The Temporal Structure of Neural Population Activity in Motor Cortex

by

Erinn Michelle Philbert Grigsby

M.S. in Biomedical Engineering, Duke University, 2015

B.S.E in Biomedical Engineering, Duke University, 2013

B.S.E in Electrical and Computer Engineering, Duke University, 2013

Submitted to the Graduate Faculty of

the Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2021

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This dissertation was presented

by

Erinn Michelle Philbert Grigsby

It was defended on

August 25, 2021

and approved by

Aaron Batista, Ph. D, Professor, Department of Bioengineering

Byron Yu, Ph. D, Professor, Departments of Electrical and Computer Engineering and Biomedical Engineering, Carnegie Mellon University

Jennifer Collinger, Ph. D, Associate Professor, Department of Physical Medicine and Rehabilitation

Neeraj Gandhi Ph. D., Graduate Program Director, Department of Bioengineering

Patrick Loughlin, Ph. D, William Kepler Whiteford Professor of Bioengineering, Department of Bioengineering

Gelsy Torres-Oviedo, Ph. D, Assistant Professor, Department of Bioengineering

Dissertation Advisors: Aaron Batista, Ph. D, Professor, Department of Bioengineering, Byron Yu, Ph. D, Professor, Departments of Electrical and Computer Engineering and Biomedical Engineering, Carnegie Mellon University

The Temporal Structure of Neural Population Activity in Motor Cortex

Erinn Michelle Philbert Grigsby, PhD

University of Pittsburgh, 2021

We are able to make robust movements, like walking and reaching, without a thought. Yet despite our ease with such coordinated movements, the underlying neural processes are subtle and intricate. A hallmark of neural information processing in the cerebral cortex is the evolution of neural activity over time. It has been proposed that temporal structure in neural activity is the result of network connectivity. If so, modification of this structure would require modifying network connectivity, which is difficult to do over short timescales. This work examines the temporal structure, or dynamics, of population neural activity to test a key hypothesis: that dynamical structure is inherent to the circuitry of motor cortex. We leveraged population analysis methods and brain-computer interface (BCI) paradigms to causally probe neural activity in the motor cortex. Rhesus monkeys performed BCI tasks in which their recorded neural activity controlled the position of a computer cursor. We established two aspects of motor cortex dynamics. First, we examined whether the characteristic time courses of neural activity persist in the motor cortex in the absence of overt movement. This would suggest that there are dynamical properties that emerge from the neural circuitry in the motor cortex rather than only being present when the muscles are being controlled. We found that dynamics did persist in the absence of movements. Second, we examined whether the animals could violate these dynamics if we gave them incentive to do so. We found that even under this pressure to change their behavior, these dynamics still persisted. These results imply that the underlying network imposes strong constraints on the time course of population activity. This supports the view that neural trajectories reflect underlying network mechanisms. By understanding these mechanisms we may gain insight into neural control of movement. Such insight may be valuable in the efforts to engineer improvements to movement following cortical or spinal injury.

Table of Contents

List of Tables

List of Figures

Preface

Acknowledgements: The road to a dissertation is an odd and often daunting process, where one constantly faces contradictory expectations. For my own part, I started strong with the contradictions. I simultaneously knew exactly what I wanted to do and also had absolutely no idea. The questions are too numerous and every question leads to more questions. There is a lot of stumbling in the dark. A sense of total control and also no control. The work is fast and constant, but the progress is slow and often feels stalled. The smallest revelations make big impacts. I developed a shocking amount of disdain for a curved virtual tube. There is a strong correlation between a trainer's mood and how well a monkey worked. I realized that Bill Wither's "Lovely Day" is the best song to listen to when working through frustration. Finally, I found that a great discussion with colleagues about science can leave me exhilarated for days. The process demands personal change and growth , which often goes unnoticed. The change is often only momentarily observed, like a shadow in the corner of your eye. This leads to a strange sense of feeling stuck, considering the same questions for years. It is only once I started the final process of collecting my thoughts that I truly saw the personal growth and accomplishment. While this journey has ended in the presentation of a single doctorate, the accomplishments presented here were not, indeed could not, have been achieved in isolation. Rather, I have been fortunate to have a community, both personal and professional, that has "hiked" this path with me.

First and foremost, I would like to thank my advisors, Aaron Batista and Byron Yu. Through their collaborative leadership and constant support, I have become a better scientist. Aaron gave me the space to explore and consider my various questions. He did not dissuade me from my riskier, more challenging questions. Rather he saw the "diamonds in the rough" and helped to hone and polish them. Byron complemented this exploration by encouraging me to think deeply about my questions, identifying the steps and information necessary to answer them. This attention to detail ensured that the work was thorough and well-considered, even when the initial experimental path was unknown. Beyond the technical skills of research, Aaron and Byron both took time to discuss with me common obstacles

in academia and the soft skills needed to overcome them. Thank you to my committee, Jen Collinger, Raj Gandhi, Pat Loughlin, and Gelsy Torres-Oviedo. Your excitement and thoughtfulness at our meetings always ensured we would have meaningful discussions.

For my lab mates and colleagues, I am grateful for all your feedback over the years and open informal science discussions. I would especially like to thank my main collaborators Alan Degenhart, Asma M, and Emily Oby. Alan is my co-first author on the work described in this dissertation and I have worked closely with him throughout my Ph.D. Alan is precise and is able to constantly reassess and mitigate limitations in his own work. While we came to this project with different skill sets and even- at times- different objectives, it is these differences that made the work and us better. Asma is a compassionate scientist whose enthusiasm for science and collaboration is infectious. She has been a wonderful friend and collaborator over the past few years and I will miss the frequency of our brainstorming discussion. Emily is a researcher who inspires me to continue to be a better researcher. Beyond being a monkey whisperer and excellent experimentalist, she is disciplined, resilient, and patient with humble wisdom that makes her a natural leader. She is engaged with every project in the Batista lab and every project is better because of her involvement. If I can be even half as good as Emily in my career, I will feel fortunate and proud.

My "lab spouse" Nicole, who is passionate and a computer wizard, has been my rock. Thank you for all the late lab nights of Indian food and excessive amounts of coffee and for letting me be the Uncle Buck to the LabView and rig systems. Special shoutout to Jessi, who was the best lab mom/friend I could ask for. She not only jumped all in with me on experimental and analysis projects, but she also "adopted" me into her friend group, the winers. Finally, thank you to Nick, Jamie, and their dog Red. They were always there for me their optimism, intelligence, silliness, and empathy grounded me in the difficult and good times. I loved our Burns Nights with awful Scottish accents, our incredibly long hikes and adventures, shared interests of dogs, ice cream, West Wing, Outlander, and games of Set. Thank you for everything.

To the rest of the Batista lab group, both current (Patrick, Emilio, Sam, and Raeed) and past (Sudarshan, Berook, Grant, DT, and Evan), thank you. I look forward to your future endeavors. The Yu group introduced me to a lot of theoretical topics and Pittsburgh restau-

rants I never knew about. The wonderful CNBC, Pitt BMES, and Pitt BioE communities helped me grow as a leader outside the scope of research.

My friends both near and far kept me sane. The winers (Anne, Erika, Jessi, Julie, Liza, Stephanie, and Yolandi) have been my family away from home. Our shared love of good food, brunch, dogs, silly memes, most of Game of Thrones, Lord of the Rings, good wine, and of course good company has brought so much joy to my life. Michelle was always ready for a car talk with Chick-fil-a or jamming to either Tenacious D's "Tribute" or Shaggy's "I Need Your Love". Maria was my cooking/baking partner and co-conspirator in making prelims and the graduate experience better. Monica, who was my brilliant dissertation writing buddy that helped me cross the finish line. To all my other friends, Shaun, Swathi, Heather, John, Dylan, Audrey, Sally, Jess, and Corey who were always open for a night of games, puzzles, and food. To my old Duke friends, Hillary, Stephen, Lauren, Steph, DeDe, the MaStars , and Stephen whom I could always call or text to celebrate each others' accomplishments, big or small.

To my parents, who were always available to talk and support me through every twist and turn of my Ph.D., even when that meant reading over grant proposals, seeing each other less, or even virtual Thanksgivings. I love you guys and thank you for believing in me. Thanks to my brother, Shane, for always being up for a game of chess and chat. To the two most ridiculous and wonderful dogs, Krystal and Riley, you both gave me a reason to go home. Krystal provided comfort and calm at the start of my Ph.D. as well as an excellent rug impression. Riley, the derpiest and silently sassy/opinionated adventure partner I could have ever gotten.

Finally, I would like to thank John, my partner, throughout this process. It is not easy to do long distance, even more so when there is an undetermined end date. Our schedules were frequently out of sync and there never was enough time when we did see each other. Despite this, you were always there for me when I needed you and we figured out small ways to show each other how we were on each other's mind (food, muppet videos, etc.) Long-distance sucks, but you are and will always be worth it. I love you. Yip-Yip.

I am incredibly proud of the work I have done during my Ph.D. But I am even prouder of the communities I am part of and the people that make them up. I hope these acknowl-

edgments begin to show my appreciation, even though they are insufficient to show the true depth of my gratitude. But the attempt was necessary. Thank you all. I look forward to the future roads we will travel together.

1.0 Introduction

We experience the world through movement. We play tennis, dance, talk with our hands, drink cups of coffee, and will even pick up a lucky penny on the street. Despite our ease performing these daily actions, they are intricate and complex, involving the processing of both sensory and motor information. Consider the lucky penny example. Because the coin is small, visual feedback and proprioceptive feedback are necessary to provide a sense of where you are in space relative to the penny. Picking up the penny will require modulating the grasp size, grasp force, reach extension and even how much to stoop down. Most importantly, we must consider how all of these components will affect each other as the movement evolves through time. This level of intricate coordination suggests that picking up the penny would be an arduous act. In reality, it is an action quickly executed while walking down the street. How is our brain able to process this information to produce such robust and consistent movements?

We are often only aware of the complexity of our movements when we either partially or fully lose them. Loss of proprioception results in a loss of body awareness causing drifting reaching behavior [1, 2]. Patients with cortical injuries, like stroke, experience a range of impairments including muscle weakness, loss of finger dexterity, reduced grasp and force regulation, difficulty initiating movement, tremor, and partial to full paralysis [3]. By understanding how the brain coordinated motor control, we could possibly improve movement impairments and the quality of life.

1.1 Computation through Dynamics

Most functions of the nervous system, such as generating movements, making decisions, and recalling a memory, evolve over time. These cognitive functions are presumably implemented by patterns of neural activity that themselves unfold over time. Indeed, many studies have shown neural activity exhibiting reliable temporal structure. The structure becomes

more evident when considering populations of neurons [4–25]. The time course of neural activity may reflect computation within the neural circuit. Decades of theoretical work with artificial networks support this view of "computation through dynamics" [10, 26–32].

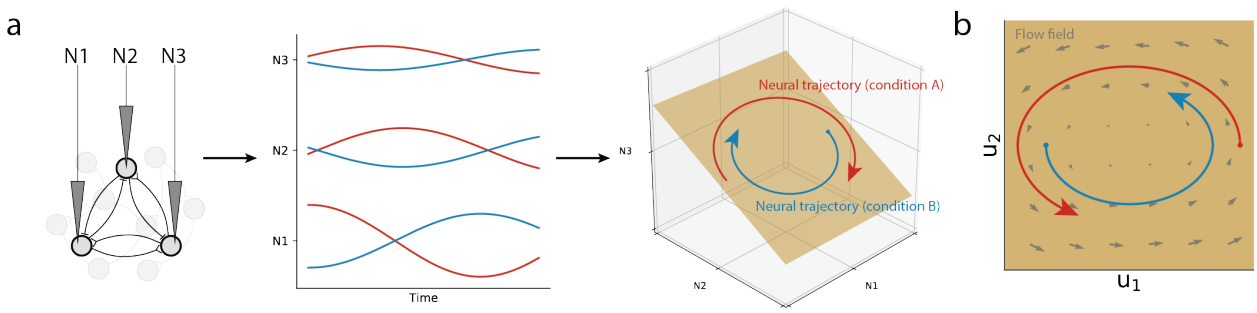

Figure 1.1: Conceptual illustration of neural trajectories and flow fields. *a)* Three hypothetical neurons (*left panel*) are recorded through time for two different conditions, A (red traces) and B (blue traces). While each neuron has its own signal in time (*central panel*), the activity can be describe within a space that incorporates all neural information. *Right panel* shows the neural trajectories plotted in a three dimensional space where each axis represents the firing rate of a single unit. This space that captures the neural activity is referred to as the neural space. *b)* A 2-D projection of the neural space that highlights a potential dynamic model or "flow field" that is driving the neural activity. The flow field is plotted as the gray vector field.

A key aspect of computation through dynamics is that the time course of neural activity can be described as a dynamical system [4, 5, 33]. This view posits that the time-varying activity of individual neurons can be described as low-dimensional neural trajectories (Fig 1.1a), and that these trajectories evolve according to an underlying dynamics model (Fig 1.1a). This dynamics model defines a neural "flow field", which describes the change in neural activity as a function of the neural state. In such a framework, differences in neural trajectories for different behavioral conditions are the result of neural activity evolving according to a fixed flow field from different initial conditions.

While this idea has provided a powerful conceptual framework with which to view neural computation, empirical evidence for such constructs is limited. While many studies have

observed temporal structure, and linked it to processes as varied as decision making [9], movement generation [4, 8, 13, 34], timing [25, 31, 35–37], and more [38, 39], causal tests of neural population dynamics have been performed only rarely [13, 36, 40, 41]. In order to test whether neural activity is in fact governed by flow field-like dynamics, some form of causal manipulation is required to directly manipulate the neural state.

Here, we use a brain-computer interface (BCI) to study the primary motor cortex (M1), an area with known robust neural population dynamics [ref]. We use the term "dynamics" to refer to fast-timescale (on the order of tens of milliseconds) changes in neural activity that are organized and consistent at the level of populations of neurons. BCI provides us with a powerful tool to causally interrogate neural population dynamics.

