$ peripheral targets at a distance $D = 7$ cm from the center target, with $\mathbf{d}_k = D[\cos(2\pi k/K), \sin(2\pi k/K)]^\mathsf{T}$. The training objective was to minimize

$$L = \sum_{k=0}^{K-1} L_k \tag{4}$$

where

$$L_k = \frac{\|\mathbf{x}_T - \mathbf{d}_k\|^2}{\delta_p^2} + \gamma_v \frac{\|\dot{\mathbf{x}}_T\|^2}{\delta_v^2} + \gamma_a \frac{\|\ddot{\mathbf{x}}_T\|^2}{\delta_a^2} + \frac{\lambda_u}{T} \sum_{t=0}^{T-1} \|\mathbf{u}_t\|^2. \tag{5}$$

Therefore, the objective was to reach the target at the end of the trial (first term) with vanishing velocity (second term) and acceleration (third term), and with an effort penalty (fourth term). Parameters δ_p, δ_v and δ_a were used to rescale the position, velocity and acceleration terms. Hyperparameters γ_v, γ_f and λ_u controlled the relative weight of the velocity, acceleration and effort costs with respect to the position loss. Hyperparameters λ_u and γ_a were nonzero only under arm control (see Table 1).

Learning

The recurrent weights W, input weights U and biases $\mathbf{b}$ were plastic while the output matrix V and the encoding matrix F were fixed. Learning was performed *via* node perturbation using the REINFORCE algorithm [44, 79]. The noise $\boldsymbol{\xi}$ independently applied to all units in Eq. 3 evoked small end effector jitters. In reinforcement learning, jitters that increase reward should be reproduced, and network parameters should be updated accordingly. Here, the reward was taken to be minus the learning objective, $R = -L$. The gradient with respect to W was estimated using

$$\nabla_W R = \frac{\sqrt{\alpha}}{\sigma} \sum_{k=0}^{K-1} \left\{ \left(R_k - \bar{R}_k\right) \sum_{t=0}^{T-1} \boldsymbol{\xi}_t^{(k)} \mathbf{r}_t^{(k)\mathsf{T}} \right\},$$

where $\boldsymbol{\xi}_t^{(k)}$ and $\mathbf{r}_t^{(k)\mathsf{T}}$ are the noise and activity when target k is presented. The reward trace $\bar{R}_k$ provided an estimate of the expected reward for target k [80] and was computed as a moving average

$$\bar{R}_k = \alpha_R \bar{R}_k^{\text{prev}} + (1 - \alpha_R) R_k, \tag{6}$$

where $\alpha_R \in [0, 1]$ is a factor that weights the relative contribution of the present reward (R_k) and the expected reward in memory ($\bar{R}_k^{\text{prev}}$) in computing the new reward estimate. For U, one simply replace $\mathbf{r}_t$ by $\mathbf{r}_t^{\text{in}}$; for $\mathbf{b}$, the summation is over $\boldsymbol{\xi}_t$ only. Parameter updates were computed after seeing all K targets (an epoch) using Adam updates with standard hyperparameters [81].

Analysis of the model

To allow clear comparison across seeds (i.e. across random initializations of the network's parameters), we normalized the BCI training loss L relative to the loss with a fixed decoder, denoted $L^{(0)}$. For each seed, we computed the $\log_{10}$ of L and $L^{(0)}$ for the last epoch of each day and we defined a new performance metric ("normalized log performance", Fig. 5C) as

$$P_d = \frac{\log_{10}(L_1^{(0)}) - \log_{10}(L_d)}{\log_{10}(L_1^{(0)}) - \log_{10}(L_{-1}^{(0)})}, \tag{7}$$

where $L_1^{(0)}$ and $L_{-1}^{(0)}$ are the losses with a fixed decoder on the first and last day, respectively. This linear transformation seeks to transform the loss for the fixed decoder to the interval $[0, 1]$.

