Sensorimotor Perturbations to Study Neural Computations of Motor Learning

Pavithra Rajeswaran

A dissertation

submitted in partial fulfillment of theQ

requirements for the degree of

Doctor of Philosophy

University of Washington

202*6*

Reading Committee:
Amy Orsborn, Chair
Azadeh Yazdan-Shahmorad
Steve Perlmutter

Program Authorized to Offer Degree:Q
Department of Bioengineering

University of Washington

Abstract

Sensorimotor Perturbations to Study Neural Computations of Motor Learning

Pavithra Rajeswaran

Chair of the Supervisory Committee:

Professor Amy Orsborn
Electrical & Computer Engineering

Motor learning depends on the brain's ability to link actions to their outcomes and learn/refine internal models through feedback. This thesis examines how sensorimotor perturbations using Brain–Computer Interfaces (BCIs) and reaching behavioral tasks paradigms reveal the neural computations that support learning. We investigated BCI learning experiments with assistive manipulations that improve performance and found that assistive manipulations alter credit assignment learning and drive compaction of learned neural representations concentrating task information in fewer neurons. Second, we adapted visuomotor perturbations, commonly used in motor learning studies, to a new interface that preserves a portion of natural redundancy to study how task relevance and altering task relevance influenced learning. These perturbation paradigms revealed distinct signatures of error based and new controller learning when task relevance is manipulated in redundant spaces. Next, we introduce an electrocorticography based brain computer interface combined with optogenetic stimulation to investigate how rest and network connectivity contribute to learning that occurs outside active practice. Finally, we compared linear and nonlinear approaches for capturing the 'dimensionality'- complexity of activity patterns across neural populations.

Together, these studies present a set of complementary tools that each uncover a different aspect of learning and offer findings that can shape future learning experiments and brain computer interface design. Future studies can integrate these tools, including redundancy-based perturba-

tion paradigms, assistive interventions, measures of network connectivity, and analyses of population structure. Combining these approaches may enable frameworks that link neural computation with large scale network organization and deepen our understanding of the principles that support flexible learning in the brain.

Acknowledgements

This PhD has been a long winding journey, and I am here because of the people who walked with me, pushed me, fed me, encouraged me, or simply reminded me how cool it is to do cutting edge neuroscience research.

To my supervisory committee, Prof. Amy Orsborn, Prof. Azadeh Yazdan, Prof. Rajesh Rao, and Prof. Steve Perlmutter. Thank you for guiding me with insight and understanding. Even in moments when I felt lost, they made me feel like I belonged here. A special place in my heart belongs to Prof. Yazdan, whose mentorship has been extraordinary. Warm, compassionate, scientifically strong, and always genuine. Watching her navigate research and life with honesty and grace made me believe that the life I hoped for was possible. Prof. Perlmutter has been a mentor whose influence began in my very first quarter. His lecture on motor learning sparked a lasting sense of curiosity that stayed with me throughout my PhD. He was always prompt and often the first to respond to my emails when I requested meetings. He invested real time in offering feedback, beginning with my qualifying reports and continuing all the way to this thesis. His guidance strengthened my writing and sharpened my scientific thinking. Prof. Rao has been a steady and grounding presence in all things computational. He met with me at pivotal moments when I needed to make decisions about research direction, and he always brought the focus back to essential questions. What do you want to do. What do you want this work to become. That clarity shaped many important choices.

And above all stands my advisor, Prof. Amy Orsborn. It is difficult to fully capture the impact she has had on my PhD and on the person I became through it. She is a true powerhouse. Watching her set ambitious goals, pursue them with focus, and excel across so many dimensions has been remarkable. Securing tenure at UW while starting her lab during Covid showed what it means to

lead with purpose and resilience.

She offered real freedom in shaping my PhD trajectory, stepping in with guidance and course correction whenever it was needed. Over time we built a relationship rooted in honesty and trust, one in which I felt comfortable bringing concerns, uncertainties, and decisions that reached far beyond the PhD itself. She taught me surgical techniques, how to work with monkeys, how to approach behavioral neuroscience, and how to build confidence in my abilities as a researcher. She opened doors through conferences, fellowship applications, exchange programs like FADEX, and many opportunities that broadened my world. She even recommended me for a panel discussion in her place, a gesture that made me feel truly seen and supported. Long before I learned to believe in myself, she believed in me, and that belief shaped my PhD more than anything else. When I had plans in life that reached beyond my PhD, she reminded me that life does not need to pause for research. Her words became an inner voice that stayed with me far more deeply than she knows. Amy and Azadeh supported me as I learned to juggle the demands of research and motherhood, and helped me show up fully for both.

To my entire PhD committee, I carry a quiet gratitude. I always left our meetings feeling happy, satisfied, a little bit proud, and far more motivated to keep moving forward. Their presence made this journey feel possible.

I am grateful for the opportunity to collaborate with Dr. Alex Payeur and Prof. Guillaume Lajoie on a core portion of my thesis. Alex shaped my scientific growth through rigor, honest feedback, and steady dedication, and our collaboration grew into a friendship that supported me through many stages of my PhD. Guillaume brought exceptional science along with joy and warmth into every meeting. His ability to step back from the details and draw attention to the larger questions helped me connect analysis and results to their broader purpose. Those conversations offered clarity and reflection and strengthened the way I think about research. This collaboration shaped how I approach experiments and the role of artificial intelligence in scientific discovery. Thanks to the Simons Foundation and NSF Accelnet for making this collaboration possible.

To my experiment team, Dr. Leo Scholl and Dr. Lydia Smith. Cutting edge neuroscience is demanding, and running experiments that study learning with neural recordings required hard work

every single day for almost two years. We faced many challenging moments together, and somehow made it through all of them. Their commitment kept the work moving forward when it mattered most. Lydia trained our monkeys and trained us to be ready for in booth experiments with patience and care. Leo guided me through viral injections, optogenetic stimulation, and conducting neural recordings. Lydia and Leo showed up on the hardest recording days and worked through long debugging sessions, often carrying far more of the invisible load than anyone ever sees. So much happened behind the scenes to collect the large datasets that form the foundation of this dissertation, and their involvement shaped this work in essential ways.

To our monkeys Affogato, Beignet, Jeev, and Seba. Their participation made this research possible. Beignet, the first monkey I trained, had a calm and curious nature, brought a steady and cute presence to our sessions. He was so eager to engage with any task we gave him. The way he learned the center out task so quickly remain vivid memories. These animals will always be remembered for their contributions to this work and to science.

To my AO Lab friends, including postdocs Dr. Leo Scholl, Dr. Marios Tringides, Dr. Sofie de Schrijver, Dr. Hao Fang, and fellow graduate students Dr. Maneeshika Madduri, Dr. Si Jia Li, Ryan Canfield, Tomohiro Ouchi, Katherine Perks, Victoria Pierce, AJ, and Mary Bun. Over time, this group made the lab feel like a place where ideas could be bold and mistakes could be survived. The space felt alive, not only because of our fancy coffee machine but because of the people in it.

Si Jia created a welcoming space in lab for me from day 1. Leo and Si Jia shared many long walks and conversations with me, moments that kept me grounded when life and research felt overwhelming. Maneeshika and Si Jia were the people I turned to when I needed clarity, motivation, or simply a pause from my own thoughts. Ryan was an incredible conference buddy who normalized my showing up at SfN with a toddler. Katherine and AJ brought fresh energy into the lab and lifted my spirits with their encouragement and kindness, often reminding me that I was doing better than I believed. To the amazing undergraduates in the lab, Emily Sperry, Catherine Rasgaitis, Mathew Sarti, Brady Brochelt and Jared Darlington. Thank you for your curiosity, hard work, and enthusiasm.

To my former roommates and close friends who have been with me long before this PhD began, Kavya Kannan, Kavya Mendu, and Sanchari Banerjee. Watching all of us earn our doctorates, grow

into our own paths has been one of the quiet joys of these years. Thank you Naveen, Kavya and Anusha for being wonderful friends and being my on call supporters as I navigated paper writing, interviews and major life decisions. To my Seattle circle, Karthik, Viji, Shreya, Shravya, Shravya, Swetha, Kavipriya, Ashwin, and Pooja. Thank you for the laughter, the hikes, the meals, and every moment that reminded me that my world extended far beyond the lab and my PhD.

And like Hogwarts, help was always provided at UW for those in need, especially in the Bio-engineering department. Thank you Kalei Combs for making sure, we received every resource and support we needed to succeed. I remain deeply grateful for a department that truly roots for its students. To the UW community, including Jasmine Zhou, Felix Schwock, Zoe Steine-Hanson, Harsha Gurnani, Patrick Zhang, John Ragland, and Penelope Lilley. Thank you for engaging my scientific curiosities and making the journey lively and fun.

During my PhD years, life refused to sit quietly in the background. It brought challenges, surprises, and one unforgettable blessing, our daughter Yashna. She is my favorite dream come true, filling me with a joy that made me feel unstoppable at times. To my husband, who is now the father of my child, and above all my best friend. He was the Samwise to my Frodo, carrying the emotional backpack when mine was overflowing. He held me together, built a life with me, and somehow made all the chaos feel like home. I cannot imagine doing life without him.

To my sister Monu, who went on her own journey to become Dr. Monisha Rajeswaran and a Forensic medical officer alongside my PhD. In my absence, you quietly carried so much and took care of everything at home in India. I owe you endlessly. You have been an inspiration, a source of admiration and my fiercest supporter. To my dad, the person I always called when life went sideways. Even though sending me across the world terrified him, he let me fly, supported me, and helped me solve problems one milkshake at a time. His quiet faith kept me moving, and his love kept me safe. And last but not the least to my mom. I did this for her. Life did not offer her the opportunities I was given, yet she instilled in me a love for education, independence, and the belief that I could do anything. She never stopped believing in me and made countless sacrifices to lift us forward. I carry her strength and love with me always.

To everyone who stood by me in ways big and small, this dissertation carries pieces of all of you.

To Nature
and to my parents

Contents

List of Figures

List of Tables

Publications from this thesis

Adaptive decoders influence learned representations

(Chapter 2)

Journal

- **Pavithra Rajeswaran**, Alexandre Payeur, Guillaume Lajoie, and Amy L. Orsborn. Assistive sensory motor perturbations influence learned neural representations. *bioRxiv*, March 2024 (under review at *Nature Communications*).

Conference abstracts

- **Pavithra Rajeswaran**, Alexandre Payeur, Guillaume Lajoie, Amy L. Orsborn. Adaptive decoders promote compaction of neural representations during long-term Brain-Computer Interface learning. *Society for Neuroscience* San Diego 2025. [Oral presentation]

- **Pavithra Rajeswaran**, Alexandre Payeur, Guillaume Lajoie, Amy L. Orsborn. Compact yet high dimensional neural representations in co adaptive brain computer interfaces. *Allen Lake Conference*, Seattle 2023.

- **Pavithra Rajeswaran**, Alexandre Payeur, Jose M. Carmena, Guillaume Lajoie, Amy L. Orsborn. Emergence of sparse unit level representations yet increased population dimensionality in brain computer interface learning. *AREADNE*, Santorini 2022.

- **Pavithra Rajeswaran**, Alexandre Payeur, Guillaume Lajoie, and Amy L. Orsborn. Compact neural representations in co adaptive brain computer interfaces. *Computational and Systems Neuroscience (COSYNE)*, Montreal, March 2023.

- **Pavithra Rajeswaran**, Jose Carmena, Amy Orsborn. Adaptive brain computer interfaces allow credit assignment learning and formation of sparse control strategies. *Society for Neuroscience* Virtual 2021.

Context dependent learning in redundant 3D visuomotor perturbations

(Chapter 3)

Journal

- **Pavithra Rajeswaran**, Leo R. Scholl, Lydia I. Smith, and Amy L. Orsborn. "Visuomotor perturbations in a redundant task evoke diverse motor learning dynamics". Manuscript in Preparation.

Conference abstracts

- **Pavithra Rajeswaran**, Leo R. Scholl, Lydia I. Smith, Amy L. Orsborn, "Visuomotor perturbations in a redundant task evoke diverse motor learning dynamics and context-dependent learning", Gordon Research Conference (GRC) on Neuroelectronic Interfaces, Galveston 2024

- **Pavithra Rajeswaran**, Leo R. Scholl, Lydia I. Smith, Amy L. Orsborn, "A novel manual control perturbation task to study formation and adaptation of internal models", Neural Control of Movement (NCM), Victoria 2023

Others

Journal

- **Pavithra Rajeswaran***, Tomohiro Ouchi*, Leo R. Scholl*, Ryan A. Canfield, Lydia I. Smith, and Amy L. Orsborn, "Mapping eye, arm, and reward information in frontal motor cortices using

electrocorticography in non-human primates", *Journal of Neuroscience*, January 2025. *authors contributed equally.

- Leo R. Scholl, **Pavithra Rajeswaran**, Ryan A. Canfield, Amy L. Orsborn, "Stimulation-based connectivity measurements reveal state-dependent, network modulation", Manuscript in preparation.

- Si Jia Li, Lauren Peterson, Leo R. Scholl, **Pavithra Rajeswaran**, Amy L. Orsborn, "Online smooth feature selection via convex optimization", Manuscript in preparation.

Perspective/Opinion piece

- **Pavithra Rajeswaran** and Amy L. Orsborn. Neural interface translates thoughts into type. Nature, volume 593, pages 197–198, May 2021.

Conference abstracts

- Leo R. Scholl, Ryan A. Canfield, **Pavithra Rajeswaran**, Amy L. Orsborn, Stimulation-based functional connectivity measurements reveal state-dependent network modulation, *Computational and Systems Neuroscience (COSYNE)*, Montreal, March 2025.

- Leo R. Scholl, Ryan A. Canfield, **Pavithra Rajeswaran**, Amy L. Orsborn, Measuring functional connectivity across days using optogenetic stimulation, *Society for Neuroscience Annual meeting*, Chicago, September 2024

- Ryan A. Canfield, Tomohiro Ouchi, Leo R. Scholl, **Pavithra Rajeswaran**, Lydia I. Smith, Amy L. Orsborn, Spatially distributed sampling of motor cortex during reaching, *Society for Neuroscience Annual meeting*, Chicago, September 2024

- Jingyuan Li, Leo R. Scholl, Trung Le, **Pavithra Rajeswaran**, Amy L. Orsborn, Eli Shlizerman, "AMAG: Additive, Multiplicative and Adaptive Graph Neural Network For Forecasting Neural Activity", NeurIPS 2023

- Jesus Cabrales Quintanilla, **Pavithra Rajeswaran**, Leo R. Scholl, Si Jia Li, Lydia I. Smith, Amy L. Orsborn , "Saccades and Motor Learning: Examining the Relationship in a Perturbation Task", Simons SURF Symposium, 2023

- Leo R. Scholl, **Pavithra Rajeswaran**, Lydia I. Smith, Amy L. Orsborn, "How eye movements evolve with learning a visuomotor rotation in macaque monkeys", Neural Control of Movement 2023

- Tomohiro Ouchi, **Pavithra Rajeswaran**, Leo R. Scholl, Lydia I. Smith, Amy L. Orsborn, "Arm and eye movement information of electrocorticographic signals in monkeys", Neural Control of Movement 2023

- Ryan A. Canfield, Leo R. Scholl, **Pavithra Rajeswaran**, Lydia I. Smith, Amy L. Orsborn, "Neural population dynamics in large-scale electrocorticography (ECoG) during reaching", Neural Control of Movement 2023

- Leo R. Scholl, **Pavithra Rajeswaran**, Lydia Smith, Amy L Orsborn. "Measuring functional connectivity across days using optogenetic stimulation", Simons Collaboration on the Global Brain 2022

Chapter 1

Introduction

The ability to move purposefully is one of the most fundamental expressions of brain function. From the contraction of a single muscle fiber to the orchestration of complex actions, movement allows animals to interact with and shape their environment. The view that the brain evolved primarily to control movement has deep roots in neuroscience [Wolpert and Ghahramani, 2000]. Under this framework, cognition, perception, and memory serve the ultimate goal of generating adaptive behavior.

What is motor learning? Fundamentally, motor skill learning refers to the incremental improvement in performance with experience [Krakauer et al., 2019; Nieuwboer et al., 2009], manifesting as increases in the speed and accuracy with which movements are executed [Shmuelof et al., 2012; Reis et al., 2009]. When the brain executes motor commands—such as those required to ride a bike—it does not rely on single neurons acting in isolation. Rather, coordinated movements are generated by groups of neurons working together [Churchland et al., 2012; Krishna V. Shenoy and Cunningham, 2013; Saxena and Cunningham, 2019]. Therefore, to truly understand motor learning, we must study plasticity and adaptation at the level of neural populations.

This process encompasses various distinct learning mechanisms, most notably *adaptation* and *de novo* learning. Adaptation refers to the brain's ability to adjust existing motor skills to new conditions (e.g., compensating for a heavy tool), while de novo learning involves the development of entirely new skills (e.g., learning to play the piano) [Krakauer, 2009; Yang et al., 2021]. Through repeated

practice, the nervous system builds and updates a *sensorimotor map*—an internal model that links intended motor actions to their expected sensory consequences [Daniel M. Wolpert and Jordan, 1995; Shadmehr and Mussa-Ivaldi, 1994; Wolpert et al., 2011]. When predictions fail, sensory feedback provides an error signal that drives the modification of future commands. This ability to predict and correct is central to adaptive control and has been formalized as optimal feedback control of movement [Todorov, 2000].

1.0.1 Computational bottlenecks in motor control

Despite our understanding of the general framework, identifying the specific neural computations solved during this process remains difficult. The brain must solve several distinct computational problems to acquire skillful control:

- **Credit Assignment:** The brain receives a vast amount of sensory information simultaneously. Determining which feedback is relevant and which neural activity patterns caused a specific error is known as the *credit assignment problem*. The nervous system must distinguish useful error signals from noise and assign credit to the appropriate synapses or neurons to update the internal model [Richards and Lillicrap, 2019].

- **Redundancy:** High-dimensional neural populations generate low-dimensional behavior. Because there are billions of neurons controlling hundreds of muscles, there are often many ways to achieve the same behavioral goal. The brain must identify and adapt task-relevant dimensions within this redundancy while managing drift in task-irrelevant dimensions [Gallego et al., 2017; Scholz and Schöner, 1999].

- **Multiple Timescales:** Learning occurs on multiple timescales. While rapid adaptation happens online during training, slower processes such as consolidation occur offline during rest or sleep [Karni et al., 1998; Xu et al., 2009]. Distinguishing how these online and offline mechanisms differentially affect neural population dynamics remains a key challenge [Dudai et al., 2015].

One of the many reasons that make studying neural computations challenging is because of the unknown mapping between neural activity an behavior. Billions of neurons distributed across

various brain regions must coordinate to generate the motor commands that control muscles and produce complex behavior [Wolpert and Ghahramani, 2000; Kalaska, 2019]. We do not yet know the specific neural activity patterns that produce these behaviors. Furthermore, it remains unclear where these patterns emerge and how sensory information and internal models are integrated within the motor cortex to generate the neural activity that drives movement [Churchland et al., 2012; Shadmehr et al., 2010; Gallego et al., 2017].

1.1 Behavioral paradigms to study motor learning

To study these computations in a controlled environment, researchers rely on behavioral paradigms that isolate specific aspects of learning.

1.1.1 Sensorimotor perturbations

Perturbation studies serve as windows into the brain's adaptability. By methodically introducing controlled disturbances to the environment or sensory input, experimenters can observe how the brain recalibrates. Classic examples include *force field adaptation*, where unexpected forces are applied to a limb during movement [Shadmehr and Mussa-Ivaldi, 1994; Shadmehr et al., 2010], and *visuomotor rotations*, where the visual feedback of a cursor is rotated relative to hand movement [Krakauer, 2009]. These paradigms force the brain to adjust its sensorimotor maps to counter the disturbance and have been instrumental in distinguishing between fast, error-driven learning and slower, retention-based processes [Smith et al., 2006; Joiner and Smith, 2013].

1.1.2 Brain-Computer Interfaces (BCIs) as a model system

Brain-Computer Interfaces (BCIs) have emerged as a revolutionary tool for studying motor learning. BCIs simplify the input–output relationship between neural activity and behavior: instead of observing billions of neurons controlling muscles through a complex spinal interface, a BCI defines a direct mapping (decoder) between the activity of a select few neurons and an external effector [Carmena et al., 2003; Golub et al., 2016]. In this setting, the musculoskeletal system can essentially

rest, removing biomechanical confounds.

Because the decoder is defined by the experimenter, BCIs allow for precise manipulation of the control loop. By changing or perturbing this decoder[Sadtler et al., 2014; Oby et al., 2019], we can directly study how the brain learns and adapts its neural activity to restore performance [Jarosiewicz et al., 2008; Orsborn and Carmena, 2013]. This dissertation leverages such controlled sensorimotor perturbations in non-human primates to understand the neural computations solved during learning, specifically addressing mechanisms like credit assignment and redundancy resolution.

1.2 Computational tools to study neural changes

Analyzing the massive amounts of neural data collected during longitudinal experiments requires advanced computational frameworks that can capture how learning unfolds across large-scale networks.

1.2.1 Quantifying Neural Dynamics

Historically, motor learning research focused on characterizing individual neurons, examining metrics such as tuning curves, modulation depth, and preferred directions [Georgopoulos et al., 1982]. However, the field has increasingly shifted toward a population-level approach, driven by the recognition that coordinated patterns of activity across neural populations explain behavior more robustly than single units alone [Saxena and Cunningham, 2019; Cunningham and Yu, 2014].

To understand how populations of neurons work together, researchers analyze the covariance patterns of neural activity using dimensionality reduction techniques such as Principal Component Analysis (PCA). Depending on the specific variance of interest, different data conditioning methods are applied; for instance, demixed PCA (dPCA) subtracts condition-independent variance to isolate task-related neural signals [Kobak et al., 2016]. These population methods revealed that for stereotyped behaviors, neural activity is constrained to low-dimensional subspaces or ”manifolds” [Gallego et al., 2017]. Crucially, most of these analyses have focused on well-learned, stable behaviors, establishing that the stability of the neural manifold closely mirrors the stability of the behavior itself.

Consequently, these neural manifolds are thought to be a fundamental feature of motor control, exhibiting stability across time and even across species for similar behaviors [Gallego et al., 2018, 2020].

Studying neural dynamics during learning, unlike for stereotyped behaviors, presents unique challenges, as the neural population activity itself is changing. Techniques such as Factor Analysis have been instrumental in decomposing neural variability into ”shared” variance (reflecting coordination among neurons) and ”private” variance (reflecting independent fluctuations). Application of these methods has revealed that during BCI learning, neurons increase their shared variance, suggesting that learning involves the refinement of coordinated population dynamics rather than independent single-unit adjustments [Athalye et al., 2017; Zippi et al., 2022].

1.2.2 Artificial Neural Networks

Artificial Neural Networks (ANNs) have emerged as powerful testbeds for studying learning under controlled conditions. Unlike biological recordings, ANNs provide a fully observable network where, for the same initializations, we can isolate the effects of specific interventions on learning trajectories. These full observability and controlled simulations allow researchers to test hypotheses about the potential link between network connectivity and learning computations. A seminal BCI study demonstrated that the brain learns new mappings much faster when they align with the pre-existing covariance structure (or manifold) of the neural population, compared to mappings that require activity outside this manifold [Sadtler et al., 2014].

Subsequent modeling work has sought to unpack the mechanisms underlying these learning constraints. Recurrent neural network (RNN) models have provided insights into how different learning rules and network architectures influence the speed and trajectory of adaptation. For instance, recent models have explored how gradient-based learning in redundant networks leads to specific patterns of solution discovery [Payeur et al., 2023] and how feedback-driven plasticity rules can constrain learning to specific manifolds [Feulner and Clopath, 2021].

More recently, Driscoll et al. [2024] demonstrated that RNNs trained on multiple tasks develop ”dynamical motifs”—reusable population dynamics such as attractors or rotations—that can be flex-

ibly reconfigured to support new behaviors. This framework suggests a compositional neural code where motor learning may involve identifying and recruiting existing dynamical structures rather than learning entirely new patterns de novo. Furthermore, RNNs have been instrumental in modeling *representational drift* and *continual learning*, showing how networks can balance the stability required for memory with the plasticity needed for adaptation [Duncker et al., 2020; Driscoll et al., 2022]. Comparing these artificial learning frameworks with biological learning datasets presents testable hypotheses about the computations the brain uses to solve motor control problems.

1.3 Interventional tools for neural dynamics study

Understanding how the brain adapts during learning requires tools that can not only observe but also causally influence neural activity [Grosenick et al., 2015]. While correlational studies (recording) reveal representation changes, moving to mechanisms requires direct intervention in the neural circuit.

1.3.1 Optogenetics

Optogenetics stands at the forefront of neuroscience tools, offering the ability to activate or inhibit neurons with millisecond precision and cell-type specificity. This technique has been instrumental in dissecting the circuit-level mechanisms of motor learning, particularly in rodent models.

Ensemble reorganization: Work by Komiyama and others has used longitudinal imaging combined with optogenetics to reveal that motor learning is accompanied by the emergence of reproducible spatiotemporal activity patterns [Peters et al., 2014]. As learning progresses, the motor cortex refines its activity: task-relevant ensembles become more reliably active, while task-irrelevant activity is suppressed [**?**Peters et al., 2017]. Recent work using holographic optogenetics has gone further, demonstrating that the targeted stimulation of these specific ”learned” ensembles is sufficient to drive the learned motor behavior, establishing a direct causal link between the consolidated neural pattern and the motor output [Wu et al., 2025].

Input-specific plasticity: Interventional tools have also revealed how upstream inputs are remapped

during learning. For instance, recent studies show that motor learning actively reshapes the influence of thalamic inputs onto motor cortex layer 2/3 neurons to enable reliable movement execution [Ramot et al., 2025]. Furthermore, optogenetic manipulations of specific inhibitory interneuron subtypes (e.g., somatostatin-expressing cells) have shown that they play a critical, subtype-specific role in regulating the stability of dendritic spines during the learning process [Chen et al., 2015].

Translation to Primates: While much of this mechanistic work has been pioneered in rodents, recent advances have translated these capabilities to non-human primates. Large-scale interfaces now allow for simultaneous optogenetic stimulation and electrophysiological recording across broad cortical areas [Yazdan-Shahmorad et al., 2016]. This enables the dissection of specific circuits, such as the frontal eye fields (FEF) during oculomotor tasks [Acker et al., 2016], and provides a bridge to understanding these mechanisms in complex, high-dimensional motor systems similar to humans.

1.3.2 Electrical stimulation

While optogenetics offers cell-type specificity, intracortical microstimulation (ICMS) remains a gold standard for probing causal links in non-human primate motor areas due to its temporal precision and ability to interface directly with clinically relevant scales. Electrical stimulation serves two critical roles in studying motor learning.

Inducing Plasticity: First, it allows researchers to artificially induce plasticity. By pairing stimulation with neural activity or behavior (spike-timing-dependent plasticity, STDP), experimenters can strengthen specific corticospinal connections [Nishimura et al., 2013]. Pioneering work demonstrated that an electronic neural implant connecting unconnected motor cortical neurons could induce stable functional reorganization, effectively ”wiring” neurons together through artificial feedback loops [Jackson et al., 2006].

Artificial Feedback and Error Signals: Second, stimulation can inject artificial information into the cortex. Recent work has shown that ICMS can deliver specific instructional cues to premotor cortex [Mazurek and Schieber, 2017] or mimic error signals that drive motor adaptation [Vyas et al., 2020b]. By delivering microstimulation to specific cortical areas during movement, researchers can mimic sensory prediction errors or perturb ongoing population dynamics. This has been used to

demonstrate that error signals in motor cortices can drive trial-by-trial adaptation, providing causal evidence for the role of these regions in error processing and credit assignment [Inoue et al., 2016; Balasubramanian et al., 2020].

1.4 Dissertation overview

This dissertation investigates the computational principles of learning within the context of neural representations. It comprises four interrelated chapters that examine how the brain learns experimenter-defined sensorimotor mappings and adapts to perturbations.

Chapter 2: Assistive perturbations influence learned neural representations This chapter investigates how assistive decoders influence neural representations during BCI learning. Combining analysis of neural recordings at both single-neuron and population levels with artificial neural network (ANN) modeling, we shed light on how machine learning techniques interact with the brain's innate learning processes. and how assistance shapes the trajectory of credit assignment.

Chapter 3: Context dependent learning in 3D visuomotor pertubations in redundant task This study introduces a reaching behavioral task perturbation paradigm to study how task relevance is learned in redundant environments. By manipulating task relevance in a redundant workspace, these experiments probe how the brain solves the credit assignment problem - identifying and refining task-relevant dimensions while navigating task irrelevant variability.

Chapter 4: Training paradigm using ECoG BCI Control to probe learning mechanisms after practice This chapter extends our investigation to an electrocorticography (ECoG)-based BCI with fixed decoders. We utilized a protocol where network connectivity was mapped using optogenetic stimulation before and after BCI training to isolate learning-induced plasticity. Additionally, we incorporated rest periods into the experimental design to probe the contribution of offline mechanisms, examining how neural representations reorganize or stabilize after active practice.

Chapter 5: A Comparative Analysis of Dimensionality Metrics on Spiking data Finally, we focus on the methodological tools for measuring neural population dynamics. While it is essential to study learning at the level of neural populations, it remains unknown how current dimensionality estimation techniques capture the covariance patterns of population variance. By applying and comparing

different dimensionality estimation techniques to neural population activity collected during a BCI reaching task, this analysis provides an unbiased metric for characterizing the complexity of neural activity.

Together, these studies aim to clarify how neural populations adapt to perturbations, assign credit, and reorganize during and after practice. The results contribute to a mechanistic understanding of motor learning and inform the design of future adaptive neuroprosthetic systems.

Chapter 2

Assistive algorithms influence neural representations in motor brain-computer interfaces

2.1 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. Over time, task-relevant information became concentrated in fewer neurons, unlike with fixed decoders. At the population level, task information also became largely confined to a few neural modes that accounted for an unexpectedly small fraction of the population variance. A neural network model suggests the adaptive decoders directly contribute to forming these more compact neural representations. Our findings suggest that assistive decoders manipulate error information used for long-term learning computations like credit assignment, which may explain the altered neural representations and inform real-world BCI design.

2.2 Introduction

Motor learning has been defined as the process by which motor skills are acquired and maintained Krakauer et al. [2019]. Learning depends on feedback error signals from multiple sources Berniker and Kording [2008] and in multiple forms Oza et al. [2024], which are corrected to improve task performance and refine movement Shadmehr et al. [2010]; Wolpert et al. [2011]; Seidler et al. [2013]. These errors must be communicated to relevant neural circuits while accounting for the complex pathways linking neural activity to the motor workspace, a challenging credit assignment problem Minsky [1961]. In reaching tasks, error-related neural activity has been observed in the motor cortex and other regions Diedrichsen et al. [2005]; Inoue et al. [2016]; Inoue and Kitazawa [2018]. Although the mechanisms by which such errors drive both widespread Dayan and Cohen [2011] and highly localized Wang et al. [2011]; Xu et al. [2009]; Fu et al. [2012] neurophysiological changes remain incompletely understood, we expect them to manifest as changes in the neural representations of task-relevant information.

Intracortical brain-computer interfaces (BCIs) provide a powerful framework to study how motor errors shape neural changes Golub et al. [2016]; Orsborn and Pesaran [2017]. BCIs define a mapping ("decoder"; Fig. 2.1A) between neural activity and movement Chapin et al. [1999]; Carmena et al. [2003]; Taylor et al. [2002] to causally interrogate learning. Perturbations of well-learned BCI mappings, analogous to perturbations of natural movement Shadmehr and Mussa-Ivaldi [1994]; **?**, have been used to probe the neural mechanisms of short-term adaptation by introducing task errors Jarosiewicz et al. [2008]; Sadtler et al. [2014]. Alternately, novel BCI mappings learned over multiple practice sessions can be used to study neural mechanisms related to skill formation Ganguly and Carmena [2009]; Koralek et al. [2012]; Athalye et al. [2020]; Orsborn and Carmena [2013]. BCI skill learning studies often use *fixed* decoders that provide limited performance, thereby placing substantial demands on neural adaptation to reduce task errors. In contrast, application-driven BCIs often use *adaptive* decoders, whose parameters are updated during use, to *improve* or *maintain* performance over time (Fig. 2.1A,B) Orsborn et al. [2012]; Li et al. [2011]; Gilja et al. [2012, 2015]; Shenoy and Carmena [2014]; Jarosiewicz et al. [2015]; Pandarinath et al. [2017]; Taylor et al. [2002]; Orsborn et al. [2014]; Benabid et al. [2019]. These assistive interventions modify the mapping between

neural activity and movement without introducing task errors (and potentially reducing them). Even when performance is merely preserved, the new mapping may alter how neural activity would need to change to further improve performance (i.e., the credit assigned to neurons). These interventions thus provide an opportunity to study how changes to neuron-to-behavior mappings that preserve performance could influence skill learning mechanisms.

BCI skill learning alters both single-neuron and population activity. Practice with a fixed decoder over several days refines performance, stabilizes neural representations Ganguly and Carmena [2009] and improves coordination among neurons Athalye et al. [2017]; Zippi et al. [2021], which is consistent with reduced neuronal variability in natural motor learning Dhawale et al. [2017]. Some changes are targeted to specific neurons: neurons used as direct "readouts" of behavior are modulated differently than "nonreadout" neurons, which may participate in computations but do not drive movement directly (Fig. 2.1B). Preferential modulation of readouts with extended practice is consistent with the possibility that credit assignment computations contribute to skill learning Zippi et al. [2021]; Ganguly et al. [2011]; Gulati et al. [2017]; Arduin et al. [2013]; Zhou et al. [2019]; Silversmith et al. [2021]. Population-level changes also appear to be primarily subserved by readouts Zippi et al. [2021].

Long-term learning may also occur during extended practice with adaptive BCI decoders. Assistive decoders adapt the neural-movement mapping to reduce errors, and also help compensate for unstable measurements by modifying which neural units are part of the readout ensemble Orsborn et al. [2014] (Fig. 2.1B). Despite this dynamic mapping, adaptive decoders can help produce skillful control of BCI over time Orsborn et al. [2014]; Silversmith et al. [2021]; Benabid et al. [2019]; Taylor et al. [2002], and reduce the amount of neural changes needed, such as shifts in neuronal tuning Orsborn et al. [2014]. Yet, longitudinal adaptive BCI experiments consistently report performance improvements beyond what would be expected from decoder adaptation alone Collinger et al. [2013]; Wodlinger et al. [2015]; Benabid et al. [2019]; Silversmith et al. [2021]; Orsborn et al. [2014]; Taylor et al. [2002]; Agudelo-Toro et al. [2024], consistent with long-term skill learning processes taking place.