1.2 Brain Computer Interface to Probe Neural Activity

Over the last several years there has been increased study into the use of the brain computer interface, or BCI, as a potential tool for helping patients with lost or damaged motor control. Both primate and human subjects have successfully used BCI to control computer cursors, virtual keyboards, and robotic limbs [42–45]. In these studies, multi-electrode array recordings in the primary motor cortex (M1) were used to provide command signals for a BCI. Primate groups noticed that by changing the mapping from the neural activity to the movement of an onscreen cursor, they could induce learning in their animals [46–48]. With a BCI framework animals are provided direct feedback about the patterns of neural activity they are generating to move the cursor. This enables experimenters to request of the animals that they generate new neural activity patterns that have specific properties. Thus experimenters are able to examine the neural basis of skill learning, neuroplasticity, and motor mechanisms through population changes [46–51].

Using a BCI, animals can be provided with direct feedback of their neural state. By asking animals to generate specific time courses of neural activity using the BCI, it is possible to test whether neural population activities are consistent with the presence of a flow field.

1.3 Research Objective and Outline

The goal of this thesis research was to demonstrate that the time course of population neural activity is a reflection of local network in addition to previously identified input constraints. In this work, we paired population analysis methods with novel BCI paradigms to causally probe the neural activity.

To begin, Chapter 2 describes the general methods used across experiments. In Chapter 3, we use a standard BCI task to determine whether characteristic time courses of M1 neural activity exist in the absence of overt movement. We predicted that the neural activity of the two conditions would take different paths when moving through the neural states. This would support the hypothesis that neural trajectories are a reflection of underlying neural mechanisms. This experiment also provided a baseline comparison for the causal experiments. Chapter 4 explores how the characteristic time course is encoded in sorted single unit activity and multi-unit neural activity.

In Chapter 5, we asked whether it was possible to generate neural trajectories inconsistent with naturally-observed time courses. We designed BCI mappings that allow animals to directly visualize their neural population dynamics. We then tested for trajectories changes in task conditions where we 1) did challenge or 2) did not challenge the animals to modify their behavior. We predicted that the network constraints would be too large for the animal to violate them within a single session.

In the final chapter (6), presents a summary of results and a discussion of future experiments and interpretations. The appendices are white pages of projects I did to facilitate the research throughout the lab.

This research addresses how the activity of neurons in the motor cortex evolves through time, or neural dynamics. By understanding these underlying dynamics, we may gain insight into neural control of movement and the role of local M1 networks. These results could lead to additional insight into the relationship between inputs and cortical connectivity, motor skill learning, and even circuitry within other cortical regions. Furthermore, this study may have valuable implications for designing clinical BCIs and rehabilitation paradigms following cortical injury like stroke. BCI design could take advantage of the brain's temporal structure,

resulting in greater control and ease of use for individuals with paralysis or amputations. For clinical rehabilitation, it will provide evidence of how cortical structure can facilitate or hinder motor control. This work strengthens the argument that population dynamics are key to the computations performed by the cerebral cortex.

2.0 General Methods

This chapter will primarily cover the common method information shared between the experiments, i.e. standard electrophysiology methods used within the Batista lab. We collected neural and behavioral data from three rhesus macaques: Monkey E (age 12, session - 98), Monkey Q (age 6.4, session - 51), and Monkey D (age- 12 , session - 78). All animal handling procedures were approved by the University of Pittsburgh Institutional Animal Care and Use Committee and were consistent with the National Institutes of Health Guide for the Care and Use of Laboratory Animals.

2.1 Electrophysiology Data

Prior to the array implant surgeries, all three animals were sufficiently trained on a center-out hold reaching task. Monkey E and monkey Q were also trained on a memory-guided reach task and an instructed path task (i.e. tubes task. See single inflection tubes in [52]). The arrays were implanted in the cortex contralateral to the animals' trained reaching arm. Monkeys E and D were implanted with a 96-channel multielectrode arrays (Blackrock Microsystem, Salt Lake City, UT), while monkey Q was implanted with a 128-channel dual array. The 96-channel multielectrode arrays were $4x4\text{mm}^2$ with one hundred 1.5mm long electrodes organized in an evenly spaced grid. The 128-channel array was two $3.2x3.2\text{mm}^2$ arrays each with 64 evenly spaced electrodes with lengths of either 1 mm or 1.5 mm. For each electrode on an array, the recorded neural activity was comprised of small clusters of neurons surrounding the electrode tip (i.e. *neural units*) and occasional well-isolated single neurons.

For monkey E, the 96-channel array was placed so that it straddled the proximal arm region of the primary motor cortex (M1) and the dorsal premotor cortex (PMd). We slightly skewed the array so that about two-thirds was placed in M1 (Fig. 2.1A). We characterized the array and found that the array exhibited planning activity and a portion that exhibited

movement-related activity. The preferred directions and modulation depths for reaching were found to be well-distributed (Fig 2.2). We were able to use 93 channels for BCI control on average (Table 2.1).

For monkey D, the array was implanted in the proximal arm region of M1 (Fig. 2.1B). Monkey D was never trained on any delay-reach task, so we were only able to characterize the perimovement activity of the array. There were more noisy and low amplitude channels for monkey D so we used fewer channels for BCI control (Table 2.1).

Finally, for monkey Q, the 1.5 mm electrode was placed in the proximal arm region of M1 and the 1 mm electrode was place in PMd (Fig 2.1C). For the PMd, we had excellent isolation of single units with some tuning for both planning and perimovement activity. (Fig 2.4,*left column*). Figure 2.4 shows the preferred direction and modulation depth for the M1 array. Unfortunately, the M1 array had much lower amplitude signals than the PMd array. Because of this we excluded all channels with thresholds lower than 30uV ($\sim$ 26 channels) and used both the M1 and the PMd array for BCI control.

Table 2.1: Summary of array information for monkeys E, Q, and D. The table lists the average(standard deviation) number of channels used for BCI, The amplitude of the channels and the number channels tuned for delay and perimovement activity. Values were calculating using all sessions where a BCI decoder was implemented.

	Channels	Delay activity chn	Movement chn
Monkey E	94.0(1.12)	69	93
Monkey Q - M1	96.7(0.48)	40	22
Monkey Q -PMd	96.7(0.48)	29	21
Monkey D	79.9(1.55)	n/a	77

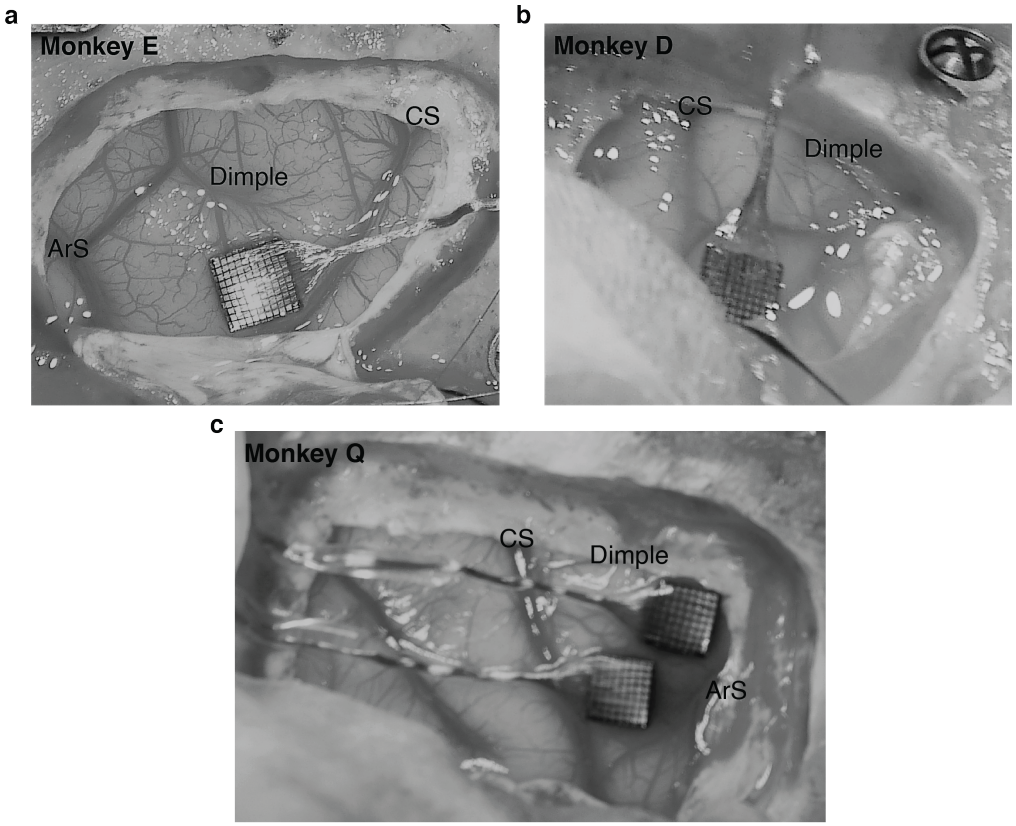

Figure 2.1: Surgical images of array placements. The central sulcus (CS), arcuate sulcus (ArC), and dimple are label when visible. *a)* Monkey E's array placement. *b)* Monkey D's M1 array placement. *c)* Monkey Q's M1 and PMd array placements.

2.2 Recording Setup

2.2.1 Neural Recordings

The neural signals from the electrodes were amplified and digitally processed using the TDT RZ2 system (Tucker Davis Technologies, Alachua, FL, USA). The digitized signals were then recorded to a computer which filtered each channel with a bandpass filter between 300Hz and 3 kHz. Spike activity was determined by setting a voltage threshold of 3 standard deviations below the mean for each channel. All threshold crossings were saved as event waveforms. The Batista lab has previously demonstrated that using threshold crossing events

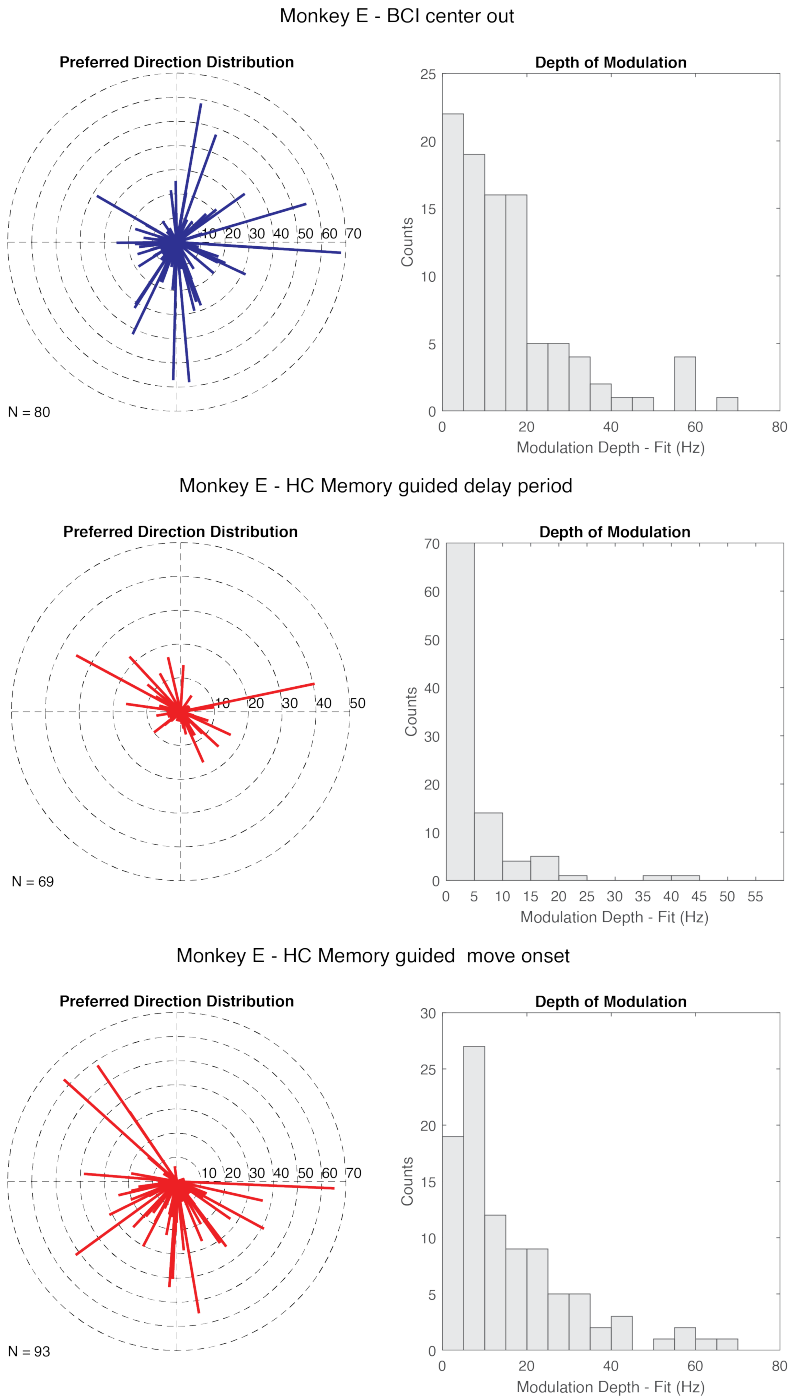

Figure 2.2: Preferred directions (PD) and modulation depth (MD) for monkey E. *top row)* BCI center out data. PD and MD were calculated over the first 400 ms after the target appeared. *middle row)* PD and MD for the last 150 ms of the delay period. *top row)* HC memory guided reach, PD and MD were calculated over the first 400 ms after the go cue.