Single-unit normalized performances (Fig. 5E) were computed by first evaluating the reach performance for each individual

unit and ranking them from the most important to the least important. Let $L_1, \ldots, L_{N_{\text{readout}}}$ be the loss of the most dominant unit (L_1) to the least dominant unit ($L_{N_{\text{readout}}}$) for a given seed and for a given value of CLDA intensity. The single-unit normalized reach performance was defined by

$$S_i = \frac{L_{N_{\text{readout}}} - L_i}{L_{N_{\text{readout}}} - L_1}. \tag{8}$$

We thus have $S_1 = 1$ and $S_{N_{\text{readout}}} = 0$.

A similar transformation was performed to obtain the normalized performance in Fig. 5F. If L_i is the loss after the ith most dominant unit has been added to the pool of readout units, then the normalized performance was

$$\hat{R}_i = \frac{L_1 - L_i}{L_1 - L_{N_{\text{readout}}}}. \tag{9}$$

Here, we have $\hat{R}_1 = 0$ and $\hat{R}_{N_{\text{readout}}} = 1$.

Combinatorial unit-adding curves (Fig. 5G) were computed by sampling all combinations of size s, for $s = 1, \ldots, N_{\text{readout}}$, and evaluating the loss when each such combination was moving the cursor. To allow comparison across CLDA intensities, we normalized each loss using the minimum and maximum losses (across all subset sizes) when a fixed decoder was used:

$$\frac{L_{\text{min}}^{(\text{fixed})} - L_c}{L_{\text{min}}^{(\text{fixed})} - L_{\text{max}}^{(\text{fixed})}}, \tag{10}$$

where L_c is the loss incurred for a specific combination.

Supplementary Information

Assistive sensory-motor perturbations influence learned neural representations

Pavithra Rajeswaran[1], Alexandre Payeur[2,3], Guillaume Lajoie[2,3], and Amy L. Orsborn[1,4,5,*]

[1]University of Washington, Bioengineering, Seattle, 98115, USA
[2]Université de Montréal, Department of Mathematics and Statistics, Montréal (QC), Canada, H3C 3J7
[3]Mila - Québec Artificial Intelligence Institute, Montréal (QC), Canada, H2S 3H1
[4]University of Washington, Electrical and Computer Engineering, Seattle, 98115, USA
[5]Washington National Primate Research Center, Seattle, Washington, 98115, USA
[*]aorsborn@uw.edu

Supplementary Methods

Neural tuning curves were defined as the mean firing rate of the neuron in a time window (0-300ms after the go cue) as a function of the eight target directions. We first identified the mean firing rate of each unit per target direction and then modeled the relationship between neural activity and movement direction via a cosine tuning model [1]:

$$f = B_1 \cos\theta + B_2 \sin\theta + B_3$$

where θ represents the target direction and B_1, B_2 and B_3 are model coefficients. The coefficients were estimated via linear regression and then used to compute each unit's modulation depth (MD) and preferred direction (PD):

$$MD = \sqrt{B_1^2 + B_2^2}$$
$$PD = \text{arctan2}(B_2/B_1)$$

We calculated the change in tuning properties within each learning series by comparing MDs on early and late training days: $\Delta\text{MD} = \text{MD}_{\text{late}} - \text{MD}_{\text{early}}$. To compare these tuning changes to our model, we calculated the change in model coefficients between early and late training day: $\Delta\mathbf{w} = \mathbf{w}_{\text{late}} - \mathbf{w}_{\text{early}}$, where $\mathbf{w}_{\text{late,early}} \in \mathbb{R}^{n_{\text{units}}}$ were calculated from the model coefficients by taking the mean across time bins and then, for each unit, selecting the mean weight for the most contributing target direction. Figure S2B shows the correlation between ΔMD and $\Delta\mathbf{w}$.

Supplementary Discussion

Compact Representation in Units Does Not Imply Compact Representation in Neural Modes

We observed compactness in both the “mode" space and the “individual unit" space. Here, we demonstrate that these observations do not simply follow from one another. We consider a two-class classification problem, eliminating the complexity of multiple targets, and we ignore the time dimension. We define x as a random column vector representing centered single-unit activities and consider a logistic regression model $y(x) = f(w^\top x + b)$, where $^\top$ denotes transpose, and w and b are the model parameters. After fitting this model to data and achieving good generalization, we assume w has been determined.