It is unclear how assistive decoder interventions impact learning and the resulting neural repre-

sentations. We hypothesized that they might alter feedback that contributes to the targeted neural changes that occur during skill learning. Better understanding this process could inform BCI applications that widely use these manipulations and provide insights into the computations guiding long-term sensorimotor learning. However, examining brain-algorithm interactions is challenging because it requires comparing learning outcomes between different paradigms (fixed vs. adaptive decoders) where variations in task performance, individuals, and training history complicate interpretations. Existing work has therefore focused on performance-based comparisons of different BCI algorithms without examining their impact on neural representations (e.g., Willsey et al. [2022]; **?**]; Gilja et al. [2012, 2015]; Jarosiewicz et al. [2015]).

We used experimental data from longitudinal BCI learning studies Ganguly and Carmena [2009]; Orsborn et al. [2014] and recurrent neural network (RNN) models to explore how adaptive decoders influence learned neural representations. The experimental data provided insights into real-world neural adaptation, while the RNN model allowed for direct comparisons that are not possible experimentally. Our findings show that learning-mediated changes were targeted to the readouts, consistent with credit assignment computations. But surprisingly, these changes were mostly concentrated in a subset of readouts when decoders adapted. In contrast, fixed-decoder experiments did not show such *compaction* with learning. Our model helped ascertain the direct involvement of decoder adaptation in the compaction process, and offered insight into how adaptive decoders could influence plasticity. Modeling also suggested opportunities to tune brain-decoder interactions to balance performance goals. Finally, data analysis of population activity revealed that learning mainly targeted low-variance components, which are often overlooked by standard methods Dyer and Kording [2023]. Together, our findings provide evidence that adaptive decoders influence neural encoding on multiple levels, shedding new light on how error manipulations shape neural activity during learning.

2.3 Methods

2.3.1 Experiment

Neural data and behavioral task

We analyzed neural activity as two male rhesus macaques (*Macaca mulatta*, monkeys S and J) learned to move a 2D cursor controlled via neural activity to perform a point-to-point reaching task while the relationship between neural activity and cursor movement was intermittently adapted across days (Fig.2.1C). The task required animals to initiate a trial by moving the cursor to the central target. After reaching the center, one of eight peripheral targets appeared. After a brief hold at the center, a target color change cued the animal to start reaching. Successfully completing a trial required the monkey to enter the peripheral target within a specified reach time ($3 - 10$ s) and hold the cursor inside the target briefly ($250 - 400$ ms). The monkey then received a liquid reward. Because we were primarily interested in neural representations of movement direction, we analyzed all trials in which the animal successfully reached the peripheral target regardless of whether the peripheral hold was successful or not.

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. Readout units (also referred to as "readouts" for brevity) were chosen based primarily on their recent history of measurement stability across days, assessed qualitatively by experimenters by observing properties of the recordings over time like spike waveforms. Functional properties, such as tuning for arm movements or offline task predictive power, were not used to inform readout unit selection. 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.

All procedures were conducted in compliance with the NIH Guide for the Care and Use of Lab-

oratory Animals and were approved by the University of California at Berkeley Institutional Animal Care and Use Committee.

BCI control with adaptive decoders

In general, "fixed" and "adaptive" decoders differ in their update policy: fixed decoders retain the parameters estimated during calibration, whereas adaptive decoders update their parameters during use to maintain or improve performance. These terms refer to the update policy rather than to the decoder architecture itself. In our datasets, the fixed decoders used a Wiener filter Ganguly and Carmena [2009], while the adaptive decoders used a position-velocity Kalman filter (KF) Kalman [1960]; Gilja et al. [2012]; Orsborn et al. [2012, 2014] to control the 2D cursor using readout neural activity. KF parameters were trained and intermittently updated using previously-developed algorithms that estimate parameters during closed-loop BCI control (closed-loop decoder adaptation, CLDA) Orsborn et al. [2012]. 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 Orsborn et al. [2012] (monkey J; see below) which updates a subset of the KF parameters and adapts iteratively on small amounts of data to improve performance, or the ReFIT algorithm Gilja et al. [2012] (monkey S) which updates all KF parameters using a single parameter adaptation based on larger amounts of data to stabilize or refine performance. Despite these small differences in timescales of adaptation, both algorithms operate on the same principle of reducing task errors by adapting decoder parameters to better match the animal's intent.

Monkeys practiced with a given decoder mapping (and subsequently updated versions of that mapping) for multiple days, defining a learning series. CLDA served two main purposes: (1) to enhance closed-loop performance from the initial decoder, and (2) to maintain performance despite shifts in neural activity, such as the loss of a unit within the BCI readout ensemble. The initial CLDA session (day 1) of a decoder series lasted approximately 5–15 minutes, providing the subject with sufficient performance to reach all targets successfully. Intermittent updates included (1) alterations to decoder parameters without changing the readout unit identities (weight change only)

if there was a noticeable performance drop, assessed as a success rate decrease of approximately 10%–20%, and (2) 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. S1A). In these cases, CLDA was briefly run for 3–5 minutes, corresponding to 1-2 updates of the KF weights. Readout unit swaps and weight adjustments were made only if previously used readout units could no longer be isolated or identified. When a performance drop from the previous day was observed but all readout units remained present, only weight changes were made without altering readout units. CLDA therefore had two primary effects: initial CLDA led to large decoder changes that boosted performance to facilitate reaches to all targets, akin to initial "calibration"; mid-series CLDA made notably smaller adjustments to decoder parameters, which primarily maintained or modestly improved performance Orsborn et al. [2014]

Monkey J and monkey S learned 13 and 6 adaptive decoder mappings in total, respectively Orsborn et al. [2014]. Both monkeys had brief (on the order of one month maximum) initial training in BCI cursor control prior to the adaptive decoder learning series studies. Our 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. For monkey J, inter-series intervals for data included in our analyses ranged from 2 to 42 days (median: 11 days; mean: 15.5 days; $n = 6$ intervals). For monkey S, intervals ranged from 1 to 40 days (median & mean: 20.5 days; $n = 2$ intervals). Longer series tend to have more CLDA events (Fig. S1D). Information about the number of readouts and CLDA events in each learning series is shown in Supplementary table 1. Monkey S had a reduced frequency of mid-series CLDA, which does not necessarily imply better task performance than Monkey J, but may potentially reflect more stable neural recordings. Several of Monkey S's series were shorter than Monkey J's, which may have contributed to the reduced number of CLDA events. The original study intentionally manipulated the degree of initial CLDA to vary BCI performance on Day 1, with each learning series having different initial performances based on the extent of decoder adaptation (Fig. 2.1C, Fig. S1C). Early success rates varied widely: 21–86% (mean: 48%) for monkey J and 35–49% (mean: 40%) for monkey S. Some series began with lower initial performance due to minimal or no CLDA applied at the

outset. Full details on adaptive decoder methods and the dataset are provided in Orsborn *et al.* Orsborn et al. [2014].

Data analysis

Learning analyses and behavioral performance metrics

We quantified long-term learning changes during a series by comparing ”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.

We assessed behavioral improvements using standard performance metrics such as mean success percentage and reach times (Fig 2.1C). 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.

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 (from go-cue to a range of 500 to 1800 ms post go-cue) was not found to qualitatively change reported findings (S2D). 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.

Logistic regression model

We analyzed changes in neural representations with multiclass logistic regression (LR) models that used neural activity (Figs. 2.2, 2.3) or transformations thereof (Fig. 2.5; see below), to predict target identity. The classifier model received flattened activity vectors composed of the activities of all units across time bins between go-cue to reach completion for each trial, and generated an output corresponding to the predicted target label. To control for variability in the number of trials com-

pleted 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 remaining trials on a day were used as a test set. Unit firing rates were z-scored across trials for each set separately. We performed 100 training-test splits on each day, randomly selecting 200 trials for training (25 trials per reach direction) and designating the remaining trials for the test set in each draw. This approach allowed us to calculate the 95% confidence interval on classification accuracy (displayed as error bars). We used the mean accuracy across the training-test splits to compare classification accuracy across series and animals (Fig. 2.2E).

We implemented our LR models using the Python library Scikit-learn noa with the L-BGFS solver and L2 regularization (hyperparameters: C = 1, max_iters = 1000). We did not observe any considerable changes in classification accuracy between LR models with or without regularization on late days (Fig. S2A). The hyperparameter C represents the inverse of regularization strength, with smaller values indicating stronger regularization, which helps prevent overfitting by penalizing large coefficients. We conducted a grid search for C ranging from C = 1e-5 to 1, across different neural populations (all, readouts, and nonreadouts). The purpose of this analysis was to confirm that our results were not merely due to the model performing unevenly across these groups. We found that classification accuracy was not more or less sensitive to the choice of C for any particular neural population (Fig. S2B). We note that this is not an analysis aiming to optimize the model's classification performance for each population.

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. 2.2B). We quantified the similarity of decoder weights across days by calculating the similarity between the model weight matrices W on consecutive days, W^d, W^{d-1} (Fig. 2.2C), similar to previous analyses Ganguly and Carmena [2009]; Orsborn et al. [2014]. This evaluation was performed by calculating the correlation coefficient between the flattened matrices of model weights (weights for each target concatenated) for 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. 2.3A, left). We then ranked units based on their classification accuracy (Fig. 2.3A, middle and Fig. 2.3B). A ranked-ordered Neuron Adding Curve (NAC) was produced by calculating the classification accuracy with the highest ranking unit, then with the first and second highest ranking units, and so on (Fig.2.3A, right and Fig.2.3C,F) Lebedev [2014]. 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. 2.3D,G).

The "combinatorial" NAC, as described in the main text, tested all possible combinations of units for each size of ensemble. We evaluated the separability between the classification accuracy distributions with and without the top N_c units (Fig. 2.3H-J) 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. 2.5A). All trials with completed reaches were used for PCA each day, and readout firing rates were z-scored per 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. 2.5B-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 splitting PCs into two groups (top 50% and bottom 50%) (Fig. 2.5F). 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 Jolliffe [2002]. 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. 2.5E), 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. 2.5H). 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. 2.5I).

2.3.2 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: $p > 0.05$. The absence of a statistics comparison line in figures between two groups indicates that no statistically significant difference was found. 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 the supplementary material, control data from $N = 6$ learning series obtained from monkey J related to the arm movement task were used (Suppl. Figs. S2E,F and S3C).

2.3.3 Model

Simulations were implemented in C++ using the Eigen library Guennebaud et al. [2010]. Analyses were performed in Python, using the Numpy and Scipy packages. Values for parameters are included in Table 2.1. Code will be made publicly available upon publication.

Arm model

Our RNN model was initially trained to control a planar torque-based arm Li and Todorov [2004] 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

$$\mathcal{M}(\mathbf{q})\ddot{\mathbf{q}} + \boldsymbol{c}(\mathbf{q}, \dot{\mathbf{q}}) + \mathcal{B}\dot{\mathbf{q}} = \boldsymbol{\tau} \tag{2.1}$$

where

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

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

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

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 2.1), 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 $\tau_f \dot{\boldsymbol{\tau}} = -\boldsymbol{\tau} + \mathbf{u}$ Todorov [2006]. In simulations, these equations were discretized in time using the standard 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 model was then trained to perform BCI control. The BCI decoder was a velocity Kalman filter—a simplification compared to Ref. Orsborn et al. [2014]— 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, $N_{\text{readout}} = 12$. These readout units were selected at random, with the condition that their mean activity across trials and targets exceeded a threshold of 1, ensuring that only active units were included. 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 Gilja et al. [2012]. 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. Orsborn et al. [2012]. To decode the cursor velocity while measuring

neural activity online, we used the update equations

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

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 $\hat{\mathbf{x}}_0 = \mathbf{0}$ with perfect precision, so that the error covariance matrix was $P_0 = 0$.

Closed-loop decoder adaptation (CLDA) was implemented following SmoothBatch Orsborn et al. [2012, 2014] using data recorded under brain control. Recorded velocities were rotated towards the current target, representing intended velocities Gilja et al. [2012]. 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 values from the last step were replaced according to

$$X \leftarrow \alpha\hat{X} + (1 - \alpha)X, \tag{2.2}$$

where $X \in \{C, Q, \mathbf{m}\}$ and $0 \leq \alpha \leq 1$ is the *CLDA intensity*; $\alpha = 0$ corresponds to a fixed decoder. In the model, CLDA was performed each day irrespective of performance, unlike in experiments, where within-series CLDA was applied only when performance declined.

Network model

The RNN contained N units fully connected by a recurrent weight matrix W (see Table 2.1). 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

$$\mathbf{v}_{t+1} = \mathbf{v}_t + \alpha_v[-\mathbf{v}_t + W\boldsymbol{\varphi}(\mathbf{v}_t) + \mathbf{i}_t + \mathbf{b}] + \sigma\sqrt{\alpha_v}\boldsymbol{\xi}_t \tag{2.3}$$

where $t = 0, 1, \ldots, T-1$. These dynamics integrate, with time constant α_v^{-1}, the following inputs: recurrent inputs from units in the network ($W\boldsymbol{\varphi}(\mathbf{v})$); an external input representing task information (**i**), e.g., the target position (see below); a bias term (**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 **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 **d**, end effector position **x** and context **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 Todorov [2000]: $\mathbf{u}_t = V\mathbf{r}_t = V\boldsymbol{\varphi}(\mathbf{v}_t)$, 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

$$L = \sum_{k=0}^{K-1} L_k \tag{2.4}$$

where

$$L_k = \frac{\|\mathbf{x}_T - \mathbf{d}_k\|^2}{\delta_p^2} + \gamma_v \frac{\|\dot{\mathbf{x}}_T\|^2}{\delta_v^2} + \gamma_a \frac{\|\ddot{\mathbf{x}}_T\|^2}{\delta_a^2} + \frac{\lambda_u}{T} \sum_{t=0}^{T-1} \|\mathbf{u}_t\|^2. \tag{2.5}$$

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_a 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 2.1).

Learning

The recurrent weights W, input weights U and biases **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 Williams [1992]; Chen and Qiao [2020]. The noise $\boldsymbol{\xi}$ independently applied to all units in Eq. 2.3 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

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

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 $\overline{R}_k$ provided an estimate of the expected reward for target k Frémaux et al. [2010] and was computed as a moving average

$$\overline{R}_k = \alpha_R \overline{R}_k^{\text{prev}} + (1 - \alpha_R) R_k, \tag{2.6}$$

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 ($\overline{R}_k^{\text{prev}}$) in computing the new reward estimate. For U, one simply replace $\mathbf{r}_t$ by $\mathbf{r}_t^{\text{in}}$; for **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 Kingma and Ba [2017].

Analysis of the model

To allow clear comparison across seeds (i.e., random initializations of the network's parameters) and CLDA intensities, 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. 2.5C) as

$$P_d = \frac{\log_{10}(L_1^{(0)}) - \log_{10}(L_d)}{\log_{10}(L_1^{(0)}) - \log_{10}(L_{-1}^{(0)})}, \tag{2.7}$$

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 sought to transform the loss for the fixed decoder to the interval $[0, 1]$.

Single-unit normalized performances (Fig. 2.5E) were computed by first evaluating the reach performance for each individual unit and ranking them from the most important to the least important. The performance of each unit was measured by restricting the Kalman filter parameters $\mathbf{m}$, C and Q to that unit and recomputing the Kalman gain K_t accordingly. This ensured that only that unit was controlling the cursor. Now, 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

$$S_i = \frac{L_{N_{\text{readout}}} - L_i}{L_{N_{\text{readout}}} - L_1}. \tag{2.8}$$

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

A similar transformation was performed to obtain the normalized performance in Fig. 2.5F. 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

$$\hat{R}_i = \frac{L_1 - L_i}{L_1 - L_{N_{\text{readout}}}}. \tag{2.9}$$

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

Combinatorial unit-adding curves (Fig. 2.5G) 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:

$$\frac{L_{\text{min}}^{(\text{fixed})} - L_c}{L_{\text{min}}^{(\text{fixed})} - L_{\text{max}}^{(\text{fixed})}}, \tag{2.10}$$

where L_c is the loss incurred for a specific combination.

The coefficients of determination in Fig. 2.5H were computed using the function `linregress` from the Python package `scipy.stats`. Each point in the scatter plots represents the change in a

recurrent weight during learning with an adaptive decoder versus the weight change of the same connection with a fixed decoder, for a given network realization (random seed). Each plot contains 100^2 weights for each of the 10 realizations.

2.4 Results

Two rhesus macaques (monkeys J and S) learned to perform point to point movements with a BCI controlled cursor (Fig. 2.1A), as previously described Orsborn et al. [2014]. Multi unit action potentials from microelectrode arrays implanted in the arm area of the primary and premotor cortices were used as inputs to a Kalman filter decoder that controlled cursor velocity and position. A subset of all recorded units was selected as input to the decoder based on recording properties (e.g., recent history of measurement stability over time) and defined the *readout* population (Fig. 2.1A) (see Methods).

Each monkey learned to control the cursor over several consecutive days, defined as a learning series. We focused on learning series lasting at least four days ($n = 7$ series for monkey J; $n = 3$ for monkey S). Learning series varied in length, initial decoder training methods, and number of decoder updates. Decoder weights were updated via Closed Loop Decoder Adaptation (CLDA), a family of algorithms that adjust decoder parameters during ongoing use by re estimating intended movement so that decoded cursor velocities realign with the target, thereby reducing task errors and visual error feedback (see Methods and Fig. S1A). In our datasets, CLDA was implemented with Smooth-Batch Orsborn et al. [2012] (monkey J) or ReFIT Gilja et al. [2012] (monkey S), two closely related algorithms with slight differences in which Kalman filter parameters are updated and the timescales of adaptation. CLDA was done at the beginning of a learning series, improving initial performance to provide sufficient cursor control throughout the workspace. CLDA was also done intermittently within series to maintain performance when neural measurements varied. Longer series generally had more decoder adaptation events (Fig. S1D). As previously shown Orsborn et al. [2014], task success rates and reach times improved over days (Fig. 2.1B, a selected series for monkey J used below in all single series examples, for consistency; see additional example series in Fig. S1B (monkey S) and Fig. S1E (monkey J)). These learning curves qualitatively resemble those reported previously

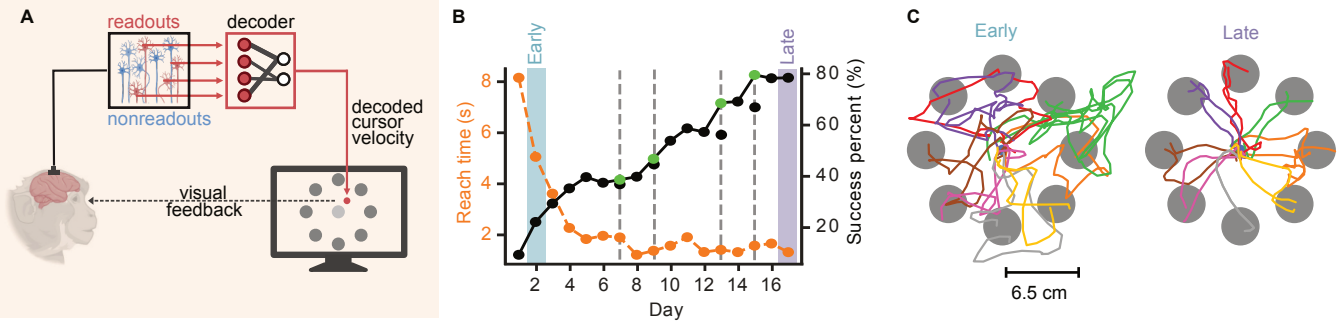

Figure 2.1: **(A)** Closed loop BCI setup. Multi unit activities from readouts (red) and nonreadouts (blue) were recorded in a monkey's motor cortex. Only the readout activity was decoded to control a cursor in a 2D center out reaching task with visual feedback. (Monkey and neuron icons from BioRender.) **(B)** Mean reach time (dashed orange line) and success percentage (solid black line) across days for an example learning series (monkey J). Early (blue) and late (purple) training phases are indicated. Vertical grey dashed lines indicate decoder adaptation, with green dots showing performance gains from decoder adaptation. **(C)** Early (left) and late (right) cursor trajectories for the series in panel B.

using a fixed decoder Ganguly and Carmena [2009]; Athalye et al. [2017]; Zippi et al. [2021].

We quantified learning within each series by defining an 'early' and 'late' training day, which corresponded to the first and last day of a learning series with at least 25 successful reaches per target direction, respectively (day 2 and 17 in Fig. 2.1B). We were primarily interested in characterizing neural representations related to movement direction, and therefore included any trial wherein the target was reached regardless of whether the hold at the target was successful. Improvements in task success rate and reach time were accompanied by more direct and accurate reaches (Fig. 2.1C , Fig. S1C and Orsborn et al. [2014]). The dataset included series where the amount of CLDA performed on day 1 varied, resulting in varying initial performance, but behavioral improvements consistent with long term skill learning were observed in all series Orsborn et al. [2014].

2.4.1 Credit assignment to readout units during long term learning with adaptive decoders

We first assessed whether adaptive decoders impacted credit assignment in motor cortex by investigating how task relevant information was distributed between readouts and nonreadouts. Because cursor kinematics changed significantly across days, we instead used an offline classification analysis to quantify how the fixed target identities were encoded. Target identity does not change over days, facilitating cross day comparisons despite behavioral changes (Fig. 2.1C). Specifically, each day we

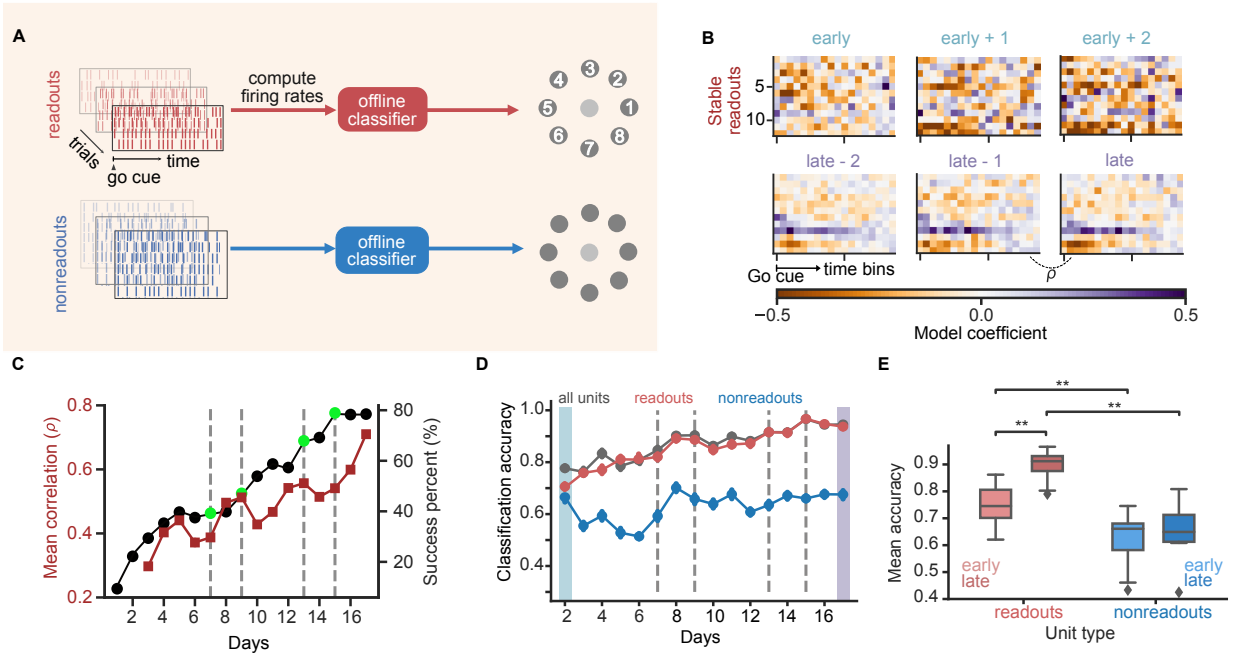

Figure 2.2: **(A)** Offline classification model trained on neural activity from readouts (red) and nonreadouts (blue). **(B-D)** Classification results for the selected series (Fig. 2.1B). **(B)** Heatmap representation of the classifier weights to predict a single target (target 2) for the stable readout population on multiple days. **(C)** Success percentage (black; reproduced from Fig. 2.1B) and mean correlation coefficient between classifier weights on consecutive days (dark red) across days. **(D)** Classification accuracy for all units (gray), readouts (red) and nonreadouts (blue). Error bars represent 95% confidence intervals on test accuracy (see Methods). **(E)** Early and late mean accuracy for readouts (shades of red) and nonreadouts (shades of blue) for all series for both monkeys. Pairwise comparisons (two sided Wilcoxon signed rank test, $n = 10$): readout early vs readout late, $^{**}p = 0.006$; readout early vs nonreadout early, $^{**}p = 0.002$; readout late vs nonreadout late, $^{**}p = 0.002$; nonreadout early vs nonreadout late, $p = 0.13$ (ns).

fitted a multiclass logistic regression to predict target identity using neural activity aligned on the go cue (Fig. 2.2A), evaluating classification accuracies using cross validation. This classifier accurately predicted target identity per trial using all units, with mean accuracies of 0.78 ± 0.13 (monkey S) and 0.83 ± 0.11 (monkey J) across all days and all learning series.

The logistic regression model weighted the contribution of each unit's firing rate across time bins between the go cue and trial end towards classifying target identity. These weights thus represent an encoding of task information, and are related to parametric analyses of neuron activity such as direction tuning (Fig. S2C). We found that optimized classifier weights varied day to day, but became more similar as learning progressed for the units that were consistently used for BCI control across the whole series (the "stable readouts") (Fig. 2.2B). The correlation coefficient of these weights on consecutive days increased during learning (Fig. 2.2C; combined across both monkeys, $N = 10$: $\rho_{\text{early}} = 0.34 \pm 0.045$, $\rho_{\text{late}} = 0.64 \pm 0.045$; $p = 0.001$, Wilcoxon signed rank test, late $>$ early), consistent with a progressive stabilization of task encoding representations that mirrors performance improvements (Fig. 2.1B). These findings are consistent with previous observations that adaptive BCIs lead to stable directional tuning in readout neurons Orsborn et al. [2014].

Target classification accuracy improved during BCI training when using all neurons, paralleling the gains in BCI performance (Fig. 2.2D). However, this improvement was mostly driven by the readouts (Figs. 2.2D, E). Classification accuracy was higher for readouts compared to nonreadouts both early and late in training despite large differences in their respective numbers (Supplementary Table 1). Similar results were obtained with other classifiers and with regularization (Fig. S2A). More importantly, only the readouts' accuracy increased significantly from early to late (Fig. 2.2E; Fig. S2G for individual animals), showing that learning related changes in classification accuracy are driven predominantly by changes within the readout population. We observed a similar separation in information encoding between readout and nonreadout populations when analyzing only the time period immediately after the go cue (0 to 200 ms), which further controls for potential differences in reach kinematics with learning (Fig. S2H). The same analysis on data during arm movements, in contrast, showed no significant separations between readout and nonreadout populations (Fig. S2E,F), ruling out factors such as variability in neural recording quality and confirming credit assignment

was specific to the BCI task Ganguly et al. [2011].

Finally, we examined the temporal relationship between decoder interventions and credit assignment. We compared classification accuracies pre and post decoder change events and observed no significant changes (Fig. S2I). This suggests that credit assignment computations were not immediately impacted by decoder changes and instead occurred over longer timescales of practice. Together our results show that assistive decoder perturbations lead to the formation of a stable BCI task neural encoding that is primarily supported by readout units, similar to BCIs with fixed decoders Ganguly et al. [2011]; Ganguly and Carmena [2009].

2.4.2 Compaction of neural representations with long term learning alongside decoder adaptation

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 after learning (Fig. 2.2B, bottom row). This suggests further credit assignment within the readout population that has not been reported in past BCI studies. Does long term learning lead certain readouts to become more important? We explored this question by conducting a rank ordered neuron adding curve (NAC) analysis Lebedev [2014]. This technique quantifies how classification accuracy improves as neurons are sequentially added to a decoding model, ranked from most to least predictive (Fig. 2.3A).

We analyzed NACs from longitudinal BCI studies with both fixed Ganguly and Carmena [2009] and adaptive decoders Orsborn et al. [2014] to assess whether task encoding was concentrated in a few neurons or distributed across the readout population across learning. On each day (early, late), we quantified the classification accuracy of a unit in isolation (Fig. 2.3B,E), and then constructed a rank ordered NAC by adding units to our decoding ensemble from most to least predictive (Fig. 2.3C,F). Since early fixed decoder performance was poor, we grouped data into epochs of 150 trials (early: first 150, late: last 150), consistent with previous analysis Athalye et al. [2017]; Zippi et al. [2021]. Adaptive decoder datasets allowed analysis across days due to sufficient early trials.

The relationship between classification accuracy and the number of ranked readout units required for prediction differed between fixed and adaptive decoders. Overall classification accuracy

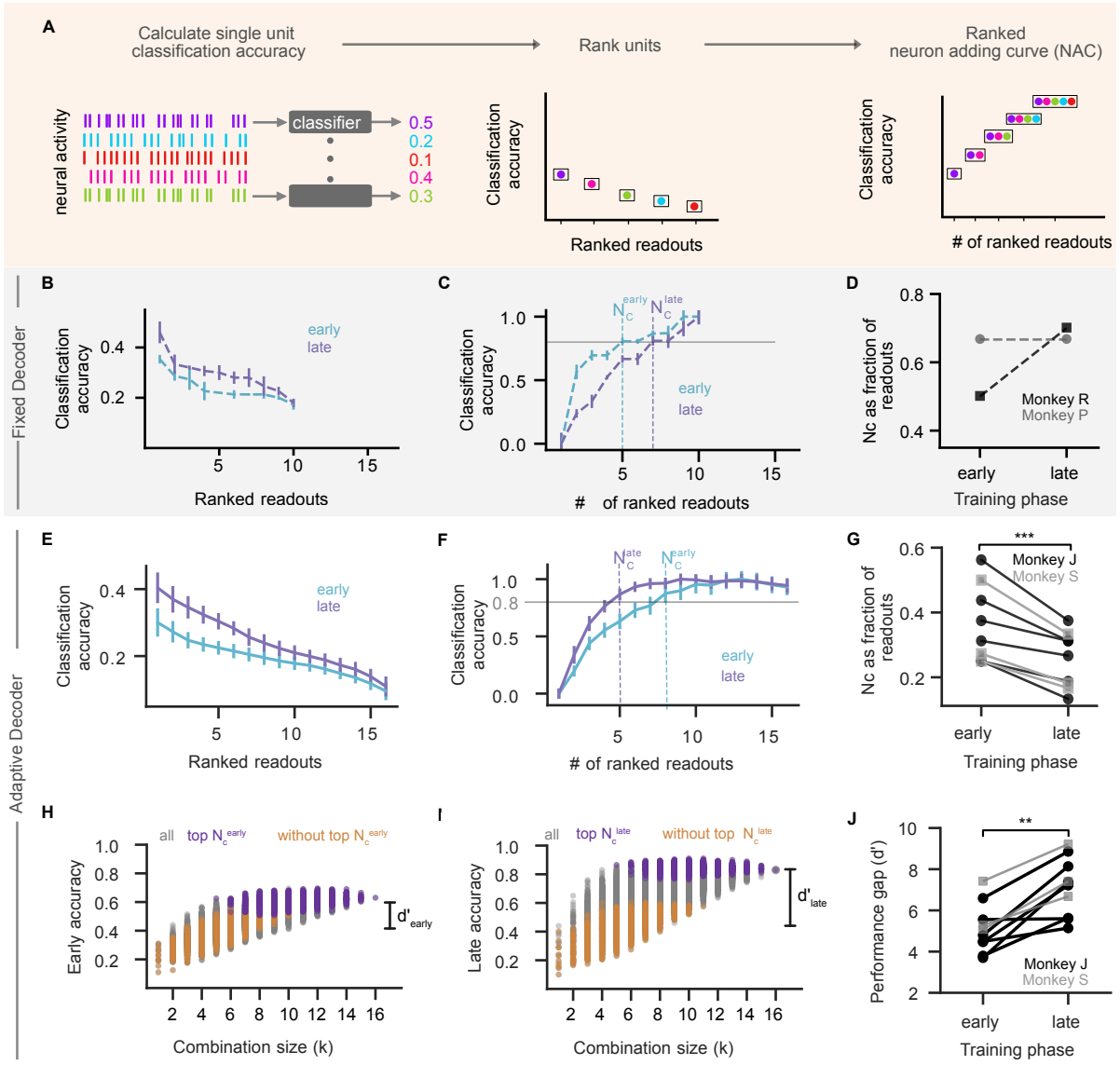

Figure 2.3: **(A)** Ranked Neuron Adding Curve (NAC) analysis (see Methods). **(B)** Ranked single unit accuracy on early (cyan) and late (purple) epoch from fixed decoder dataset (example from monkey R). See text for the definition of "epoch". **(C)** NAC for early and late epoch using ranking from B, normalized to each epoch's peak performance. Vertical dashed lines indicate number of units to reach 80% of peak accuracy (N_c). **(D)** Comparison of N_c^{late} and N_c^{early} (monkey P, grey circles; monkey R, black squares). **(E-F)** Same as B C for the selected series for the adaptive decoder, where early and late refer to days. **(G)** Comparison of N_c^{late} and N_c^{early} as a fraction of total readouts (late < early, $^{***}p = 9.8 \times 10^{-4}$, $n = 10$, one sided Wilcoxon signed rank test). Monkey J: black circles; monkey S: grey squares. **(H-I)** Combinatorial NACs for early (H) and late (I): accuracy for readout combinations of different sizes. Combinations may contain all top N_c units (deep purple), no top N_c units (orange), or a subset of them (grey). Distance between purple and orange distributions (performance gap) measured by discriminability index d'. **(J)** Comparison of early vs late d' 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 G.

improved between early and late learning (Fig. S3A). We therefore examined the number of units contributing to task encoding by normalizing each day's NACs (Fig. 2.3C,F) and then computing the number of neurons (N_c) required to reach 80% of the normalized classification accuracy at each learning stage. In fixed decoders, late N_c was greater than or equal to early N_c for the two available series (Fig. 2.3C,D). Our modeling results below further supported this observation, compensating for the limited data (Fig. 2.4F, left). In contrast, adaptive decoders showed the opposite trend, with a consistently smaller late N_c than early N_c (Fig. 2.3F,G, Fig. S3E). This effect in adaptive decoder data was even larger when including all recorded units (readouts and nonreadouts, Fig. S3B). We did not observe this effect in arm movement tasks, suggesting that the changes were specific to BCI learning and not due to recording quality (Fig. S3C). Beyond this within series (early vs late) trend, N_c qualitatively decreased across chronologically ordered series for each animal (Fig. S3D). There was no clear relationship between the reduction in N_c and the number of readout units added or removed by decoder perturbations (Fig. S3F), consistent with credit assignment computations occurring on longer timescales rather than tied to any individual decoder intervention. Compaction showed no significant relationship with either initial performance or performance improvements across learning series (Fig. S3G,H).