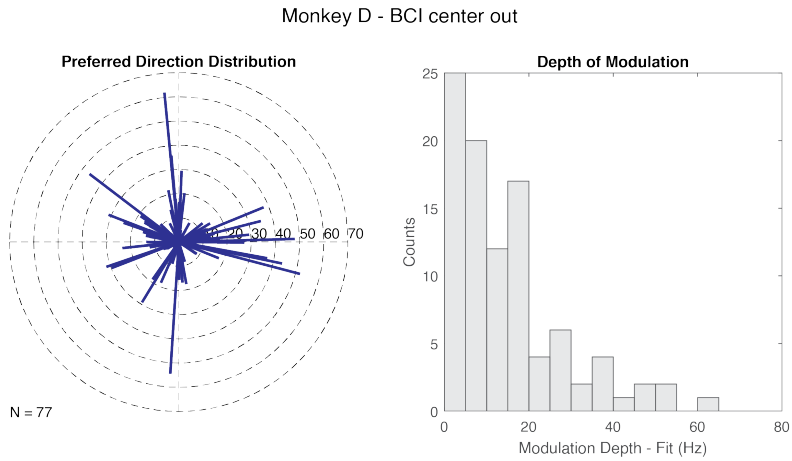

Figure 2.3: Preferred directions (PD) and modulation depth (MD) for monkey D during the BCI center out task.

is effective for BCI control [53]. Although we did not analyzed the raw voltage, i.e. broadband data, we did save some broadband data for every session. For monkey E we saved 10 channels of broadband data each day and for monkeys D and Q we saved all channels.

At the beginning of each experimental session, electrode thresholds were calculated while the animal sat calmly in a darkened room. Each electrode's threshold crossings were examined prior to the experiment. An electrode was excluded for use in the BCI if it either exhibited non-neural waveforms or was shorted to another electrode. See the first column of table 2.1 for average number of electrodes used for each experiment. The data were recorded approximately 8–18 months after array implantation for monkey E, approximately 9-18 months after array implantation for monkey D, and 3-6 months after implant for monkey Q.

2.2.2 Behavior Monitoring

During BCI experiments, animals sat head-fixed in a primate chair in front of a visual display with both arms loosely restrained. To reduce hand movements during the BCI

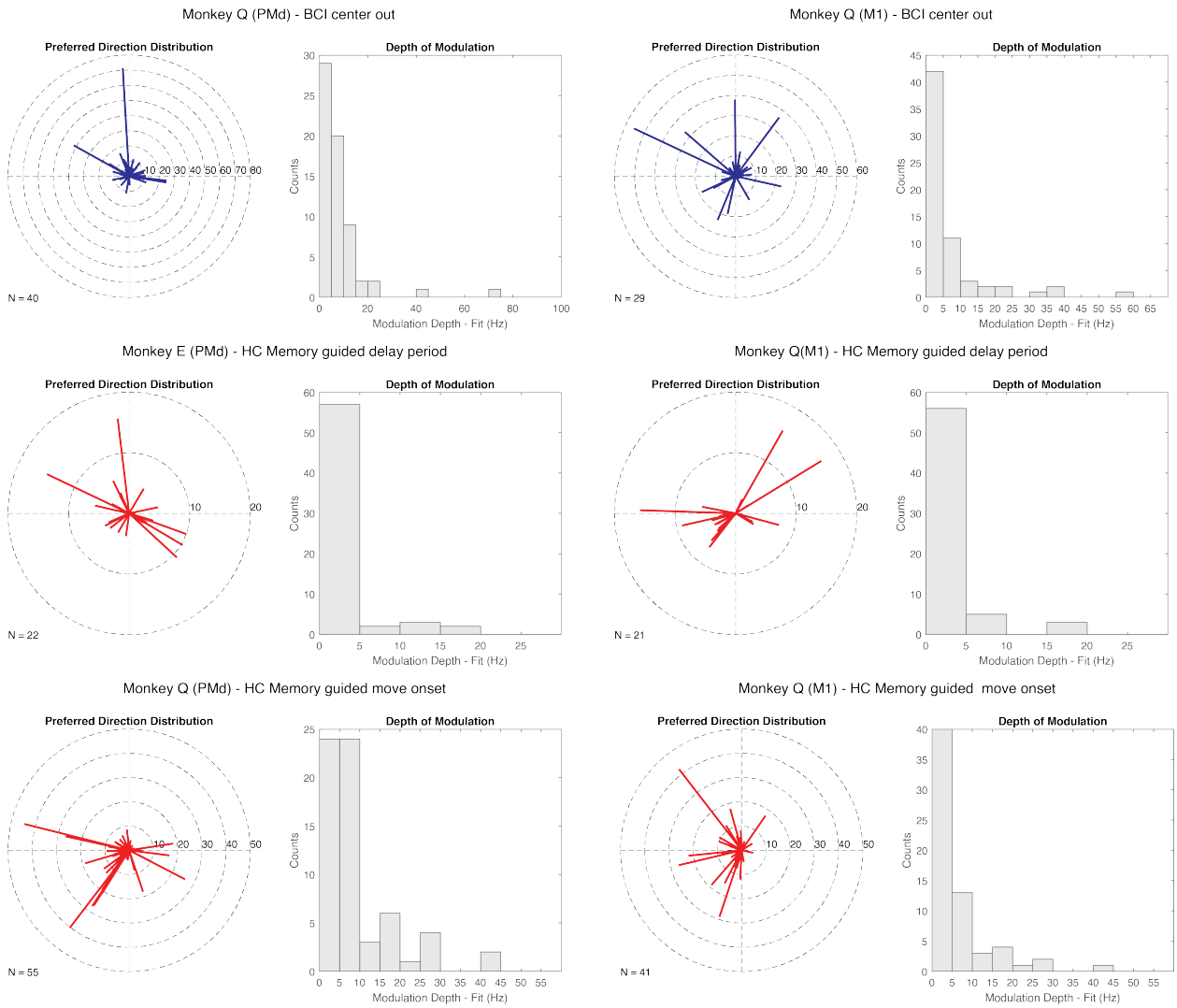

Figure 2.4: Preferred directions (PD) and modulation depth (MD) for monkey Q. The PMd data is plotted in the left column and M1 in the right. *top row)* BCI center out data. PD and MD were calculated over the first 400 ms after the target appeared. *middle row)* PD and MD for the last 150 ms of the delay period. *top row)* HC memory guided reach, PD and MD were calculated over the first 400 ms after the go cue.

trials and ensure consistent hand position, a pressure touch bar was placed under the hand contralateral to the recording array and an additional wrist restraint was used. The animals were required to maintain a hold of this bar throughout the trials and releasing the bar automatically failed the trial. Additionally, the forces applied to the bar were recorded in over half the sessions and were found to be minimal. Figure 2.5 shows the average forces applied during a session. An LED marker (PhaseSpace Inc.) was also placed on the contralateral hand to monitor movement. The sampling rate of the phasespace data was 120 Hz for Monkey Q and 480 Hz for Monkeys E and D. The animals' gaze direction was also recorded (SR Research Ltd), but not analyzed.

2.3 Behavioral Tasks

For all BCI tasks, prior to the presentation of the targets and cursor, the animals were presented with visual feedback of the state of the touchbar. In order to initiate the task, the animals had to maintain a stable 250 ms hold of the bar. In the case of the contact sensor (monkey E), this meant the animal was required to simply contact the top of the touch bar. In the case of the force transducer (monkeys E, D, and Q), touch bar contact was defined as exerting a light force downward on the bar. Animals were required to maintain contact with the touch bar throughout the trial. Releasing the bar or exerting force outside of the specified window at any point after trial initiation, automatically failed the trial. Following the force bar hold period, the BCI cursor and a peripheral target were presented. The cursor remained fixed at the center of the workspace for the first 500 ms of the trial (referred to below as the "freeze period"), after which it was placed under BCI control (See fig. 2.6 for target layout and the two target task design).

2.3.1 Center-out

Animals performed a two-dimensional center-out cursor task under BCI control. Following the freeze period, the animal was then given 4 seconds to acquire the target with the

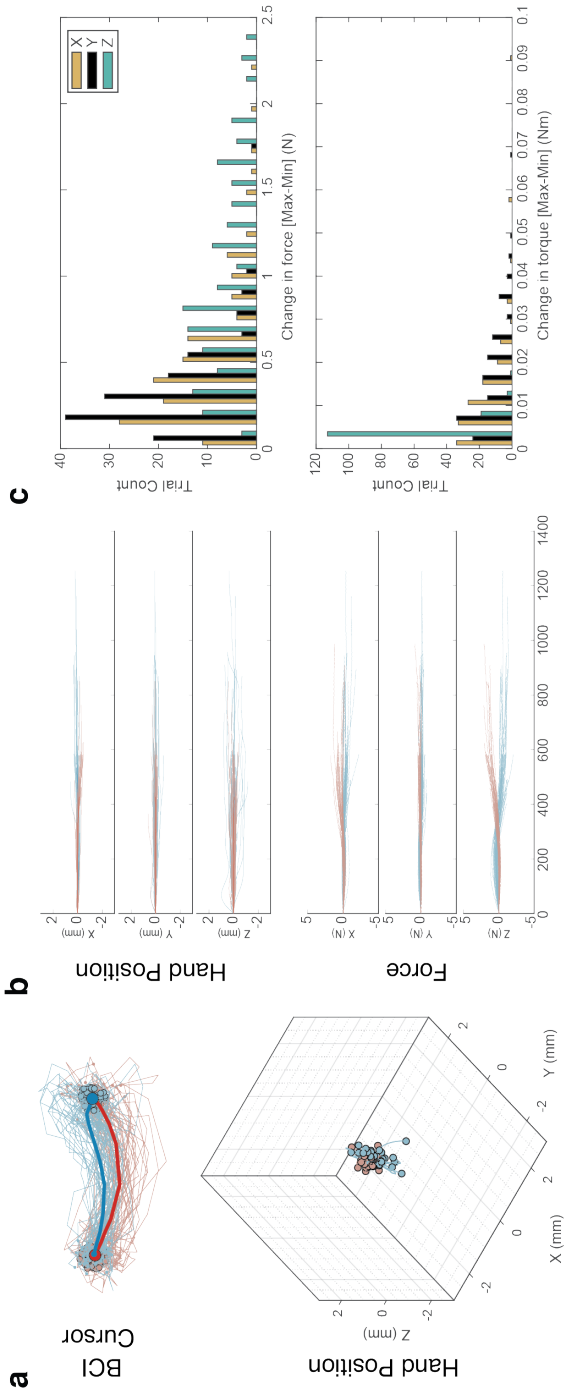

Figure 2.5: Hand movement and forces during BCI.

cursor in order to receive a liquid reward. No target hold time was enforced. The peripheral targets were selected from one of eight evenly spaced locations. Targets were presented in a pseudo-random order, such that each of the 8 targets was presented once before any target was repeated.

2.3.2 Two-target

The monkeys were trained to perform a two-step target acquisition task under BCI control. The animals had to acquire two targets that were always directly opposite each other relative to the center of the workspace. Targets were referred to as a "target pair". The first step was to acquire one of the targets. The second step was to acquire the other target of the pair. For each step, the animal had 4 seconds to acquire the target and no target hold time was enforced. Following completion of the second step, the animal received a liquid reward. For the majority of the sessions, the target pairs were presented one at a time. The order of the target pair presentation was pseudo-random.

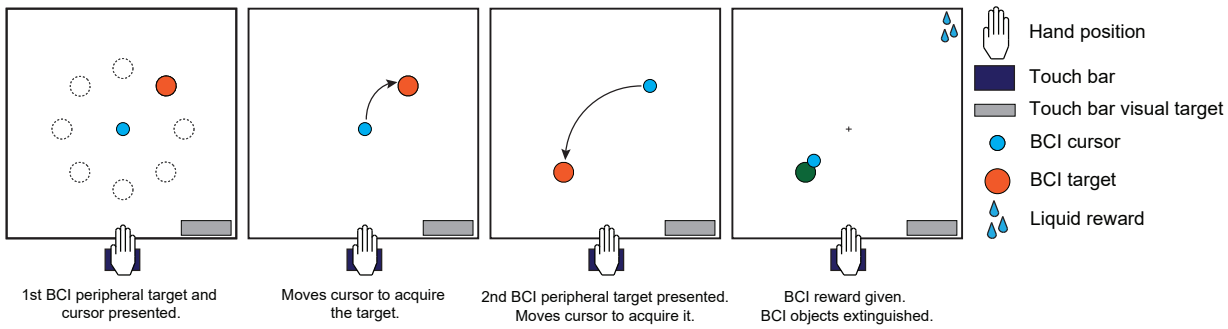

Figure 2.6: Two target task design.

2.3.3 Conditional Grid

The monkeys were trained to perform a modified two-step target acquisition task under BCI control. The first step was to acquire one target of the target pair, consistent with two

target task. For the second acquire step, there were multiple possible target locations (Fig 2.7A). For each step, the animal had 4 seconds to acquire the target and no target hold time was enforced. Following completion of the second step, the animal received a liquid reward. The number of possible second targets depended on the target grid used. The majority of sessions presented in this paper used a 4-target grid, where the targets were evenly spaced around the center of the workspace. The first step targets were chosen pseudo-randomly. The second step targets as weighted so that there were 100 trials to the second target of the target pair and 40 totatl trials to the other grid targets (Fig 2.7B).

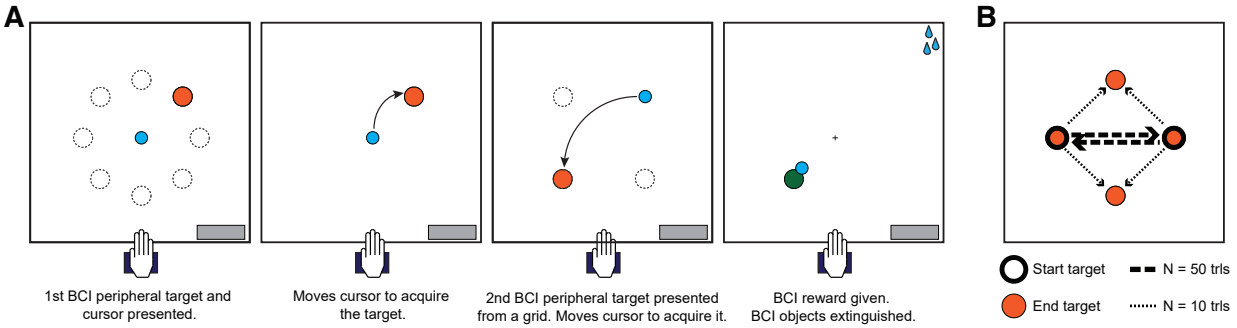

Figure 2.7: Conditional grid BCI task design. *a)* Task layout for the 45° target pair. *b)* The 4-target grid used for hte majority of the sessions.