Drawing inspiration from Fig. 3B, we define a representation as compact if the weight vector w is sparse, specifically having a few dominant elements (ideally one, for the sake of this discussion). We introduce matrix A containing principal vectors as columns, forming an orthonormal matrix ($AA^\top = A^\top A = I$, where I is the identity matrix). This allows us to reformulate the logistic regression model in terms of principal components and their projections: $y(x) = f((A^\top w)^\top (A^\top x) + b)$, where $A^\top x$ represents the principal components of x, and $A^\top w$ is the projection of w onto the principal vectors.

If w is sparse, its projection $A^\top w$ will effectively highlight the contribution of the dominant unit(s) in the space of principal vectors. However, whether the resulting weight vector in this transformed space remains compact (sparse) depends on the characteristics of the principal vectors and, thus, on the covariance matrix of the data. This observation underscores that a compact representation in “unit space" does not automatically imply a compact representation in the “mode space”.

The essence of this discussion is that the transformation of the basis (through rotation or otherwise) does not necessarily preserve the sparsity of the representation. Thus, the compactness of representations across different spaces (unit vs. mode) is not a trivial matter and requires careful consideration of the underlying data structure and transformation methods used.

Supplementary Figures

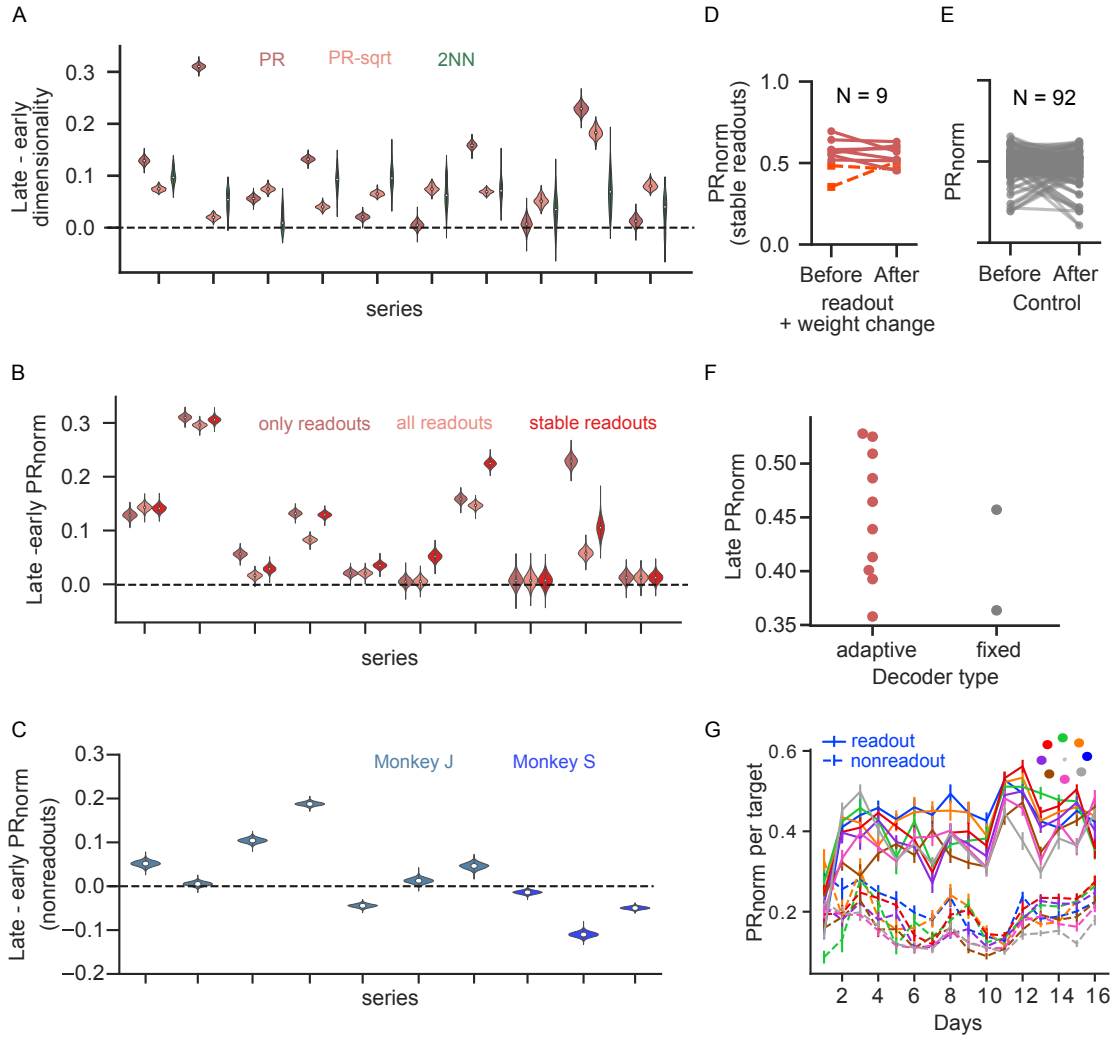

Figure S1. Supplementary figure related to figure 2: dimensionality analysis (A) Change in dimensionality of readout units between late and early computed using Participation ratio (PR, dark red), Participation Ratio after applying square root transform (PR-sqrt, pink), two-nearest neighborhood (2NN, green). Each distribution estimated using 1000 bootstraps. (B) Late - early PR_{norm} for readouts used each day (dark red), all units used as readouts in a learning series (pink), only stable readouts (red). (C) Late - early PR_{norm} for nonreadout populations. (D) PR_{norm} computed using only stable readout units before and after decoder weight changes alongside readout unit changes across all series for both monkeys J and S ($p = 0.496(n.s), N = 9$, Wilcoxon signed rank test). (E) PR_{norm} computed using all readouts for pairs of days where no decoder adaptation occurred ($p = 0.19(n.s), N = 92$, Wilcoxon signed rank test) (F) Late PR_{norm} ranges for adaptive decoder series (Monkey J and S, $N = 10$) and fixed decoder series (from [2], Monkey P and R, $N = 2$). Statistical analyses to compare between fixed and adaptive decoders was not performed due to the small sample size available for fixed decoder studies. (G) PR_{norm} trends per target direction for readouts (solid lines) and nonreadouts (dashed lines) for representative series. In panels A, D and f, each distribution was estimated using 1000 bootstraps. The first 7 series is from Monkey J and last three series is from Monkey S.