Next, we examined whether this reduction in N_c carried over to the encoding of kinematic variables in addition to target identity. We found that a smaller subset of readout units, identified using the classification model, was sufficient to accurately reconstruct cursor velocities offline late in learning compared to early (Fig. S3I K and Supplementary Methods). Qualitatively, reconstruction accuracy was noticeably lower when we used readouts other than the top N_c units (Fig. S3I). This suggests that the same units which increasingly contributed to target encoding during learning also contributed to skilled movement. We term this reduction in the number of units needed to accurately decode behavior over time "compaction" to indicate that the brain appears to progressively encode task information in a smaller number of neural units when learning is assisted, in contrast to fixed decoder scenarios (Fig. 2.3D,G).

Lastly, we asked whether compaction reflected a situation where task information was encoded within a specific subset of readout neurons, i.e. the units identified by the NAC analysis. The al-

ternative would be a more general increase in encoding efficiency where other combinations of N_c units would yield similar classification performance. To address this question thoroughly, we performed a combinatorial variant of the NAC 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) or did not contain any of the top units (orange) (Fig. 2.3H,I). Early and late in learning, combinations with top units held the most classification power (Fig. 2.3H,I). Comparing the distributions between early and late revealed 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') (see Methods). This performance gap increased with learning across series (Fig. 2.3J), showing that learning related changes are targeted to a specific subset of readout neurons in BCIs with adaptive decoders.

2.4.3 Assistive decoder adaptation contributes to learning more compact representations: modeling analysis

Our analysis revealed that neural representations became more compact over multiple days of BCI training with assistive decoder perturbations (Fig. 2.3E G), but not without them (Fig. 2.3B D). However, experimental BCI 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. We therefore simulated BCI task acquisition using an artificial neural network (Fig. 2.4A) to directly compare learning strategies for identically initialized neural networks trained with fixed or adaptive decoders.

We developed a minimal yet biologically plausible RNN model representing a motor cortical circuit that captured the basic learning dynamics of BCI experiments (Fig. 2.4A). We first trained the network on a center out arm movement task (Fig. 2.4B, left) to establish a plausible inductive bias for subsequent BCI training and to mirror experiments Orsborn et al. [2014]. The network received context information about task type (arm vs BCI) along with the effector and target positions. After

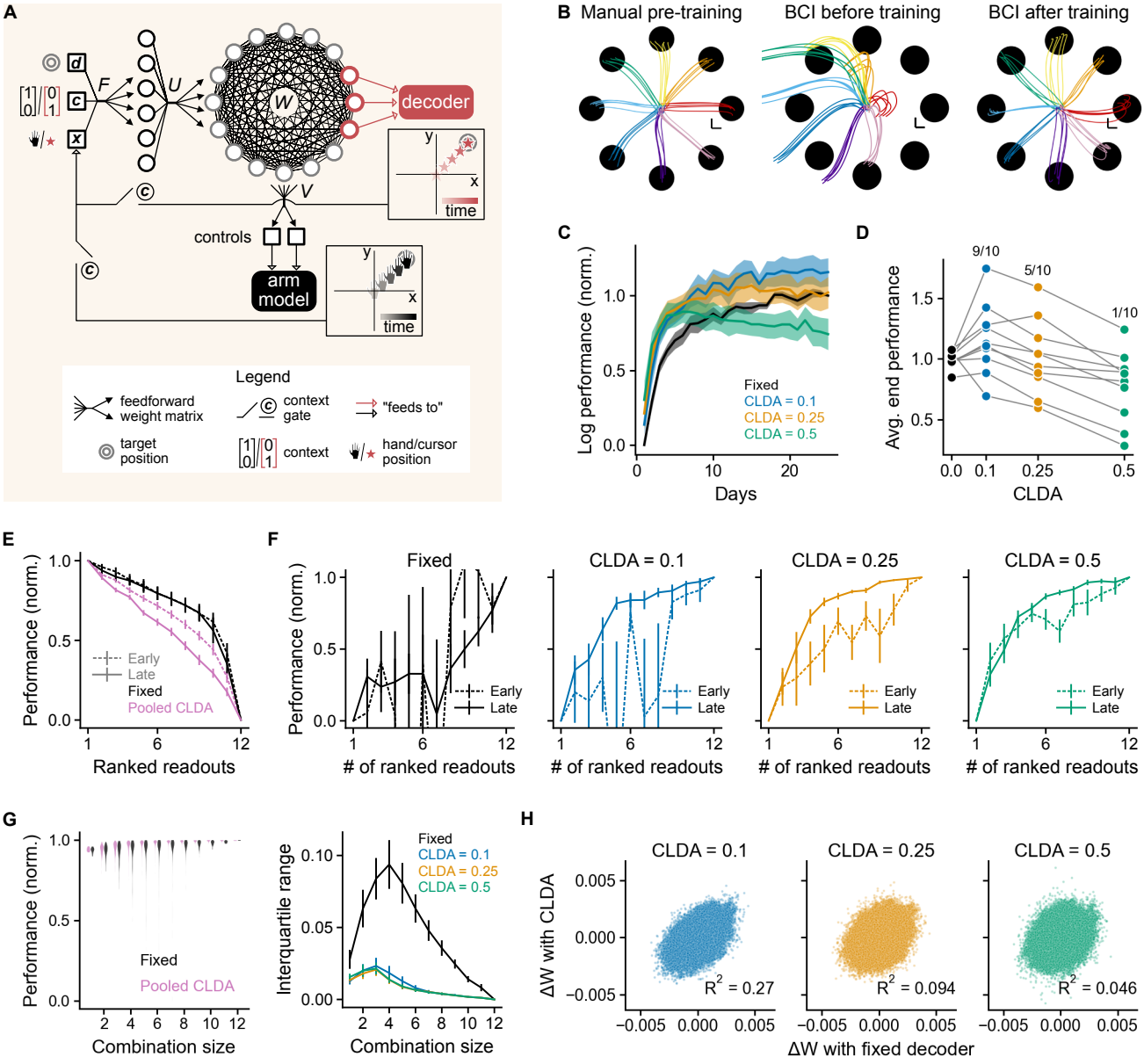

Figure 2.4: (A) The RNN received visual feedback about target (**d**) and end effector (**x**) positions. Based on context (**c**), it controlled either an arm model with all units or a BCI cursor with readout units (red). F, U, W and V indicate the encoding, input, recurrent and output weight matrices, respectively. See text for details. **(B)** Learning stages. Left: train on arm movement task. Middle: fit decoder parameters to arm trajectories. Right: set context to 'BCI' (here CLDA intensity = 0.25) and train on BCI task. Scale bars = 1 cm. **(C)** Normalized log performance (Methods, Eq. 2.7) for different CLDA intensities. **(D)** Normalized log performance averaged over last 5 days. Ratios at top are the fraction of seeds with better performance with adaptive decoders compared with fixed decoders. **(E)** Ranked single unit normalized performance (Methods, Eq. 2.8). Results for adaptive decoders were pooled across CLDA intensities (pink lines). **(F)** Neuron adding curves (NACs) using online performance (Methods, Eq. 2.9). **(G)** Late combinatorial NAC 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 vs 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.

initial training, 12 units out of 100 were selected as readouts and a velocity Kalman filter (see Methods) was fitted to manual reach trajectories (Fig. 2.4B, middle). The context then switched to 'BCI' and the network was trained on the BCI task (Fig. 2.4B, right). In all contexts the REINFORCE algorithm Williams [1992] updated recurrent (W, Fig. 2.4A) and input (U) weights, while the encoding (F) and output (V) matrices remained fixed after random initialization. REINFORCE does not rely on backpropagating task errors through the effectors, making it more biologically plausible.

Adaptive decoding in the model followed the principle of CLDA algorithms used in experiments Orsborn et al. [2014], focusing on decoder updates when readout ensemble remained unchanged (see Methods). CLDA was performed on each "day", corresponding to seeing each target 100 times. Regardless of CLDA intensity (a factor representing how much the decoder changed; Eq. 2.2), adaptive decoders accelerated early performance improvements compared to fixed decoders (Fig. 2.4C). Moderate CLDA intensities maintained or improved final BCI performance (Fig. 2.4D) while preserving generalization properties comparable to fixed decoder training (Fig. S4A). However, excessive adaptation degraded final performance (Fig. 2.4D, CLDA $= 0.5$), likely because CLDA's objective (reach towards the target) conflicted with the task objective (reach the target with vanishing velocity in a specified time) late in training (Methods, Eqs. 2.4-2.5). Stopping high intensity CLDA once a performance threshold was reached (Fig. S4B) or increasing the number of days between adaptations (Fig. S4C) mitigated this issue. Overall, in the model, decoder adaptation facilitated rapid BCI task acquisition, as in experiments. Our results also underscore potential shortcomings of long term CLDA applications at high intensity.

Next, we explored whether adaptive decoder learning led to more compact representations in the model. While this was assessed offline in the data (Fig. 2.3), the model allowed for an online evaluation using the task loss as the performance metric (Methods, Eqs. 2.4-2.5). Each individual readout unit controlled the cursor in turn and its reach performance was evaluated both 'early' (end of first day) and 'late' (end of last day). Readout units were then ranked according to their performance. Early and late ranked single unit performances (Methods, Eq. 2.8) differed only for adaptive decoders (Fig. 2.4E and Fig. S4D,E), with faster decrease late in training compared to early (Fig. 2.4E, pink lines). Thus, the dominant units became relatively more dominant with learning on

average under network decoder co adaptation. Using these rankings, we then computed a ranked NAC (Methods, Eq. 2.9). Consistent with our experimental findings, adaptive decoders contributed to the compaction of representations of task performance (Fig. 2.4F and Fig. S4F) and kinematic variables (Fig. S4G). With a fixed decoder, ranked NACs were highly variable across seeds, with no sign of compaction as 11 (out of 12) ranked readout units were required to reach 80% of the maximal performance late in training (Fig. 2.4F, left). Notably, stopping CLDA early or increasing the interval between CLDAs not only preserved performances (as mentioned above), but also tended to prevent (stopping, Fig. S4H) or reduce (increased intervals, Fig. S4I) the emergence of compact representations. Overall, our model confirms and expands on our observations that decoder adaptation singles out certain units for BCI control and progressively shapes more compact neural representations.

Using the online loss metric, we next obtained combinatorial unit adding curves (Fig. 2.4G, left) and computed the corresponding interquartile ranges (IQR; Fig. 2.4G, right). These IQRs measured the dispersion of performance for each combination size (Methods, Eq. 2.10). The IQRs were significantly greater for a fixed decoder, which was caused by few units with very low contributions to the overall performance (Fig. 2.4E and Fig. S4D). Therefore, while decoder adaptation promoted more compact representations with a few dominant units, it also mitigated the impact of unreliable ones.

The model suggests that neural plasticity alone, without decoder adaptation, does not lead to compaction. Thus, decoder adaptation must interact with the model's plasticity to generate 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. 2.4H). All connections among and across readout and nonreadout units were included. 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 as CLDA intensity increased, indicating that the total weight changes with an adaptive decoder became increasingly uncorrelated from those obtained with a fixed decoder. These results suggest that brain decoder co adaptation drives distinct neural plasticity patterns and contributes to compaction.

2.4.4 Compact representations encode task information in low variance population modes

Our neural data analysis and model revealed that compaction at the unit level in neural representations emerges as the result of long term learning processes occurring alongside an adaptive decoder. Does this reshaping of sensorimotor encoding at the level of individual neurons reflect a broader reorganization within the neural population? Many studies represent coordinated neural activity with principal components (PCs) capturing the most prominent directions of variability Cunningham and Yu [2014]; Pandarinath et al. [2018]; Gallego et al. [2018]. In this view, each PC can be treated as an independent component or "mode" of neural population activity. Importantly, compaction at the unit level does not guarantee compaction at the population level (see Supplementary Discussion).

We explored how compaction changed population structure by performing a variation of our neuron adding curve analysis in which population level modes served as features for classification rather than individual units (Fig. 2.5A). For early and late days, we performed principal component analysis, ranked PCs based on their classification accuracy (Fig. 2.5B,E) and computed a rank ordered PC adding curve (Fig. 2.5C,F). Similar to the compaction observed at the unit level, we found that the number of PCs required for 80% classification accuracy decreased with learning (Fig. 2.5E,F, S5D). This observation confirms compaction of representations at the level of populations. In contrast, applying the same PC based analysis to the fixed decoder dataset revealed no reduction in the number of PCs required to reach 80% classification accuracy across learning (Fig. 2.5B D). This parallels our single unit analyses, where fixed decoders also did not exhibit compaction, and further supports that compaction in PC space emerges specifically in the presence of adaptive decoders.

We further observed that variance became more broadly distributed across PCs in the readout population after learning, suggesting an increase in dimensionality (Fig. 2.5H, top). To validate this observation, we calculated the dimensionality of readout activity using the participation ratio, a covariance based dimensionality measure Recanatesi et al. [2019]; Gao et al. [2017], normalized by the number of readouts (PR_{norm}). Unlike in natural motor learning studies where neural variability tends to shrink Dhawale et al. [2017] or fixed decoder BCI studies where "shared" dimensionality decreases Athalye et al. [2017]; Zippi et al. [2021], dimensionality remained the same or increased during BCI learning with adaptive decoders (Fig. S5A,C). Re analyzing fixed decoder data using PR_{norm}

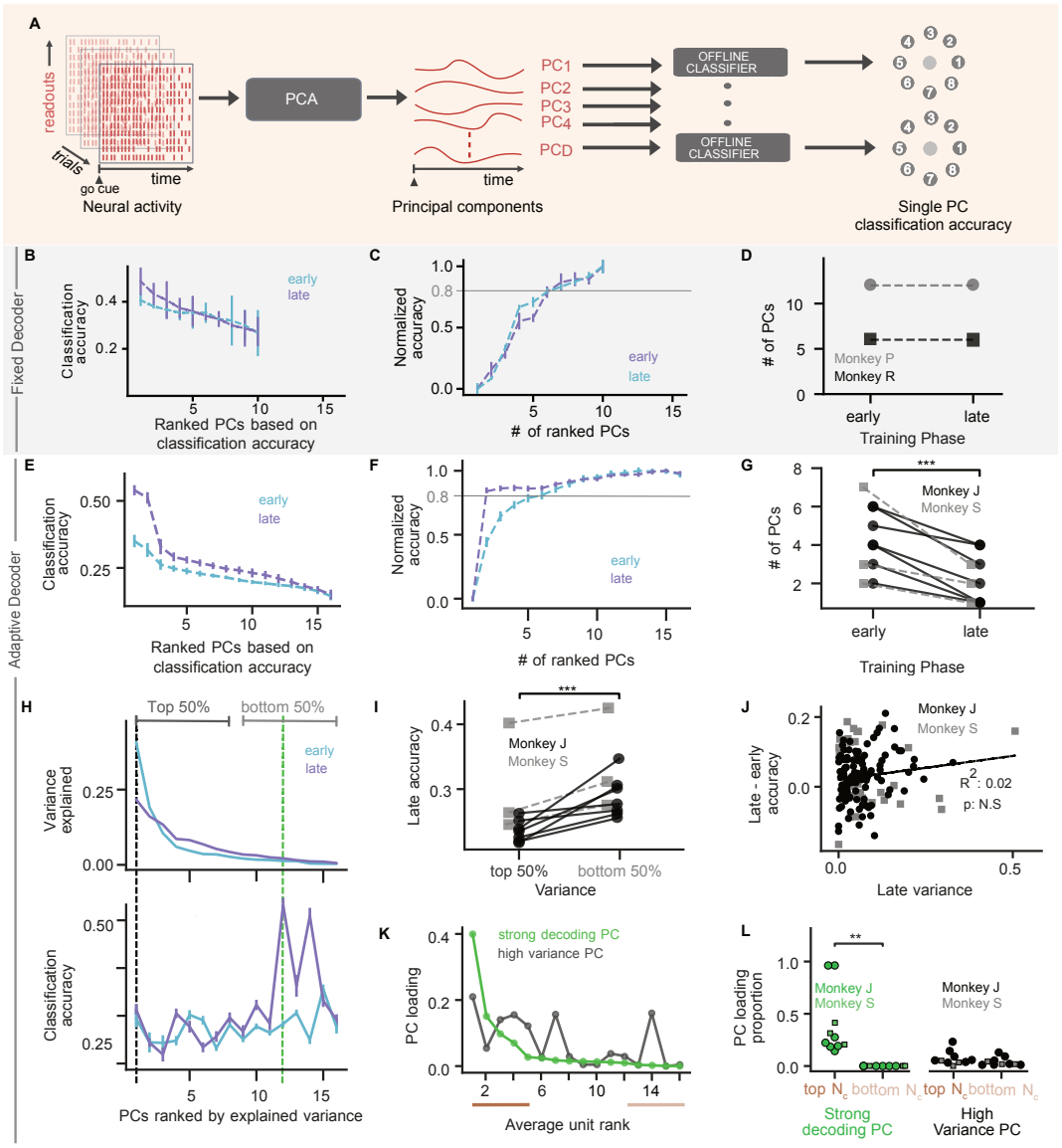

Figure 2.5: (**A**) Decoding analysis with PC modes (see text). (**B**) Early (cyan) and late (purple) ranked classification accuracy of individual PCs with fixed decoder (monkey R). (**C**) Ranked PC adding curve using panel B ranking. (**D**) Number of PCs needed to reach 80% accuracy (monkey P: grey circles; monkey R: black squares). No statistical test: $n = 2$ and no effect. E G Same as B D but for the adaptive decoder, using the selected series in E,F. (**G**) Monkey J: black circles; monkey S: grey squares. Compaction statistics: $^{***}p = 9.7 \times 10^{-4}$, $n = 10$, late $<$ early, Wilcoxon signed rank test. **(H)** Variance (top) and accuracy (bottom) explained by each PC, ranked by variance explained. Error bars: 95% confidence intervals from 10^4 random draws of trials. Strongly decoding PC (dashed green line) has highest accuracy. **(I)** Late accuracy from top 50% vs bottom 50% PCs. Top 50% is lower than bottom 50%: $^{**}p = 0.001$, $n = 10$, one sided Wilcoxon signed rank test. **(J)** Change in accuracy (late $-$ early) versus late variance; each symbol represents a PC. No significant trend: $R^2 = 0.01$, $p = 0.26$, $n = 10$. **(K)** Square of PC loadings for each readout unit for the strong decoding (green) and high variance PCs (black). Units on the x axis are ranked by their average accuracy. **(L)** Sum of squares of PC loadings from top 4 and bottom 4 ranked readouts from strong decoding and high variance PCs. Strong decoding PC: $^{**}p = 0.002$; high variance PC: $p = 0.38$ (ns), two sided Wilcoxon signed rank test, $n = 10$.

confirmed that dimensionality changes with learning differ in the presence of adaptive decoders (Fig. S5B,C).

How can neural representations become more compact while dimensionality does not decrease? We linked the variance captured by population level modes with task information across learning. While variance was more distributed across modes after learning (Fig. 2.5H, top), modes capturing the highest population variance were not necessarily the most predictive (Fig. 2.5H, bottom). We quantified the relationship between variance explained and task information for population modes by dividing PCs into two groups for each learning series: the top 50%, which explained approximately 80 to 84% of population variance, and the bottom 50%, which explained 16 to 22% of population variance. Across all learning series, the average classification accuracy on late days was higher in the bottom 50% group compared to the top 50% group (Fig. 2.5I), showing that task information was more often found in low variance modes after learning. Moreover, the changes in a PC's classification accuracy during learning were not correlated with its variance explained (Fig. 2.5J).

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 strongly decoding PC revealed starkly different patterns of unit contributions (Fig. 2.5K). 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. To quantify this effect across all learning series, we compared the sum of squared PC loadings from the top N_c task predictive units to the bottom N_c task predictive units for the strongest decoding PC and highest variance PC in each series (Fig. 2.5L). 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. 2.5L). Thus, the most predictive PCs in neural representations learned with adaptive decoders were not only associated with moderate to low variance but also composed primarily of the most task predictive units. This indicates that task relevant information can be compactly encoded in low variance population modes through the coordinated contributions of highly informative units.

2.5 Discussion

We found that assistive BCI perturbations led to a compaction of neural representations, concentrating task information into fewer neurons and neural modes that capture a relatively small portion of the total population variance while supporting a non decreasing dimensionality. Both experimental data and our model suggest that compaction does not occur without these assistive perturbations. By revealing new impacts of BCI *decoders* on the brain's *encoder*, our work also opens new opportunities to design BCI algorithms that deliberately steer the brain decoder system toward better task representations.

Our findings highlight the complexity of "error" signals in BCI learning. Past BCI studies emphasized links between neural changes and task performance. Neural activity may only reorganize when decoder perturbations introduce task errors and violate existing neural correlations Sadtler et al. [2014]; Oby et al. [2019]; and adapting a decoder to decrease task errors reduces neural changes over time Orsborn et al. [2014]. Unlike most prior studies Jarosiewicz et al. [2008]; Sadtler et al. [2014]; Zhou et al. [2019]; Golub et al. [2018]; Oby et al. [2019], here the decoder changes created intermittent changes to the mapping between neural activity and behavior that either improved task performance or left it unchanged. Adaptive decoders, therefore, manipulate the mapping between *intent* and *control*, even when task performance is not affected. We found that these decoder changes impacted learned neural representations (experiment, Fig. 2.3B G and model, Fig. 2.4E G) and synaptic level changes (model; Fig. 2.4H). This is consistent with our hypothesis that decoder adaptation interacts with error driven skill learning computations, and suggests that learned representations in BCI tasks are influenced by error signals beyond that of task objectives. Motor learning studies suggest that skill acquisition involves both fast (within session) and slow (multi day) adaptation Karni et al. [1998]; Zhou et al. [2019]; Krakauer et al. [2019]; Dayan and Cohen [2011] with different error sensitivities Smith et al. [2006]. Fast and slow adaptation likely involve different neural changes, such as fast adaptation via minimal synaptic changes Perich et al. [2018]; Golub et al. [2018]. One possibility that can be explored in future studies is that the initial performance boost from adaptive decoders could supplant rapid learning mechanisms driven by task errors, while slow mechanisms that shape synaptic changes remain and are impacted by the algorithm. This is

consistent with our observation that compaction emerged over time, rather than explicitly tied to individual decoder update events (Fig. S2I, S3F), and trends of more compact representations *across* learning series (Fig. S3D). Indeed, adaptive decoders may alter slower credit assignment processes using error signals generated through mechanisms like an "internal model" Wolpert and Flanagan [2001]; Golub et al. [2015].

Our results also indicate that credit assignment computations during extended BCI practice operate at scales spanning single neurons as well as coordinated neural populations, and that these computations are impacted by adaptive decoders. Importantly, unit level compaction does not necessarily imply population level compaction (see Supplementary Discussion), making it noteworthy that dominant units preferentially contribute to strongly decoding modes (Fig. 2.5H I). Our offline classification of target identity with neural data (Fig. 2.3, Fig. 2.5) and online assessment of neuronal contributions to control in the model (Fig. 2.4) revealed that adaptive decoders strongly influence which neurons or neural modes participate in the task, leading to exaggerated credit assignment. These task informative neurons appear to form an exclusive coordinated population, since task informative population modes have minimal contributions from other neurons. This observation could be related to our supplemental analysis comparing variance and dimensionality changes during learning between fixed and adaptive decoders. Neural variability and dimensionality have been found to reduce with practice (along with behavioral variability) in both motor learning Dhawale et al. [2017] and BCI learning with fixed decoders Athalye et al. [2017]; Zippi et al. [2021]. We found that this is not the case when the decoder adapts, instead leaving population dimensionality and variability largely unchanged (Fig. S5BC). Together, these results suggest that assistive decoders may shape neuron level learning processes in a way that helps maintain population dimensionality. Our study adds to observations that neural variance and task information have a nuanced relationship Russo et al. [2018]; Kaufman et al. [2016], and that subtle phenomena, which may depend on different credit assignment mechanisms, can be missed by linear based variance analyses Dyer and Kording [2023].

Our study reveals new challenges and opportunities in designing long term BCIs. Given measurement instabilities over a BCI's lifetime, adaptive decoding is unavoidable and widely used in

recent high performance human BCIs (e.g., Silversmith et al. [2021]; **?**]; **?**]; **?**). Our results highlight that long term learning mechanisms will likely occur alongside decoder changes, making long term BCI systems inherently *co adaptive* Madduri et al. [2023]; Shenoy and Carmena [2014]. Moreover, our results demonstrate the inherent entangling of *decoding* and *encoding* in co adaptive systems. The adaptive decoding algorithms used in our experiments Orsborn et al. [2012] closely related to widely used adaptive algorithms in research and clinical BCIs Taylor et al. [2002]; Gilja et al. [2012, 2015]; Jarosiewicz et al. [2015]; Collinger et al. [2013]; Silversmith et al. [2021]; Benabid et al. [2019] produced compaction of neural representations. How the co adaptation process we observe in our study may relate to co adaptation in BCIs used for daily life is a critical outstanding question. Our results suggest that the brain decoder system ultimately relies on a smaller set of neural features for this task, which will impact BCI performance. For instance, compact representations may improve computational efficiency, and could lead to task encodings that are more easily separated from other behaviors to reduce interference from other task demands Orsborn et al. [2014]. Compact encoder decoder systems may also protect against unreliable units (Fig. 2.4G). Yet, the dominance of a few features can prevent more subtle information from influencing learning Pezeshki et al. [2021] and compact task encodings may increase sensitivity to neural recording loss. Our model highlights the opportunity to balance these potential trade offs by properly fine tuning the decoder adaptation algorithm. Such models will likely be valuable for more principled designs of co adaptive BCIs Madduri et al. [2023]. Understanding the mechanisms by which assistive decoders shape task encoding and their functional implications will ultimately make it possible to design BCIs that provide high performance for someone's lifetime.

Table 2.1: Parameters of the model. For the arm model, we combined data from Ref. Cheng and Scott [2000] and from the morphology of the subjects in Ref. Orsborn et al. [2014].

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

Chapter 3

Context influences learning in redundant 3D visuomotor perturbations

3.1 Introduction

People routinely perform motor actions that involve more degrees of freedom than are strictly necessary. For instance, reaching for a cup can be achieved through countless arm trajectories and with many final arm postures. This phenomenon, termed redundancy, is pervasive across motor behaviors, from simple reaches to complex tool use Scholz and Schöner [1999]; Todorov and Jordan [2002]. Whereas our bodies can generate richly variable movements through many degrees of freedom, the sensory and task feedback—particularly visual feedback—used to guide movement is often much lower-dimensional. For instance, controlling a 2D cursor on a screen with a freely moving arm introduces a mismatch between the richness of the motor system's output (action space) and the low-dimensional sensory feedback (task outcome) available to guide it Law et al. [2014]; So et al. [2012]. Learning motor skills in such redundant environments requires identifying which actions are causally linked to task outcomes—the *task-relevant* space of actions. The mechanisms by which we perform this credit assignment computation to learn task relevance spaces in redundant motor skills remain poorly understood.

Classic motor learning task paradigms primarily restrict redundancy by using controlled tasks

and environments that tightly align movement and feedback spaces Shadmehr and Mussa-Ivaldi [1994]; Krakauer [2009]; Krakauer et al. [2019]. For instance, movements in a 2-dimensional manipulandum to control a 2-dimensional task ensure that all movements generated by the subject produce task-relevant feedback, allowing errors to be unambiguously mapped to specific actions. These studies have demonstrated error-based updating of internal models Wolpert and Flanagan [2001]; Shadmehr et al. [2010]. While instrumental in characterizing the principles of error-based learning, these paradigms cannot readily probe computations involved in real-world motor skills, such as discovering task-relevant actions within a larger null space.

Paradigms that maintain some level of redundancy have shed light on important motor learning principles by revealing how variability is structured relative to task goals. Redundancy in a task introduces the additional structure of sub-spaces of movement that do and don't influence task performance (task-relevant and task-irrelevant components of movement, respectively) Müller and Sternad [2004]; Scholz and Schöner [1999]. Learning in a redundant space requires identifying and generating actions specifically within the task-relevant space. This process is thought to depend on exploration across the action space—generating movement variability along multiple dimensions, observing the resulting feedback, and using this feedback to guide learning Dhawale et al. [2017]; Therrien et al. [2016]. This exploration allows the brain to gradually infer the structure of the task and identify task-relevant subspaces Heald et al. [2021]; Sadtler et al. [2014]; Golub et al. [2018]. Once these subspaces are learned, theories such as optimal feedback control and the uncontrolled manifold framework suggest that the brain does not minimize variability uniformly across all dimensions. Instead, it selectively reduces variability along task-relevant dimensions while maintaining or even amplifying variability along task-irrelevant ones Todorov and Jordan [2002]; Müller and Sternad [2004]. This selective regulation allows the learner to exploit redundancy efficiently, using variability not as noise but as an active mechanism to explore, discover, and refine task-relevant control strategies. Similar principles have been observed in songbirds, where vocal variability is thought to support trial-and-error learning during song acquisition and is actively regulated depending on the learning stage Fee and Goldberg [2011]; Dhawale et al. [2017].

Recent work has begun to address the role of redundancy in learning using interfaces that ex-

plicitly incorporate input redundancy Law et al. [2014]; So et al. [2012]. For example, Kobayashi and Nozaki (2024) demonstrated that the baseline structure of redundancy constrains how movements are controlled and adapted Kobayashi and Nozaki [2024]. These studies introduced task paradigms that differ fundamentally from classic non-redundant motor adaptation tasks, offering new opportunities to probe how the brain learns in high-dimensional spaces. However, we still lack paradigms that manipulate task relevance directly and dynamically. Existing studies have largely focused on static redundancy or fixed mappings, making it unclear whether learners reuse existing sensorimotor models ("structure learning") when task relevance changes or whether they must construct new ones de novo Heald et al. [2021]; Yang et al. [2021]. Furthermore, the role of prior experience in shaping how the motor system navigates these shifting relevance landscapes remains unknown.

We designed new experiments to study how sensorimotor perturbations that change the task-relevant dimensions influence learning. We used a sensorimotor interface paradigm that allowed us to manipulate task relevance by introducing redundancy between the space of actions and task feedback. Specifically, subjects used 3D hand movements to control 2D cursor movements. Monkeys first trained on a "baseline plane" mapping, where horizontal and vertical hand motions controlled the cursor, rendering front-back movements task-irrelevant. We then introduced perturbations that preserved or altered the task-relevant planes. Relevance Preserving (RP) perturbations kept the same task-relevant space while introducing visuomotor shifts in the mapping, while Relevance Altering (RA) perturbations redefined a new task-relevant plane, requiring movement in previously irrelevant dimensions. We hypothesized that RP perturbations would be learned via adaptation of the existing planar controller, whereas RA perturbations would require exploration to form a new controller, resulting in distinct behavioral signatures Yang et al. [2021]; Telgen et al. [2014].

With relevance preserving (RP) perturbations, we observed rapid, within-day learning marked by aftereffects and scaling with perturbation magnitude, hallmarks of error-based adaptation Shadmehr and Mussa-Ivaldi [1994]; Krakauer [2009]. In contrast, relevance altering (RA) perturbations showed slower initial learning, little cross-context generalization, and behavior suggestive of de novo learning of a new controller Telgen et al. [2014]; Yang et al. [2021]. Notably, once monkeys gained experience with RA mappings, subsequent RA learning was faster, consistent with "struc-

ture learning" where prior experience facilitated exploration of the redundant workspace. Crucially, learning was modulated by the useful feedback available: the ratio of cursor-to-hand motion predicted adaptation, and monkeys increased movement variance when task relevance was altered, pointing to a shift from error correction to variance-based or reward-based learning mechanisms Dhawale et al. [2017]; Therrien et al. [2016]. These findings demonstrate that diverse learning dynamics emerge from a single paradigm depending on how task relevance is altered, suggesting distinct neural mechanisms for adaptation versus structure learning.

3.2 Materials and Methods

3.2.1 Animal selection and surgical procedures

All animal procedures were approved by the IACUC at the University of Washington. Two male rhesus macaques (macaca mulatta) were selected as experimental subjects. Monkey 1 (Beignet, 9 years old, 11.3 kg) and Monkey 2 (Affogato, 10 years old, 10.6 kg) were pair-housed at the Washington National Primate Research Center. The animal facility maintained a 12h light/12h dark cycle. Both monkeys had scheduled access to water when on the study. On each day of data collection and for several days beforehand, animals would receive fluids during AM training for successful reaches (average 200 mL per day) followed by PM access to 300-500 mL water. Training time and PM water time were kept consistent across days. Prior to data collection, animals underwent surgery to implant a titanium headpost surgery and recording chambers. Chambers were designed for semi-chronic access to the brain, as described in (Ouchi et al., 2025). Electrode arrays were implanted over left motor cortices during a portion of the experiments. Neural data recorded from these arrays were not used in the present study.

3.2.2 Experimental Design

Monkeys were trained to perform arm movements to control the movements of a cursor displayed on a two-dimensional display (Fig. 3.1A). Cursor control and visual stimuli were implemented using a custom version of BMI3D (`http://github.com/aolabNeuro/brain-python-interface`). The dis-

play (16:9 LCD, 21.5 cm vertical size) was positioned 28–30 cm from the animal's eyes, and monkeys moved their right arm freely just below the screen within a $20 \times 20 \times 20$ cm motion capture volume. The monkeys' left arm and head movements were restrained and visibility of their right hand was limited by a polycarbonate sheet placed horizontally between the display and the movement volume. The position of the cursor was determined by a linear projection of the monkey's right hand position (see Hand to Cursor mapping). We used a marker-based optical tracking system (NaturalPoint, Inc., Corvallis, OR) to capture the three-dimensional position of the animal's hand in real time at a rate of 240 Hz. Monkeys wore a lightweight glove equipped with infrared-reflective markers to enable motion tracking. All data streams, including hand position, cursor position, and task events, were temporally aligned using synchronized clocks via a data-acquisition system (eCube, White Matter Research, Montreal CA).

3.2.3 Delayed center out task

Monkeys performed a delayed center-out reaching task. Each trial was self-initiated by holding the cursor inside a central target for 200 ms. Then, one of eight peripheral targets, equally spaced at 45° intervals around the center (0°, 45°, 90°, 135°, 180°, 225°, 270°, and 315° relative to the horizontal axis), appeared 6.5 cm from the center of the screen. The cursor was required to remain inside the center target during a variable delay of 100 - 600 ms until the center target disappeared (the "go cue"). After the go cue, reaches to the peripheral target did not have any experimentally imposed time constraints, and rewards were delivered for successful reaches only after holding within the peripheral target for 200 ms. After a trial ends with or without reward, animals returned to the center to initiate the next trial. Trial initializations were self-paced.