2.3.4 Memory-guided Reach

Monkeys were trained to perform a memory guided reach task. At the start of the trial, the animal was required to acquire and maintain a start position. This position was either reaching to a target at the center of the workspace and maintaining a hold or holding the force bar with a neutral arm position and minimal force applied. Following the 350 ms start target hold, the instructed reach target was flashed on screen for a 150 ms. The animal would then maintain the start target hold until a go-cue was given 250-1500 ms later. The go-cue was the appearance of all possible reach targets shaded in gray. This style of go-cue ensured that the animal's attention was directed to the targets. It also ensured that the

reaches were precise. The animal then had to initiate a reach within 500 ms (i.e. a reaction state) of the go-cue in order to not fail the trail. We also failed the trials if the animals initiating their reaches prior to the go-cue. Finally, the animals had to maintain a short hold on the end target to be rewarded.

2.4 Extracting Neural Trajectories

We applied dimensionality reduction techniques to the neural data to identify a lower dimensional latent space that could accurately represent the data. The activity of cortical neurons is correlated [54–57]. With population recordings, it is evident that correlations, or co-modulation patterns, can encompass ensembles of many neurons [47, 58, 59]. Correlations imply that the neurons are coupled through some mechanism, such as local synaptic connectivity or common input.

Threshold crossing events were transformed into smoothed low-dimensional neural trajectories using a causal form of Gaussian Process Factor Analysis (GPFA) [58]. Threshold crossing events on each electrode were binned in non-overlapping 45ms time windows. Threshold crossings for q electrodes at a given point in time $\mathbf{u}_t \in \mathcal{R}^{q\times 1}$ were transformed to a p-dimensional latent state $\mathbf{z}_t \in \mathcal{R}^{p\times 1}$. In brief, GPFA assumes a linear-Gaussian relationship between latent states and spike counts

$$\mathbf{z}_t|\mathbf{u}_t \sim \mathcal{N}(C\mathbf{z}_t + \mathbf{d}, R) \tag{2.1}$$

where $C \in \mathcal{R}^{q\times p}$ specifies the relationship between the latent state and spike counts, $\mathbf{d} \in \mathcal{R}^{q\times 1}$ is the mean activity of each electrode, and $R \in \mathcal{R}^{q\times q}$ is a diagional matrix specifying the spike count private variance. Latent states are assumed to be Gaussian Processes, with the neural trajectory for the ith latent state at time steps 1 to T, $\mathbf{z}_i = [z_{i,1}, \ \ldots \ , z_{i,T}]^\top$, defined as

$$\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, K_i) \tag{2.2}$$

where $K_i \in \mathcal{R}^{T \times T}$ is a covariance matrix defining the relationship between the ith latent state at different points in time. We used a causal implementation of GPFA in which the latent state at the tth time step is only determined by neural activity from the previous T time steps. For the majority of our sessions, our casual implementation used 7 time bins, so $t-6$ to t or approximately 315ms into the past. There were a few sessions where we used a 22-bin truncation, but we found that weights for time bins after $t-6$ were small relative to recent time steps (Fig 2.8).

We used $p = 10$ dimensions for all experiments, as this has been found to capture most of the shared variability in motor cortex during BCI control [47,60]. The spike count history used to extract neural trajectories was reset at the beginning of each trial. There were no edge effects related to this reset, as the touch bar hold time (250ms) and the cursor freeze period (500ms) occurring at the beginning of each trial provided a full spike count history buffer before the onset of BCI control.

Neural trajectories were extracted in real-time by our system as:

$$\hat{\mathbf{z}}_t = C^\top R^{-1}(\mathbf{u}_t - \mathbf{d}) \tag{2.3}$$

$$\bar{\mathbf{z}} = [\hat{\mathbf{z}}_{t-\tau}^\top, \ \ldots \ , \hat{\mathbf{z}}_t^\top]^\top \tag{2.4}$$

$$\mathbf{z}_t = \bar{M}\bar{\mathbf{z}} \tag{2.5}$$

where $\bar{M} \in \mathcal{R}^{p \times p\tau}$ is a "smoothing kernel" describing the influence of past spiking activity on the latent state at time t. Parameters C, R, $\mathbf{d}$ and $\bar{M}$ were fit using expectation-maximization as described in [58] using data from target onset to target acquisition (for successful trials) or timeout (for failed trials).

2.5 BCI Decoder

Animals controlled the position of a cursor in a two-dimensional virtual environment using control signals derived from their neural trajectories. A BCI mapping was used to

transform low-dimensional neural trajectories $\mathbf{z}_t$ into 2D cursor positions $\mathbf{x}_t \in \mathcal{R}^{2\times 1}$. BCI mappings were of the form

$$\mathbf{x}_t = W\mathbf{z}_t + \mathbf{c} \tag{2.6}$$

where $W \in \mathcal{R}^{2\times p}$ is a weight matrix, and $\mathbf{c} \in \mathcal{R}^{2\times 1}$ is a positional offset. Intuitively, the BCI mapping can be viewed as a technique for providing animals with visual feedback of their neural trajectories in a particular 2D projection defined by W. The intuitive mapping is described in detail below.

2.5.1 Intuitive mapping

The intuitive BCI mapping was designed to provide animals with proficient cursor control, such that they were able to move the cursor quickly and accurately to targets placed throughout the workspace. Broadly, identifying the intuitive mapping involved finding parameters W_{int} and $\mathbf{c}_{int}$ that best predicted the assumed intent of the animals given their neural trajectories. More specifically, mapping parameters were defined as

$$W_{int} = XZ^\top(ZZ^\top)^{-1} \tag{2.7}$$

$$\mathbf{c}_{int} = -W_{int}\left(\frac{1}{N_{st}}\sum_{i=1}^{N_{st}} \mathbf{z}_{i,st}\right) \tag{2.8}$$

where $Z = [\mathbf{z}_1, \ \ldots \ , \mathbf{z}_N]$ and $X = [\mathbf{x}_1, \ \ldots \ , \mathbf{x}_N]$ are matrices of observed latent activity and associated intended cursor positions during calibration, $\mathbf{z}_{i,st}$ is the ith observation of latent state $\mathbf{z}$ at the start of the trial.

To individually optimize the BCI control for each monkey, we slightly adjusted how we defined the calibration data. For monkey E, calibration data for each trial consisted of sets of $\mathbf{z}_i, \mathbf{x}_i$ for two time epochs in the trial. The first time epoch consisted of the first 3 time bins (approx. 135ms) of the freeze period. During this period, we assumed that the animal was not attempting to move the cursor, and set $\mathbf{x}_i = [0, 0]^\top$. The second time epoch was 5 time bins in duration (approx. 225ms) immediately preceding acquisition of the BCI target. During these time bins, we assumed that the animal was intending that the cursor was at

the target location (e.g., $\mathbf{x}_i = [90, 0]^\top$). We found that these assumptions generally worked well for monkey E, whose neural activity was highly stereotyped during calibration.

For Monkey Q, we were also able to align the second time epoch to the 5 time bins proceeding BCI target acquisition. This worked well for the animal. The first time epoch had to be shifted to slightly later in the session as the animal habitually adjust his grip on the force bar at the start of each trial. This resulted in very inconsistent baseline activity measurements. To account for this, we define the first epoch as the 3 time bins prior to the go-cue being given, when the animal was maintaining a constant hold on the force bar. This ensured that the animal's movement was minimal

For monkey D, we found that neural responses were less consistent on a trial-to-trial basis, with temporal responses that varied across targets. We therefore sought to infer the idealized intended kinematic profile for the ith trial $X_i = [\mathbf{x}_{i,1}, \ \ldots \ , \mathbf{x}_{i,T}]$ from the structure of neural responses during the trial Z_i. To do this, we first assumed that the animal was intending to move the cursor from the center of the workspace to the target, but the timing and consistency of this intent might vary from trial to trial. We found that neural responses during the freeze period of calibration trials consisted of an initial target-invariant response, where neural activity increased but was similar across target directions, and a subsequent target-specific response, where neural activity was discriminable for different targets.

In order to estimate the target-specific response, we first sought to remove the target-invariant component of neural activity. We defined the target invariant signal over the freeze period (500ms in duration) by first interpolating neural trajectories for each trial over a 1ms time grid. The target-invariant signal was defined as

$$Z_{inv} = \frac{1}{N} \sum_{i=1}^{N} Z_i \tag{2.9}$$

where $Z_{inv} \in \mathcal{R}^{p \times T}$ is the target-invariant response, Z_i is the latent state for trial i, T is the number of time steps in the interpolated latent states (500), and N is the number of trials. We then subtracted this component from the neural responses during each trial:

$$\hat{Z}_i = Z_i - Z_{inv} \tag{2.10}$$

We next sought to infer the time course of the neural response for each target on a trial-by-trial basis. Principal components analysis (PCA) was applied to the average neural trajectory $\hat{Z}$ for each target direction. Because the target-invariant response was removed from these trials, a single PCA dimension captured most of the variability in $\hat{Z}$ for each target. Neural trajectories for all trials for a given target were then projected onto this PCA dimension to yield a set of temporal profiles $P_{targ} = [\mathbf{p}_1, \ \ldots \ , \mathbf{p}_{N_{targ}}] \in \mathcal{R}^{T \times N_{targ}}$, where $\mathbf{p}_i$ is the temporal activation profile for the ith trial and N_{targ} is the number of trials to the current target. Temporal activation profiles for each trial were then sub-sampled over the original time grid (i.e., interpolation was removed) and then normalized to produce the intended cursor trajectory:

$$\hat{X}_i = \mathbf{x}_{targ} \frac{\mathbf{p}_i^\top - p_{min}}{p_{max}} \tag{2.11}$$

where $\hat{X}_i \in \mathcal{R}^{2 \times T_i}$ is the intended cursor position for the ith trial, $\mathbf{x}_{targ}$ is the target position, p_{min} and p_{max} are temporal profile normalization constants, and T_i is the number of time steps in the ith temporal profile. Normalization constants p_{min} and p_{max} were used scale the temporal response; p_{min} was defined as the average temporal activation profile over the first 100ms of each trial, and p_{max} was the average of the max temporal activitaion for each trial. The resultant set of inferred cursor positions and their corresponding neural trajectories $\hat{X}_i, Z_i$ were used to find W_{int} and $\mathbf{c}_{int}$ as described above.

2.5.2 BCI Calibration Procedure

At the start of each experiment, we collected a block of calibration trials. This data was used to determine the parameters of the intuitive mapping with proficient BCI control.

The first calibration step involved 32 trials of the animals passively observing of cursor movements while we recorded from M1. Previous studies have shown that passive observation modulates M1 activity [61, 62]. During these trials, the cursor automatically moved in a straight line to the target at a constant velocity ($\sim 0.15 m/s$). The targets were presented in pseudo-random order ($\sim$ 4 trials per target). After this initial observation block, we used the data calibrated the initial set of decoder parameters. The animals performed four

32-trial blocks the center out BCI task. We refer to this procedure as 'gradual training' because we gradually give the monkey more and more control of the cursor. The decoder parameters were re-calculated after each block using data accumulated across all BCI blocks. In the first BCI block, we completely restricted the cursor's perpendicular movement, but with each update of the decoder parameters we increased the animals' perpendicular control. The amount of perpendicular control with each decoder was 0.001, 0.25, 0.5, 1, and 1. The final intuitive decoder was calibrated using 160 trials of BCI data.

2.6 Flow Field Analysis

A flow-field-based analysis was used to compare trajectories across experimental conditions during the two-target task. In brief, this analysis estimated the change in trajectory as a function of the trajectory state, or $\dot{\mathbf{x}} = f(\mathbf{x})$. This analysis was used to compare both cursor trajectories and neural trajectories in different two-dimensional projections (See 5.4.1)

To estimate the flow field for a given condition, we first divided the two-dimensional space into a set of square voxels. When estimating flow fields for cursor trajectories, we used a voxel width of 20mm. When estimating flow fields for neural trajectories, we used a voxel width of one latent unit. To calculate the velocity for a given voxel, we averaged the velocity of all states located in that voxel. The resultant field consisted of a set of 2D velocity vectors, one for each voxel. Average velocity vectors were only considered as valid if they were based on a minimum of 2 data points.

We used two metrics to compare flow fields across conditions: (1) the number of valid overlapping voxels (flow field overlap), and (2) the mean-squared error (MSE) of all flow vectors (flow difference). To compute flow field overlap, we counted the number of overlapping valid voxels for the two conditions. This resulted in a single value for each flow field comparison. For the flow difference, we calculated the mean-squared error (i.e. $mean((\vec{v_1} - \vec{v_2})^2)$) of the two average conditions for each overlapping voxel. This resulted in a distribution of flow differences. We used the mean of this distribution for comparison, unless otherwise stated.

2.6.1 Cursor Trajectory Flow Fields

For the flow field analysis shown in Figures 3.2, 5.6 and 5.7, we compared flow fields for the following conditions during the two-target task: (1) cursor control using the intuitive mapping (intuitive-actual), (2) predicted rotated mapping trajectories, where neural activity during intuitive mapping cursor control was passed through the rotated mapping offline (rotated-predicted), and (3) cursor control using the rotated mapping (rotated-actual). We then computed flow difference (Fig. 5.6C) and flow field overlap for the intuitive-actual vs rotated-actual and rotated-predicted vs rotated-actual comparisons.