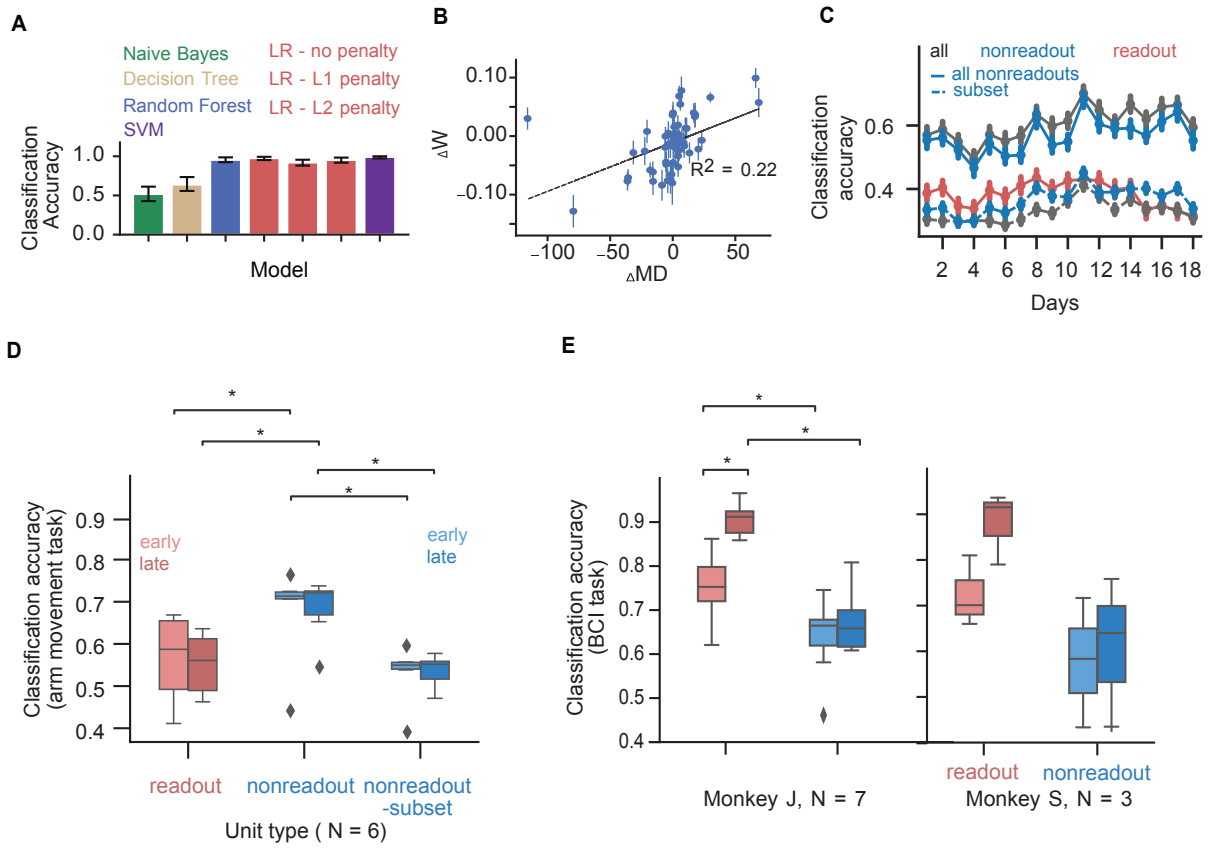

Figure S2. Supplementary figure related to figure 3: credit assignment (A) Classification accuracy obtained from different classifier models when trained on neural data from last day of our representative series. Error bars represent the 95% confidence interval from 1000 train-test splits of the data. (B) Change in logistic regression model coefficients between early and late day as a function of changes in modulation depth of units from representative series. (C) Classification analysis from all units (black), readouts (red) and nonreadouts (blue) when the animal performed an arm movement task for the representative series. There were significantly more units in the 'all' and 'nonreadout' populations compared to the 'readout' population, which contributes to the large difference in overall classification accuracy. Dashed blue and black lines show the classification accuracy achieved from a subset of nonreadout and all units, respectively, randomly drawn to match the number of readout units. (D) Early vs late classification accuracy for readout (shades of red) and non-readout populations(shades of blue) across all series for arm movement task (readout-early vs nonreadout-early : $p = 0.031(*)$;readout-late vs nonreadout-late : $p = 0.031(*)$; nonreadout-early vs nonreadout subset- early: $p = 0.031(*)$; nonreadout-late vs nonredout-late subset: $p = 0.031(*)$, readout-early vs nonreadout-subset early: $p = 0.21(n.s)$; readout-late vs nonreadout-subset late: $p = 0.84(n.s)$, $N = 6$, Wilcoxon signed rank test) (E) Early vs late classification accuracy for readout (shades of red) and nonreadout populations (shades of blue) shown separately for each monkey for BCI task (Monkey J ($N = 7$: nonreadout-early vs. nonreadout-late: $p = 0.29(n.s)$; readout-early vs. readout-late: $p = 0.0156(*)$; readout-late vs. nonreadout-late: $p = 0.0156(*)$; readout-early vs. nonreadout-early: $p = 0.0156(*)$; Monkey S $N = 3$: : nonreadout-early vs. nonreadout-late: $p = 0.5(n.s)$; readout-early vs. readout-late: $p = 0.25(n.s)$; readout-late vs. nonreadout-late: $p = 0.25(n.s)$; readout-early vs. nonreadout-early: $p = 0.25(n.s)$, Wilcoxon signed rank test)