3.2.4 Hand to Cursor mapping

Our experimental paradigm used the spatial distinctions and redundancy between 3D hand movement and 2D task (cursor) space. This setup allowed us to define and perturb the mapping (M) from

3D hand position space (H) to 2D cursor position space (C) with a linear transformation:

$$C = MH \tag{3.1}$$

where: H is the hand position vector denoting X, Y and Z movement axis, C is the cursor position vector denoting X and Y axis, and M is a linear mapping matrix that is specific to each experiment. Because there are more input dimensions than output dimensions, the task introduces redundancy – multiple hand movement trajectories produce the same cursor movement trajectory. Changes to the hand-to-cursor mappings may or may not change the task-relevant plane within the full 3D movement space.

Monkeys first learned a mapping which we define as baseline, $M_{baseline}$. This mapping defined cursor space to be a 1:1 linear projection of the horizontal (X) and vertical (Y) components of hand movement:

$$M_{baseline} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{3.2}$$

This baseline mapping, thus, defines the XY hand movement plane as the task relevant plane. Movements along the depth (Z) axis (i.e., the third column of H) have no contribution to cursor movement and were thus task-irrelevant.

3.2.5 Relevance preserving perturbations

RP perturbations were applied to the baseline mapping to define RP mappings M_{RP}. These altered the baseline mapping in a way that kept the task-relevant plane the same as that of the baseline mapping, but altered the hand-cursor mapping within that plane (Fig. 3.2A). In the space of possible transformations, we specifically studied relevance preserving perturbations that applied a fixed rotation angle θ about the hand Z axis (e.g., $\pm 20°$, $\pm 45°$, $\pm 90°$). This corresponds to applying a two-dimensional rotation matrix R_z to the baseline mapping $M_{baseline}$:

$$M_{RP} = R_z(\theta) M_{baseline} \tag{3.3}$$

We tested rotations ranging from $20°$ to $90°$, and use rotation angle as a measure of perturbation size since it captures the deviation – within the original task plane – from the baseline mapping. Importantly, because the perturbed task plane remained fully aligned with the learned baseline plane, maximal feedback was available when monkeys continued to move within this familiar subspace.

3.2.6 Perturbations altering task relevant plane

The redundancy in our task provides a second category of perturbations that alter the hand-cursor relationship while also changing the task-relevant movement space within the 3D hand movement space (Fig. 3.3A). RA mappings were defined by multiplying the baseline mapping by a 3D rotational matrix R_{3D}:

$$M_{RA} = M_{baseline} R_{3D} \tag{3.4}$$

By rotating the baseline mapping itself in 3D hand space, these perturbations changed which hand axes projected into the 2D task plane. We hypothesized that these perturbations would require exploration beyond the movements monkeys made in the baseline mapping. We studied three RA mappings with varying magnitudes ($20°$, $45°$, and $90°$). For $90°$ RA perturbations, the magnitude defined the degrees of rotation applied around the X-axis. For $20°$ and $45°$ RA perturbations, the magnitude defined a compound rotation where the perturbation was first applied about the X-axis and subsequently about the Y-axis.

3.2.7 Training paradigm

After training on the baseline mapping, we studied how monkeys learned different perturbed mappings. These mappings were randomly selected at the start of the experiment from a predefined set of rotations, and each mapping was held fixed for the duration of a learning series across consecutive days. We continued a learning series until success percentage hit a performance plateau. In some cases of RP perturbations, learning series continued or reoccurred after a break based on other constraints.

Learning series were composed of daily sessions with a consistent block structure where the

animal experienced the baseline mapping, a perturbed mapping, and then a return to the baseline (A-B-A) (Fig. 3.2B). Within a learning series, the B block mapping was the same each day. B block included least 20 and up to 500 trials, depending on the animal's engagement and tolerance to the perturbation. The session concluded with an A' block to assess washout and return to baseline behavior. During the B block, animals often exhibited signs of frustration and reduced motivation. The number of trials in the perturbation block was chosen each day by the experimenter, balancing the desire to maximize exposure to the perturbation with the need to collect sufficient A' block data for assessing recovery. Within each block, we define early and late periods as the first and last 20 trials. Consistently, Fig. 3.2B,E,H, Fig. 3.3B,E,H and all statistics use these definitions. Table 1 provides a summary of the perturbations experienced by each monkey in chronological order.

3.2.8 Behavioral Data Analysis

Hand movement data and cursor positions were low-pass filtered with a 15 Hz cutoff using a zero-phase Butterworth filter and downsampled to 100 Hz. These preprocessed data were used in all subsequent analyses of movement kinematics and task-related behavior. We quantified behavioral performance using mean success percentage and reach time (Figure 1C). Success percentage was calculated as the proportion of successfully-initiated trials that resulted in a reward. We note that this removes from consideration trials where the monkeys failed to successfully hold at the center target. We assume these reflect errors that are less relevant to learning the hand-cursor mapping (e.g., failing to withhold a reach for the delay period). Reach time was defined as the duration (seconds) between the cursor exiting the center target and entering the peripheral target. For Figure 1C, we show behavioral performance across 10 consecutive baseline days randomly selected before perturbations were introduced.

Success percentage related metrics

We estimated performance improvements using success percentage - the number of successful trials within a 20-trial rolling window (Fig. 3.2B-D, 3.3B-D, 4A-B, 5B). This was implemented in python by convolving a binary success vector with based on a normalized convolutional kernel of width 20.

To preserve the original array length, the algorithm extended the first and last valid values at the array boundaries before convolution. We compared mean success percent during baseline mapping session A to early and late phases of the perturbation and washout block for statistical comparisons across series.

Time to learn

We compared the rate of learning across different series by examining the number of trials required to consistently achieve a certain level of task performance, which we define as time to learn (Fig. 3.2D, 3.3D, 5A). Time to learn was calculated using the success rate, estimated via the rolling window average, aggregating trials across perturbation blocks from all days in a series. We then computed the trial at which the success rate remained at or above 80% for 21 consecutive trials.

Within session vs Offline gain

Each day, we computed a rolling success percentage that is bound between 0 to 100 across a rolling window of 21 trials. Within session gain was computed as the change from the initial to the peak success percent within the perturbation block (Fig. 4D, F). Offline gain was computed as the difference between performance at the start of a session and the mean performance of the preceding session (Fig. 4E, F, H, I).

Cursor trajectory analyses

Cursor trajectories during the reach were analyzed, focusing on the period from the go cue to target acquisition. In figs. 3.2C and 3.3B, the early and late phases of each block were defined as the first 20 and last 20 trials, respectively. This definition was applied consistently across both perturbation and washout blocks to evaluate aftereffects and learning.

Cursor error

To quantify within-session learning and recovery, we computed errors in the cursor trajectories during the reach (time between the go cue and reaching the peripheral target). Cursor error was

defined as the deviation from a straight line connecting the start position at the center to the target (i.e., the direction orthogonal to the movement direction). For each trial, the average deviation across all time points during the reach was used as the movement error. The temporal evolution of movement error was estimated using a rolling average window. The mean and standard deviation were calculated with an overlapping window size of 21 and a minimum of 1 trial to smooth the movement error (Fig. 3.2D).

Quantifying aftereffects

We analyzed patterns of cursor movement generalization between the perturbed and baseline mapping by examining cursor errors during the A' (washout) block. Movement error was computed for each trial as described above. To enable cross-series comparisons (Fig. 3.2E & 3.3E), we used the mean movement error over the first 60 trials of the A' block as a measure of washout.

Hand trajectory analyses

To understand the complexity of hand movements, we employed Principal Component Analysis (PCA) on 3D hand reach segments. We focused on the hand movement variance during reaches by re-centering each reach segment such that the start at a common origin. This removed trial-to-trial variance and drift in the spatial location of the hand, which could and sometimes did drift in task-irrelevant dimensions. We performed PCA on each day's hand kinematics separately and computed the variance explained by each principal component. The distribution of explained variance across components was computed for ten consecutive baseline days corresponding to the same days shown in Fig. 1C.

Task-relevant and task-irrelevant variability analyses

We analyzed how the overall 3D hand movements related to the task space by calculating variance in both the task-relevant and irrelevant spaces. In our task, task-relevant variance corresponds to calculating the fraction of overall movement variance that lies in the task plane defined by the mapping M (e.g., the XY plane for the baseline and all relevance preserving mappings). For each trial,

task relevant variance was computed as the sum of the variances of hand position along the X and Y axes during the reach segment (from go cue to reward). The proportion of task relevant variance shown in Figures 1F, 3.2I and 3.3I was defined as the ratio of this task plane variance to the total variance across all 3 axes:

$$P_{task} = \frac{\sigma^2_{task}}{\sigma^2_{total}} \tag{3.5}$$

For relevance altering (RA) perturbations, the mapping rotated the task relevant plane out of the original XY plane. To compute task relevant variance in these conditions, hand trajectories were projected onto the rotated plane using the corresponding perturbation specific mapping matrix. Variance was then computed along the two projected axes that contributed to the cursor's X and Y movements. These values were used to compute the proportion of task relevant variance in the same manner as above, replacing the original X and Y axes with the rotated task relevant dimensions. In our task, task-irrelevant variance corresponds to the axis orthogonal to the task-relevant plane (e.g., the Z-axis for the baseline and all relevance preserving mappings). For each trial we computed the variance of hand position along the Z axis in cm. For visualization we plotted a rolling mean of this variance across a window of 21 trials with a minimum of one data point required for estimating rolling mean. We quantified the timescale of variance changes by fitting the trial-by-trial variance data to an exponential decay function:

$$V(t) = Ae^{-t/\tau} + B \tag{3.6}$$

where A is the initial amplitude of the decay, τ is the decay constant, and B is the steady-state null axis variance. Model fits were obtained with the curve fit routine in scipy.optimize. The decay constant was converted to a half life $t_{1/2}$, which provided the number of trials required for variance to fall halfway from its initial level toward its asymptote. Half life values were estimated using all training sessions in which a hand to cursor mapping was used. This included training sessions that involved random target reaching as well as center out movements.

Feedback density

Feedback density was the ratio of cursor to hand variance per trial. Cursor variance was computed in 2D, hand variance in 3D, over the movement window for each reach. For across day analyses, mean feedback density per session was used.

Exploration Volume

The intersection between the baseline and perturbed planes was computed by sampling uniformly across each plane within ± 10 cm bounds, projecting these into 3D space, forming a combined point cloud and computing its convex hull volume using SciPy. Sampling resolution was set to 2 mm grid spacing. A convex hull was computed from the merged set, and its volume served as the exploration volume. Matrix operations and point generation used NumPy, and convex hull estimation used SciPy. We normalized PC3 variance RA mapping to exploration volume for fig. 5D.

3.2.9 Statistical analysis

Statistical significance was defined as $p < 0.05$. All statistical tests are reported in the Results with sample sizes and test types indicated for each comparison. Reported p values were mapped to significance levels using the following convention: $p < 0.001$ (***), $p < 0.01$ (**), $p < 0.05$ (*), and $p > 0.5$ (n.s). Repeated measures, such as success percentage, movement error, and variance were compared across behavioral blocks using paired t-tests. The direction of each test is stated alongside reported significance values in the Results. Linear regression was used to evaluate relationships between perturbation magnitude and learning related measures including time to learn, washout magnitude and null axis variance. One sample t-tests were used to assess whether within day learning, offline gains and performance changes between consecutive days were significantly above zero. All analyses were implemented in Python using the scipy.stats library. When quantifying across series effects, data from both monkeys were pooled and individual monkeys were treated as repeated instances of the same experimental design, which is standard practice in non-human primate studies.

3.3 Results

3.3.1 Learned a planar solution to a baseline 3D hand to 2D cursor mapping

We first examined how monkeys learned to control movements in our redundant task, where their hand movements in three-dimensional space controlled two-dimensional cursor movements (Fig. 3.1A). Both monkeys first learned a baseline mapping ($M_{baseline}$), which translated horizontal and vertical hand position to cursor movements, while front–back hand motion had no effect on cursor position (Fig. 3.1B). After training, animals attained high task proficiency, evidenced by high success rates (Fig 3.1C, top) and straight cursor trajectories (Fig 3.1C, bottom) across 10 consecutive days (randomly selected from the training period before perturbations were introduced).

The task allowed movement in three dimensions, but we found that the monkeys rapidly identified the lower-dimensional task-relevant space and restricted their movements primarily to a plane. Visual inspection of three-dimensional hand trajectories showed that they lay almost entirely within a 2D subspace that was well-aligned with the task-relevant plane (Fig. 3.1D). Principal component analysis (PCA) of 3D reaches showed that the first two components captured over 95% of the movement variance, and the third component contained very little variance (Fig. 3.1E). Across training days, the variance within the plane defined by the top two PCs closely matched that of the baseline task plane (XY plane), with points clustering near the unity line (Fig. 3.1F). Each animal performed more than 10,000 trials (XX trials monkey B, YY trials monkey A) before perturbations were introduced, during which they rapidly reduced task-irrelevant movements. Their task-irrelevant variance had a half-life of only 122.4 trials for Monkey A and 693 trials for Monkey B, and remained low thereafter (Fig. 3.1G). These results show that even in a redundant control space, the monkeys quickly learned the planar structure of the movement space that controls the cursor, accurately identified the task-relevant plane, and rapidly reduced task irrelevant movements.

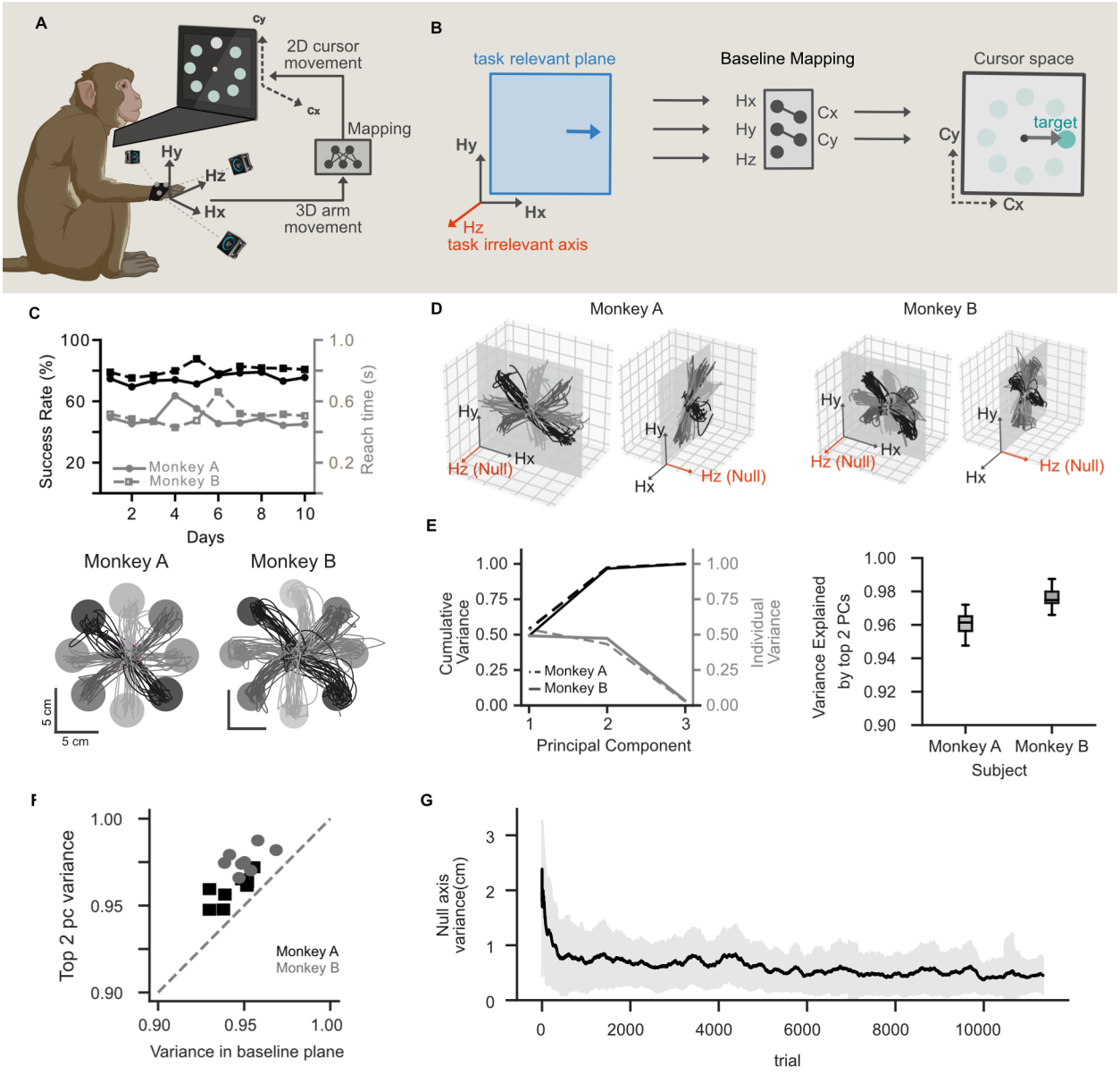

Figure 3.1: Experimental design, Monkeys learned a baseline mapping Monkeys learned a planar solution to baseline 3Dhand-to2D-cursor mapping (A) Experiment schematic: Monkey moves his right arm in a 3D space. 3D hand movements are mapped to 2D cursor movements using an experimenter defined mapping. (B) Schematic showing how the same hand movement projects onto different task-relevant planes. In the blue plane, feedback fully reflects the movement, while in the orange plane, the projection is altered, leading to partial or ambiguous feedback. (C) (Top) Success rate (black) and reach time (orange) across days for Monkeys A (circle) and B (square) (Bottom) Example cursor trajectories color coded by target direction from monkey A right and monkey B left. Scale bars denote 5 cm. Data from 10 consecutive baseline days randomly selected before perturbations were introduced. (D) Example 3D hand trajectories corresponding to cursor trajectories in D from Monkey A (right) and monkey B (left). In all views, task relevant plane (XY plane) is shown in gray. (E) (left) Cumulative variance (blue) and variance explained by each PC (black) for hand movements. (right) Cumulative variance explained by top 2 PCs shown for two monkeys. Mean and error bars (standard deviation) estimated across the 10 days shown in C. (F) Percentage variance captured in baseline plane (see methods) vs variance explained by top 2 PCs. Each datapoint shows data from a single day. Gray dotted line indicates unity line. (G) Task irrelevant variance across trials. Solid line indicates rolling mean and shaded region indicates rolling standard deviation computed from a rolling window of 21 trials.

3.3.2 Relevance-preserving perturbations were learned within a day and showed generalization

After training in the baseline mapping, we introduced relevance-preserving (RP) visuomotor perturbations, which rotated the hand-to-cursor mapping but maintained the same task-relevant plane as the baseline mapping (see Methods; Fig. 3.2A). Introducing RP perturbations transiently disrupted task performance, causing an initial drop in success rate that rapidly recovered within the session (Fig. 3.2B, top, Fig 3.2C, early B success on day 1 < baseline, p.= 0.007 (**); late B success on day 1 < baseline, p = 0.19 (n.s), paired t-test, N = 15 RP learning series across two monkeys). When the original baseline mapping was reinstated (A' block), we observed a transient performance drop, which again resolved within the session (Fig 3.2B top, Fig 3.2C, early A' success on day 1 < baseline, p.= 0.00645 (**); late A' success on day 1 < baseline, p = 0.186 (n.s), paired t-test, N = 15 RP learning series across two monkeys).

Monkeys were exposed to the same RP perturbation across days within a learning series, allowing us to further examine learning dynamics across days. Task performance was less disrupted by the introduction of RP perturbations across days (Fig 3.2B bottom, early B success percent on day 2 – day 1 > 0, p = 0.0023 (**), one sample t-test, N = 9 RP learning series with multiple days), consistent with learning both within and across days. We quantified the speed of learning by calculating the number of trials required to consistently reach 80% task success (time to learn; see Methods). Time to learn increased with perturbation magnitude (Fig 3.2D, R^2 = 0.408, p = 0.00778, linear regression), consistent with observations in many visuomotor perturbation experiments without input redundancy (citations).

We next analysed the structure of cursor trajectories during learning. Introducing the RP perturbations resulted in notably curved trajectories early in learning, which gradually became straighter over the course of each session (Fig. 3.2E). Returning to baseline (A' block) resulted in trajectories that curved in the direction opposite of the perturbation, consistent with aftereffects commonly observed in visuomotor adaptation (citations). We quantified this effect by calculating the deviation relative to a straight-line trajectory (cursor error, see Methods: Quantifying aftereffects). Cursor error increased when perturbations were introduced (Day 1: early cursor error in B > mean movement

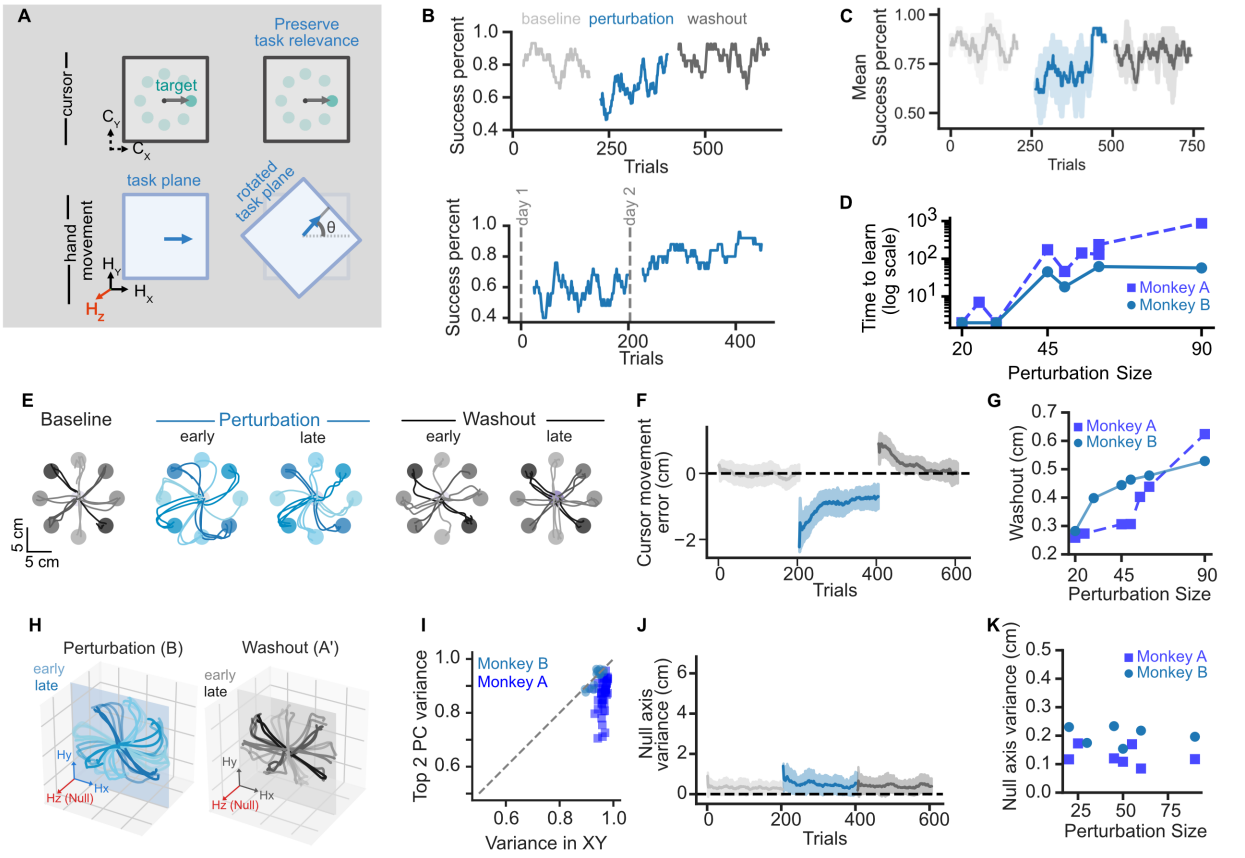

Figure 3.2: Overview of perturbations Figure 2 Relevance preserving perturbations learned within day and show generalization (A) Perturbation schematic: relevance preserving perturbations used the same task relevant plane as baseline mapping but with a visuomotor rotation (B) (Top) Success percent vs trials during baseline (A, grey), perturbation (B, blue) and washout (A', dark grey) blocks on day 1 for example 45 degree perturbation from monkey B. (Bottom) Success percent during perturbation block for two consecutive days. (C) Mean success percent across all perturbation sizes and all days, example from monkey B. (D) Time to learn (log scale) vs. perturbations size for monkey A (light blue) and monkey B (dark blue) (E) Cursor trajectories during baseline, early and late perturbation, early and late washout blocks for example 45 degree perturbation from monkey B. Each color represents a target direction. (F) Cursor movement error (cm) across trials during ABA' blocks similar to B. (G) Washout vs perturbation size summary for monkey A (light blue) and monkey B (dark blue) (H) Hand movement trajectories in 3D during perturbation and washout blocks for monkey B. Each color represents a target direction. Lighter shades denote early and darker shade denotes late trials. (I) Null-axis variance (cm) across trials for the same blocks as in (B). (J) Null axis variance (cm) vs perturbation size summary for monkey A (light blue) and monkey B (dark blue). (K) Variance in XY plane vs variance in top 2 PCs during perturbation block for monkey A (light blue) and monkey B (dark blue). For C, F and I, Solid line shows the rolling mean and shaded area indicates rolling standard deviation.

error in A, p = 0. 000095 (***); late movement error in B > mean movement error in A, p = 0.00039 (***), paired t-test, N = 15 RP learning series). While trajectories became straighter across days (early cursor error on first day > last day, p = 0.0159 (*), paired t-test, N = 9 RP learning series with multiple training days across 2 monkeys), cursor error during perturbations did not fully return to baseline levels (Last day: early cursor error in B > mean cursor error in A, p = 0.00031 (***) ; late cursor error in B > mean cursor error in A, p = 0.00539 (**) , paired t-test, N = 9 RP learning series). Moreover, cursor error reversed sign upon returning to baseline mapping (A' block), consistent with aftereffect generalization patterns observed in visuomotor adaptation (citations) (example series shown in Fig 3.2F). Cursor trajectories in A' started with a higher movement error, which gradually reduced (early movement error in A' > mean movement error in A, first day: p = 0. 000036 (***), last day: p = 0.00007(***); late movement error in A' > mean movement error in A, first day: p = 0.031 (*), last day: p = 0.148 (n.s), N = 9 RP learning series). The magnitude of these aftereffects, assessed by average movement error in the first 60 trials of washout, was correlated with the size of the preceding RP perturbation (Fig. 3.2G, R^2 = 0.8699, p = 0.0001, linear regression).

While RP perturbations changed cursor and hand movements within the task-relevant plane, we did not observe significant changes in the monkeys' hand movements in the task-irrelevant dimension (Fig. 3.2H, Day 1: early null variance in B vs. mean null variance in A, p = 0. 356 (n.s); late null variance in B vs. mean null variance in A, p = 0.643 (n.s), paired t-test two sided, N = 15 RP learning series). Qualitatively, hand trajectories remained confined to the task-relevant plane defied by the baseline mapping (XY plane) for all mappings (Fig. 3.2H). Indeed, most of the total hand movement variance remained in the task-relevant plane for the baseline and RP mappings (XY plane), and the total variance captured by the dominant PC plane in B block varied between 0.7 to 0.95 (Fig. 3.2I). Correspondingly, variability along the null axis (Z) of the task relevant plane remained low across all blocks (mean null axis variance in A, B and A' blocks on day 1, F = 3.19, p = 0.051, one-way ANOVA, N = 15 RP learning series) (Fig. 3.2J). Moreover, exposure to RP perturbations did not alter task-irrelevant movements upon returning to the baseline block (A') (Day 1: early null variance in A' vs. mean null variance in A, p = 0. 989 (n.s); late null variance in A' vs. mean null variance in A, p = 0.986 (n.s), paired t-test two sided, N = 15 RP learning series). Variability along the null axis similarly did

not show any relationship with perturbation size (Fig. 3.2K, slope = 0, R = -0.145, p = 0.637 (n.s), linear regression), confirming that aftereffects were restricted solely to the task-relevant plane.

3.3.3 Relevance-altering perturbations required multi-day learning and showed no generalization

We next examined learning in relevance-altering (RA) perturbations, which rotated the task-relevant plane into a new orientation in 3D hand space and altered which hand movements produced task-relevant feedback (Fig. 3.3A). Introducing RA perturbations substantially reduced the proportion of successful trials (Fig. 3.3B, top). Unlike in RP perturbations, monkeys were generally unable to recover baseline-level performance within a single session (Fig 3.3B top, early B success on day 1 < baseline mean success, p.= 0.0014 (**); late B success on day 1 < baseline mean success, p = 0.0129 (*), paired t-test, N = 7 RA learning series across two monkeys). However, across multiple days, performance gradually improved (Fig. 3.3B, bottom, early B success on day 1 < early B success on day 2, p = 0.0037 (**); early B success on day 2 < early B success on last day, p = 0.0011 (**), early B success on last day < baseline mean success, p = 0.0931 (n.s), N = 6 RA learning series with consecutive days), suggesting slow acquisition of a new control strategy. Success percent quickly returned to near-baseline levels when the baseline plane was reintroduced during A' block on the first day following perturbation where little to no learning occurred (Fig. 3.3C, early A' success < baseline, p = 0.262 (n.s); late A' success < baseline, p = 0.0076 (**), paired t-test, N= 7 RA learning series across two monkeys) as well as last day when RA mapping are learned (early B success on last day < baseline mean success, p = 0.0931 (n.s), early A' success < baseline mean success, p = 0.105, N = 7 RA learning series). The time to learn relevance altering perturbations did not show any consistent relationship with the perturbation size (Fig. 3.3D, R^2 = 0.1, slope = 1.96, p = 0.442 (n.s), linear regression), a notable contrast to relevance-preserving perturbations (Fig. 3.2D).

Cursor trajectories showed neither typical curvatures in B block nor revealed aftereffects in the A' block(Fig 3.3E). Cursor movement error increased significantly and was highly variable during the RA perturbation block B (early movement error in B vs. mean movement error in A, first day: p = 0. 00498 (**), last day: p = 0.0453 (*); late early movement error in B vs. mean movement error

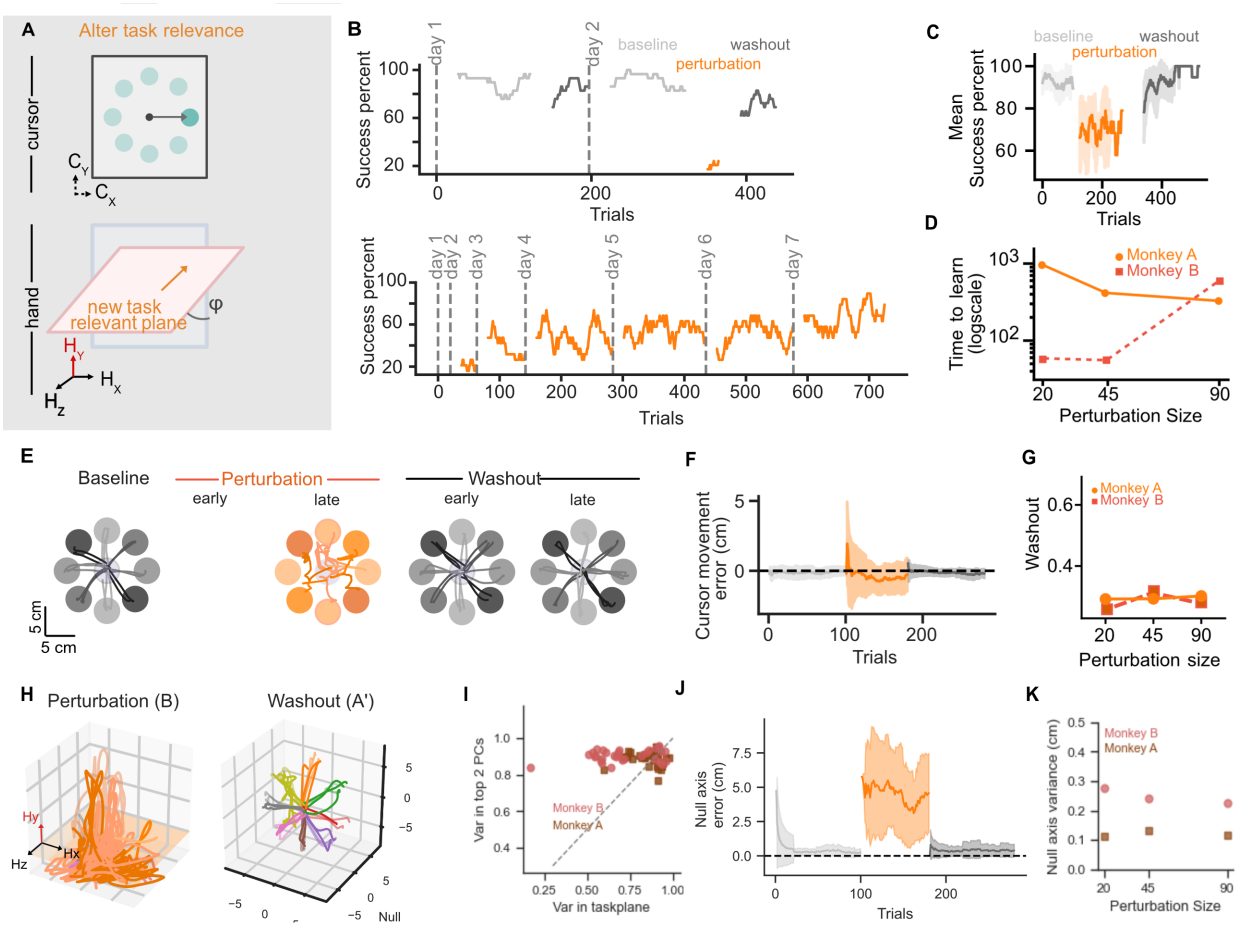

Figure 3.3: Relevance altering perturbations required multi-day learning and showed no generalization (A) Perturbation schematic. Task-relevance altering perturbations introduced a new task-relevant plane rotated out of the baseline plane by an angle ϕ, changing the mapping between hand and cursor motion. (B) (Top) Success percent across trials during baseline (A, gray), perturbation (B, orange), and washout (A', dark gray) blocks on day 1 for an example 90° perturbation (Monkey B). (Bottom) Success percent during B block across multiple days. (C) Mean success percent across all perturbation sizes and all days for Monkey B. (D) Time to learn (log scale) vs. perturbation size for Monkey A (light orange) and Monkey B (dark orange). (E) Example cursor trajectories during baseline, early and late perturbation, and early and late washout blocks for an example 90° perturbation (Monkey B) on day 2. Each color represents a target direction. (F) Cursor movement error (cm) across trials during the ABA' blocks, similar to (B). (G) Washout magnitude vs. perturbation size for Monkey A (light orange) and Monkey B (dark orange). (H) 3D hand trajectories during perturbation (B) and washout (A') blocks for Monkey B. Each color represents a target direction. (I) Variance captured by the top two principal components (PCs) vs. variance in the task plane for Monkey A (light orange) and Monkey B (dark orange). (J) Null-axis variance (cm) across trials for the same ABA' blocks as in (B). (K) Null-axis variance (cm) vs. perturbation size summary for Monkey A (light orange) and Monkey B (dark orange). For panels C, F, and J, the solid line shows the rolling mean, and the shaded area indicates the rolling standard deviation.

in A, first day: p = 0.00762 (**) last day: p = 0.147(n.s), paired t-test two sided, N = 7 RA learning series). No consistent aftereffects were observed during the A' block, as cursor trajectories returned to baseline orientation without showing curvature or directional bias (Fig 3.3F, early movement error in A' vs. mean movement error in A, first day: p = 0. 327 (n.s), last day: p = 0.1 (n.s); late early movement error in A' vs. mean movement error in A, first day: p = 0.432 (n.s), last day: p = 0.567 (n.s) , paired t-test two sided, N = 7 RA learning series). The magnitude of aftereffects during the A' block, assessed by average movement error in the first 60 trials as washout, did not show any relationship with perturbation size (Fig. 3.3G, R = -0.286, p = 0.281 (n.s), linear regression). Together, these data show distinct generalization patterns between RA and RP perturbations, and suggest that RA perturbations lead to formation of a new sensorimotor map rather than modifying the existing baseline mapping.