2.6.2 Neural Trajectory Flow Fields

For the neural flow field analysis shown in Fig. 5.6D-E, we sought to determine how the visual feedback provided to the animal impacted neural trajectories in the full 10D GPFA space. To do this, we estimated flow fields independently for each single target direction (e.g., A→B) for the following conditions: (1) cursor control using the intuitive mapping (Indicated as "intuitive" in this document), (2) cursor control using the rotated mapping (rotated), (3) a time-reversed version of the neural trajectories during cursor control using the intuitive mapping (intuitive-reversed), and (4) neural trajectories during cursor control using the intuitive mapping for the alternate target direction (i.e., B→A, intuitive-alternate).

To compare flow fields over the entire 10D space, we first projected neural trajectories into random 2D subspaces:

$$\tilde{\mathbf{z}}_t = M^\top \mathbf{z}_t \tag{2.12}$$

where $M \in \mathcal{R}^{p\times 2}$ is a random orthonormal matrix. Flow fields for the 4 conditions (intuitive, rotated, intuitive-reversed, and intuitive-alternate) were then estimated. For each condition, we randomly split the total number of available trials (~ 50) into two sets of 20 trials. We estimated the flow field for each set. This provided two estimates of the flow field for each condition, allowing for an estimate of the within-condition variability.

We then performed 4 flow field comparisons: (1) intuitive vs rotated, where the visual feedback provided to the animal varied across conditions (variable-feedback), (2) between

the two sets of trials for the intuitive condition, where the feedback was the same (fixed-feedback), (3) intuitive vs intuitive-reversed (time-reversed), and (4) intuitive vs intuitive-alternate (alternate-target). The fixed-feedback comparison provided a measure of the degree of variability expected when estimating flow fields within a condition. The time-reversed comparison provided a measure of the degree of flow field similarity expected if the flow was maximally different but overlapping. The alternate-target comparison provided a measure of the degree of flow field similarity expected when the flow fields were different and non-overlapping, consistent with our observations of trajectory asymmetries. With the exception of the fixed-feedback case, all comparisons were between flow fields for a single random split of trials. We calculated flow field overlap and flow difference magnitude for five random trial splits per projection, and combined results for both target pairs (A→B and B→A) for each projection.

Flow comparisons were performed for 400 different random projections per experiment. This resulted in four distributions per metric (flow field overlap and flow difference magnitude) per experiment, one for each of the 4 flow field comparisons (variable-feedback, fixed-feedback, time-reversed, and alternate-target). To compare these distributions across experiments, flow field overlap o and flow difference m were normalized:

$$\hat{o} \quad = \frac{o-\bar{o}_{alt}}{\bar{o}_{same}-\bar{o}_{alt}}\hat{m} \quad = \frac{m-\bar{m}_{same}}{\bar{m}_{rev}-\bar{m}_{same}} \tag{2.13}$$

where $\bar{o}_{alt}$ and $\bar{o}_{same}$ are the mean number of overlap voxels for the alternate-target and fixed-feedback distributions, and $\bar{m}_{same}$ and $\bar{m}_{rev}$ are the mean flow difference for the fixed-feedback and time-reversed distributions. A normalized flow field overlap value of 0 indicated that the region of the state space occupied by the trajectories was highly non-overlapping between distributions, while a value of 1 indicated no difference in overlap between distributions. A normalized flow difference of 0 indicated that there was no change in flow difference between distributions, while a value of 1 indicated that flow difference were maximally different between comparison conditions.

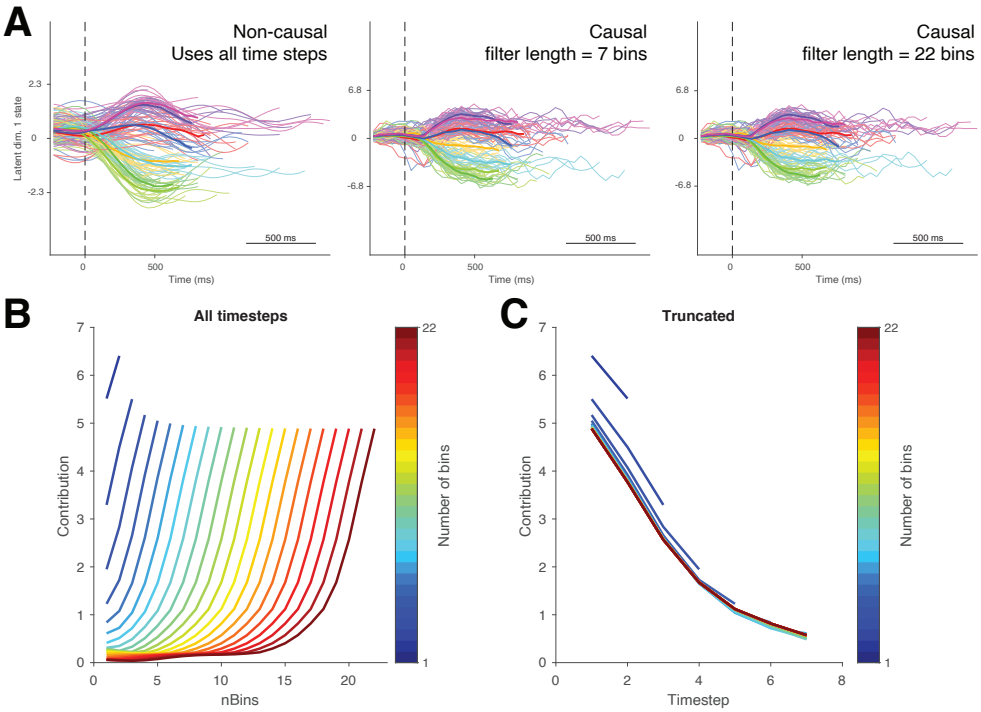

Figure 2.8: Characterizing GPFA smoothing. A comparison of how varying the size of T changes trajectory smoothing. *a)* The impact of difference GPFA implications. *left panel)* The standard non-causal implementation of GPFA [4]. The entirety of trials are used to determine temporal smoothing. The trajectories appear smoother and larger. *center panel)* Causal GPFA. Only the previous 7 bins are used to filter the neural trajectories. This means trajectories are less smooth. *right panel)* Causal GPFA with a 22 time bin filter. The trajectories look identical to the trajectories in the center panel. This is because there are diminishing returns for the contributions. *b)* A conceptual drawing of the contribution weight as you increase the number of time bins, T. The color bar shows the number of time bins used for smoothing. The x-axis is the time step along the trajectory. The most recent time bins have the greatest smoothing contribution and earlier time bins trail off to trivial weights. *c)* The effect of changing T for a truncated filter at the start of the trials. For the first few time bins of a trajectory, the limited amount of previous data causes the filter contribution weight to shift. The contribution changes occur for $T < 7$ before stabilizing for all filter sizes.

3.0 Neural Population Activity Time Courses are Not Time-Reversible

3.1 Summary

If a central aspect of cortical information processing is computation through dynamics, then we would expect to see dynamics present whenever cortex is active, and not just in some contexts. Here, we asked whether neural population dynamics in primary motor cortex are present when the cells are active, even if the arm is not moving. We leveraged a brain-computer interface (BCI) to probe these dynamics by directly mapping recorded neural activity to a computer cursor's position. Animals moved the cursor repeatedly between a pair of onscreen targets, producing highly overlapping cursor trajectories. When examined in the high dimensional neural space, we observed consistent and distinct structure to the time-evolution of neural population activity during BCI control when the arm was not moving.

3.2 Introduction

Behavior unfolds over time, as does the neural activity that governs movement. To date, dynamics in primary motor cortex (M1) have primarily been reported in the context of preparing and executing arm movements [12, 14, 20–22, 34, 63]. Work from the Churchland lab has shown that there is a consistent oscillatory dynamic in the motor cortex response during reaches. This suggests that there may be generalized dynamical properties in M1 and other motor areas. These dynamics, however, were only looked for in a set of overly trained stereotyped arm reaches. It is therefore unclear if the consistent time courses of neural trajectories observed during overt movements are due to constraints imposed by the underlying neural circuitry (neural dynamics), or if they are simply related to controlling the limbs through time (effector dynamics).

Here, we used a BCI paradigm to ask whether neural activity in the motor cortex of rhesus macaques was consistent with the presence of a neural dynamic. With BCI, we were

able to separate neural and effector dynamics by probing the neural activity in the absence of overt movement. Animals performed BCI tasks in which their recorded neural activity was used to control the position of a computer cursor. We first identified a 2D projection of neural population activity designed to yield proficient control. Using this "intuitive" BCI mapping, animals moved a cursor repeatedly between a pair of onscreen targets (A and B). Animals had reliable control of the cursor, rapidly producing trajectories that were straight and largely overlapping between the two conditions (A→B and B→A).

Using this paradigm, we asked whether neural activity took different paths when moving between shared neural states. The presence of distinct trajectory paths, or trajectory asymmetries, in the high-dimensional neural space would imply that the time course of the activity is distinct. We observed that in the high-dimensional neural trajectories of the two conditions were separable and distinct. We quantified the trajectory separation in the 10D space using two metrics: the discriminability of the midpoints and the distance between the trajectories through the state space. For all sessions and animals, we found that the discriminability of the condition midpoints was well above chance (50%), demonstrating that the trajectories are well separated across trials. We next assessed how close the neural activity patterns of the two conditions were when they produced similar cursor positions. We found that a large separation in the neural space *between* the two conditions throughout cursor control. A large separation was not observed *within* individual conditions. These metrics show that there are consistently distinct time courses of the neural trajectories even in the absence of arm movements.

3.3 Methods

All experiments were collected using the intuitive BCI mapping (see methods 2.5). Animals performed either a two target task or conditional grid task. There were two potential start targets (A and B) that defined a diametrically opposing target pair. The target pair was selected as one of four target pairs from an eight-target cartiesan map (0°, 45°, 90° and 135°. See Fig 2.6, *first panel*). We analyzed 111 sessions (Monkey E - 50, Monkey D - 40,

and Monkey Q - 21) where there were at least 100 trials of the two target task between the target pair.

3.3.1 Calculating Discriminability Index

Discriminability index (d') measured the separation of the two condition's 10D midpoints. This metric accounts for both the mean and standard deviation of the condition distributions. For every trial, we defined the midpoint as the point closest to the center between the two end targets. we first calculated the projection vector ($\vec{p}$) between the two end targets and center of the two targets (c).

$$\vec{p} = \frac{\overline{Z_A} - \overline{Z_B}}{|\overline{Z_A} - \overline{Z_B}|} \tag{3.1}$$

$$c = \frac{\overline{Z_A} + \overline{Z_B}}{2} \tag{3.2}$$

where $\overline{Z_A}$ and $\overline{Z_B}$ were averages of the first time step of observed latent activity for the given start target. The center (c) and all data ($z(t)$) were then projected onto this vector to determine distance along the line. The midpoint of the trajectory was identified as the point closes to the projected distance c.

$$c_p = c * \vec{p} \tag{3.3}$$

$$z_p(t) = z(t) * \vec{p} \tag{3.4}$$

$$t_{midpoint} = \arg\min_t z_p(t) - c_p \tag{3.5}$$

We calculated the average midpoint for each condition and defined an axis as the vector between the averages. The distance along this axis was calculated for every trial's midpoint.

$$\vec{a} = \frac{\overline{Z_{AB}} - \overline{Z_{BA}}}{|\overline{Z_{AB}} - \overline{Z_{BA}}|} \tag{3.6}$$

$$z_a = z(t_{midpoint}) * a \tag{3.7}$$

where $\overline{z_{AB}}$ and $\overline{z_{BA}}$ are the average midpoints. For each condition we determined the mean (μ_{AB} and μ_{BA}) and variance of the midpoint distribution. We then used this to calculated d' as

$$d' = \frac{\mu_{AB} - \mu_{BA}}{\sqrt{0.5 * (\sigma_{AB}^2 + \sigma_{BA}^2)}} \tag{3.8}$$

With d', larger values mean that the distributions are more separable. Since it can be difficult to interpret the threshold for good d', we translated d' to percentage of classification accuracy, separability accuracy (SA). This metric measured the best case scenario to classify the conditions using only the midpoint. To determine this value, we calculated a ROC curve to measure true positive rate vs false positive rate for a single condition. From the ROC, we then calculated the maximum SA for each condition.

$$(SA) = 100 * max(\frac{n_{AB,true} + n_{BA,true}}{n_{AB} + n_{BA}}) \tag{3.9}$$

where $n_{AB,true}$ and $n_{BA,true}$ are the number of true positive hits for the conditions and n_{AB} and n_{BA} are the total number of trials per condition.

3.3.2 Calculating Neural Distance

To calculate the neural distance between conditions we first calculated the average trajectory($\overline{Z_i}$) of each condition in neural space (10D) by aligning each trial to trajectory onset. Because the cursor space is a direct readout of the neural space in the 2D, we also created 2D cursor trajectory averages (Z_i).

We randomly assigned which one condition average trajectory to compare against (X_A). Using Euclidean distance nearest neighbor method on the 2D averages, we identified for each index along X_A the closest point on X_B and its corresponding Z_B. Within the 10D space, we then calculated the distance for each index of Z_A and its corresponding nearest neighbor value.

$$d(Z_A, Z_{NB}) = \sqrt{\sum_{i=1}^{10} (z_{A,i} - z_{NB,i})^2} \tag{3.10}$$

Where z_{NB} are the observed latent activity for the nearest neighbor values X_{NB}. In order to compare these distances across session, we normalized each distance by the distance between the end targets ($\overline{Z_A}$ and $\overline{Z_B}$). Separation distances near zero have little difference between the condition. Distances larger than one are greater than the end point separation.

To assess within condition variability, we randomly split the total of number of condition A trials into two sets of 20. We compared these split averages ($\hat{X}_{A,1}$ and $\hat{X}_{A,2}$) to X_A and found the distance between $\hat{Z}_{A,1}$ and $\hat{Z}_{A,2}$. We ran 50 permutations of the random split to calculate a distance distribution for each index of Z_A.