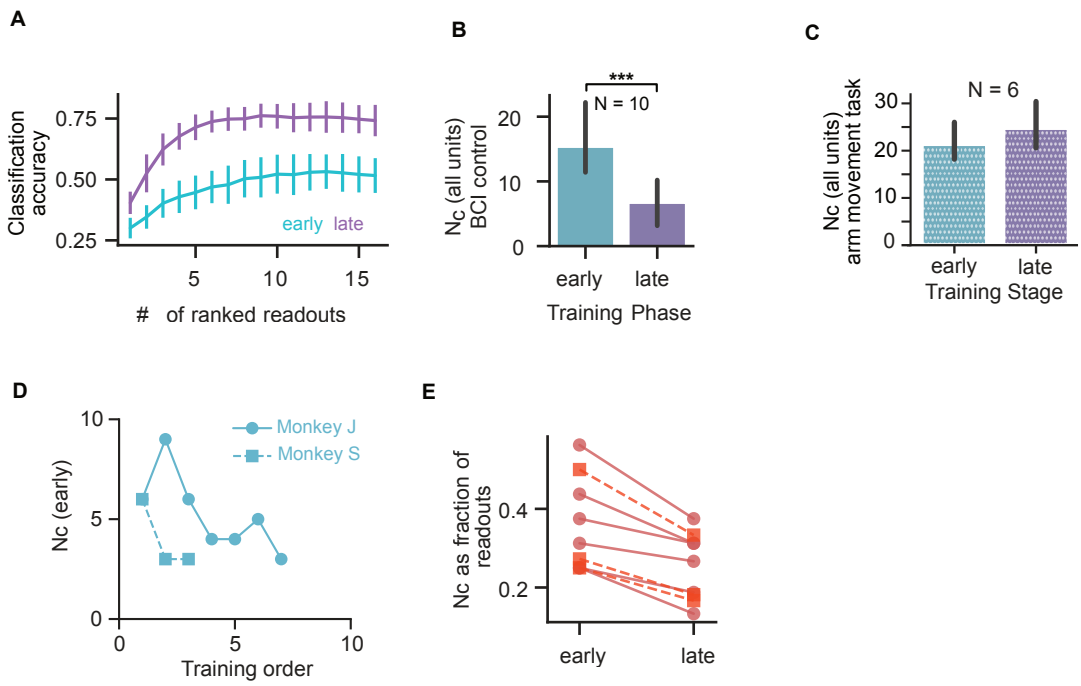

Figure S3. Supplementary figure related to figure 4: compactness (A) Classification accuracy (non-normalized) as units are added in ranked order (ranked NAC) on early (cyan) vs late (purple) day for the representative series (Unnormalized version of Fig 4C). (B) Comparison of N_C^{late} vs N_C^{early} across series while using all units (N_C^{late} is less than N_C^{early}, $p = 0.00098$, $N = 10$, Wilcoxon Signed rank test). (C) Similar to (B) for arm movement task ($p = 0.067(n.s)$, $N = 6$, Wilcoxon Signed rank test) (D) N_c (number of units required to decoder 80% of maximum classification accuracy) for readouts early in training as a function of training order, for Monkey J (solid lines with round markers) and Monkey S (dashed lines with square markers). (E) N_c plotted as a fraction of readouts used for BCI control.