Hand kinematics during B showed exploration in 3D subspace (Fig. 3.3H) and return to the original 2D baseline plane in A', indicating a re-engagement of the previously learned controller (Fig. 3.3H). Further, to evaluate whether monkeys had learned the correct task-relevant plane, we compared the variance of cursor movements captured in the new task plane versus the variance explained by the first two principal components of hand kinematics. This analysis revealed that, over time, animals aligned their control strategies with the appropriate rotated task plane, suggesting that learning involved identifying and engaging with the correct subspace in 3D hand space (Fig. 3.3I). We quantified hand movement along the null axis, orthogonal to the baseline plane. Hand movement variance in the Z axis increased with learning in the perturbation block (Fig 3.3J, early null axis variance in B > mean null axis variance in A, first day: p = 0.0896 (n.s), last day: p = 0.00413(**) ; late null axis variance in B > mean null axis variance in A, first day: p = 0.0139 (*), last day: p = 0.0296 (*) paired t-test, N = 6 RA learning series), as expected due to the 3D exploration. Z axis variance returned to negligible levels upon return to baseline mapping in the A' block (Fig 3.3J, early null axis variance in A' > mean null axis variance in A, first day: p = 0.139 (n.s), last day: p = 0.629 (n.s) ; late null axis variance in A' > mean null axis variance in A, first day: p = 0.154 (*), last day: p = 0.0771 (n.s) paired t-test, N = 6 RA learning series). In the A' block, we did not observe any characteristic aftereffects in task irrelevant axis in hand trajectories. Hand movement variance along z axis during

the perturbation block did not have any relationship with the perturbation size (Fig. 3.3K, R^2 = -0.139, p = 0.792 (n.s), linear regression).

3.3.4 Distinct learning timescales in RP vs RA perturbations

We compared performance changes within and between sessions for relevance-preserving (RP) and relevance-altering (RA) perturbations, which revealed clear differences in the learning timescales between perturbation types (Fig. 4). To illustrate these dynamics, we examined representative sessions from RP and RA perturbations. RP perturbations produced rapid, feedback-driven adaptation within each block, with animals reaching near-baseline success rates within a day (Fig. 4A). In contrast, RA perturbations exhibited slower within-day gains and large performance jumps between consecutive days (offline gains) (Fig. 4B) and initially had no or lower success during B than RP perturbations (Fig. 4C, first day success rates for RA < RP mappings, p = X, Mann Whitney U test, N = 15 RP learning series, 7 RA learning series across both monkeys).

Introducing RP and RA perturbations had different impacts on task success. This behavioral difference opens up the possibility that two perturbation types imposed distinct learning demands. Despite differences in initial performance, both RP and RA perturbations showed performance improvements within a single training session (Fig. 4D, mean within session performance change > 0, RP: 20.66% $\pm$ 15.9%, p = 1.65e-10 (****), N= 41 learning days across 15 RP learning series from two monkeys; RA: 21.42% $\pm$ 16.53%, p= 6.87e-8 (****), N = 29 learning days across 7 RA learning series from two monkeys). In contrast, RP and RA perturbations showed distinct patterns of offline gains (Fig. 4E). Only RA perturbations showed statistically significant offline performance changes (mean offline gain in performance > 0, RP: 2.85% $\pm$ 26.70%, p = 0.302 (n.s), N= 41 learning days across 15 RP learning series from two monkeys; RA: 12.26% $\pm$ 12.15%, p= 8.145e-5 (****), N = 29 learning days across 7 RA learning series from two monkeys). Across sessions, offline gains were inversely correlated with within-day learning (R^2 = 0.50, p = 0.011; Fig. 4F), suggesting that offline learning primarily occurred after sessions where learning was limited.

We next asked what factors might influence the distinct learning dynamics between RA and RP perturbations. We hypothesized that feedback density – the extent to which learned movement

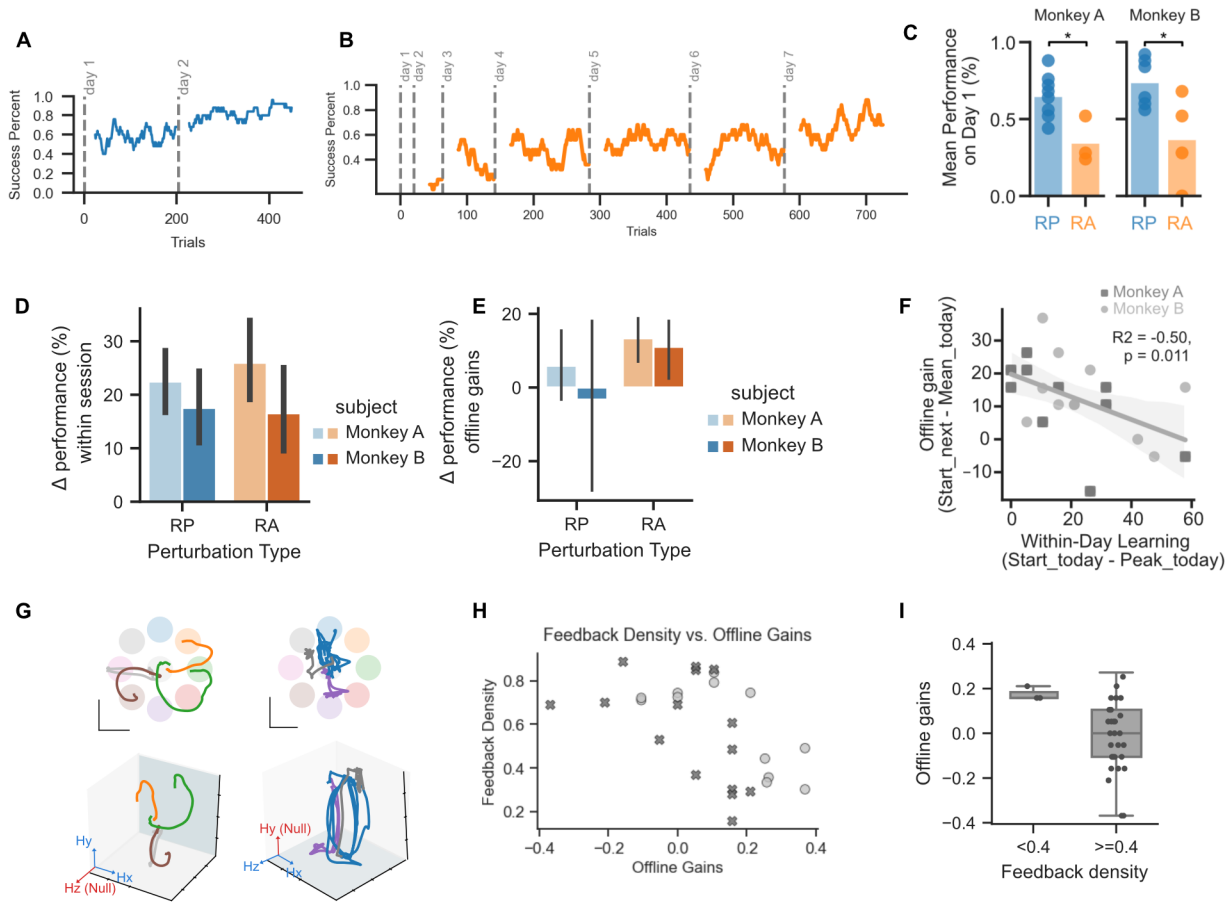

Figure 3.4: Diverse learning timescales in relevance preserving vs relevance altering perturbations (A) Success percent across trials during a relevance-preserving (RP) perturbation for Monkey B across two consecutive days. (B) Success percent across trials during a relevance-altering (RA) perturbation for Monkey B across multiple days. (C) Mean day 1 success percent during perturbation block across perturbation types –RP (blue) and RA (orange) .(Mann-Whitney U test, RP > RA, p value: monkey A = 0.0204, monkey B = 0.0333) (D) Change in performance within session for relevance-preserving (RP, blue) and relevance-altering (RA, orange) perturbations, shown separately for each subject. (E) Offline performance gains for RP and RA perturbations for Monkey A (blue) and Monkey B (orange). (F) Relationship between offline gains and within-day learning across all sessions ($R^2 = -0.50$, $p = 0.011$) from monkey A (grey circles) and monkey B (dark grey squares) (G) Cursor trajectories (top) and corresponding 3D hand trajectories (bottom) during late learning in RP (left) and RA (right) (H) Feedback density plotted against offline gains across sessions. Each point represents a single day. (I) Comparison of offline gains for sessions with low (< 0.4) and high (≥ 0.4) feedback density.

variance from the baseline mapping produced feedback – shaped how animals learned each perturbation. As quantified above (Fig. 3.2), animals generated movements that were primarily within the learned 2D baseline plane during RP perturbations. This plane is the task-relevant plane in both baseline and RP perturbations, which means most of the animals movement variance resulted in informative cursor feedback. RA perturbations, by necessity, resulted in animals generating hand movements outside of the baseline plane (Fig. 3.3). Example hand movement and cursor trajectories from early in the first RA and RP sessions illustrate this difference in Fig. 4G.We quantified feedback density by computing the ratio of cursor movement variance to hand movement variance, which captures how much of the movement space contributes to observable feedback (see Methods). Sessions with low feedback density, where little of the hand movement projected onto cursor motion, were associated with larger offline gains (Fig. 4H–I). These findings suggest that the availability of task-relevant feedback dictates how movement variance is used for learning. When feedback is sufficient, learners refine control through rapid, online error correction. When feedback is limited and inconsistent, learning relies on generating exploratory variance and offline processes that reorganize control toward feedback-relevant dimensions.

3.3.5 History dependent modulation of movement variability

Relevance-altering (RA) perturbations did not show a consistent relationship between learning time and perturbation size (Fig. 3.3D), suggesting that the difficulty of learning was not determined solely by the geometric magnitude of the perturbation. We therefore hypothesized that other aspects of the learning experience, such as the history of experience with RA perturbations, contributed to RA perturbation learning dynamics (Fig. 5). The time to learn RA perturbations decreased systematically with training order, with later RA mappings learned more quickly than earlier ones. No such trend was observed for relevance-preserving (RP) perturbations (Fig. 5A), which were well-explained by perturbation size (Fig. 3.2D). This pattern demonstrated a history dependence in learning that was specific to relevance-altering mappings: prior experience with altered task relevance accelerated future learning, independent of perturbation size.

Direct comparisons between two relevance-altering (RA) perturbations of similar magnitude in

Monkey B confirmed this history-dependent effect and provided insight into the mechanism by which prior experience shapes behavior to facilitate subsequent learning. During the first exposure to an RA perturbation that used an XZ task-relevant plane (right–left hand movements controlled horizontal cursor motion, and front–back movements controlled vertical motion), Monkey B exhibited slow, multi-day learning with limited within-session improvement on early days. In contrast, when later exposed to another RA perturbation of similar magnitude that used a "minus XZ" task-relevant plane (right–left hand movements again controlled horizontal cursor motion, but back–front rather than front–back movements controlled vertical motion), the monkey showed substantially faster acquisition and higher initial success (Fig. 5B).

Comparing the dynamics of 3D hand movement between these two RA perturbations revealed differences in how the monkey generated movements to identify the task-relevant plane. During the first exposure to a RA perturbation, monkey B continued to make primarily planar, 2D reaches within the 3D workspace (Fig. 5C, left). In contrast, during the second exposure, the monkey immediately generated movements that spanned the 3D space before gradually converging back to a stable 2D subspace over the course of several days (Fig. 5C, right). This shift in movement generation dynamics suggests that prior experience led the monkeys to alter their exploration strategies to more efficiently identify the task-relevant plane. In this redundant setting, 3D exploration specifically increases the likelihood of receiving useful feedback to first identify the task-relevant plane before then refining control to a lower-dimensional, task-relevant plane.

Because perturbation order was randomized, the two monkeys experienced different sequences of relevance-altering perturbations, which led to distinct exploration strategies. Monkey A encountered smaller perturbations first ($20°$, $45°$, then $90°$), and showed relatively high variance along the third principal component (PC_3) even for small rotations when normalized to the exploration volume. This indicates that, despite the modest geometric shift from the baseline plane, Monkey A explored sufficiently in 3D space to identify the altered task-relevant plane (Fig. 5D left). In contrast, Monkey B transitioned directly from the baseline XY plane to an XZ plane during the first RA exposure with a large change in task relevance. Although Monkey B learned this first XZ mapping slowly, upon a second exposure of similar magnitude he immediately generated substantial 3D variance to sample

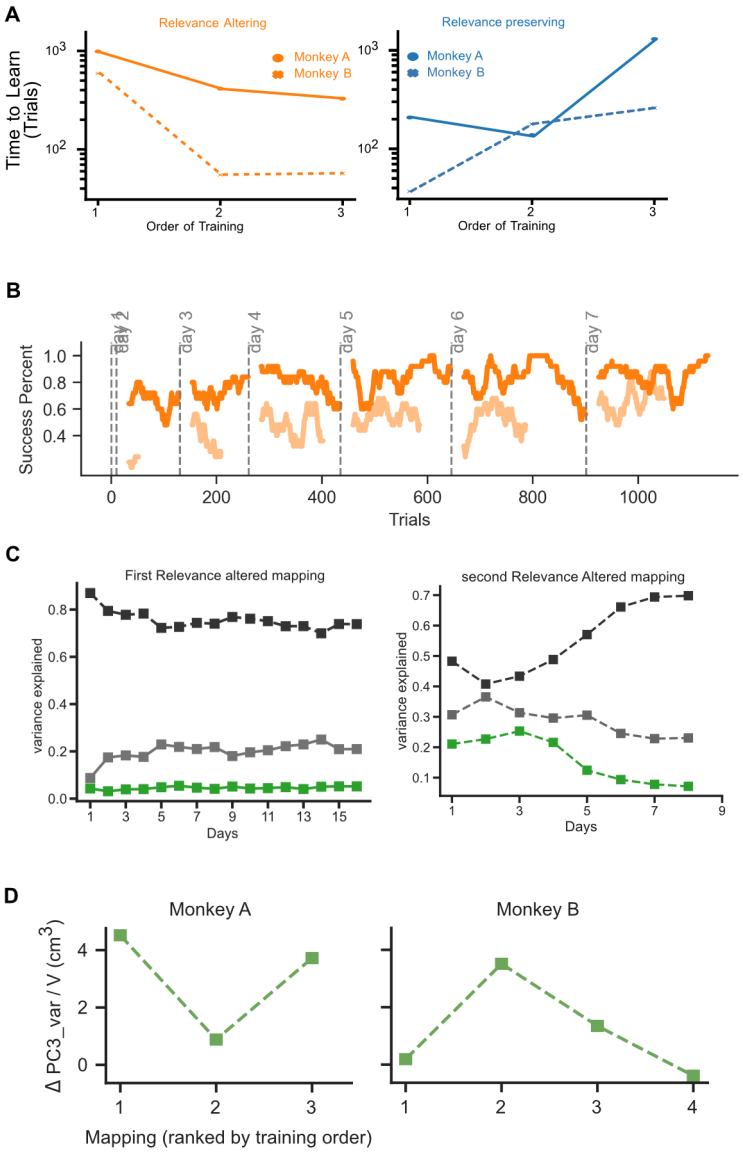

Figure 3.5: Prior learning influences exploration (A) Time to learn (log scale) plotted as a function of training order for relevance-altering (RA, orange) and relevance-preserving (RP, blue) perturbations for Monkey A (solid lines) and Monkey B (dashed lines). (B) Success percent across trials and days for Monkey B for first relevance altering perturbation (light orange) and similar magnitude second relevance altering perturbation (orange). (C) Variance explained by the first (black), second (gray), and third (green) PC for first (left) and second (right) relevance altering perturbation learned by monkey B. (D) Change in the variance captured by the third PC (Δ PC_3 variance normalized by exploration volume) across mappings ranked by training order for monkey A (left) and monkey B (right).

the new control volume. For subsequent RA perturbations of smaller magnitudes (45° and 20°), exploration scaled down accordingly (Fig. 5D right), suggesting that Monkey B adjusted exploration effort based on prior experience and the geometric demands of the mapping.

These complementary learning trajectories illustrate that the two monkeys discovered different strategies for exploring redundant motor spaces. One animal (Monkey B), exposed first to a large perturbation, learned to explore broadly and reused that strategy efficiently during subsequent mappings. The other (Monkey A), who experienced incremental increases in perturbation magnitude, expanded exploration only as needed when task demands became more complex. Together, these findings reveal that the history of exposure to altered task relevance shapes how the motor system regulates exploration, either by scaling exploratory breadth dynamically with task demands or by maintaining a readiness to generalize from previous large-scale perturbations.

3.4 Discussion

3.4.1 A Single Paradigm to Study Adaptation and De Novo Learning

Our results establish a unified redundant motor interface capable of isolating two distinct regimes of motor learning—adaptation and *de novo* skill acquisition—depending on how task relevance is manipulated. During baseline training, monkeys rapidly reduced variance in the task-irrelevant (null) dimension, effectively constraining their 3D hand movements to a 2D plane aligned with the cursor mapping. This behavior offers a critical perspective on the Uncontrolled Manifold (UCM) hypothesis, which posits that variability should be allowed to accumulate along task-irrelevant dimensions to minimize control effort Scholz and Schöner [1999]; Müller and Sternad [2004]. While our task theoretically permitted infinite variance along the null axis, the animals actively minimized it. This aligns with Optimal Feedback Control (OFC) theories, where the controller minimizes not just task error but also metabolic cost and control effort Todorov and Jordan [2002]. By reducing variance in the null space, the motor system effectively simplified the redundant control problem into a lower-dimensional, non-redundant one Todorov and Jordan [2002].

This baseline structure heavily influenced subsequent learning strategies. Perturbations that

preserved the task-relevant plane (RP) operated within the established control manifold. These perturbations elicited rapid, within-day learning, robust aftereffects, and scaling with perturbation magnitude—classic signatures of error-based adaptation observed in traditional visuomotor rotation studies Shadmehr and Mussa-Ivaldi [1994]; Krakauer [2009]. Because the task structure (the XY plane) remained invariant, the animals could efficiently update their internal models using vector error signals Shadmehr et al. [2010].

In contrast, Relevance Altering (RA) perturbations rendered the learned manifold obsolete, requiring the recruitment of dimensions that had been previously suppressed. This engaged a fundamentally different learning process characterized by slow, multi-day acquisition and a lack of aftereffects, hallmarks of *de novo* skill learning Telgen et al. [2014]; Yang et al. [2021]. This dichotomy confirms that ”variability” and ”redundancy” are not static properties but functional tools; the brain dynamically regulates them to perform credit assignment. When task relevance is altered, error-based mechanisms are insufficient, and the system must revert to variance-based exploration to discover the new control subspace Dhawale et al. [2017].

3.4.2 Offline Learning and Distinct Timescales

The dissociation between RP and RA perturbations was further evident in their learning timescales. RP learning occurred primarily online (within-session), consistent with the fast timescales of synaptic update in forward model adaptation Smith et al. [2006]. Conversely, RA learning relied heavily on offline gains between sessions. While animals showed limited improvement during the first exposure to an RA perturbation, performance jumped significantly between days (Figure 4).

This pattern suggests that the structural reorganization required for RA learning relies on consolidation processes that occur during rest or sleep Klinzing et al. [2019]; Gulati et al. [2014]; Bönstrup et al. [2019]. We hypothesize that offline processing becomes critical when online feedback density is low. Early in RA learning, hand movements confined to the baseline plane generated sparse cursor motion, providing a ”gradient starvation” problem where feedback was insufficient to drive rapid gradient-descent learning Pezeshki et al. [2021]. Under these conditions, the brain likely relies on offline replay to re-evaluate recent exploration and stabilize the new task manifold Rubin et al.

[2022]; Eichenlaub et al. [2020]. This finding highlights a critical limitation in current BCI and rehabilitation protocols: when teaching novel, high-dimensional controllers, continuous training may be less effective than spaced training that leverages offline consolidation windows.

3.4.3 Implications for Learning Constraints and Neural Manifold Theory

Our findings challenge strict interpretations of neural manifold theory, which suggest that movements outside the established neural subspace are difficult to learn Sadtler et al. [2014]; Golub et al. [2018]. While we observed that RA (outside-manifold) perturbations were indeed slower to learn than RP (within-manifold) perturbations, they were not strictly constrained by covariance related factors. Monkeys could freely move in 3 dimensional space. However, learning to map specific motor commands already in motor repertoire to new context took time.

Learnability was highly context-dependent. Specifically, we observed a ”learning-to-learn” effect: prior experience with an RA perturbation accelerated the learning of subsequent, distinct RA mappings (Figure 5). This history dependence suggests that the difficulty of learning outside the manifold is not a fixed biological constraint but a dynamic feature of the exploration strategy Heald et al. [2021]. Monkey B, for instance, learned to expand exploration into 3D space immediately upon the second RA exposure. This indicates the animal acquired a meta-structure of the task environment—learning that ”the solution is a plane, but its orientation can change” rather than just a specific motor mapping.

Chapter 4

ECoG BCI learning with Fixed Decoders with Rest

4.1 Introduction

Brain-Computer Interfaces (BCIs) have emerged as groundbreaking tools in neuroscience research and clinical neuroprosthetics, offering direct pathways for communication and control between the brain and external devices [Wolpaw et al., 2002; Hochberg et al., 2006]. While intracortical devices have historically provided the highest resolution, ElectroCorticography (ECoG)-based BCIs offer a compelling alternative, capturing rich spatiotemporal brain activity with high signal stability and reduced invasiveness [Leuthardt et al., 2004; Schalk et al., 2008]. A critical advantage of this mesoscale resolution is the ability to track large-scale neural dynamics over extended periods [Volkova et al., 2019]. By integrating these ECoG arrays with a fixed decoding algorithm, we encounter a novel opportunity to investigate the underlying mechanisms of learning. This ”fixed decoder” paradigm forces the brain to adapt to a static control manifold, allowing us to isolate and examine the specific neural reorganizations that drive skill acquisition, unobscured by compensatory changes in the decoding strategy [Ganguly and Carmena, 2009; Sadtler et al., 2014; Orsborn et al., 2014].

To deeply investigate these mechanisms, this study utilized a sophisticated multimodal experimental platform capable of simultaneous optical stimulation and electrical recording across the

cortex [Yazdan-Shahmorad et al., 2016; Scholl et al.]. In the specific brain areas where we recorded neural activity—spanning primary motor and premotor cortices—we utilized targeted viral injections to express light-sensitive channelrhodopsins. This setup allowed us to optically stimulate distinct neural populations while simultaneously recording broad electrical activity, effectively mapping effective connectivity between groups of neurons at the mesoscale [Yazdan-Shahmorad et al., 2016].

A central aim of our work was to synergize this optogenetic profiling with a rigorous BCI learning paradigm. By interleaving BCI training with optogenetic stimulation, we established a framework to observe how network connectivity changes with practice and, critically, how those changes continue to evolve after practice ends. This longitudinal approach offers a dynamic window into the structural and functional plasticity that accompanies the acquisition of neuroprosthetic control [Ganguly et al., 2011; Gulati et al., 2014].

Furthermore, we recognized that rest periods are not merely passive intervals but are active phases known to facilitate performance improvements and memory consolidation [Diekelmann and Born, 2010]. Specifically, the phenomenon of "neural replay"—the offline reactivation of task-relevant firing sequences—is hypothesized to be a key mechanism driving this consolidation [Wilson and McNaughton, 1994; Foster and Wilson, 2006]. To leverage this, our protocol included specific periods of rest between BCI sessions. We maintained neural recordings during these intervals to allow for the study of offline mechanisms, providing the data necessary to investigate how the brain consolidates learning when not actively engaged in the task [Gulati et al., 2017; Eichenlaub et al., 2020].

4.2 Method

4.2.1 Experimental Design

Animal Model

Our experimental design involves one adult male rhesus macaque (*Macaca mulatta*): Monkey 1 (Beignet, 9 years old, 11.3 kg)[cite: 1552]. The subject was selected based on availability, well-documented neuroanatomy, and prior training in manual control tasks. Prior to the study, the animal underwent a thorough health evaluation to ensure fitness for participation. All procedures were

approved by the Institutional Animal Care and Use Committee (IACUC) at the University of Washington and comply with the National Institutes of Health guide for the care and use of Laboratory animals[cite: 1550].

Implantation and Viral Injection Procedure

The animal was implanted with a custom titanium chamber in the left hemisphere, stereotaxically targeted to span portions of the primary motor cortex (M1), premotor cortex (PM), frontal eye fields (FEF), and dorsolateral prefrontal cortex (DLPFC). The chamber was custom-shaped to the skull curvature using MRI data and attached using self-tapping titanium screws. A 2.6 cm diameter craniotomy was performed, followed by a durotomy.

Following the durotomy, an adeno-associated virus targeting excitatory cells (AAV9-CaMKIIa-hChR2(H134R)-mCherry, Addgene) was injected to express channelrhodopsin. An artificial dura with 32 punched holes arranged in a grid pattern was used to guide injections. A Hamilton syringe was angled perpendicular to the cortical surface, and injections were made at 7 sites at depths of 3, 2, and 1 mm. At each depth, 1.0 µl of viral vector was injected over 30 seconds, followed by a one-minute wait period (ten minutes after the final depth).

Following injections, a removable silicone artificial dura was implanted to seal the craniotomy. Post-operative care included analgesics and antibiotics. The animal was given 8 weeks to recover and allow for viral expression before behavioral training resumed.

4.2.2 Neural Recording and Optogenetic Hardware

Neural data were recorded using a 244-electrode µECoG array (762 µm inter-electrode pitch, 229 µm contact size)[cite: 1557]. The array was embedded in a silicone artificial dura designed to be optically opaque but featured a grid of 32 holes matching the viral injection sites. Fiberoptic cannulas (200 µm diameter) were aligned to these holes using a custom fixture glued atop the array. This setup allowed for simultaneous electrical recording and optical access to the cortical surface.

Signals from 240 active electrodes were amplified and digitized at 25 kHz using an eCube recording system (White-Matter Inc., Seattle WA). The recordings were time-stamped and synchronized

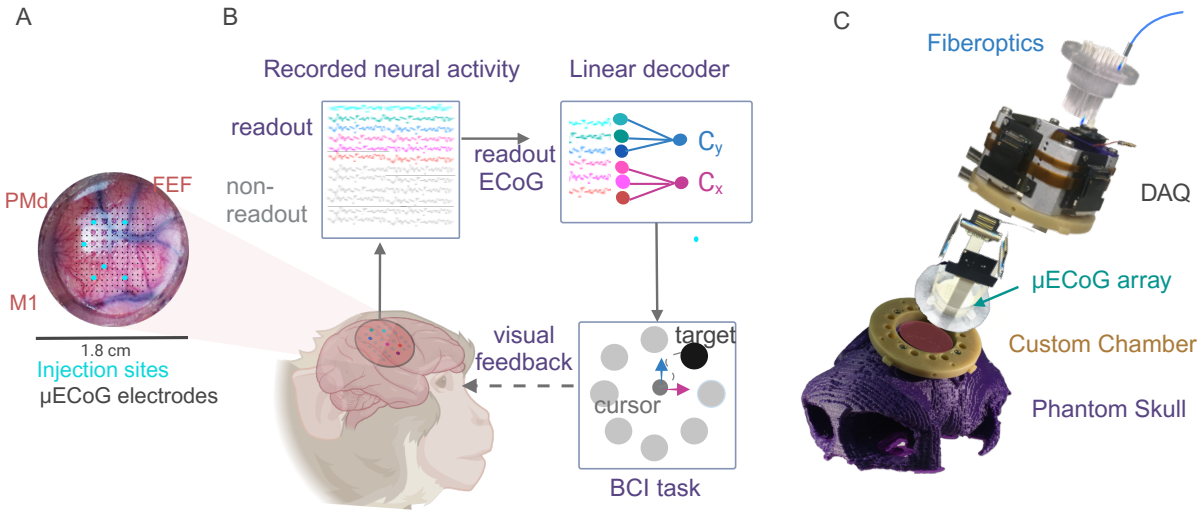

Figure 4.1: A. Neural Recording Sites: Topographical brain map displaying the electrode array placement on the cortical surface. B. BCI Task Interface: Neural activity is recorded and processed through a linear decoder to translate brain signals into cursor movement. C. Recording and Stimulation setup: Hardware assembly featuring the micro-ECoG array connected to fiberoptics and the eCube Data Acquisition (DAQ) system.

with behavioral data and laser stimulation events using the eCube data acquisition system.

Fig 4.1 provides a top-down view of the brain, marking the areas involved in the BCI task. The setup includes the μECoG array and fiberoptics connected to the Data Acquisition (DAQ) system.

The implant was semi-chronic in nature; the array was inserted once and remained in place for the entirety of the BCI learning series. The design included four access ports that allowed for regular saline flushing to maintain brain health throughout the study. The array was only explanted after the conclusion of the learning series.

4.2.3 Brain-Computer Interface Task

The BCI apparatus consists of a computer interface for the behavioral task, the implanted ECoG array, and a linear decoder. The cursor was controlled by a custom version of BMI3D and displayed on a 21.5 cm 16:9 LCD display. Display frame timing was synchronized via a microcontroller.

We collected high gamma band activity (80-200 Hz), indicative of motor planning, to serve as input for a linear decoder updating at 10 Hz. We specifically selected six readout units based on their proximity to the viral injection sites: three sites from M1 and three sites from PMd. Importantly,

these sites were chosen strictly for their location relative to optogenetic expression; no other signal quality criteria were used for readout selection.

The decoder was implemented as a linear positional decoder that mapped neural features to cursor positions. Decoder accumulated neural features in batches and updates estimates at 10 Hz. Neural features were first z-scored using statistics (mean and standard deviation) calculated from a three-minute baseline recording each day. To mitigate artifacts, input observations exceeding a standard deviation threshold were rejected. Valid features were then smoothed using a moving average window and linearly mapped to control states (position or velocity) via a weighted matrix. Finally, the output was scaled to the display workspace coordinates.

Task Structure

The subject performed a 2D center-out cursor task with 8 peripheral targets equally spaced at 45° intervals around the center (0°, 45°, 90°, 135°, 180°, 225°, 270°, and 315° relative to the horizontal axis). Each trial was self-initiated and began with the subject holding the cursor inside a central target for 200 ms. Upon a successful hold, one of eight peripheral targets appeared, spaced at 45-degree intervals around the center at a distance of 6.5 cm. A variable delay of 100 to 600 ms followed, after which the center target disappeared (Go Cue. To achieve a success, the subject was required to navigate the cursor to the peripheral target and hold for 200 ms. Successful trials resulted in a liquid reward.

4.2.4 Training and Stimulation Protocol

Initial Behavioral Training

An initial training period acclimated the animal to the task using positive reinforcement. Behavioral training sessions spanned 1.5 years, progressing from manual reaching tasks to the BCI task. During the initial BCI phase, a velocity control Kalman Filter decoder was used, followed by a fixed linear mapping for position control.

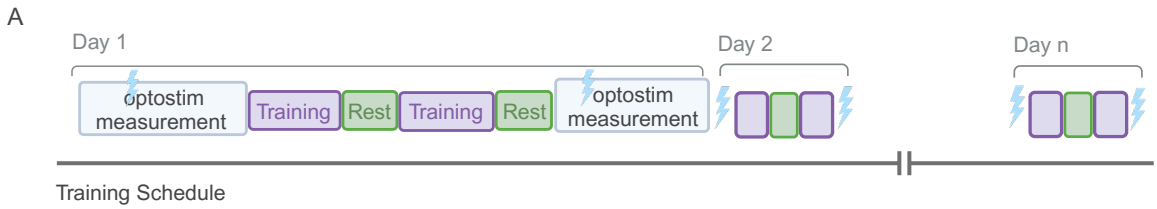

Figure 4.2: A. Training and data collection schematic for ECoG BCI learning series.

Optogenetic Stimulation

We performed optogenetic stimulation to measure network connectivity. A 450 nm (blue) laser, coupled to a 200 µm optic fiber, was used to stimulate specific sites. Laser power was set to achieve 10 ± 2 mW at the fiber tip. Stimulation was delivered using Poisson-randomized inter-stimulation intervals (ISI) drawn from an exponential distribution with a mean of 500 ms (minimum 100 ms). Pulse widths were randomly selected between 1 and 30 ms to avoid network habituation.

Evoked Response Analysis

To quantify the neural response to optogenetic stimulation, we analyzed the µECoG activity in the 250 ms following each laser pulse. Data were normalized by subtracting the mean activity from the 250 ms prior to laser onset. The maximum deviation (max ERP) was calculated for every electrode to map the spatial extent and magnitude of the evoked response. Volume of activation was compared to chance distributions generated from non-stimulated epochs to determine significance.

4.2.5 Behavioral Analysis Methods

To quantify the performance improvements in our BCI task, we utilized three standard performance metrics: success rate, reach time, and path length. These metrics collectively provide a comprehensive view of the subject's proficiency and learning curve over time.

Success rate was calculated as the ratio of the number of successful reaches and holds at the peripheral target to the total number of trials initiated. This metric directly reflects the monkey's

ability to complete the task and is indicative of skill acquisition.

Reach time was measured from the GoCue signal to the moment the cursor entered the peripheral target. This interval was recorded for all trials where the cursor successfully reached the target, encompassing both successful trials and those unsuccessful due to holding errors at the peripheral target.