3.4 Results

3.4.1 Cursor Behavior was Consistent between A→B and B→A Conditions

Animals used the intuitive BCI mapping to move a cursor between a pair of onscreen targets (A and B) for 100 trials. As expected, the animals had great control and the resulting cursor trajectories that were fast and consistent (Fig 3.1). We also observed a large amount of trajectory overlap between conditions. We quantified this overlap using a flow field analysis of the cursor trajectories. We performed two flow field comparisons: intuitive vs intuitive-alternate (alternate-target) and intuitive vs intuitive-reversed (time-reversed). The time-reversed flow field provides a measure for unconstrained or reversible trajectories. We found that the MSE was comparable between the two comparisons (Fig 3.1B, the green and blue histograms). This suggests that the cursor behavior is reversible.

In addition, to measuring the flow field difference, we also assessed the acquire times (Fig 3.1C) and straightness of path (Fig 3.1D) for each direction. As seen in the acquire time histograms, there is often a significant difference in the acquire times between the two conditions. Indeed, we found that there was difference in the condition acquires across all sessions for each animal (Table 3.1). While there was a target pair discrepancy for each animal, which faster condition was entirely animal dependent. We assessed the normalized trajectory length between any target pair condition and found that there was not a significant

Table 3.1: Acquire times for the two target trials. The average and standard deviation of acquire time in ms for each animal. The acquire times were calculated using on successful trials. The values are written $mean(STD)$ and ordered by target pair.

	Monkey E	**Monkey Q**	**Monkey D**
0°	477(220)	576(312)	506(271)
180°	557(231)	652(370)	461(239)
45°	450(228)	528(230)	403(222)
225°	533(245)	660(337)	608(331)
90°	432(206)	595(344)	462(244)
270°	473(246)	584(273)	483(252)
135°	513(238)	507(248)	461(219)
315°	460(231)	706(370)	483(307)

Table 3.2: N-way ANOVA for trajectory length. ANOVA was run on all sessions using two groups, subject and target pair. Bolded values was significant.

Source	*d.f. (n-1)*	*F*	*Prob.F*
Target Pair	3	1.36	0.2569
Animal	2	8.02	**0.0004**
Target Pair*Animal	6	1.48	0.1851

difference in trajectory length. Indeed, we ran an Anova over all condition and animal condition pairings and found that only the animal source had a significant impact on the trajectory lengths (Table 3.2). We extended these behavioral metric comparison for all intuitive two target task session. The trends persisted (Fig 3.2). We found that the time-reverse and alternate-target comparison was consistent across sessions and target pairs.

We found the online cursor trajectories demonstrated high levels of overlap for the majority of the sessions. Considering cursor position and trajectory length alone, this might suggest that the activity is malleable. However, the small discrepancies in the acquire times suggest that the activity may be different within the high-D neural space.

3.4.2 Neural Time Courses Exhibit Separation in the Neural Space

We first sought to identify structure in neural trajectories suggestive of flow-field-like dynamics. Specifically, we looked for trajectory asymmetries, in which trajectories took different paths between shared neural states. We observed that in the high-dimensional neural state space the neural trajectories of the two target task were separable and distinct (Fig. 3.3A). To characterize how separated the two neural trajectories were from each other in high D neural space, we calculated the d', discriminability index, using the trajectory midpoints (Fig. 3.3B/C). Because the range of d' is unbounded, we also calculated separability accuracy, SA, as a more intuitive measure (Fig. 3.4). This metric simply calculates the great-

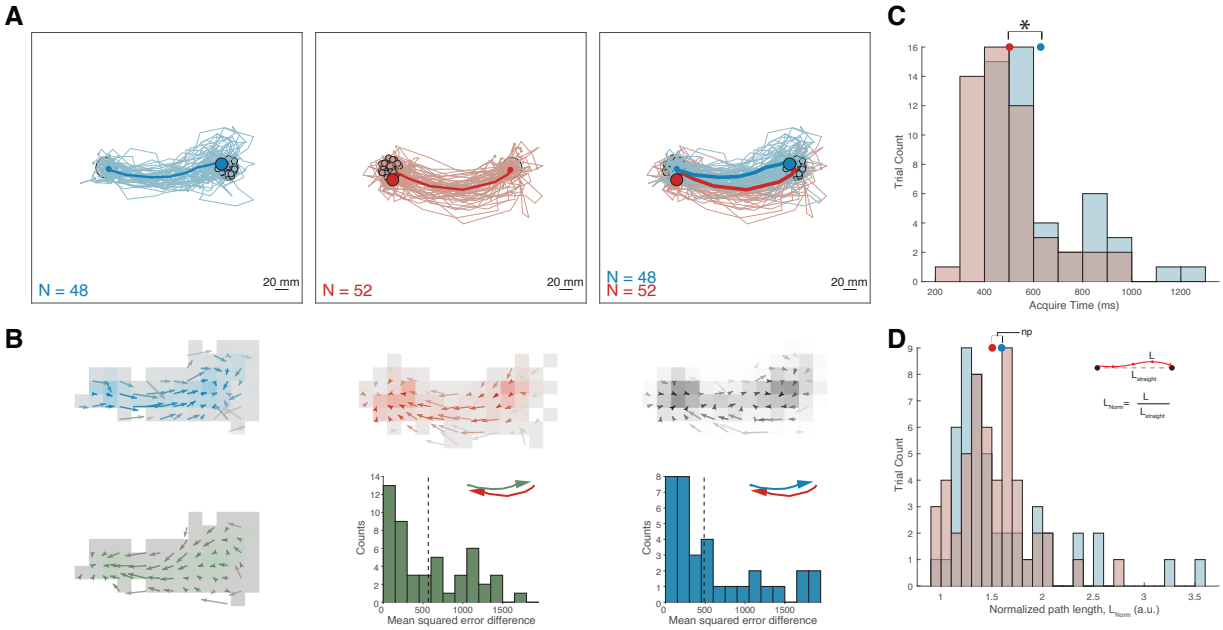

Figure 3.1: Behavior characterization of session 54E. *a)* The cursor trajectories for the conditions A→B and B→A. Trajectories are plotted separately and then together for better comparison. *b)* The flow field comparison between the time-reversed and alternate-target comparisons. The blue flow field is for the A→B condition, green is time reversed A→B, red is B→A, and black is the combined flow field for A→B and B→A. The green and blue histograms show the distribution of MSE for the time-reversed and alternate-target configuration respectively. *c)* The distribution of the acquire times for the two conditions. They were significantly different from each other (t-test $p = 0.001$). *d)* The distribution of the total path length for every trial of each condition. The difference between the distributions was not significant.

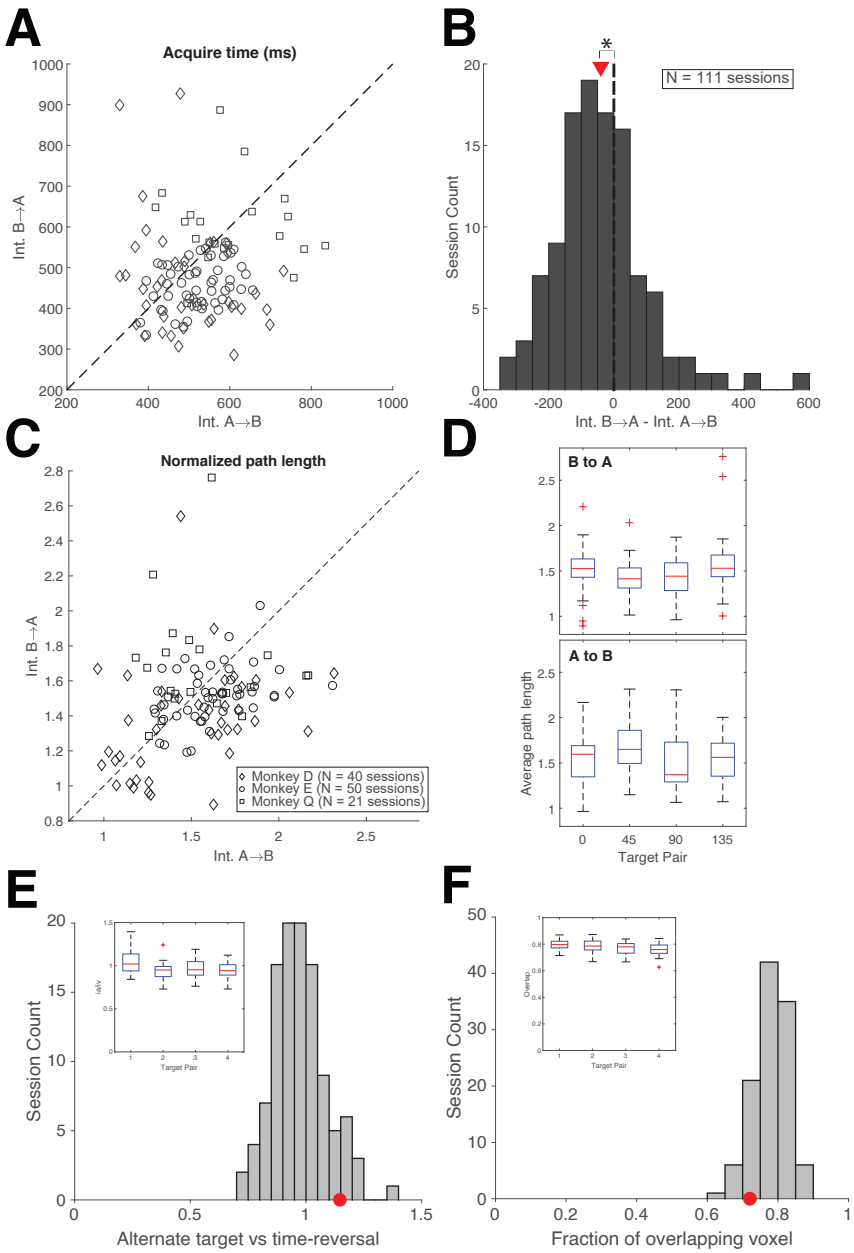

Figure 3.2: Summary of the intuitive control behavior metrics. *a)* Scatter plot of acquire times for all sessions. b) Histogram of the difference in condition acquire times (p=0.0046, significant). *c)* Scatter plot of path length for all sessions. *d)* Path length metric separated by direction and target pair. There was not a significant interaction between these groupings. e) Distribution of the MSE for the flow field analysis. d) Proportion of shared voxels for the flow field analysis.

est accuracy we could have in classifying the midpoints. SA is therefore bounded between 0%, 50% (chance), and 100% (well separated). We found that the conditions were always discriminabile (Fig. 3.3D) For all sessions and animals, we found that the discriminability index was well above chance (50%), demonstrating that the trajectories are well separated across trials (Fig. 3.3D).

Having observed the high separability of individual trial midpoint, we next assessed how distinct the condition trajectories were in high-D neural space. It is possible that the two conditions are separable, but the distance between their activity is minimum. Large amounts of overlapping activity patterns would suggest that the observed time course constraints are not from M1 circuitry. We quantified overlap as the 10D distance between comparison points along the neural trajectories (Fig 3.3E). To identify comparison points, we used a nearest neighbor metric on the average two-dimensional cursor trajectories. This allowed us to identify neural activity patterns that produced similar cursor positions (see method 3.3.2) If the time course of activity is not constrained then we might expect similar activity patterns to produce similar cursor positions.

We found the separation was small near the target activity, but large along the middle of the trajectories (Fig 3.3F). We normalized the separation distance by the end target distance. This allowed for comparison across sessions and provided a intuition for the distance measure. Separation distances at zero meant the activity patterns were the same. Distances larger than one were greater than the separation for the targets. As a control, we calculated the separation distance within a single condition (*blue trace in F*). The separations between the two conditions were greater than the within condition control. Furthermore, the maximum distance between the conditions were well above zero for all animals (Fig 3.3G) These results show that there was little shared activity between the conditions. These distinct trajectory paths could be a reflection of network constraints.

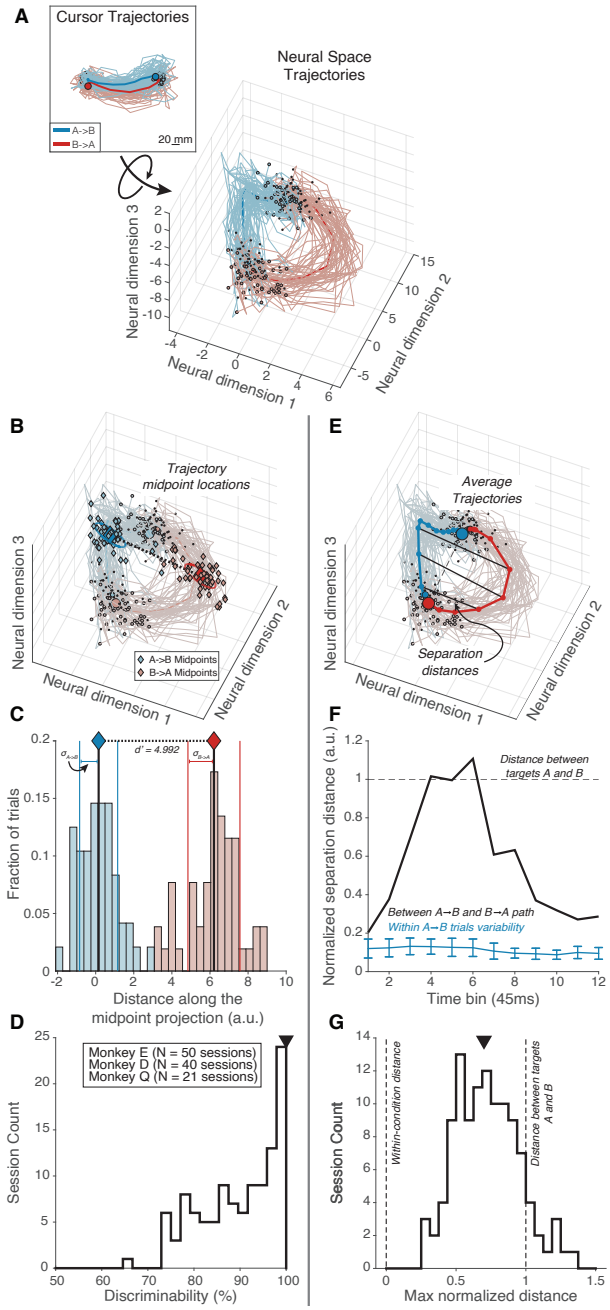

Figure 3.3: Neural trajectories are well separated in the neural state space. *a)* Two-target cursor trajectories for an example session (left) projected into a 3D representation of the 10D orthonormalized neural space (right). The thin and thick lines are individual trials and condition averages respectively. The circles mark the end of trajectories. *b)* Midpoints for all trials in the neural state space. The dashed line connects the average midpoints and their covariance ellipses. The midpoints are colored by condition. *c)* The distributions midpoints along the projection axis and colored by condition. The average distance (diamonds) and standard deviations are also shown. d' was used to calculate discriminability index. *d)* The discriminability index calculated for all sessions. *e)* The average trajectories in neural space. Nearest neighbor pairs are connected with solid black lines. The start targets are connected with a dashed line. *f)* The normalized distance between each nearest neighbor pair. The blue line represents the within-condition distances. *g)* The max normalized distance for all sessions. The black arrow marks the example session.