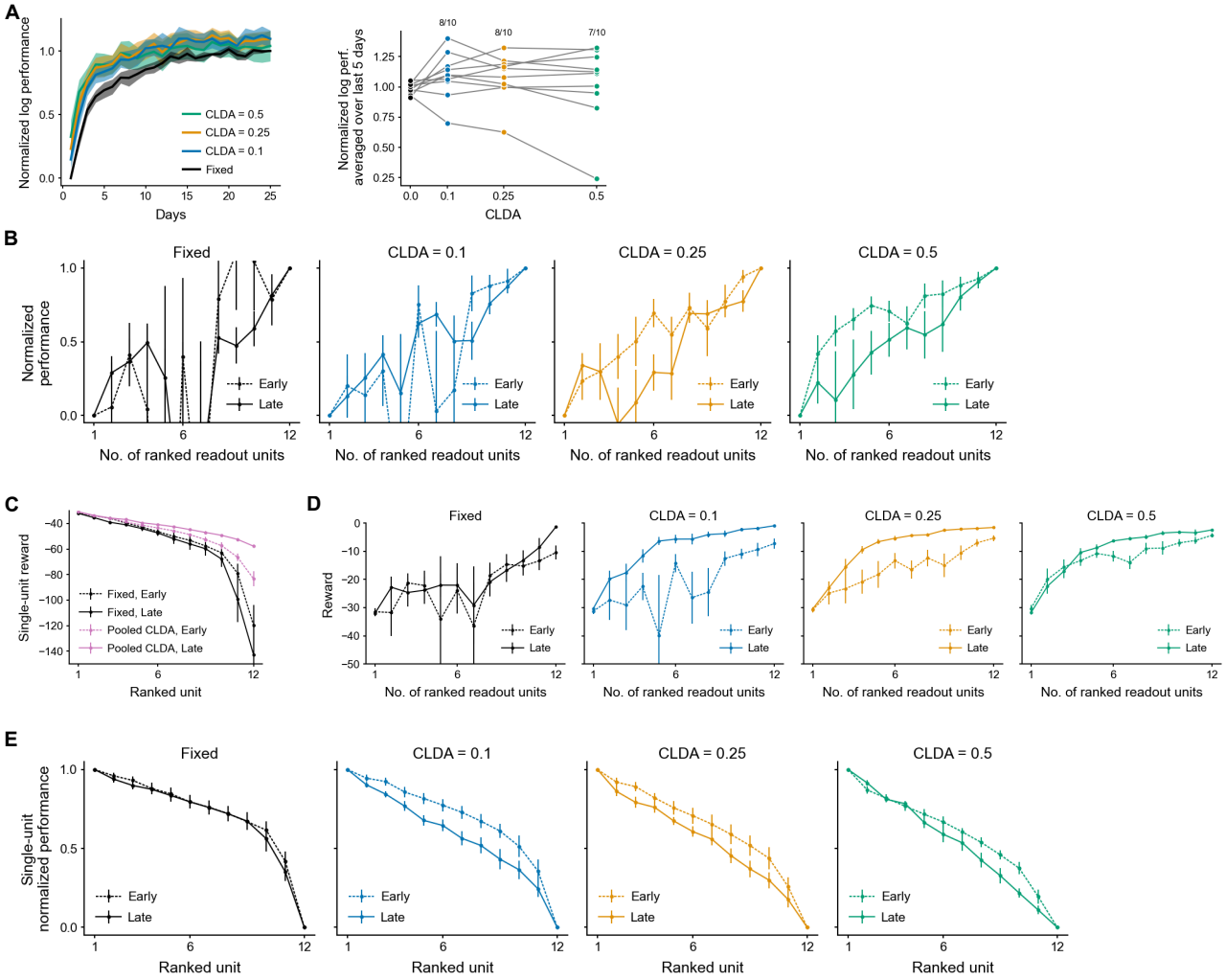

Figure S4. Supplementary figure related to Fig. 5: model. Training performance (A) and compactness (B) when stopping CLDA after a loss criterion was reached (i.e., when loss < 2). Compare with Figs. 5C-D and 5F in the main text. (C-E) Results obtained using the exact same protocol as in Fig. 5 in the main text (i.e., no stopping of CLDA). (C-D) Unnormalized single-unit performances (C) and unit-adding curves (D). Reward, which is the negative of the loss, as a function of number of ranked readout units. Corresponds to Fig. 5E-F in the main text. (E) Single-unit performances for each CLDA intensity. Un-pooled version of Fig. 5E in the main text.

References

[1] A. P. Georgopoulos et al. "On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex". en. In: *Journal of Neuroscience* 2.11 (Nov. 1982). Publisher: Society for Neuroscience Section: Articles, pp. 1527–1537. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.02-11-01527.1982.

[2] Karunesh Ganguly and Jose M. Carmena. "Emergence of a Stable Cortical Map for Neuroprosthetic Control". en. In: *PLOS Biology* 7.7 (July 2009). Publisher: Public Library of Science, e1000153. ISSN: 1545-7885. DOI: 10.1371/journal.pbio.1000153.