Path length was assessed by calculating the total distance traveled by the cursor along the reach trajectory. This measurement provides insights into the efficiency and fluidity of the movement strategy employed by the subject, with shorter paths suggesting more direct and controlled movements.

4.3 Results

This study successfully demonstrated the viability of a multimodal platform integrating ECoG-based BCI control with chronic optogenetic setup to map network connectivity in a non-human primate.

4.3.1 Behavioral Learning in the ECoG BCI Task

Over the course of the study, we observed notable trends in the performance metrics of the BCI task across an eight-day period. The success rate, as shown in Fig 4.3 panel A, fluctuated initially but demonstrated an upward trend, suggesting that the subjects were able to improve their proficiency at the task over time. After an initial adaptation phase, there was a marked improvement, with success rates climbing to over 60%. Fig 4.3 Panel B presents the reach time data, which varied from day to day. Despite some variability, there was a general decrease in reach time by the end of the observation period. This decrease indicates that the subjects were becoming more efficient at reaching the target, albeit with some days showing increased times, possibly reflecting the subjects' daily variability in performance or concentration. Path length metrics in Fig 4.3 panel C indicate a decreasing trend over the initial days, followed by stabilization. This metric suggests that the subjects were learning to optimize their movement strategy to create more direct paths over time. Although there were fluctuations, the overall path length by day eight was shorter than at the start

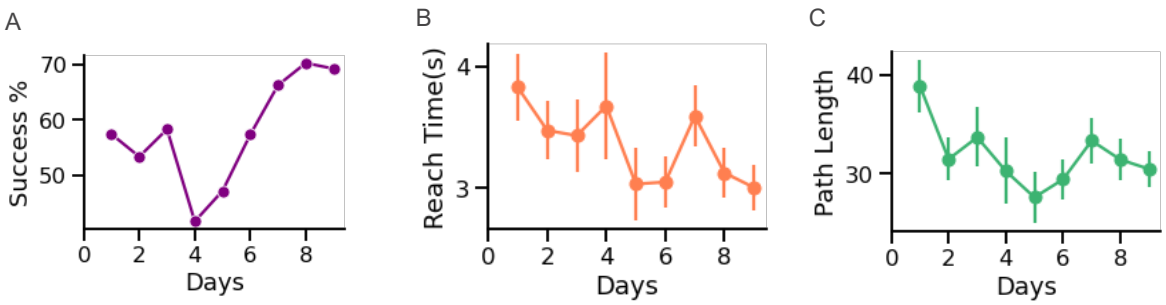

Figure 4.3: A. Success rate percentage across an eight-day period, with a general upward trend indicating learning over time. B Reach time in seconds, with a trend toward decreased times, suggesting increased efficiency. C. Path length of reach trajectories, with a decrease over time followed by stabilization, reflecting optimization in movement strategy.

of the trial period.

4.3.2 Longitudinal Optogenetic Mapping of Effective Connectivity

Every day before and after BCI training, we performed optogenetic stimulation to map network connectivity at sites our readout electrodes in M1 and PMd. Optical pulses delivered to specific injection sites (Fig. 4.4A) reliably evoked electrical potentials both near and far from stimulation sites across the µECoG array (Fig. 4.4B), confirming the system's ability to measure synaptically driven network responses. We generated spatial heatmaps of these evoked responses to visualize connectivity structure (Fig. 4.4C). Crucially, the semi-chronic implant maintained signal stability throughout the learning period. Connectivity maps for a representative stimulation site showed a consistent spatial structure from Day 1 to Day 9 (Fig. 4.4C), validating the system's capacity for longitudinal monitoring of network properties during skill acquisition.

4.4 Discussion

This study establishes a methodological framework for bridging the gap between large-scale neural connectivity and the specific computations driving motor learning. By integrating a semi-chronic micro-ECoG (µECoG) interface with wide-field optogenetic stimulation, we created a system capable of longitudinally tracking both the functional output of the cortex (via BCI control) and its un-

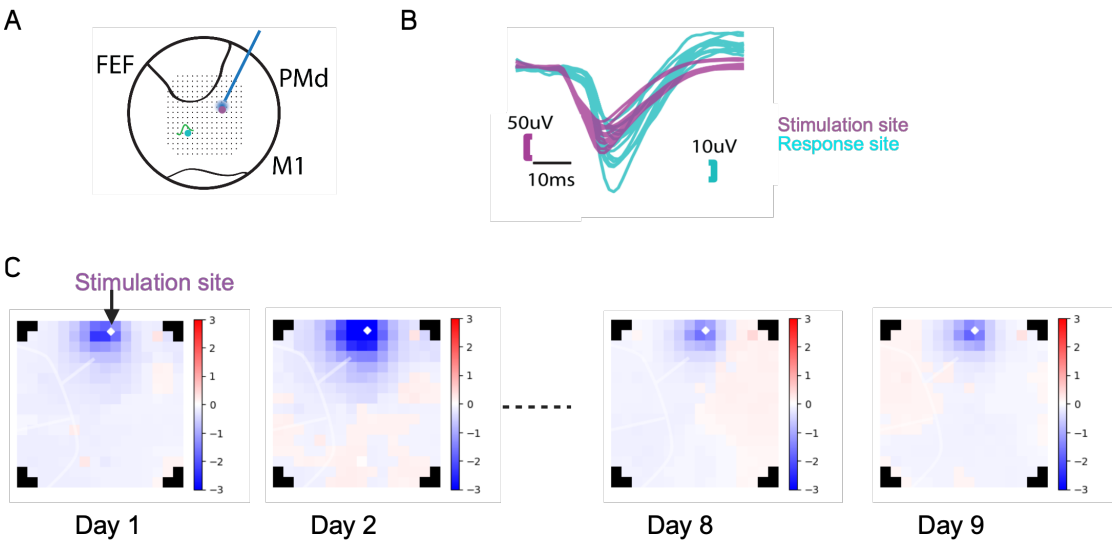

Figure 4.4: A.Schematic showing our optogenetic stimulation and ECoG neural recording setup. B Example evoked responses near stimulation site and response site. C. Spatial maps capturing evoked responses across all channels for the first two days and last two days of BCI training.

derlying effective connectivity (via stimulation-evoked responses) [Yazdan-Shahmorad et al., 2016]. This multimodal approach addresses a critical challenge in systems neuroscience: linking the structural reorganization of neural circuits to the emergence of skilled behavior [Orsborn and Pesaran, 2017].

4.4.1 Linking Connectivity to Computation

A central hurdle in understanding learning is determining how synaptic plasticity translates into improved computational performance [Dayan and Cohen, 2011]. While traditional electrophysiology reveals correlations in spiking activity, and anatomical tracers reveal static structure, neither fully captures the dynamic, causal interactions that evolve during skill acquisition [Magrans de Abril et al., 2018]. Our method overcomes this by using optogenetic stimulation as a causal probe [Petreanu et al., 2012]. By delivering precise optical pulses to specific neural populations, we can map the "effective connectivity"—the functional strength of connections—across the cortical surface. Because this probing is interleaved with a fixed-decoder BCI task, we can directly associate changes in these connectivity maps with specific improvements in decoding features and behavioral metrics. This design allows us to ask not just if the network changes, but how specific pathways are

reweighted to support the generation of the high-gamma control signals required by the decoder [Ganguly et al., 2011; Koralek et al., 2012]. Furthermore, observing these changes within the ”fixed decoder” paradigm enables us to isolate neural reorganization from decoder adaptation, providing a clearer window into the neural constraints on learning [Sadtler et al., 2014; Golub et al., 2018; Hennig et al., 2021].

4.4.2 Enabling the Study of Offline Learning Mechanisms

A unique strength of this dataset is the incorporation of rest periods with continuous recording, enabling the rigorous study of offline learning mechanisms. The role of sleep and rest in memory consolidation is well-established [Diekelmann and Born, 2010; Klinzing et al., 2019], with neural replay—the reactivation of waking firing patterns—serving as a hypothesized mechanism for synaptic stabilization [Wilson and McNaughton, 1994; Foster and Wilson, 2006]. However, observing replay at the mesoscale, particularly in primate motor cortex, remains a frontier. While recent studies have identified BCI-related reactivation in spikes [Gulati et al., 2014, 2017], our µECoG setup captures broad spatial patterns of activity that may reflect ”replay” of the manifold dynamics associated with BCI control [Rubin et al., 2022; Eichenlaub et al., 2020]. By applying dimensionality reduction and decoding techniques to resting-state data, future analyses of this dataset could determine whether the ”neural trajectory” of the cursor is spontaneously reactivated during rest and whether the fidelity of this replay predicts subsequent performance gains [Bönstrup et al., 2019; Jiang et al., 2017].

4.4.3 Future Directions: Plasticity and Stability

This experimental design opens several avenues for future inquiry regarding the stability of neural representations. Recent work suggests that while individual neurons may exhibit representational drift, the low-dimensional latent dynamics of the population remain stable [Gallego et al., 2020; Driscoll et al., 2022; Rule et al., 2019]. Our method allows us to investigate whether this stability extends to the effective connectivity of the network. We can examine whether the connectivity maps remain stable even as the precise neural units driving the BCI might drift, or conversely, if stable behavior relies on the continuous homeostatic retuning of these connections. Additionally,

this setup is ideally suited to test hypotheses regarding ”within-manifold” versus ”outside-manifold” learning, specifically by determining if learning new BCI dynamics relies on global network shifts or localized plasticity restricted to the readout nodes [Sadtler et al., 2014; Feulner and Clopath, 2021; Payeur et al., 2023].

Ultimately, this work moves beyond simply observing neural activity during a task. By actively probing the network connectivity with optogenetics and evoking behavior with a BCI, we have designed a toolset to causally dissect the circuit-level mechanisms that enable the brain to learn, consolidate, and master new skills.

Chapter 5

Dimensionality Metrics for Neural Data: A Comparative Analysis

5.1 Introduction

In the intricate landscape of neuroscience, the study of population dynamics stands out for its ability to provide a comprehensive view of neural activity Cunningham and Yu [2014]; Saxena and Cunningham [2019]; Vyas et al. [2020a]. This approach, focusing on the collective behavior of neuronal populations, unveils a more nuanced picture than what can be discerned from individual neurons in isolation. Central to understanding these dynamics is the concept of dimensionality, a measure that quantifies how a relatively small number of latent variables can capture a significant portion of neural variance Cunningham and Yu [2014]; Gao and Ganguli [2015]. This observation is crucial, as it suggests that the complex information processed by vast networks of neurons might be represented within a more compact, yet rich, dimensional space Gallego et al. [2017]; Jazayeri and Ostojic [2021].

However, accurately estimating the intrinsic dimensionality of neural data remains a significant challenge Altan et al. [2021]; Facco et al. [2017]. The neural system is a highly complex network characterized by lower-dimensional embeddings, nonlinear interactions, and noisy measurements Gao et al. [2017]. While various estimation methods have been proposed—ranging from linear

techniques like Principal Component Analysis (PCA) Jolliffe [2002] to complex nonlinear manifold learning Tenenbaum et al. [2000]; Roweis and Saul [2000]—they are often evaluated on disparate datasets. This makes direct comparison difficult; a method that performs well on motor cortex data may not translate effectively to sensory areas, and different metrics may imply different underlying structures within the same data Altan et al. [2021].

Crucially, the majority of existing dimensionality analyses have focused on well-learned tasks. In these contexts, neural representations are typically found to be stable, confined to a fixed manifold, and optimized for execution Gallego et al. [2018, 2020]; Churchland et al. [2012]; Kaufman et al. [2016]. There is significantly less understanding of how these metrics behave during the dynamic and often more variable process of learning, where the neural code is in flux and may involve exploring new neural patterns or reassociating existing ones Sadtler et al. [2014]; Golub et al. [2018]; Oby et al. [2019]; Hennig et al. [2021]. To utilize dimensionality as a metric for learning, we must first understand how different estimators react to the nuances of neural data. For instance, linear methods like PCA are computationally efficient but may oversimplify dynamics by assuming linearity. Conversely, nonlinear methods may capture intricate patterns but are often computationally intensive and sensitive to noise Altan et al. [2021]. Furthermore, properties such as data resolution and thresholding choices can drastically alter dimensionality estimates—at smaller scales, a dataset might appear low-dimensional, while higher resolutions reveal complexity aligning with the embedding space Recanatesi et al. [2020]. Without a rigorous comparison of these methods on a unified dataset, it is risky to apply them to the study of learning, where distinguishing between signal evolution and estimation artifacts is critical.

5.2 Research question

To address these challenges, this chapter aims to systematically evaluate the pros and cons of various dimensionality estimation techniques. Our goal is to understand these methods in detail so that we can ultimately apply them to learning in an informative and useful way.

Rather than viewing these tools in isolation, we seek to determine which methods are most robust for capturing the changing landscape of neural activity. By exploring a variety of approaches—from

linear dimensionality reduction to advanced nonlinear methods—within the same neural datasets, we can isolate the effects of the method from the properties of the data. This comparative analysis is a necessary prerequisite for using dimensionality as a reliable metric to track neural plasticity Perich et al. [2018]; Sun et al. [2022], aiming to answer: Which dimensionality metrics are sufficiently robust and interpretable to characterize the evolution of neural codes during learning?

5.3 Methods

Dimensionality estimation is the process of determining the minimum number of variables, or dimensions, that are needed to capture the essential structure or variability of a dataset. In the context of neural data analysis, dimensionality estimation is often used to determine the number of independent sources of activity or the number of features that are needed to represent the neural data.

5.3.1 Dimensionality metrics

There are various methods for estimating the dimensionality of neural data, and each method has its strengths and weaknesses. We compared dimensionality scores estimated by different methods explained below on the same spiking neural dataset.

PCA

Principal Component Analysis (PCA) is a widely used statistical technique for estimating the dimensionality of a dataset, often utilized for purposes such as dimensionality reduction and denoising of data. The procedure begins by computing the covariance matrix of the dataset to understand the variance shared among its features. Eigen decomposition is then applied to this covariance matrix to extract the principal components (PCs), which are the directions of maximum variance in the data.

The number of dimensions retained is determined based on a threshold set for the explained variance, a critical step where the right balance must be struck. This threshold, however, depends on the specific application and the desired balance between dimensionality reduction and informa-

tion retention. A major limitation of PCA is its dependence on choosing an appropriate variance threshold; setting this threshold either too high or too low can lead to either excessive or insufficient data reduction. Additionally, PCA inherently assumes linear relationships within the data, which might not hold true for complex datasets like neural data, known for their non-linear characteristics. Another limitation is PCA's disregard for the local structure of the data, focusing only on global patterns, which can sometimes lead to the loss of significant local information. PCA can be biased towards detecting only the linear structure of the data, and may not capture the non-linear structure in neural data.

Participation ratio

The participation ratio measures the extent to which each neuron contributes to the total variance of the data Recanatesi et al. [2019]; Gao et al. [2017]. A high participation ratio suggests that many neurons are active and the data is high-dimensional, while a low participation ratio indicates that the data is low-dimensional. This method is computationally simple non-parametric method and doesn't make any assumptions about the structure of the data but still cannot capture the nonlinearity in neural data.

Factor analysis

Factor analysis assumes that the data is generated by a set of latent factors, which can be identified by analyzing the correlation structure of the data. The dimensionality is estimated by determining the number of factors that explain a significant proportion of the variance in the data Athalye et al. [2017]. Factor analysis can capture non-linear relationships between variables, but may be sensitive to noise and outliers. It can capture non-linear dependencies in the data but is computationally intensive.

Parallel analysis

Parallel analysis is a statistical method that compares the eigenvalues of the observed data to those generated by random data with the same dimensions Altan et al. [2021]. The dimensionality is

estimated by identifying the number of eigenvalues that exceed the values generated by the random data. This method is less susceptible to noise and outliers than other methods and can be used with non-normal data.

Two-nearest neighborhood

The twoNN method is a simple and efficient method for estimating the intrinsic dimension of a dataset Altan et al. [2021]; Facco et al. [2017]. The method works by first finding the nearest neighbors of each data point. The number of neighbors is set to 2. The intrinsic dimension of the dataset is then estimated as the average number of unique neighbors over all data points. This method estimates the intrinsic dimensionality of the data by computing the average number of nearest neighbors required to cover the data manifold. This method is useful for data that has non-linear structure and is less affected by noise than other methods.

Correlation dimension

Correlation dimension estimation is a method for measuring the complexity of a dataset. It is based on the idea that the correlation between two points in a dataset decreases as the distance between them increases Altan et al. [2021]. To estimate the correlation dimension, we first need to calculate the correlation between all pairs of points in the dataset. This can be done using a variety of methods, such as the Pearson correlation coefficient or the mutual information. Once we have calculated the correlation between all pairs of points, we can plot the correlation as a function of the distance between the points. The correlation dimension is then estimated as the slope of the line that best fits the data. A higher correlation dimension indicates that the dataset is more complex. This is because more complex datasets have more structure and therefore more correlation between the points. The correlation dimension is a measure of the complexity of the data structure. It estimates the number of points required to cover a fractal set with a given dimensionality. This method can capture non-linear relationships between variables and is less susceptible to noise and outliers than other methods

5.3.2 Synthetic Dataset

Our goal with the mock benchmark dataset is to use it to compare different dimensionality metrics with known true dimension.

M1_sphere Dataset

The `M1_sphere` dataset, provided by the `skdim.datasets` module, is a synthetic dataset primarily used for the analysis of dimensionality estimation methods. It consists of samples distributed uniformly on the surface of a unit sphere. This dataset is particularly useful for testing algorithms under conditions where data is spread uniformly in a high-dimensional space but inherently lies on a lower-dimensional manifold. The generation of the `M1_sphere` dataset involves sampling points from a multidimensional Gaussian distribution and normalizing them to lie on the surface of a sphere. This creates a spherical distribution of points in a high-dimensional space, representing a simple yet effective model for evaluating the performance of dimensionality reduction techniques and manifold learning methods. The `M1_sphere` dataset is ideal for testing the effectiveness of algorithms in capturing global, linear structures in the data.

Mn1_nonlinear Dataset

The `Mn1_nonlinear` dataset from the `skdim.datasets` module is designed to challenge dimensionality estimation methods with a more complex, nonlinear structure. It is a synthetic dataset composed of samples that lie on a nonlinear manifold embedded in a higher-dimensional space. This dataset is particularly valuable for examining the performance of algorithms in the presence of nonlinear relationships and complex data structures. The `Mn1_nonlinear` dataset is generated through a process that introduces nonlinearity and intertwines the dimensions in a way that mimics real-world data scenarios where linear assumptions do not hold. It serves as a crucial tool for testing and comparing the robustness of various dimensionality reduction techniques, especially those designed to uncover and preserve the intrinsic nonlinear structures in data. This dataset assists in assessing the ability of algorithms to handle nonlinearity and complex data distributions effectively.

5.3.3 Spiking Dataset

In our research, we analyze a spiking dataset derived from the Orsborn et al. [2014], which features multi-unit threshold crossing firing rates from 96 channel intracortical electrodes. This dataset was collected when monkeys learned a center out reaching task using a brain computer interface with adaptive decoder with implants in the primary motor cortex (M1). We chose spiking activity binned at 100 ms for computing and comparing dimensionality estimates.

5.4 Results

In this paper, we compare how dimensionality estimations differ under different noise and non-linearity conditions in a mock neural dataset with known true dimensions and then look at how preprocessing and learning could affect dimensionality estiamtes in spiking data.

5.4.1 All methods estimate dimensionality close to true in linear, noise free data

Our results indicate a comparison of dimensionality estimation methods applied to a noise-free linear dataset. Fig 5.1A shows a 3D scatter plot of the dataset, confirming its linear structure without any added noise. Fig 5.1 B compares various dimensionality estimation methods, including both linear (PCA 0.8 explained variance, PCA 0.95 explained variance, PR, PA, FA, MLE) and nonlinear (2-NN, cor-int, FSA, MLE) approaches. It's observed that all methods, irrespective of being linear or nonlinear, closely estimate the true dimensionality of the dataset, demonstrating their effectiveness in a noise-free linear context.

5.4.2 Linear methods overestimate dimensionality in low-dimensional non-linear embedding

The results from the "M1_swissroll" dataset, as visualized in Fig 5.2 A, exhibit a non-linear, swiss roll topology, characterized by points that form a spiraling two-dimensional manifold within the three-dimensional space. In Fig 5.2 B, a range of dimensionality estimation methods are evaluated on this dataset without the presence of noise. The true dimensionality is signified by the dotted

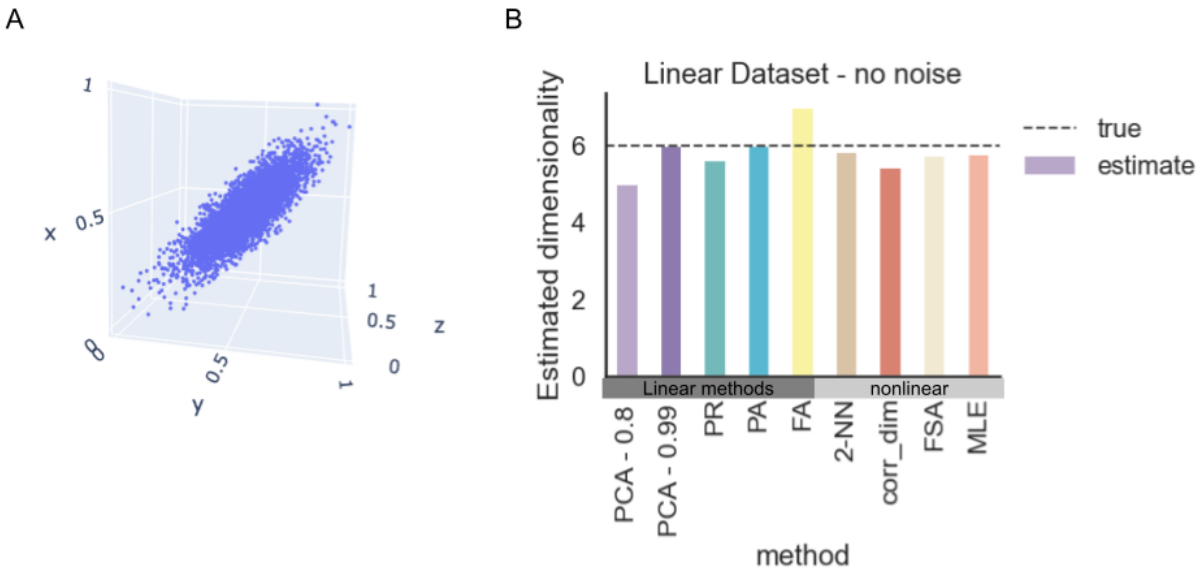

Figure 5.1: A. 3D scatter plot of dataset, B. Dimensionality estimates from methods (PCA with thresholds 0.8 and 0.99 for explained variance , Participation Ratio, Parallel Analysis, Factor Analysis, correlation dimension, Two-Nearest Neighbor, Fisher Separability , Maximum Likelihood estimate) . The true dimensionality is indicated by the dashed line, and the estimated dimensionalities are represented by the solid bars.

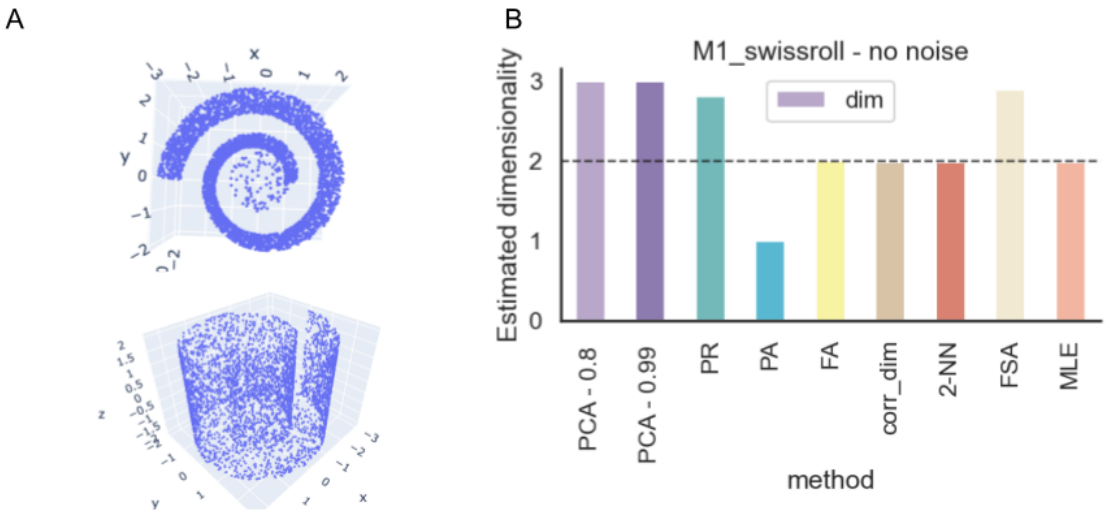

Figure 5.2: A. 3D scatter plot of dataset, B. Dimensionality estimates. The true dimensionality is indicated by the dashed line, and the estimated dimensionalities are represented by the solid bars.

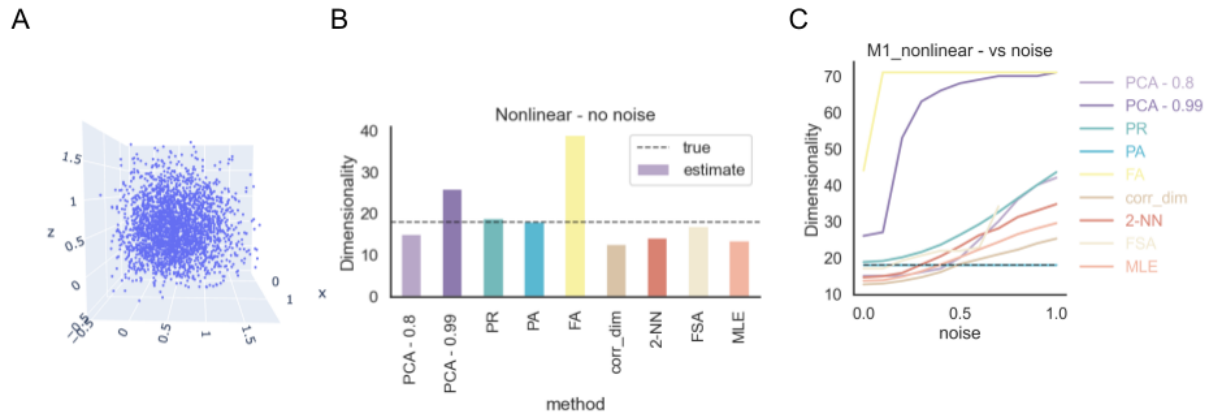

Figure 5.3: A. 3D scatter plot of dataset, B. Dimensionality estimates. The true dimensionality is indicated by the dashed line, and the estimated dimensionalities are represented by the solid bars. C. Effect of noise vs dimensionality estimations.

line, against which each method's estimation is compared. It can be observed that PCA, at various explained variance thresholds (0.8 and 0.99), along with PR, PA, and FA methods, tend to underestimate the true dimensionality. In contrast, cor-dim and 2-NN provide closer estimates, while MLE notably overestimates. These results highlight the challenges inherent in dimensionality estimation for datasets with complex, non-linear structures.

5.4.3 Parallel analysis estimates close to true dimension in nonlinear high dimensional embedding

In the analysis of the "M1_nonlinear" dataset, Fig 5.3 A displays a 3D scatter plot, indicating a dispersed distribution of data points, consistent with a non-linear manifold. Fig 5.3 B illustrates the performance of various dimensionality estimation methods on this dataset without noise. The dashed line represents the true dimensionality, and it is evident that most methods underestimate it to varying degrees, with the exception of MLE, which overestimates. Surprisingly, Linear methods PA and PR gave close to true estimates when non-linear embedding was in a higher dimensional space.

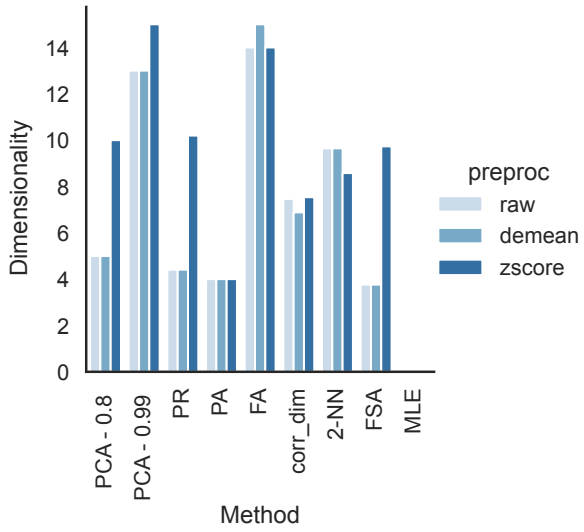

Figure 5.4: A. Dimensionality estimates across different preprocessing techniques. The bar chart compares the estimated dimensionality of a dataset using various methods (PCA with explained variance thresholds 0.8 and 0.99 , Participation Ratio, Parallel Analysis, Factor Analysis, correlation dimension, Two-Nearest Neighbor, Fisher Separability , Maximum Likelihood estimate) under three preprocessing conditions: raw data, demeaned data, and z-score normalized data. The estimates highlight the influence of preprocessing on perceived data complexity, with z-scoring consistently showing higher dimensionality.

Parallel analysis performs well. This agrees with observations about non-linear methods underestimating dimensionality when $D > 10$, as observed in Altan et al. [2021]. Fig 5.3 C further explores the robustness of these methods against increasing levels of noise. It is observed that methods such as PCA (with thresholds 0.8 and 0.99), FA, and MLE exhibit an increase in estimated dimensionality with higher noise levels, whereas methods like PR, PA, cor-dim, 2-NN, and FSA maintain a relatively consistent estimation across noise levels. This suggests a varying degree of sensitivity among the methods to noise when estimating the dimensionality of non-linear data.

5.4.4 Spiking Data: Preprocessing affects dimensionality estimates

Fig 5.4 illustrates the impact of different preprocessing techniques on the estimated dimensionality of a dataset, as determined by various analytical methods. Three preprocessing approaches are compared: raw data, demeaning, and z-scoring. Across methods such as PCA (at 0.8 and 0.99 thresholds), PR, PA, FA, cor-dim, 2-NN, FSA, and MLE, z-scoring generally yields the highest dimensionality estimates, while raw data typically results in the lowest. Demeaning appears to result in dimensionality estimates that are consistently intermediate between raw and z-scored data. This

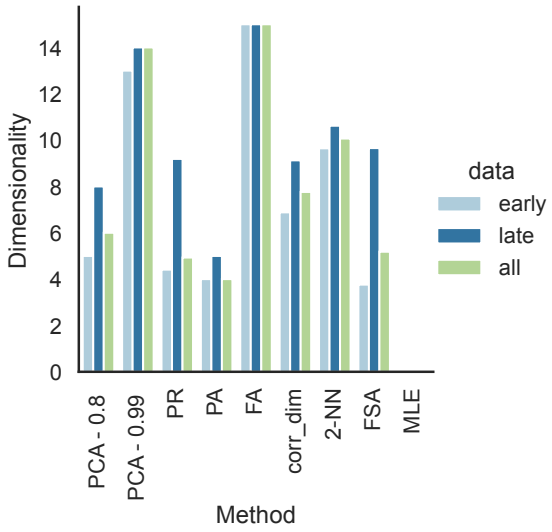

Figure 5.5: A. Dimensionality estimates across different preprocessing techniques. The bar chart compares the estimated dimensionality of a dataset using various methods (PCA with explained variance thresholds 0.8 and 0.99 , Participation Ratio, Parallel Analysis, Factor Analysis, correlation dimension, Two-Nearest Neighbor, Fisher Separability , Maximum Likelihood estimate) during three conditions: early learning stage, late learning stage and combined early and late data. The estimates highlight the influence of preprocessing on perceived data complexity, with z-scoring consistently showing higher dimensionality.

pattern suggests that the choice of preprocessing can significantly influence the perceived complexity of the dataset as captured by dimensionality estimation techniques.

5.4.5 Changes in data dimensionality during learning stages

Fig 5.5 presents a comparative analysis of dimensionality estimates across different stages of data collection: early, late, and the entire dataset (all). The dimensionality, denoted as 'dim', is estimated using methods: PCA with two thresholds (0.8 and 0.99), PR, PA, FA, cor-dim, 2-NN, FSA, and MLE. It's observed that for most methods, the estimated dimensionality is higher in the early stages of data collection compared to the late stage, with the exception of FA and MLE where late stage estimates are higher or equal to early ones. The overall trend suggests a temporal variation in the complexity of the dataset, with different methods reflecting this change to varying degrees. The 'all' category, representing the aggregate data, typically shows dimensionality estimates that are intermediate between the early and late stages, indicating the averaging effect of combining temporal datasets.

5.5 Discussion

In pursuit of an unbiased method for estimating the dimensionality of neural datasets, our investigation has yielded key insights. As population dynamics becomes central to our understanding of neural computation Cunningham and Yu [2014]; Vyas et al. [2020a], the choice of estimation method is no longer trivial. Our comparative analysis using both synthetic benchmarks and biological data reveals that there is no single metric that can be applied to any condition; rather, the choice of method depends on noise levels, nonlinearity, and, crucially, data preprocessing.

5.5.1 Trade-offs between linear and nonlinear estimators

A primary challenge in neural data analysis is the inherent nonlinearity of the system. Our results on synthetic manifolds confirm that standard linear methods, such as PCA and Factor Analysis, tend to overestimate dimensionality when the data lies on a nonlinear manifold (Figures 5.2 and 5.3). This overestimation occurs because linear basis sets must tile the curved manifold with multiple flat hyperplanes to capture the variance, effectively interpreting curvature as additional dimensions Altan et al. [2021].

Conversely, while nonlinear methods like the Two-Nearest Neighbor (2-NN) estimator Facco et al. [2017] and correlation dimension underestimated dimensionality in low-noise nonlinear settings, they exhibited varying sensitivity to noise. Notably, our analysis identified Parallel Analysis (PA) as a surprisingly robust candidate. Despite being a linear method based on eigenvalue decomposition, PA closely aligned with the true dimension even in higher-dimensional nonlinear embeddings (Fig 5.3). This supports recent findings suggesting that PA effectively filters out the ”noise floor” of the eigenvalue spectrum, making it less susceptible to the inflation of dimensionality often seen with standard PCA variance thresholds Altan et al. [2021].

5.5.2 Preprocessing

One of our key practical findings is the profound impact of preprocessing on dimensionality estimates (Fig 5.4). We observed that Z-scoring (standardizing firing rates) consistently yielded higher

dimensionality estimates compared to raw or demeaned data.