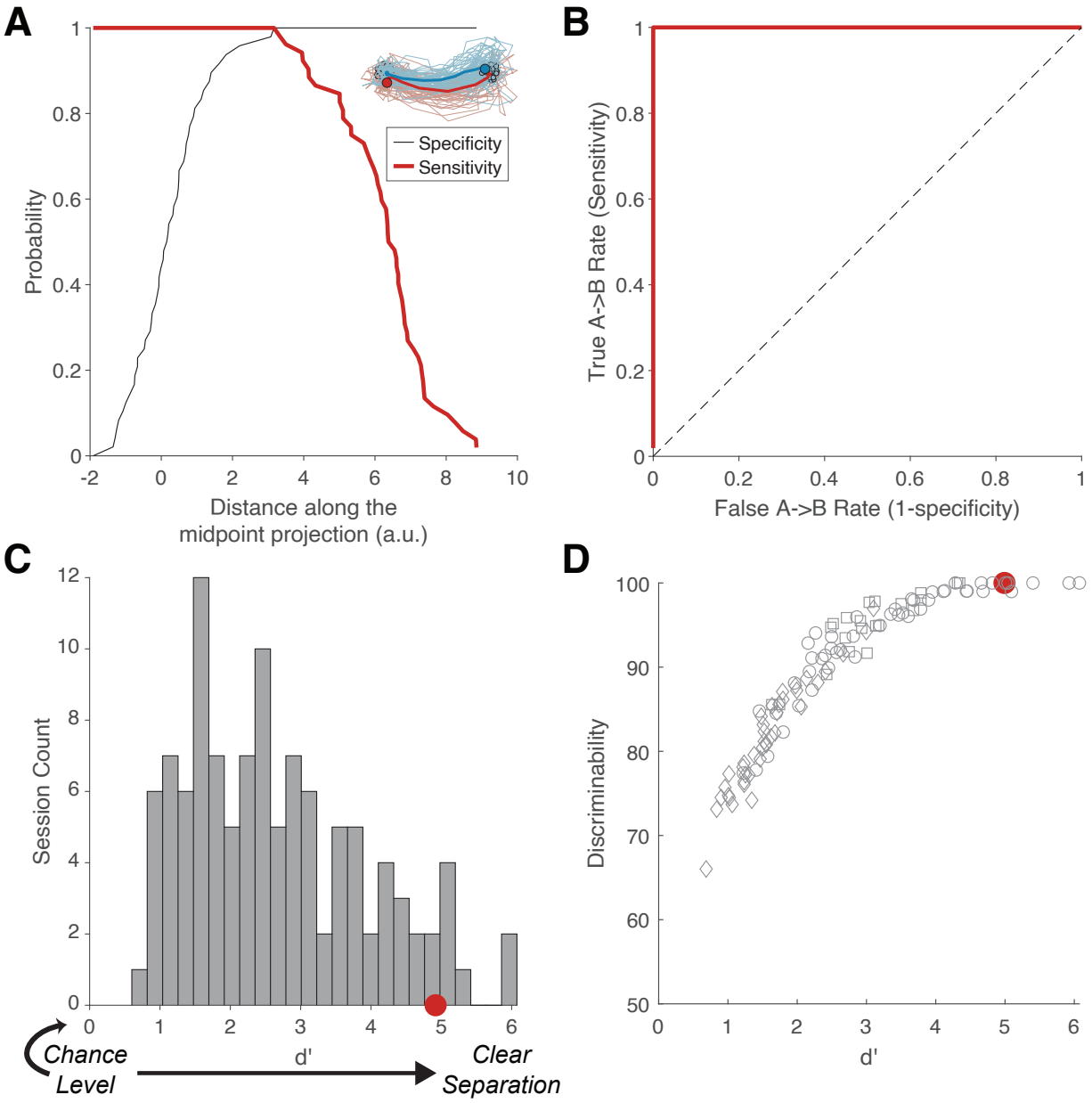

Figure 3.4: Example ROC and summary of d'. *a)* A sensitivity-specificity plot for the example session 54E. *b)* The ROC curve for the example session. *c)* Histogram of the d' primes for all sessions. The example session is plotted along the x-axis. *d)* The relationship between d' and our discriminability index. The example session is again labeled.

3.5 Discussion

If the time course of activity is constrained by neural circuitry, we would expect the constraints to be present even in the absence of overt movement. Here, we leveraged BCI to probe neural population activity in M1 without making arm movements. We observed that while the BCI cursor trajectories were largely overlapping between the two task conditions (A→B and B→A), the underlying neural trajectories were non-overlapping in the 10D neural activity space. This would suggest that there are dynamical properties that emerge from the neural circuitry in M1 rather than only being driven by feedback from the muscles. We quantified the trajectory separation and found it to be significant for all sessions.

4.0 Characteristic Population Activity Time Courses can be Identified With or Without Spike Sorting

4.1 Summary

Before starting the collection of our experimental data with Monkey E, we assessed whether we should use spike sorted data. We had observed neural asymmetries when using threshold-crossing activity. We were curious if these asymmetries were evident in well-isolated single unit activity. Additionally, whether multi-unit or single unit data better identified the asymmetries. To address this question, we performed a post-hoc analysis on pilot data from Monkey N. We found that asymmetries were present in both sorted and unsorted data.

4.2 Introduction

Oby at el demonstrated that the using threshold-crossing activity enables better BCI control [53]. The addition of lower amplitude multi-unit signals increases signal information, improving decoder performance. In our pilot data from Monkey N, we had observed neural asymmetries. These asymmetries varied in strength across target pairs. Is it possible that these additive feature of multi-unit activity could obscure the population dynamics? Suppose the network connectivity constraints only affect a subspace of the neural activity. If we combine the firing rates from the constrained population's neuron with a malleable population's neuron, we may mask evidence for constraints. To assess the impact of spike sorting on neural dynamics, we ran post-hoc analysis on pilot data from Monkey N. We identified the projection with the greatest amount of asymmetry for each target pair. We then sorted the neural data from multi-unit to well isolated single unit. We identified and compared the asymmetries for various combinations of sorted activity. We found that the neural asymmetries persisted for all tested configurations, even we when only used a small

population of single units. Qualitatively, the asymmetries identified with sorted data were similar to unsorted asymmetries. This confirmed that spike sorting would not provide additional benefit for our experiments. Finally, the presence of dynamics in both sorted and unsorted activity suggest that there are network constraints on all neural activity.

4.3 Methods

All the data was collected using threshold crossing activity to control the decoder. During the experiment, we used a threshold of 3.0xRMS for each channel (see method 2.2.1 for more details). The figures within this chapter are all based on post-hoc analysis of a single pilot experiment (N20161231) from Monkey N. The animal performed 200 trials of a modified conditional grid. In the task, the start target was randomly selected from any of the four targets presented on the screen. The end target was then set as one of four remaining targets. With this design, we would identify six possible target pair asymmetries to probe for neural dynamics.

Note that in this experiment, the animal used a Kalman filter decoder rather than a linear regression decoder (see method 2.5) to control cursor position. While this difference in decoder type will produce smoother cursor trajectories, it will not impact the latent space analysis.

4.3.1 Spike Sorting Criteria

All sorting was done post-hoc using OpenSorter (Tucker Davis Technologies, Alachua, FL, USA). Our standard recording setup maintains three sortIDs: one for any valid threshold crossing, one for any outlier waveforms, and one for any noise signals. When sorting the channels, we exclude any waveforms originally identified from the outlier and noise sorts. In other words, we only sorted the data that was used for BCI control. This ensured that we were directly comparing sorted and unsorted asymmetries. Waveforms were manually classified according to three metrics: separation in the PCA space, waveform shape, and

voltage threshold (Fig 4.1). Each unit was then scored from 0 (multi-unit activity) to 4 (well-isolated waveforms). We classified a waveform unit as well-isolated if it passed all three metrics. That is, a well-isolated unit (sort 4) had clear separation of the cluster in PCA space, a low variance waveform shape, and a large peak-to-peak amplitude (see the cyan unit in Fig 4.1). Sort 3 was for well-isolated units with greater variance in their waveform shape (red unit in Fig 4.1). Sort 2 classified units were well-characterized by one metric and difficult to parse for other metrics. An example for this would be a unit where the waveform shape is well defined but its PCA cluster is difficult to distinguish. Waveforms classified as 1 were low amplitude with waveform shapes slightly more distinct from the multi-unit activity. Sort 0 contain any multi-unit activity or any additional unidentified threshold crossings (red unit in Fig 4.1).

4.4 Results

We sorted the neural activity on the array, excluding three channels due to noise. Of the remaining 93 channels, we found 59 channels with a single isolated unit, 14 channels with two units and two channels with three single units. All channels also had multi-unit activity present. The single-unit and multi-unit waveforms were distinct across the array (Figure 4.2). Sort 2 was the most prevalent of the single unit activity while well-isolated units were the rarest (Table 4.1).

We first established a baseline to assess the asymmetries. We used the original unsorted neural data to find trajectory asymmetries for each of the target pairs. We implemented an optimization algorthim to identify projections with the largest asymmetries (see method 5.3.1). All six target pairs had large asymmetries (Fig. 4.3).

To assess the impact of spike sorting, we identified five sorting configurations. The configurations were 1) multi-unit activity only, 2) all sorted activity, 3) single unit activity only (sorts 1-4), 4) sorts 2-4 only and 5) well-isolated single units (sorts 3-4). We applied the same procedure on each configuration to identify asymmetries. For all configurations we found asymmetries across the target pairs. Using only multi-unit activity (1), we found

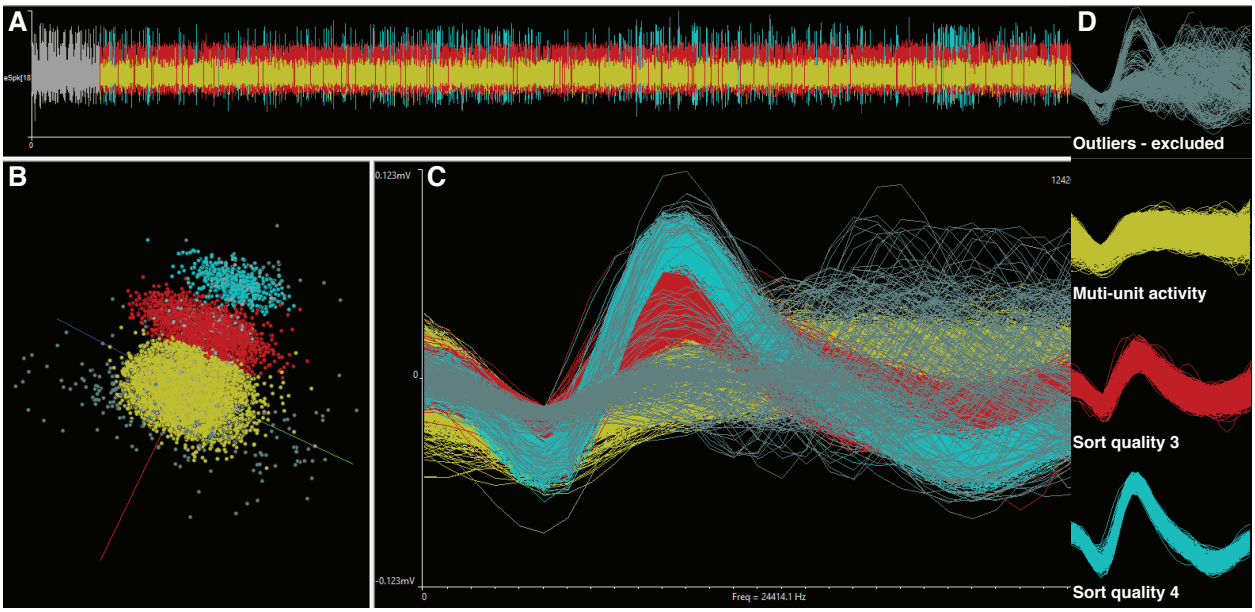

Figure 4.1: Spike sort criteria. A screenshot of the sorted information for channel 18 of Monkey N. *a)* A temporal snapshot of the identified snippet information. Snippets are colored by sort. *b)* The separation of sorts in the PCA space *c)* The snippet information of all sorts plotted on top of each other. *d)* The individual sorts observed on this channel labeled according to how the sorts were classified: Grey - outliner, yellow - multiunit, red - sort quality 3 and cyan - sort quality 4.

that asymmetries identical to the baseline asymmetries (Fig 4.3). This was expected given the large overlap of neural activity used. For the all sorted configuration (2), we created a neural space separating the multi-unit and single unit activity for each channel. We found that some asymmetries became stronger (ex: the red, green and purple, Fig. 4.3C) and others weaker (cyan, navy, and grey). This demonstrated that combining activity can both dampen and highlight asymmetries. This effect is likely driven by the relationship between single and multi-unit activity. We suspect that combining correlated activity enhances the asymmetries, while uncorrelated activity diminishes the asymmetries.