This effect likely arises because Z-scoring amplifies the variance of neurons with low firing rates. In raw data, high-firing neurons dominate the covariance matrix, often confining the estimated manifold to the ”dominant” modes of population activity. By normalizing activity, Z-scoring boosts the contribution of low-firing neurons—which may contain subtle task-relevant signals, but also significantly more Poisson-like noise Gao et al. [2017]. Consequently, methods that rely on variance structures (like PCA and FA) interpret this amplified noise as additional dimensions. This suggests that for high-SNR recordings, raw or demeaned data may provide a more conservative estimate of the ”dominant” manifold, whereas Z-scoring should be reserved for analyses where the contribution of sparse or low-firing units is hypothesized to be computationally critical.

5.5.3 Could dimensionality be a signature of learnability?

The ultimate goal of validating these metrics was to apply them to the study of neural plasticity. Our analysis of the spiking dataset from Orsborn et al. [2014] revealed a distinct temporal trend: dimensionality estimates were generally lower during the early stages of learning compared to the late stages (Fig 5.5).

The change in dimensionality over time aligns with the hypothesis that learning involves a transition from neural exploration to consolidation. During early learning, the motor cortex may explore a wider state space to identify activity patterns that map successfully to the desired output Sadtler et al. [2014]; Golub et al. [2018]. As the subject refines their skill, the neural population likely converges onto a stable, lower-dimensional manifold that efficiently generates the required control signals Gallego et al. [2020]; Athalye et al. [2017]. The fact that Factor Analysis (FA) and Maximum Likelihood Estimation (MLE) showed a different trend (higher or equal estimates in late stages) may indicate that these specific methods are more sensitive to the fine-grained, task-specific variance that emerges once the large-scale exploratory variance has settled. This divergence underscores the necessity of using a suite of estimators rather than a single metric when characterizing the dynamic process of learning.

5.5.4 Conclusion

Accurately gauging the complexity of neural data is a prerequisite for understanding how the brain processes information. Our results suggest that for general neural recordings, Parallel Analysis offers a robust balance between resisting noise and capturing structure. However, the drastic effects of preprocessing highlight that ”dimensionality” is not an absolute property of the data, but a function of how we choose to view it. Future work applying these metrics to learning must carefully report preprocessing steps and potentially utilize comparative approaches—contrasting linear and nonlinear estimates—to disentangle changes in manifold geometry from changes in neural noise.

Chapter 6

Conclusion

This thesis investigated the computational principles that support sensorimotor learning and to develop tools that allow us to study neural dynamics during learning in non human primates. Learning is difficult to study because both behavior and brain activity evolve together, and the mechanisms that govern these changes extend across many levels of the motor system. The work presented here approached this challenge by simplifying the problem from both the behavioral and neural perspectives. On the behavioral side, tasks were designed to isolate a small set of motor dimensions within which learning could be rigorously measured. On the neural side, brain computer interfaces provided a controlled framework that maps population activity onto low dimensional control signals, allowing us to observe how learning reshapes cortical dynamics.

Across the projects described in this thesis, I examined how the brain assigns credit and adapts computations in response to changes in task demands. The studies on assistive algorithms demonstrated that manipulations of error can reshape learning. By altering the feedback structure available to the subject, we found that neural populations adopt distinct strategies to achieve the same behavioral goal. In complimentary work, I introduced perturbations that changed the relevance of different movement dimensions and showed that even simple reaching tasks engage multiple learning processes. Perturbations that preserved the known task relevant space were learned rapidly and produced clear after effects, consistent with classical adaptation. In contrast, perturbations that altered the task relevant space required slow, exploration driven learning without strong after effects,

suggesting the formation of new internal models. These results reveal that the computational demands of credit assignment, shaped by the available feedback, strongly influence which learning processes are recruited.

Beyond behavioral signatures of learning, this thesis contributes tools for studying the structure of cortical networks and how they reorganize with experience. Using optogenetic stimulation with micro electrocorticography recordings, we mapped directed functional connectivity across motor areas during and after behavior. These methods allow us to track network level changes at a resolution that is not accessible through behavioral measures or standard recording alone. In parallel, we developed analysis frameworks to understand population dimensionality and its relation to learning. Together, these approaches offer a richer view of how motor learning unfolds across both neural state space and network organization.

The brain controls billions of neurons that drive hundreds of muscles, creating thousands of possible degrees of freedom. To study learning in such a complex system, we simplified the sensorimotor loop: on the behavioral side, we used carefully designed tasks that reduced the motor degrees of freedom from thousands to just a few controllable dimensions; and on the neural side, we used brain–computer interfaces to map activity from billions of neurons down to a few dozen directly linked to behavior — a powerful abstraction that still preserves the essential computations underlying learning.

In summary, this thesis showed that assistive interventions that manipulate error and altering task relevance can change how the brain solves the same problem. We also built tools to map and track network connectivity, linking behavioral learning with underlying plasticity. Altogether, these studies expand our understanding of how credit-assignment computations shape learning across neural populations.

6.1 Collaborations

My thesis work has resulted in and is a result of some exciting collaborations within and outside of the Orsborn lab. I am grateful to collaborate and expand not only the boundaries of science but also the scientific contributions of the monkeys that contributed significant data for advancing neuro-

science research. We sought multidisciplinary collaborations combining insights from neuroscience, artificial neural networks (ANN), and optogenetics. Collaborative projects include

6.1.1 How similarly does the brain and ANN learn to solve a BCI task?

In this project that strengthened our successful collaborations with Prof. Guillaume Lajoie and Dr. Alex Payeur at Mila Quebec AI Institute, we led investigations into the parallels between brain learning and ANN learning for solving a BCI task. The efforts eventually contributed to Chapter 2 of my thesis and a poster presentation at COSYNE 2023. We build ANN models that can continually learn and use biologically plausible learning rules and closely resemble experimental conditions. The ANN model provides a crucial testbed for us to dissect and infer learning dynamics by allowing us access to all neurons in the model and we could perform manipulations beyond what is possible in experiments.

6.1.2 Optogenetics stimulation to map network connectivity

In research led by Dr. Leo Scholl at Orsborn lab, we used optogenetic stimulation paired with μECoG recording to measure directed functional connectivity. We successfully injected viral constructs to drive expression of Channelrhodopsin in seven sites across motor areas in 2 monkeys and used this to map connections across motor cortical areas. Following viral injections, we stimulated at varying pulse width (2-20 ms) and measured stimulation-evoked responses. Characterizing the response strength and latency across our array allows us to estimate functional connectivity. Using stimulation provides a directed measure that may more accurately capture connections. Our chronic recordings allowed us to measure connections across days to monitor plasticity induced during and immediately after behavioral training. This rich dataset paired with behavioral training could be valuable to study learning induced plasticity mechanisms.

6.1.3 Spatial mapping of arm, eye, and reward information using electrocorticogram in non human primates

This research led by Tomohiro Ouchi (PhD student, Orsborn lab) uses the ECoG data collected during arm reaching task of 2 monkeys (related to chapter 3) to study how arm and eye movement information are distributed across brain areas. Though ECoG signals are assumed to be averaged signals, this project shows spatial and temporally specificity of task information in M1, PMd and FEF brain areas, highlighting the need for new analysis methods to tease apart information present in the neural data for BCI decoder implications. Publication from this collaboration: Ouchi et al. [2025]

6.1.4 ECoG for targeting data-driven neuropixels insertion

In research led by Ryan Canfield (PhD student, Orsborn lab), we implement a data-driven approach for neuropixels insertion to sample units from task-relevant regions based on spatial maps derived from ECoG recordings. Motor cortex is often studied by indiscriminately sampling neurons and inferring population-level computations. Neurons in motor cortex comprise structured networks defined by anatomy. Coarse sampling across large areas may blend or miss information. Here, this work uses data driven sampling to reveal new insights into motor cortical computations.

6.1.5 Eye movements to elucidate learning strategies

In this project led by Dr. Leo Scholl (Orsborn lab) and contributed to by Joseph Asfouri and Jesus Cabrales , we studied how eye movements contribute to motor learning in macaque monkeys during behavioral perturbations used in Chapter 3. We noticed an increase in saccades when perturbations are introduced and an eventual decrease as perturbations are learned.

6.2 Open Questions

Although the work in this thesis advances our understanding of how motor learning operates in redundant and structured tasks, many questions remain. A central open question concerns the fundamental unit of computation: does the brain assign credit to individual neurons, or are errors

resolved at the level of collective population dynamics? Bridging the gap between local synaptic rules and population-level learning remains a critical challenge.

Furthermore, while the studies here primarily focused on cortical activity, motor learning relies on distributed circuits involving cerebellum, basal ganglia, and subcortical regions. Understanding how these distinct areas coordinate local and population-level plasticity is an important next step.

Another open question concerns generalization. The same cortical networks support a remarkable range of motor behaviors. How does a population that has learned one task restructure itself to learn another, and what limits this flexibility? Addressing these questions may require combining perturbation based tools, large scale neural recordings, and closed loop manipulations that target specific elements of the network.

Finally, this line of research opens the possibility of guiding or accelerating learning. If altering feedback structure or stimulating specific network connections can shift the strategy the brain adopts, then principled interventions may enhance skill acquisition in both healthy and clinical populations. Realizing this potential will require a deeper understanding of the computations that govern learning across the entire sensorimotor hierarchy.

Bibliography

sklearn.linear_model.LogisticRegression.

Leah Acker, Erica N Pino, Edward S Boyden, and Robert Desimone. 2016. FEF inactivation with improved optogenetic methods. *Proceedings of the National Academy of Sciences*, 113(46):E7297–E7306. Publisher: National Acad Sciences.

Andres Agudelo-Toro, Jonathan A. Michaels, Wei-An Sheng, and Hansjörg Scherberger. 2024. Accurate neural control of a hand prosthesis by posture-related activity in the primate grasping circuit. *Neuron*, 112(24):4115–4129.e8.

Ege Altan, Sara A. Solla, Lee E. Miller, and Eric J. Perreault. 2021. Estimating the dimensionality of the manifold underlying multi-electrode neural recordings. *PLOS Computational Biology*, 17(11):e1008591.

Pierre-Jean Arduin, Yves Frégnac, Daniel E. Shulz, and Valérie Ego-Stengel. 2013. "Master" Neurons Induced by Operant Conditioning in Rat Motor Cortex during a Brain-Machine Interface Task. *Journal of Neuroscience*, 33(19):8308–8320. Publisher: Society for Neuroscience Section: Articles.

Vivek R Athalye, Jose M Carmena, and Rui M Costa. 2020. Neural reinforcement: re-entering and refining neural dynamics leading to desirable outcomes. *Current Opinion in Neurobiology*, 60:145–154.

Vivek R. Athalye, Karunesh Ganguly, Rui M. Costa, and Jose M. Carmena. 2017. Emergence of Coordinated Neural Dynamics Underlies Neuroprosthetic Learning and Skillful Control. *Neuron*, 93(4):955–970.e5.

Karthikeyan Balasubramanian, Vasileios Papadourakis, Wei Liang, Kazutaka Takahashi, Matthew D. Best, Aaron J. Suminski, and Nicholas G. Hatsopoulos. 2020. Propagating Motor Cortical Dynamics Facilitate Movement Initiation. *Neuron*, 106(3):526–536.e4.

Alim Louis Benabid, Thomas Costecalde, Andrey Eliseyev, Guillaume Charvet, Alexandre Verney, Serpil Karakas, Michael Foerster, Aurélien Lambert, Boris Morinière, Neil Abroug, Marie-Caroline Schaeffer, Alexandre Moly, Fabien Sauter-Starace, David Ratel, Cecile Moro, Napoleon Torres-Martinez, Lilia Langar, Manuela Oddoux, Mircea Polosan, Stephane Pezzani, Vincent Auboiroux, Tetiana Aksenova, Corinne Mestais, and Stephan Chabardes. 2019. An exoskeleton controlled by an epidural wireless brain–machine interface in a tetraplegic patient: a proof-of-concept demonstration. *The Lancet Neurology*, 18(12):1112–1122. Publisher: Elsevier.

Max Berniker and Konrad Kording. 2008. Estimating the sources of motor errors for adaptation and generalization. *Nature Neuroscience*, 11(12):1454–1461.

Marlene Bönstrup, Iñaki Iturrate, Ryan Thompson, Gabriel Cruciani, Nitzan Censor, and Leonardo G. Cohen. 2019. A Rapid Form of Offline Consolidation in Skill Learning. *Current biology: CB*, 29(8):1346–1351.e4.

Jose M. Carmena, Mikhail A. Lebedev, Roy E. Crist, Joseph E. O'Doherty, David M. Santucci, Dragan F. Dimitrov, Parag G. Patil, Craig S. Henriquez, and Miguel A. L. Nicolelis. 2003. Learning to Control a Brain–Machine Interface for Reaching and Grasping by Primates. *PLOS Biology*, 1(2):e42. Publisher: Public Library of Science.

John K. Chapin, Karen A. Moxon, Ronald S. Markowitz, and Miguel A. L. Nicolelis. 1999. Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex. *Nature Neuroscience*, 2(7):664–670.

Jiahao Chen and Hong Qiao. 2020. Motor-Cortex-Like Recurrent Neural Network and Multi-Tasks Learning for the Control of Musculoskeletal Systems. *IEEE Transactions on Cognitive and Developmental Systems*, pages 1–1.

Simon X. Chen, An Na Kim, Andrew J. Peters, and Takaki Komiyama. 2015. Subtype-specific plasticity of inhibitory circuits in motor cortex during motor learning. *Nature Neuroscience*, 18(8):1109–1115. Publisher: Nature Publishing Group.

Ernest J. Cheng and Stephen H. Scott. 2000. Morphometry of Macaca mulatta forelimb. I. Shoulder and elbow muscles and segment inertial parameters. *Journal of Morphology*, 245(3):206–224.

Mark M. Churchland, John P. Cunningham, Matthew T. Kaufman, Justin D. Foster, Paul Nuyujukian, Stephen I. Ryu, and Krishna V. Shenoy. 2012. Neural population dynamics during reaching. *Nature*, 487(7405):51–56. Number: 7405 Publisher: Nature Publishing Group.

Jennifer L. Collinger, Brian Wodlinger, John E. Downey, Wei Wang, Elizabeth C. Tyler-Kabara, Douglas J. Weber, Angus JC McMorland, Meel Velliste, Michael L. Boninger, and Andrew B. Schwartz. 2013. High-performance neuroprosthetic control by an individual with tetraplegia. *The Lancet*, 381(9866):557–564.

John P Cunningham and Byron M Yu. 2014. Dimensionality reduction for large-scale neural recordings. *Nature Neuroscience*, 17(11):1500–1509.

Zoubin Ghahramani Daniel M. Wolpert and Michael I. Jordan. 1995. An internal model for sensorimotor integration. *Science*, 269(5232):1880–1882.

Eran Dayan and Leonardo G. Cohen. 2011. Neuroplasticity Subserving Motor Skill Learning. *Neuron*, 72(3):443–454.

Ashesh K. Dhawale, Maurice A. Smith, and Bence P. Ölveczky. 2017. The Role of Variability in Motor Learning. *Annual Review of Neuroscience*, 40(1):479–498.

Jörn Diedrichsen, Yasmin Hashambhoy, Tushar Rane, and Reza Shadmehr. 2005. Neural Correlates of Reach Errors. *Journal of Neuroscience*, 25(43):9919–9931.

Susanne Diekelmann and Jan Born. 2010. The memory function of sleep. *Nature Reviews Neuroscience*, 11(2):114–126. Publisher: Nature Publishing Group.

Laura N. Driscoll, Lea Duncker, and Christopher D. Harvey. 2022. Representational drift: Emerging theories for continual learning and experimental future directions. *Current Opinion in Neurobiology*, 76:102609.

Laura N. Driscoll, Krishna Shenoy, and David Sussillo. 2024. Flexible multitask computation in recurrent networks utilizes shared dynamical motifs. *Nature Neuroscience*, 27(7):1349–1363. Publisher: Nature Publishing Group.

Yadin Dudai, Avi Karni, and Jan Born. 2015. The Consolidation and Transformation of Memory. *Neuron*, 88(1):20–32.

Lea Duncker, Laura Driscoll, Krishna V Shenoy, Maneesh Sahani, and David Sussillo. 2020. Organizing recurrent network dynamics by task-computation to enable continual learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 14387–14397. Curran Associates, Inc.

Eva L. Dyer and Konrad Kording. 2023. Why the simplest explanation isn't always the best. *Proceedings of the National Academy of Sciences*, 120(52):e2319169120. Publisher: Proceedings of the National Academy of Sciences.

Jean-Baptiste Eichenlaub, Beata Jarosiewicz, Jad Saab, Brian Franco, Jessica Kelemen, Eric Halgren, Leigh R. Hochberg, and Sydney S. Cash. 2020. Replay of Learned Neural Firing Sequences during Rest in Human Motor Cortex. *Cell Reports*, 31(5):107581.

Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. 2017. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific Reports*, 7(1):12140.

M. S. Fee and J. H. Goldberg. 2011. A hypothesis for basal ganglia-dependent reinforcement learning in the songbird. *Neuroscience*, 198:152–170.

Barbara Feulner and Claudia Clopath. 2021. Neural manifold under plasticity in a goal driven learning behaviour. *PLOS Computational Biology*, 17(2):e1008621. Publisher: Public Library of Science.

David J. Foster and Matthew A. Wilson. 2006. Reverse replay of behavioural sequences in hippocam-

pal place cells during the awake state. *Nature*, 440(7084):680–683. Publisher: Nature Publishing Group.

Nicolas Frémaux, Henning Sprekeler, and Wulfram Gerstner. 2010. Functional Requirements for Reward-Modulated Spike-Timing-Dependent Plasticity. *Journal of Neuroscience*, 30(40):13326–13337.

Min Fu, Xinzhu Yu, Ju Lu, and Yi Zuo. 2012. Repetitive motor learning induces coordinated formation of clustered dendritic spines in vivo. *Nature*, 483(7387):92–95.

Juan A. Gallego, Matthew G. Perich, Raeed H. Chowdhury, Sara A. Solla, and Lee E. Miller. 2020. Long-term stability of cortical population dynamics underlying consistent behavior. *Nature Neuroscience*, 23(2):260–270.

Juan A. Gallego, Matthew G. Perich, Lee E. Miller, and Sara A. Solla. 2017. Neural Manifolds for the Control of Movement. *Neuron*, 94(5):978–984. Publisher: Elsevier.

Juan A. Gallego, Matthew G. Perich, Stephanie N. Naufel, Christian Ethier, Sara A. Solla, and Lee E. Miller. 2018. Cortical population activity within a preserved neural manifold underlies multiple motor behaviors. *Nature Communications*, 9(1):4233. Number: 1 Publisher: Nature Publishing Group.

Karunesh Ganguly and Jose M. Carmena. 2009. Emergence of a Stable Cortical Map for Neuroprosthetic Control. *PLoS biology*.

Karunesh Ganguly, Dragan F. Dimitrov, Jonathan D. Wallis, and Jose M. Carmena. 2011. Reversible large-scale modification of cortical networks during neuroprosthetic control. *Nature Neuroscience*, 14(5):662–667. Publisher: Nature Publishing Group.

Peiran Gao and Surya Ganguli. 2015. On simplicity and complexity in the brave new world of large-scale neuroscience. *Current Opinion in Neurobiology*, 32:148–155.

Peiran Gao, Eric Trautmann, Byron Yu, Gopal Santhanam, Stephen Ryu, Krishna Shenoy, and Surya

Ganguli. 2017. A theory of multineuronal dimensionality, dynamics and measurement. preprint, Neuroscience.

A. P. Georgopoulos, J. F. Kalaska, R. Caminiti, and J. T. Massey. 1982. On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *Journal of Neuroscience*, 2(11):1527–1537. Publisher: Society for Neuroscience Section: Articles.

Vikash Gilja, Paul Nuyujukian, Cindy A. Chestek, John P. Cunningham, Byron M. Yu, Joline M. Fan, Mark M. Churchland, Matthew T. Kaufman, Jonathan C. Kao, Stephen I. Ryu, and Krishna V. Shenoy. 2012. A high-performance neural prosthesis enabled by control algorithm design. *Nature Neuroscience*, 15(12):1752–1757.

Vikash Gilja, Chethan Pandarinath, Christine H. Blabe, Paul Nuyujukian, John D. Simeral, Anish A. Sarma, Brittany L. Sorice, János A. Perge, Beata Jarosiewicz, Leigh R. Hochberg, Krishna V. Shenoy, and Jaimie M. Henderson. 2015. Clinical translation of a high-performance neural prosthesis. *Nature Medicine*, 21(10):1142–1145. Publisher: Nature Publishing Group.

Matthew D Golub, Steven M Chase, Aaron P Batista, and Byron M Yu. 2016. Brain–computer interfaces for dissecting cognitive processes underlying sensorimotor control. *Current Opinion in Neurobiology*, 37:53–58.

Matthew D. Golub, Patrick T. Sadtler, Emily R. Oby, Kristin M. Quick, Stephen I. Ryu, Elizabeth C. Tyler-Kabara, Aaron P. Batista, Steven M. Chase, and Byron M. Yu. 2018. Learning by neural reassociation. *Nature Neuroscience*, 21(4):607–616.

Matthew D Golub, Byron M Yu, and Steven M Chase. 2015. Internal models for interpreting neural population activity during sensorimotor control. *eLife*, 4:e10015. Publisher: eLife Sciences Publications, Ltd.

Logan Grosenick, James H. Marshel, and Karl Deisseroth. 2015. Closed-loop and activity-guided optogenetic control. *Neuron*, 86(1):106–139.

Gaël Guennebaud, Benoît Jacob, and others. 2010. Eigen v3.

Tanuj Gulati, Ling Guo, Dhakshin S Ramanathan, Anitha Bodepudi, and Karunesh Ganguly. 2017. Neural reactivations during sleep determine network credit assignment. *Nature Neuroscience*, 20(9):1277–1284.

Tanuj Gulati, Dhakshin S. Ramanathan, Chelsea C. Wong, and Karunesh Ganguly. 2014. Reactivation of emergent task-related ensembles during slow-wave sleep after neuroprosthetic learning. *Nature Neuroscience*, 17(8):1107–1113. Number: 8 Publisher: Nature Publishing Group.

James B. Heald, Máté Lengyel, and Daniel M. Wolpert. 2021. Contextual inference underlies the learning of sensorimotor repertoires. *Nature*, 600(7889):489–493. Number: 7889 Publisher: Nature Publishing Group.

Jay A. Hennig, Emily R. Oby, Darby M. Losey, Aaron P. Batista, Byron M. Yu, and Steven M. Chase. 2021. How learning unfolds in the brain: toward an optimization view. *Neuron*, 109(23):3720–3735. Publisher: Elsevier.

Leigh R. Hochberg, Mijail Serruya, Gerhard M. Friehs, Jon A. Mukand, Maryam Saleh, Abraham H. Caplan, Almut Branner, David R. Chen, Richard D. Penn, and John P. Donoghue. 2006. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*.

Masato Inoue and Shigeru Kitazawa. 2018. Motor Error in Parietal Area 5 and Target Error in Area 7 Drive Distinctive Adaptation in Reaching. *Current Biology*, 28(14):2250–2262.e3.

Masato Inoue, Motoaki Uchimura, and Shigeru Kitazawa. 2016. Error Signals in Motor Cortices Drive Adaptation in Reaching. *Neuron*, 90(5):1114–1126.

Andrew Jackson, Jaideep Mavoori, and Eberhard E. Fetz. 2006. Long-term motor cortex plasticity induced by an electronic neural implant. *Nature*, 444(7115):56–60. Publisher: Nature Publishing Group.

Beata Jarosiewicz, Steven M. Chase, George W. Fraser, Meel Velliste, Robert E. Kass, and Andrew B. Schwartz. 2008. Functional network reorganization during learning in a brain-computer interface paradigm. *Proceedings of the National Academy of Sciences of the United States of America*, 105(49):19486–19491.

Beata Jarosiewicz, Anish A. Sarma, Daniel Bacher, Nicolas Y. Masse, John D. Simeral, Brittany Sorice, Erin M. Oakley, Christine Blabe, Chethan Pandarinath, Vikash Gilja, Sydney S. Cash, Emad N. Eskandar, Gerhard Friehs, Jaimie M. Henderson, Krishna V. Shenoy, John P. Donoghue, and Leigh R. Hochberg. 2015. Virtual typing by people with tetraplegia using a self-calibrating intracortical brain-computer interface. *Science Translational Medicine*, 7(313):313ra179–313ra179. Publisher: American Association for the Advancement of Science.

Mehrdad Jazayeri and Srdjan Ostojic. 2021. Interpreting neural computations by examining intrinsic and embedding dimensionality of neural activity. *Current Opinion in Neurobiology*, 70:113–120.

Xi Jiang, Isaac Shamie, Werner K. Doyle, Daniel Friedman, Patricia Dugan, Orrin Devinsky, Emad Eskandar, Sydney S. Cash, Thomas Thesen, and Eric Halgren. 2017. Replay of large-scale spatio-temporal patterns from waking during subsequent NREM sleep in human cortex. *Scientific Reports*, 7(1):17380. Number: 1 Publisher: Nature Publishing Group.

William M. Joiner and Maurice A. Smith. 2013. Long-term retention explained by a model of short-term learning in the adaptive control of reaching. *Journal of Neurophysiology*, 110(11):2703–2716.

I.T. Jolliffe. 2002. *Principal Component Analysis*, second edition edition. Springer Series in Statistics. Springer, New York, NY.

John F. Kalaska. 2019. Emerging ideas and tools to study the emergent properties of the cortical neural circuits for voluntary motor control in non-human primates. *F1000Research*, 8:749.

R. E. Kalman. 1960. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45.

A. Karni, G. Meyer, C. Rey-Hipolito, P. Jezzard, M. M. Adams, R. Turner, and L. G. Ungerleider. 1998. The acquisition of skilled motor performance: fast and slow experience-driven changes in primary motor cortex. *Proceedings of the National Academy of Sciences of the United States of America*, 95(3):861–868.

Matthew T. Kaufman, Jeffrey S. Seely, David Sussillo, Stephen I. Ryu, Krishna V. Shenoy, and Mark M.

Churchland. 2016. The Largest Response Component in the Motor Cortex Reflects Movement Timing but Not Movement Type. *eneuro*, 3(4):ENEURO.0085–16.2016.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*.

Jens G. Klinzing, Niels Niethard, and Jan Born. 2019. Mechanisms of systems memory consolidation during sleep. *Nature Neuroscience*, 22(10):1598–1610. Publisher: Nature Publishing Group.

Dmitry Kobak, Wieland Brendel, Christos Constantinidis, Claudia E Feierstein, Adam Kepecs, Zachary F Mainen, Xue-Lian Qi, Ranulfo Romo, Naoshige Uchida, and Christian K Machens. 2016. Demixed principal component analysis of neural population data. *eLife*, 5:e10989. Publisher: eLife Sciences Publications, Ltd.

Toshiki Kobayashi and Daichi Nozaki. 2024. Implicit motor adaptation patterns in a redundant motor task manipulating a stick with both hands. *eLife*, 13:RP96665. Publisher: eLife Sciences Publications, Ltd.

Aaron C. Koralek, Xin Jin, John D. Long Ii, Rui M. Costa, and Jose M. Carmena. 2012. Corticostriatal plasticity is necessary for learning intentional neuroprosthetic skills. *Nature*, 483(7389):331–335. Number: 7389 Publisher: Nature Publishing Group.

John W. Krakauer. 2009. Motor learning and consolidation: the case of visuomotor rotation. *Advances in Experimental Medicine and Biology*, 629:405–421.

John W. Krakauer, Alkis M. Hadjiosif, Jing Xu, Aaron L. Wong, and Adrian M. Haith. 2019. Motor Learning. In *Comprehensive Physiology*, pages 613–663. John Wiley & Sons, Ltd.

Mark M. Churchland Krishna V. Shenoy and John P. Cunningham. 2013. Computational neuroscience: Dynamics of population codes in motor cortex. *Nature*, 503:78–84.

Andrew J. Law, Gil Rivlis, and Marc H. Schieber. 2014. Rapid acquisition of novel interface control by small ensembles of arbitrarily selected primary motor cortex neurons. *Journal of Neurophysiology*, 112(6):1528–1548.

Mikhail A. Lebedev. 2014. How to read neuron-dropping curves? *Frontiers in Systems Neuroscience*, 8.

Eric C. Leuthardt, Gerwin Schalk, Jonathan R. Wolpaw, Jeffrey G. Ojemann, and Daniel W. Moran. 2004. A brain-computer interface using electrocorticographic signals in humans. *Journal of Neural Engineering*, 1(2):63–71.

Weiwei Li and Emanuel Todorov. 2004. Iterative linear quadratic regulator design for nonlinear biological movements systems. In *Proceedings of the First International Conference on Informatics in Control, Automation and Robotics*, pages 222–229.

Zheng Li, Joseph E. O'Doherty, Mikhail A. Lebedev, and Miguel A. L. Nicolelis. 2011. Adaptive Decoding for Brain-Machine Interfaces Through Bayesian Parameter Updates. *Neural Computation*, 23(12):3162–3204.

Maneeshika M. Madduri, Samuel A. Burden, and Amy L. Orsborn. 2023. Biosignal-based co-adaptive user-machine interfaces for motor control. *Current Opinion in Biomedical Engineering*, 27:100462.

Ildefons Magrans de Abril, Junichiro Yoshimoto, and Kenji Doya. 2018. Connectivity inference from neural recording data: Challenges, mathematical bases and research directions. *Neural Networks*, 102:120–137.

Kevin A. Mazurek and Marc H. Schieber. 2017. Injecting Instructions into Premotor Cortex. *Neuron*, 96(6):1282–1289.e4.

Luca Mazzucato, Alfredo Fontanini, and Giancarlo La Camera. 2016. Stimuli reduce the dimensionality of cortical activity. *Frontiers in systems neuroscience*, 10:11.

Marvin Minsky. 1961. Steps toward Artificial Intelligence. *Proceedings of the IRE*, 49(1):8–30.

Hermann Müller and Dagmar Sternad. 2004. Decomposition of variability in the execution of goal-oriented tasks: three components of skill improvement. *Journal of Experimental Psychology. Human Perception and Performance*, 30(1):212–233.

Alice Nieuwboer, Lynn Rochester, Liesbeth Müncks, and Stephan P. Swinnen. 2009. Motor learning in Parkinson's disease: limitations and potential for rehabilitation. *Parkinsonism & Related Disorders*, 15:S53–S58.

Yukio Nishimura, Steve I. Perlmutter, Ryan W. Eaton, and Eberhard E. Fetz. 2013. Spike-timing dependent plasticity in primate corticospinal connections induced during free behavior. *Neuron*, 80(5):1301–1309.

Emily R. Oby, Matthew D. Golub, Jay A. Hennig, Alan D. Degenhart, Elizabeth C. Tyler-Kabara, Byron M. Yu, Steven M. Chase, and Aaron P. Batista. 2019. New neural activity patterns emerge with long-term learning. *Proceedings of the National Academy of Sciences*, 116(30):15210–15215.

Amy Orsborn and Jose Carmena. 2013. Creating new functional circuits for action via brain-machine interfaces. *Frontiers in Computational Neuroscience*, 7.

Amy L. Orsborn, Siddharth Dangi, Helene G. Moorman, and Jose M. Carmena. 2012. Closed-loop decoder adaptation on intermediate time-scales facilitates rapid BMI performance improvements independent of decoder initialization conditions. *IEEE transactions on neural systems and rehabilitation engineering: a publication of the IEEE Engineering in Medicine and Biology Society*, 20(4):468–477.

Amy L. Orsborn, Helene G. Moorman, Simon A. Overduin, Maryam M. Shanechi, Dragan F. Dimitrov, and Jose M. Carmena. 2014. Closed-loop decoder adaptation shapes neural plasticity for skillful neuroprosthetic control. *Neuron*, 82(6):1380–1393.

Amy L. Orsborn and Bijan Pesaran. 2017. Parsing learning in networks using brain-machine interfaces. *Current Opinion in Neurobiology*, 46:76–83.

Tomohiro Ouchi, Leo R. Scholl, Pavithra Rajeswaran, Ryan A. Canfield, Lydia I. Smith, and Amy L. Orsborn. 2025. Mapping Eye, Arm, and Reward Information in Frontal Motor Cortices Using Electrocorticography in Nonhuman Primates. *Journal of Neuroscience*, 45(12). Publisher: Society for Neuroscience Section: Research Articles.

Anushka Oza, Adarsh Kumar, Apoorva Sharma, and Pratik K. Mutha. 2024. Limb-related sensory prediction errors and task-related performance errors facilitate human sensorimotor learning through separate mechanisms. *PLOS Biology*, 22(7):e3002703.

Chethan Pandarinath, Paul Nuyujukian, Christine H Blabe, Brittany L Sorice, Jad Saab, Francis R Willett, Leigh R Hochberg, Krishna V Shenoy, and Jaimie M Henderson. 2017. High performance communication by people with paralysis using an intracortical brain-computer interface. *eLife*, 6:e18554. Publisher: eLife Sciences Publications, Ltd.

Chethan Pandarinath, Daniel J. O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D. Stavisky, Jonathan C. Kao, Eric M. Trautmann, Matthew T. Kaufman, Stephen I. Ryu, Leigh R. Hochberg, Jaimie M. Henderson, Krishna V. Shenoy, L. F. Abbott, and David Sussillo. 2018. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature Methods*, 15(10):805–815.

Alexandre Payeur, Amy L. Orsborn, and Guillaume Lajoie. 2023. Neural manifolds and learning regimes in neural-interface tasks. Pages: 2023.03.11.532146 Section: New Results.

Matthew G. Perich, Juan A. Gallego, and Lee E. Miller. 2018. A Neural Population Mechanism for Rapid Learning. *Neuron*, 100(4):964–976.e7.

Andrew J. Peters, Simon X. Chen, and Takaki Komiyama. 2014. Emergence of reproducible spatiotemporal activity during motor learning. *Nature*, 510(7504):263–267. Publisher: Nature Publishing Group.

Andrew J. Peters, Jun Lee, Nathan G. Hedrick, Keelin O'Neil, and Takaki Komiyama. 2017. Reorganization of corticospinal output during motor learning. *Nature Neuroscience*, 20(8):1133–1141. Publisher: Nature Publishing Group.

Leopoldo Petreanu, Diego A. Gutnisky, Daniel Huber, Ning-long Xu, Dan H. O'Connor, Lin Tian, Loren Looger, and Karel Svoboda. 2012. Activity in motor–sensory projections reveals distributed coding in somatosensation. *Nature*, 489(7415):299–303. Publisher: Nature Publishing Group.

Mohammad Pezeshki, Oumar Kaba, Yoshua Bengio, Aaron C Courville, Doina Precup, and Guillaume Lajoie. 2021. Gradient Starvation: A Learning Proclivity in Neural Networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 1256–1272. Curran Associates, Inc.

Assaf Ramot, Felix H. Taschbach, Yun C. Yang, Yuxin Hu, Qiyu Chen, Bobbie C. Morales, Xinyi C. Wang, An Wu, Kay M. Tye, Marcus K. Benna, and Takaki Komiyama. 2025. Motor learning refines thalamic influence on motor cortex. *Nature*, 643(8072):725–734. Publisher: Nature Publishing Group.

Stefano Recanatesi, Serena Bradde, Vijay Balasubramanian, Nicholas A. Steinmetz, and Eric Shea-Brown. 2020. A scale-dependent measure of system dimensionality. Pages: 2020.12.19.423618 Section: New Results.

Stefano Recanatesi, Gabriel Koch Ocker, Michael A. Buice, and Eric Shea-Brown. 2019. Dimensionality in recurrent spiking networks: Global trends in activity and local origins in connectivity. *PLoS Computational Biology*, 15(7):e1006446.

Janine Reis, Heidi M. Schambra, Leonardo G. Cohen, Ethan R. Buch, Brita Fritsch, Eric Zarahn, Pablo A. Celnik, and John W. Krakauer. 2009. Noninvasive cortical stimulation enhances motor skill acquisition over multiple days through an effect on consolidation. *Proceedings of the National Academy of Sciences*, 106(5):1590–1595.

Blake A Richards and Timothy P Lillicrap. 2019. Dendritic solutions to the credit assignment problem. *Current Opinion in Neurobiology*, 54:28–36.

Sam T. Roweis and Lawrence K. Saul. 2000. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326.

Daniel B. Rubin, Tommy Hosman, Jessica N. Kelemen, Anastasia Kapitonava, Francis R. Willett, Brian F. Coughlin, Eric Halgren, Eyal Y. Kimchi, Ziv M. Williams, John D. Simeral, Leigh R. Hochberg, and Sydney S. Cash. 2022. Learned Motor Patterns Are Replayed in Human Motor Cortex during Sleep. *Journal of Neuroscience*, 42(25):5007–5020. Publisher: Society for Neuroscience Section: Research Articles.

Michael E. Rule, Timothy O'Leary, and Christopher D. Harvey. 2019. Causes and consequences of representational drift. *Current opinion in neurobiology*, 58:141–147.

Abigail A. Russo, Sean R. Bittner, Sean M. Perkins, Jeffrey S. Seely, Brian M. London, Antonio H. Lara, Andrew Miri, Najja J. Marshall, Adam Kohn, Thomas M. Jessell, Laurence F. Abbott, John P. Cunningham, and Mark M. Churchland. 2018. Motor cortex embeds muscle-like commands in an untangled population response. *Neuron*, 97(4):953–966.e8.

Patrick T. Sadtler, Kristin M. Quick, Matthew D. Golub, Steven M. Chase, Stephen I. Ryu, Elizabeth C. Tyler-Kabara, Byron M. Yu, and Aaron P. Batista. 2014. Neural constraints on learning. *Nature*, 512(7515):423–426. Publisher: Nature Publishing Group.

Shreya Saxena and John P Cunningham. 2019. Towards the neural population doctrine. *Current Opinion in Neurobiology*, 55:103–111.

G Schalk, K J Miller, N R Anderson, J A Wilson, M D Smyth, J G Ojemann, D W Moran, J R Wolpaw, and E C Leuthardt. 2008. Two-dimensional movement control using electrocorticographic signals in humans. *Journal of Neural Engineering*, 5(1):75.

Leo Scholl, Pavithra Rajeswaran, Lydia Smith, and Amy L. Orsborn. Measuring functional connectivity across days using optogenetic stimulation.

J. P. Scholz and G. Schöner. 1999. The uncontrolled manifold concept: identifying control variables for a functional task. *Experimental Brain Research*, 126(3):289–306.

Rachael D. Seidler, Youngbin Kwak, Brett W. Fling, and Jessica A. Bernard. 2013. Neurocognitive Mechanisms of Error-Based Motor Learning. *Advances in experimental medicine and biology*, 782:10.1007/978–1–4614–5465–6_3.

Reza Shadmehr and Ferdinando A. Mussa-Ivaldi. 1994. Adaptive representation of dynamics during learning of a motor task. *Journal of Neuroscience*, 14(5):3208–3224.

Reza Shadmehr, Maurice A. Smith, and John W. Krakauer. 2010. Error Correction, Sensory Prediction, and Adaptation in Motor Control. *Annual Review of Neuroscience*, 33(1):89–108.

Krishna V. Shenoy and Jose M. Carmena. 2014. Combining decoder design and neural adaptation in brain-machine interfaces. *Neuron*, 84(4):665–680.

Lior Shmuelof, John W. Krakauer, and Pietro Mazzoni. 2012. How is a motor skill learned? Change and invariance at the levels of task success and trajectory control. *Journal of Neurophysiology*, 108(2):578–594.

Daniel B. Silversmith, Reza Abiri, Nicholas F. Hardy, Nikhilesh Natraj, Adelyn Tu-Chan, Edward F. Chang, and Karunesh Ganguly. 2021. Plug-and-play control of a brain–computer interface through neural map stabilization. *Nature Biotechnology*, 39(3):326–335. Number: 3 Publisher: Nature Publishing Group.

Maurice A. Smith, Ali Ghazizadeh, and Reza Shadmehr. 2006. Interacting Adaptive Processes with Different Timescales Underlie Short-Term Motor Learning. *PLOS Biology*, 4(6):e179. Publisher: Public Library of Science.

Kelvin So, Karunesh Ganguly, Jessica Jimenez, Michael C. Gastpar, and Jose M. Carmena. 2012. Redundant information encoding in primary motor cortex during natural and prosthetic motor control. *Journal of Computational Neuroscience*, 32(3):555–561.

Xulu Sun, Daniel J. O'Shea, Matthew D. Golub, Eric M. Trautmann, Saurabh Vyas, Stephen I. Ryu, and Krishna V. Shenoy. 2022. Cortical preparatory activity indexes learned motor memories. *Nature*, 602(7896):274–279. Number: 7896 Publisher: Nature Publishing Group.

Dawn M. Taylor, Stephen I. Helms Tillery, and Andrew B. Schwartz. 2002. Direct cortical control of 3D neuroprosthetic devices. *Science*.

Sebastian Telgen, Darius Parvin, and Jörn Diedrichsen. 2014. Mirror Reversal and Visual Rotation Are Learned and Consolidated via Separate Mechanisms: Recalibrating or Learning De Novo? *Journal of Neuroscience*, 34(41):13768–13779.

J. B. Tenenbaum, V. de Silva, and J. C. Langford. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science (New York, N.Y.)*, 290(5500):2319–2323.

Amanda S. Therrien, Daniel M. Wolpert, and Amy J. Bastian. 2016. Effective reinforcement learning following cerebellar damage requires a balance between exploration and motor noise. *Brain: A Journal of Neurology*, 139(Pt 1):101–114.

Emanuel Todorov. 2000. Direct cortical control of muscle activation in voluntary arm movements: a model. *Nature Neuroscience*, 3(4):391–398.

Emanuel Todorov. 2006. Optimal Control Theory. In *Bayesian Brain: Probabilistic Approaches to Neural Coding*. The MIT Press.

Emanuel Todorov and Michael I. Jordan. 2002. Optimal feedback control as a theory of motor coordination. *Nature Neuroscience*, 5(11):1226–1235.

Ksenia Volkova, Mikhail A. Lebedev, Alexander Kaplan, and Alexei Ossadtchi. 2019. Decoding Movement From Electrocorticographic Activity: A Review. *Frontiers in Neuroinformatics*, 13.

Saurabh Vyas, Matthew D Golub, David Sussillo, and Krishna V Shenoy. 2020a. Computation Through Neural Population Dynamics. page 27.

Saurabh Vyas, Daniel J. O'Shea, Stephen I. Ryu, and Krishna V. Shenoy. 2020b. Causal Role of Motor Preparation during Error-Driven Learning. *Neuron*, 106(2):329–339.e4.

Ling Wang, James M. Conner, Jessica Rickert, and Mark H. Tuszynski. 2011. Structural plasticity within highly specific neuronal populations identifies a unique parcellation of motor learning in the adult brain. *Proceedings of the National Academy of Sciences*, 108(6):2545–2550.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256.

Matthew S. Willsey, Samuel R. Nason-Tomaszewski, Scott R. Ensel, Hisham Temmar, Matthew J. Mender, Joseph T. Costello, Parag G. Patil, and Cynthia A. Chestek. 2022. Real-time brain-machine interface in non-human primates achieves high-velocity prosthetic finger movements using a shallow feedforward neural network decoder. *Nature Communications*, 13(1):6899.

M. A. Wilson and B. L. McNaughton. 1994. Reactivation of hippocampal ensemble memories during sleep. *Science (New York, N.Y.)*, 265(5172):676–679.

B. Wodlinger, J. E. Downey, E. C. Tyler-Kabara, A. B. Schwartz, M. L. Boninger, and J. L. Collinger. 2015. Ten-dimensional anthropomorphic arm control in a human brain-machine interface: difficulties, solutions, and limitations. *Journal of Neural Engineering*, 12(1):016011.

Jonathan R. Wolpaw, Niels Birbaumer, Dennis J. McFarland, Gert Pfurtscheller, and Theresa M. Vaughan. 2002. Brain-computer interfaces for communication and control. *Clinical Neurophysiology: Official Journal of the International Federation of Clinical Neurophysiology*, 113(6):767–791.

Daniel M. Wolpert, Jörn Diedrichsen, and J. Randall Flanagan. 2011. Principles of sensorimotor learning. *Nature Reviews Neuroscience*, 12(12):739–751.

Daniel M. Wolpert and J. Randall Flanagan. 2001. Motor prediction. *Current Biology*, 11(18):R729–R732. Publisher: Elsevier.

Daniel M. Wolpert and Zoubin Ghahramani. 2000. Computational principles of movement neuroscience. *Nature Neuroscience*, 3(11):1212–1217. Publisher: Nature Publishing Group.

An Wu, Qiyu Chen, Bin Yu, Soyoung Chae, Zijing Tan, Assaf Ramot, and Takaki Komiyama. 2025. Targeted stimulation of motor cortex neural ensembles drives learned movements. Pages: 2025.01.06.631504 Section: New Results.

Tonghui Xu, Xinzhu Yu, Andrew J. Perlik, Willie F. Tobin, Jonathan A. Zweig, Kelly Tennant, Theresa Jones, and Yi Zuo. 2009. Rapid formation and selective stabilization of synapses for enduring motor memories. *Nature*, 462(7275):915–919.

Christopher S Yang, Noah J Cowan, and Adrian M Haith. 2021. De novo learning versus adaptation of continuous control in a manual tracking task. *eLife*, 10:e62578. Publisher: eLife Sciences Publications, Ltd.

Azadeh Yazdan-Shahmorad, Camilo Diaz-Botia, Timothy L Hanson, Viktor Kharazia, Peter Ledo-

chowitsch, Michel M Maharbiz, and Philip N Sabes. 2016. A large-scale interface for optogenetic stimulation and recording in nonhuman primates. *Neuron*, 89(5):927–939. Publisher: Elsevier.

Xiao Zhou, Rex N. Tien, Sadhana Ravikumar, and Steven M. Chase. 2019. Distinct types of neural reorganization during long-term learning. *Journal of Neurophysiology*, 121(4):1329–1341.

Ellen L. Zippi, Albert K. You, Karunesh Ganguly, and Jose M. Carmena. 2021. Selective modulation of population dynamics during neuroprosthetic skill learning. Pages: 2021.01.08.425917 Section: New Results.

Ellen L. Zippi, Albert K. You, Karunesh Ganguly, and Jose M. Carmena. 2022. Selective modulation of cortical population dynamics during neuroprosthetic skill learning. *Scientific Reports*, 12(1):15948. Publisher: Nature Publishing Group.

Chapter A

Appendix One

A.1 Supplementary information related to chapter 2

Supplementary Methods

Neural tuning curves

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 Georgopoulos et al. [1982]:

$$f = B_1 \cos\theta + B_2 \sin\theta + B_3 \tag{S1}$$

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):

$$MD = \sqrt{B_1^2 + B_2^2} \tag{S2}$$

$$PD = \text{arctan2}(B_2/B_1) \tag{S3}$$

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 $\Delta\mathbf{w}$.

Comparison of offline velocity estimates

We used the same velocity Kalman filter decoders from the experiments to estimate cursor velocities offline. We compared velocity estimates obtained using all readouts and using only designated subsets of readouts (see below and Fig. S3G-I for detail). To quantify the similarity between the estimated velocities, we calculated the following time-average matching:

$$M = 1 - \frac{1}{T}\sum_{t=1}^{T} \begin{cases} 0 & \text{if } \|\mathbf{V}_t^{(\text{all})}\| = \|\mathbf{V}_t^{(\text{subset})}\| = 0 \\ \frac{\|\mathbf{V}_t^{(\text{all})} - \mathbf{V}_t^{(\text{subset})}\|^2}{\|\mathbf{V}_t^{(\text{all})}\|^2 + \|\mathbf{V}_t^{(\text{subset})}\|^2} & \text{otherwise} \end{cases} \tag{S4}$$

where $\mathbf{V}_t^{(\text{all})}$ and $\mathbf{V}_t^{(\text{subset})}$ represent the cursor velocities estimated using all readouts and the subset, respectively. We defined the summand as zero when both velocities are zero. This ensures that $0 \leq M \leq 1$. We compared the velocities estimated using all readouts and the top N_c^{late} units—from the target-encoding analysis—on late day (Fig. S3G). To assess how the addition of units improved the accuracy of cursor velocity estimates, we computed M as we progressively added more units ranked according to the neuron adding curves obtained from the target-encoding analysis on early and late days (Fig. S3H). Finally, we compared the matching score M for the two training phases when using a number of units equal to N_c^{late} on both the early and late day (Fig. S3I).

Dimensionality estimation using Participation ratio

We computed the neural dimensionality for readouts and nonreadouts using the participation ratio (PR) Mazzucato et al. [2016]; Gao et al. [2017]; **?**. 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

$$\text{PR} = \frac{[\text{tr}(C_{\text{units}})]^2}{\text{tr}(C_{\text{units}}^2)}. \tag{S5}$$

We estimated population dimensionality for readouts and nonreadouts separately each day. 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

$$\text{PR}_{\text{norm}} = \frac{\text{PR} - 1}{n_{\text{units}} - 1}. \tag{S6}$$

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. S5A) 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 change in dimensionality between early and late learning (Fig. S5C) was estimated by $\Delta\text{PR} = \text{PR}_{\text{late}} - \text{PR}_{\text{early}}$. We computed the 95% confidence intervals for PR calculations (Fig. S5C) by re-sampling the original trials with replacement, matching target identity distributions (N = 10^4).

Dimensionality analysis for BCI learning with a fixed decoder (Fig. S5B, C) was performed on the dataset from Ganguly and Carmena [2009]. It used the same method as above except that training epochs were used instead of days Athalye et al. [2017]; Zippi et al. [2022]. 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 Athalye et al. [2017]; Zippi et al. [2022], 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.

Supplementary Tables

Subject	# of readouts (early)	# of readouts (late)	# of readout units replaced	# of weight change events	# of ensemble change events	# of nonreadouts (early - late)
J	16	16	1	4	1	36 - 36
J	16	16	4	3	1	39 - 39
J	16	16	3	4	2	41 - 41
J	16	16	2	2	1	38 - 38
J	16	15	-2 +1	2	1	37 - 38
J	16	15	-1	1	1	28 - 29
J	16	16	3	2	1	29 - 29
S	12	12	0	2	-	118 - 118
S	11	11	4	2	2	66 - 66
S	12	12	0	1	-	121 - 121

Table S1: Information about the number of readouts and nonreadouts, the number of readout units replaced during ensemble change events, the number of weight change events and the number of ensemble change events for every series analyzed in the main text. The '+' and '-' indicate the number of units newly added or removed, respectively. The '-' symbol in the ensemble change events column indicates no ensemble changes for that series. The range in the nonreadouts column denotes the total number of nonreadouts in the early and late stages.

Supplementary Discussion

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

We observed compaction 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. 3, 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 the 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 "mode space".

Impact of Neural Recording Methods

Differences in neural recording methods, such as threshold crossings versus sorted units, may also influence co-adaptation processes and the stability of neural representations. For instance, we used threshold crossings in monkey J. Threshold crossings may have different stability properties over days than sorted units, which might lead to differences across monkeys. However, isolating con-

tributions of neural recording properties is beyond the scope of what is feasible with these current data. Future studies exploring how neural recording stability, specifically, influences neural plasticity and decoder adaptation algorithms will be critical for designing BCIs that maintain performance over time.

Supplementary Figures

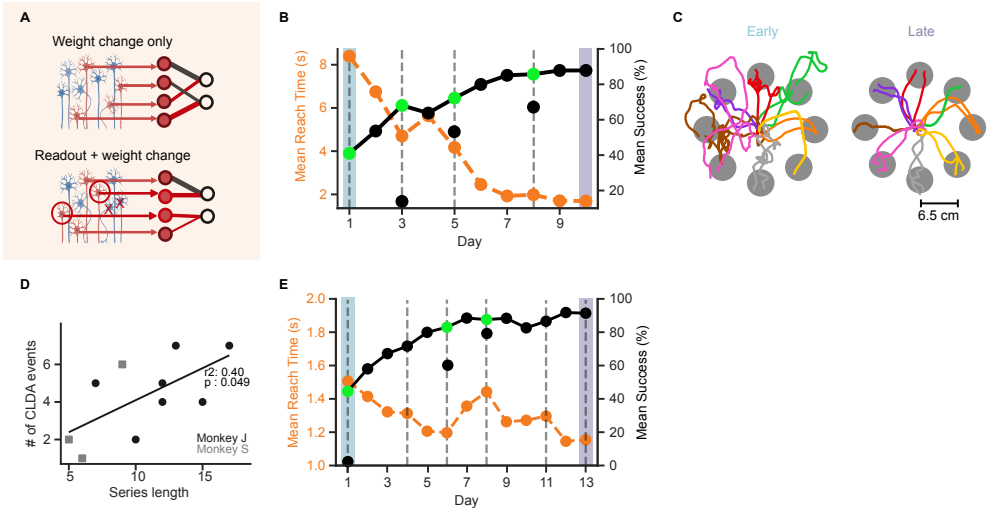

Figure S1: Supplementary figure related to figure 1: example behavior from monkey S and J (A) BCI decoder adaptation. During learning, the decoder was adapted in two ways: changing the weights relating readout units to velocity while keeping the identify of readout units constant (Weight change only), and altering both the contributing readout units and their corresponding decoder weights (Readout + weight change). **(B)** Success percentage (solid black line) and mean reach time (dashed orange line) across days for an example learning series for monkey S (seba010911_011811). GreenIndigo dots show performance boost after decoder adaptation. 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). **(C)** Cursor trajectories during early (left) and late (right) training phases for the example series in panel B. **(D)** Number of total decoder adaptation events (including both weight changes and readouts + weight changes) plotted against series length for monkey J (black circles) and monkey S (grey squares). **(E)** Same as (B) for another example series from monkey J where decoder was adapted on day 1 for better performance.

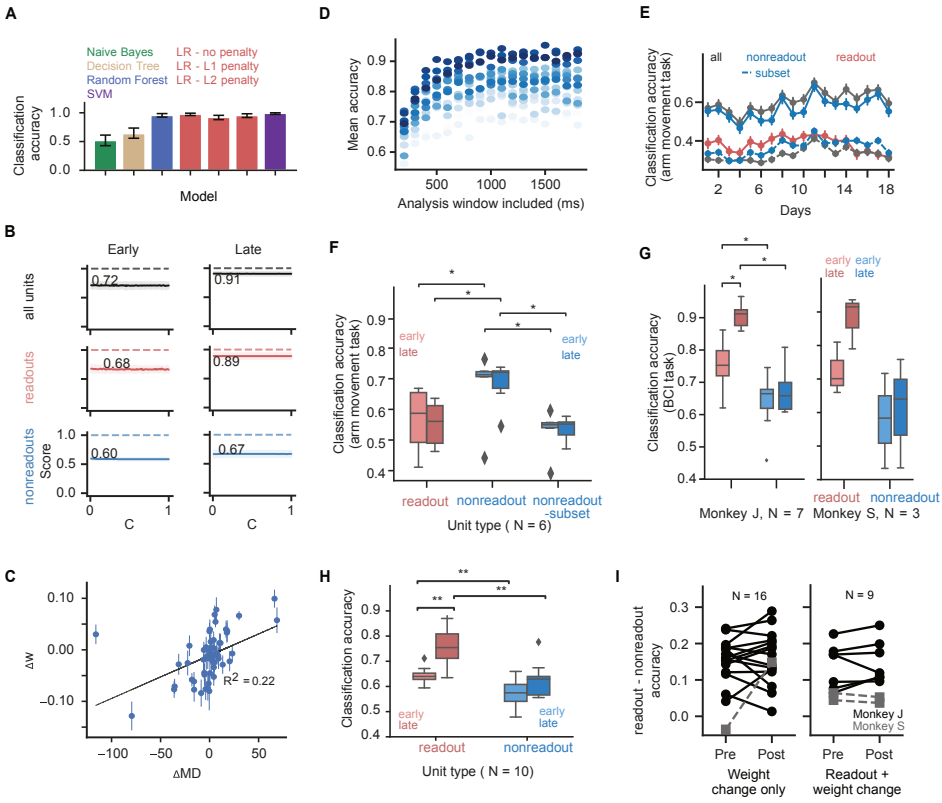

Figure S2: Supplementary figure related to figure 2: credit assignment (A) Classification accuracy obtained from different classifier models when trained on neural data from last day of our example series. Error bars are 95% confidence intervals on test accuracy. (B) Early and late classification scores for all units (black), readouts (red) and nonreadouts(blue) plotted as hyperparameter C (related to regularization strength) varies from 10^{-5} to 1. (C) Change in logistic regression model coefficients ($\Delta \mathbf{w}$) between early and late day as a function of changes in modulation depth of units from example series. (D) Mean classification accuracy (25 train-test splits) vs. analysis window length included after the go cue. Each dot is one day in a learning series. blue shading indicates early (light blue) to late (dark blue) days. (ED) Classification analysis from all units (black), readouts (red) and nonreadouts (blue) for arm movement task for the example 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. (FE) 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 nonreadout-late subset: $^{*}p = 0.031$, readout-early vs nonreadout-subset early: $p = 0.2$ (n.s); readout-late vs nonreadout-subset late: $p = 0.8$ (n.s), $n = 6$, Wilcoxon signed-rank test). (GF) 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: $p = 0.3$ (n.s); readout-early vs. readout-late: $^{*}p = 0.016$; readout-late vs. nonreadout-late: $^{*}p = 0.016$; readout-early vs. nonreadout-early: $^{*}p = 0.016$. 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. (HG) Early vs late classification accuracy for readout (shades of red) and nonreadout populations (shades of blue) from BCI task using neural activity from the go cue to 200 ms after go cue. Nonreadout-early vs. nonreadout-late: $p = 0.1$ (n.s); readout-early vs. readout-late: $^{**}p = 0.008$; readout-late vs. nonreadout-late: $^{**}p = 0.004$; readout-early vs. nonreadout-early: $^{**}p = 0.008$. Two-sided Wilcoxon signed-rank test, $n = 10$. (IH) Readout-nonreadout classification accuracy, grouped by the type of change (with (right) and without (left) readout unit changes) across all series for monkey J (black circles) and monkey S (grey squares). Weight change only: $p = 0.3$ (ns), $n = 16$; readout + weight change: $p = 0.25$ (ns), $n = 9$. Pre < post, one-sided Wilcoxon signed-rank test.

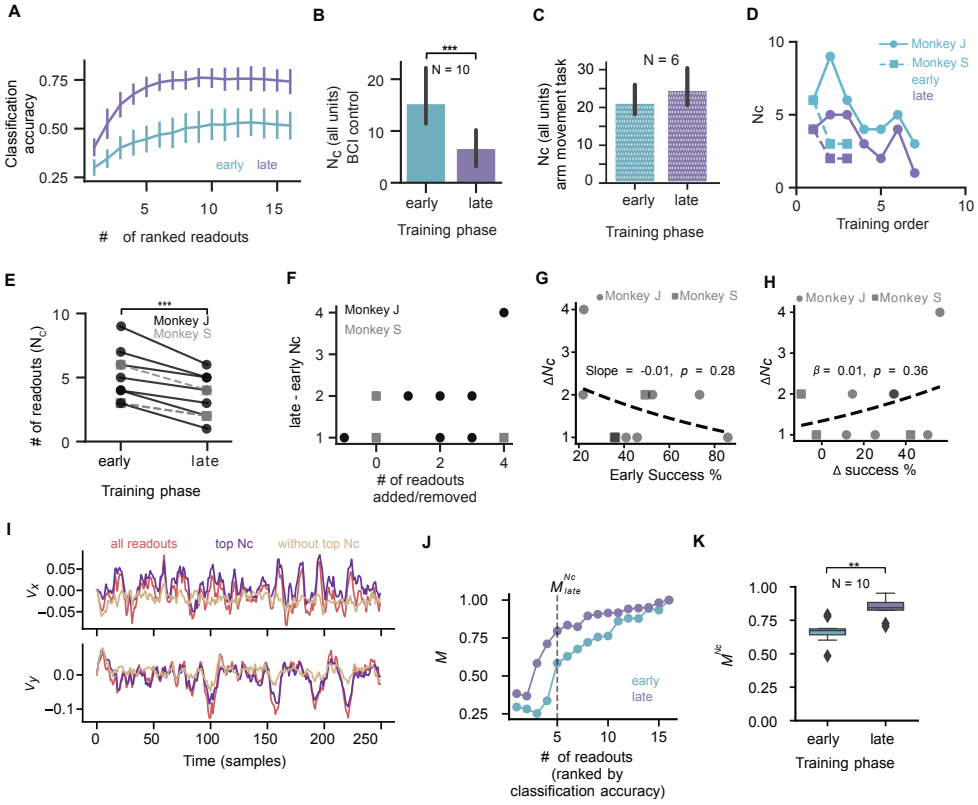

Figure S3: Supplementary figure related to figure 3: compaction (A) Classification accuracy (not normalized) as units are added in ranked order (ranked NAC) on early (cyan) vs late (purple) day for the representative series (Unnormalized version of Fig 3F). (B) Comparison of N_c^{late} vs N_c^{early} across series while using all units (N_c^{late} is less than N_c^{early}, $^{***}p = 9.8 \times 10^{-4}$, $n = 10$, Wilcoxon signed-rank test). (C) Similar to (B) for arm movement task ($p = 0.07(n.s)$, $n = 6$, Wilcoxon signed-rank test). (D) $N_c{}^{\text{early}}$ for readouts early (cyan) and late (purple) days across chronologically-ordered series for monkey J (solid lines, round markers) and monkey S (dashed lines, square markers). (E) 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 (black circles); Monkey S (grey squares).. (F) Late−early N_c vs. number of readouts added or removed in a learning series ($R^2 = 0.36$, $p = 0.3$ (n.s). (G) ΔN_c vs. early success % for monkey J (circle) and monkey S (square) ($p = 0.28$, Poisson regression and two-sided permutation test). (H) ΔN_c vs. change in success % between late and early day for monkey J (circle) and monkey S (square) ($p = 0.36$, Poisson regression and two-sided permutation test) (IG) X and Y cursor velocities reconstructed offline using the Kalman filter decoder with all readouts (red) ,and the top N_c^{late} readout units (purple) identified by target classification analysis and readout units except top N_c^{late} (tan) for the representative series. (JH) Matching (M) between the velocities reconstructed using all units vs. as units are added in ranked order (ranked NAC) on early (cyan) vs late (purple) day for the representative series. (KI) Velocity matching (M^{N_c}) when using top Nc units on early vs late day ($^{**}p = 0.0019$, $n = 10$, Wilcoxon signed-rank test).

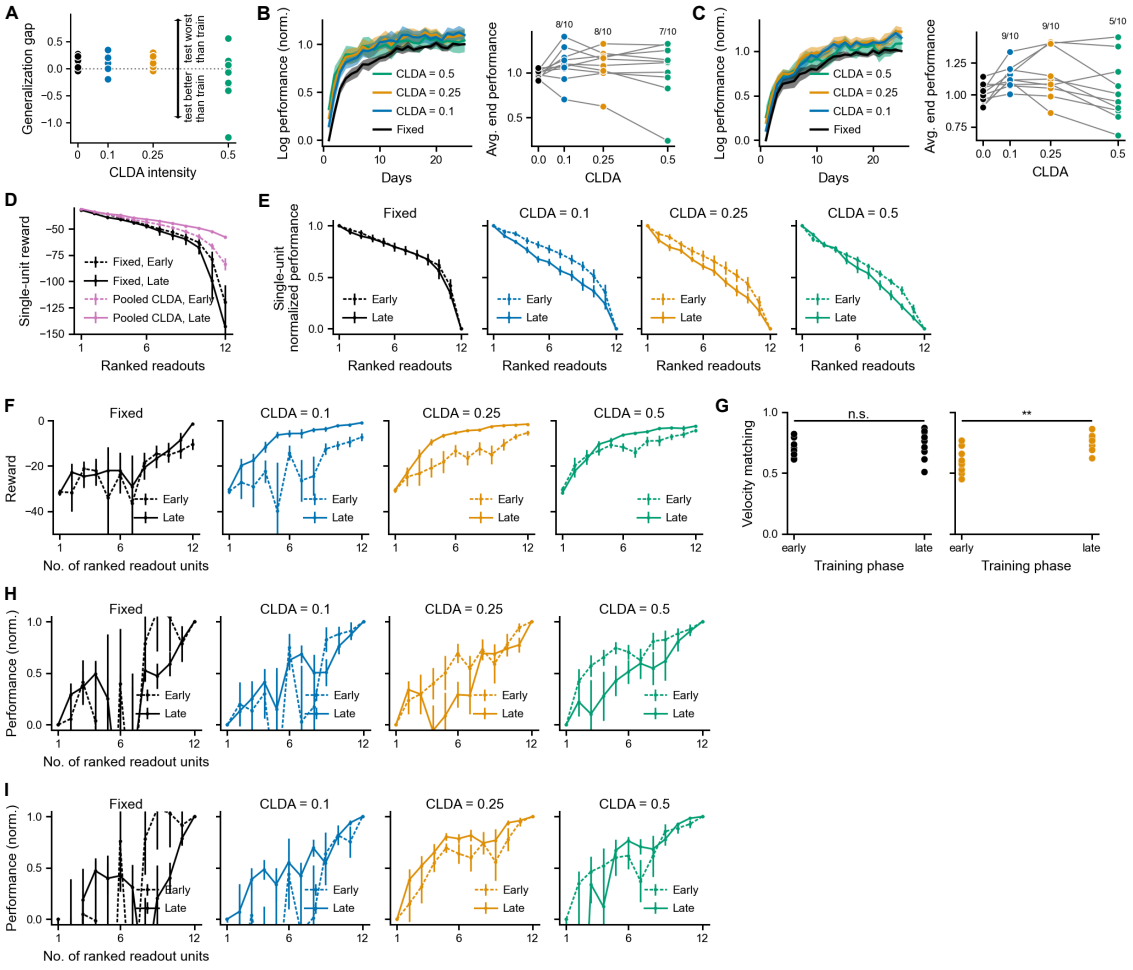

Figure S4: Supplementary figure related to Fig. 4: model. (A) Generalization gap is performance on training set minus performance on test set. Gap for the adaptive decoder is not significantly different from that for the fixed decoder ($p > 0.1$, $n = 10$, Wilcoxon signed-rank test, CLDA $>$ fixed). Variance for CLDA is also not larger than variance for the fixed decoder ($p > 0.1$, $n = 10$, paired permutation test for variance(CLDA> 0) > variance(fixed)) (B) Training performance when stopping CLDA after a loss criterion was reached (i.e., when loss < 2). (C) Training performance with CLDA every 5 days. (D-E) Unnormalized pooled (C) and normalized un-pooled (D) single-unit performances using the reward which is the negative of the loss. (F) Unnormalized NACs. (G) Matching between the online velocity inferred using the top 6 best units—according to the loss-based performance metric (see Fig. 4F)—and using all readout units. Left: fixed decoder. Right: adaptive decoder with CLDA = 0.25, which produced the most salient compaction according to Fig. 4F ($p = 0.002$ (**), $n = 10$, Wilcoxon signed-rank test). (H) Compaction results when stopping CLDA. (I) Same as H, with CLDA every 5 days.

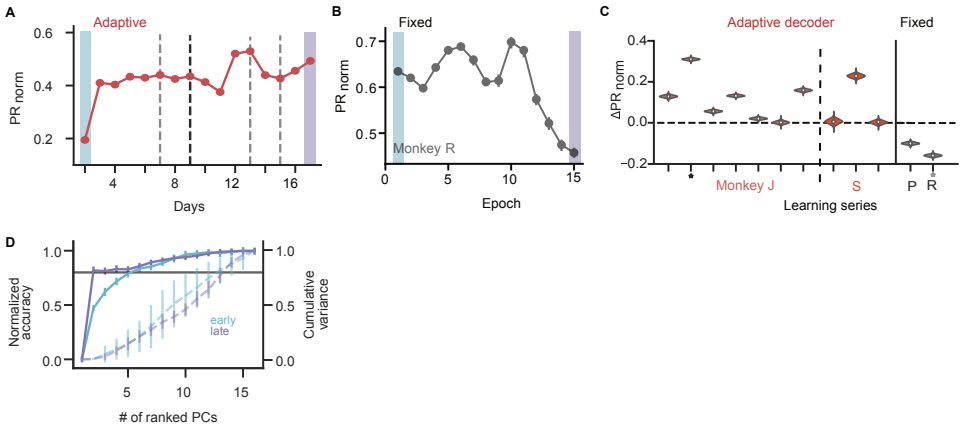

Figure S5: Supplementary figure related to Fig. 5: Population analysis (A) Normalized participation ratio (PR_{norm}) for an example adaptive decoder learning series (same as in Fig. 1D). Error bars are 95% confidence intervals estimated via bootstrapping (see Methods). Note that most error bars are hidden by data point markers. (B) Same as (A) for fixed decoders (monkey R; data from Ganguly and Carmena [2009]). Note: the x-axis denotes epoch, which are groups of constant number of trials, consistent with previous analyses Athalye et al. [2017]. (C) ΔPR_{norm} distributions for learning series with adaptive decoders ($n = 7$ for monkey J and $n = 3$ for monkey S). Fixed data is from two monkeys (P and R) using fixed decoders (data from Ganguly and Carmena [2009]). * denotes the example series shown in panels A and B. (D) Ranked PC adding curve (same as Fig. 5F) and cumulative explained variance of ranked PCs for early (blue) and late (purple) days for selected series from adaptive decoder dataset.