Figure 4.2: Spike sorted waveforms for Monkey N. A summary of the sorted waveforms where the mean and standard deviation of the waveforms were plotted. Waveforms are separated by electrode number and colored according to their sort quality. Navy - sort 0 (multi-unit), blue - 1, teal - 2, gold - 3, and yellow - 4;

Table 4.1: Sort quality summary for the example session from Monkey N

Sort quality	**Number of units**
0 - Multiunit activity	93
1	24
2	46
3	15
4 - Best isolation	8

Our well-isolated population also showed the asymmetries consistent with the presence of constraints. However, the asymmetry strength decreased as we decreased the number units (Fig 4.3D-F). To assess this strength discrepancy, we analyzed the firing rates and tuning for all sorts. Single units had lower average firing rates than the multi-unit activity (Fig. 4.4). The reduced firing rates meant that we observed lower modulation for single unit activity than multi-unit (Fig 4.5). Furthermore, we did not appear to have a uniform distribution of tuning across the units. These results suggest that our well-isolated population was too small to capture the dynamics. A larger sample size of well-isolated units would likely identify asymmetries comparable to the baseline asymmetries.

4.5 Discussion

We examined the impact of spike sorting on identifying dynamic structure. We sorted the neural activity from a single session and classified the sorts from multi-unit to well-isolated single units. We found that trajectory asymmetries were present in both the sorted and unsorted data. Furthermore, the identified asymmetries were comparable between the two groups. Finally, we identified a neural space using a very limited population of extremely

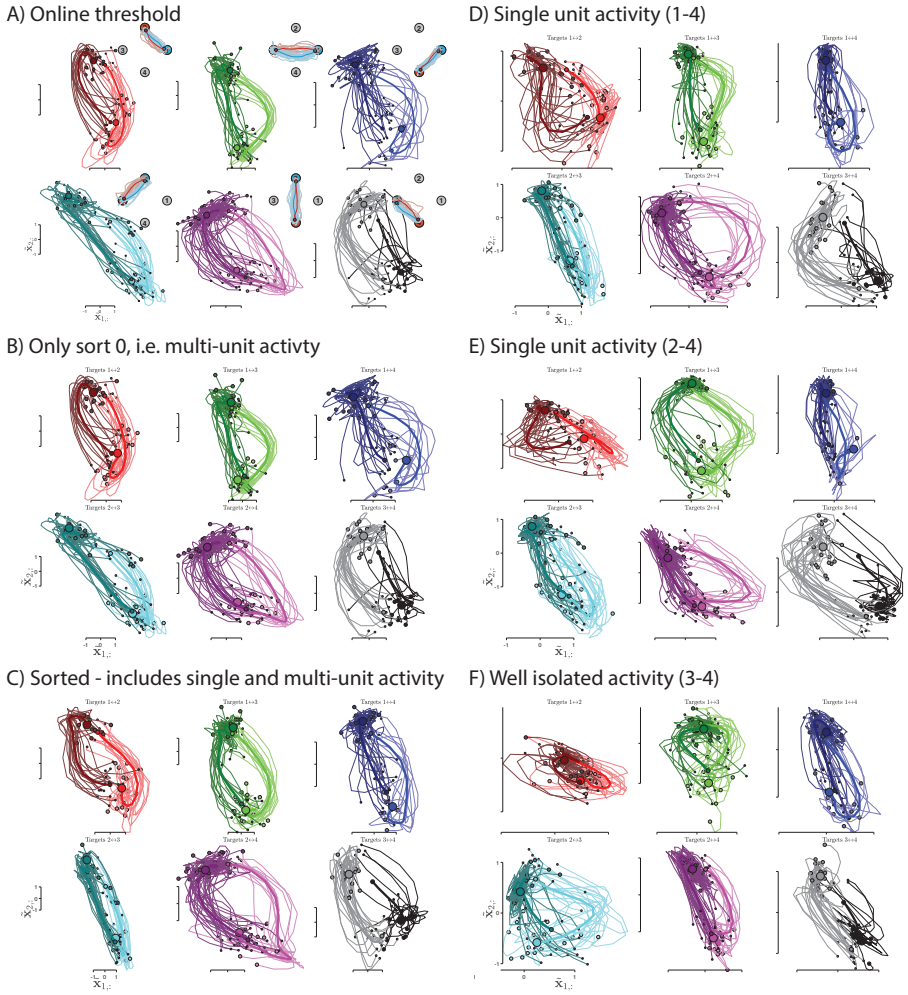

Figure 4.3: Neural dynamics for different spike sorting conditions. Each panel shows the optimal 2D projection for that condition. These projections highlight the separation of the neural trajectories and were identified using only the given condition's data. *a)* Online control method. No sorting of the neural activity. All activity that meets the 3.0xRMS threshold are combined into a single sort of each channel. The insets show the online cursor trajectories for all trials of each condition. The average trajectories are also plotted in bold. **b)** Multi-unit activity only. We only used sorts that didn't have a clear neural waveform. **c)** We treated all sorts as individual units to identify the neural space and projection. **d)** Only uses the single unit activity. **e)** A tighter criteria for including single unit activity. **f)** Neural space and projection were identified using only the most well isolated units.

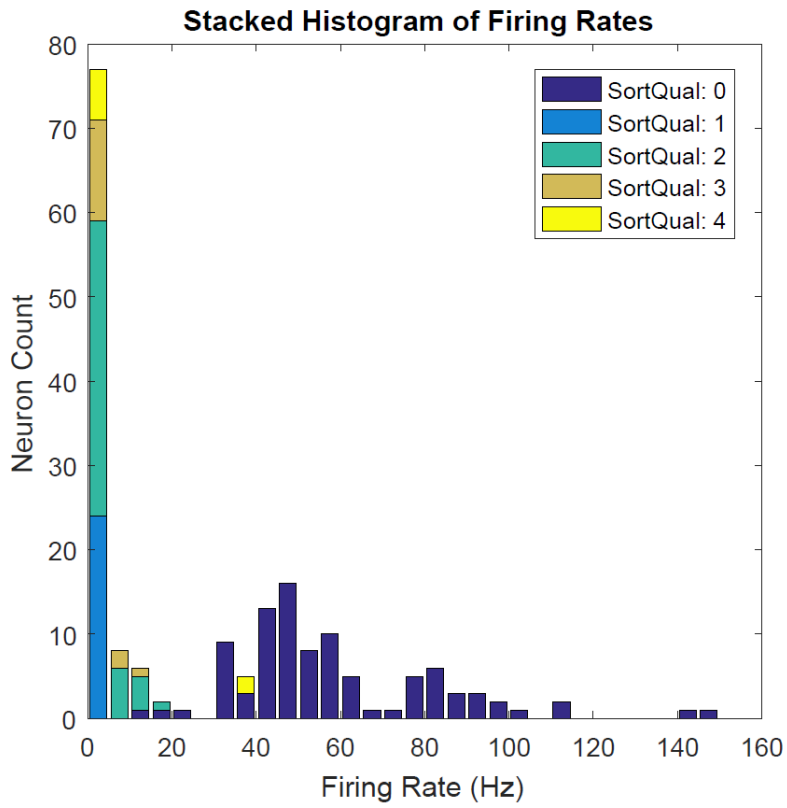

Figure 4.4: Distribution of sorted firing rates. A stacked histogram of the average firing rates for each sort identified in the Monkey N example session. Single-unit activity was low firing while multi-unit activity was variable.

well-isolated units. Asymmetries were still present, though they were weaker asymmetries. We found that these well-isolated unit had lower modulation and tuning. This likely caused the discrepancy in asymmetry strength. This result is consistent with a recent study published by Trautmann et al, where they found that similar neural population dynamics for sorted and unsorted data [64].

Since a key aspect of our experiments requires probing perturbed control, we need the animals to have the best control before the perturbation. Otherwise, how could we separate between neural constraints and poor-control? Lower tuning could poorly impact the animal's control of the cursor. It could also limit which target pairs we could test. Thus, we concluded

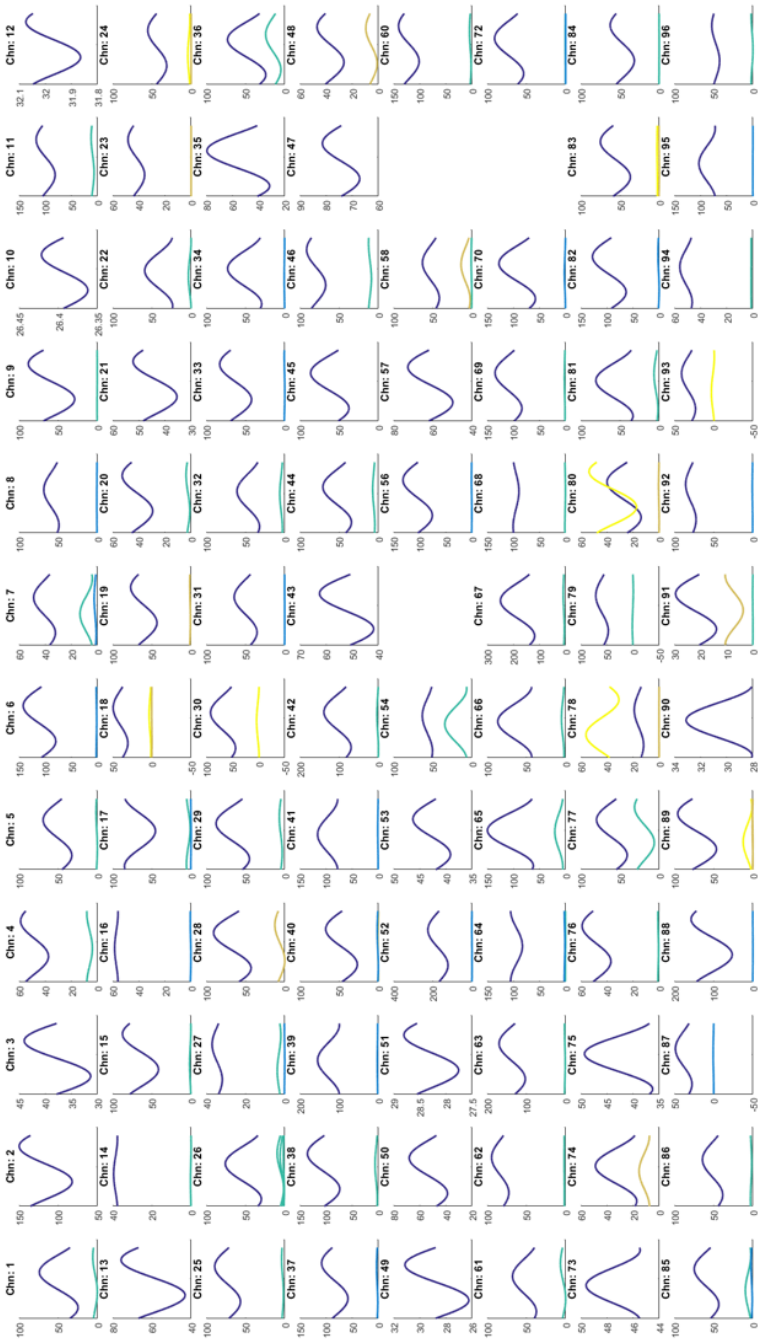

Figure 4.5: Spike sorted tuning curves for Monkey N. A summary of the average sorted tuning curves for all units. Tuning curves are separated by electrode number and colored according to their sort quality. Navy - sort 0 (multi-unit), blue - 1, teal - 2, gold - 3, and yellow - 4;

that it was best to continue to use multi-unit threshold crossing data to run the asymmetry experiments.

Our initial results suggest that the network constrains all neural activity. If constraints only acted on a subspace of activity, we would have expected some sort configuration would have failed to find any asymmetries. We did not observe that but rather we found large asymmetries across target pairs and sort qualities. This was a very preliminary study and there are several directions to consider for future experiments. The first step will be to extend this analysis to other sessions and animals. Is there a difference in results when PMd units are also on the array (like for Monkeys E and Q)? Assuming all other features are equal, is there a relationship between the sort quality and asymmetry strength? Finally, would violating the asymmetry (see Chapter 5) be more difficult when only using single unit activity? Understanding these relationships, may help us further tease apart the network mechanisms acting on the time course of neural activity.

5.0 The Persistence of Constraints on the Neural Time Course

5.1 Summary

Having demonstrated that dynamics are present in M1, we next sought to assess whether they were inherent to the M1 circuit. We predicted that if the observed dynamics are inherent than they will persist even when challenged to change. To do this, we changed the BCI mapping so that the animals received visual feedback of the dynamics. Presenting the visual feedback was not enough to change the trajectory shape. We next encouraged the animals to follow paths that went against the grain of the temporal sequence. We found that animals were unable to modify the time course of their neural activity, despite incentive to do so. These results support the theory that neural trajectories reflect underlying network mechanisms.

5.2 Introduction

While many groups have demonstrated the existence of neural dynamics throughout the brain, few studies have causally tested them [9, 13, 31, 38, 40, 41, 65]. This has been due to a limitation in methods to finely perturb the neural activity. Studies using electrical [40] or optogenetic [13] stimulation have limited specificity and variability respectively in the evoked population responses. Additionally, complex behavioral studies are not possible in rodent models [41], thus limiting how we can relate rodent result to human movement. By taking advantage of how we define our BCI mapping, we could causally perturb the population neural activity. Through the use of BCI, we are able to directly probe the neural activity of M1.

We defined a "rotated" mapping that maximize the neural activity separation that we characterized in chapter 3. We then tested two conditions to see if the neural activity would change. First, we tested if the animals would straighten the cursor trajectories when given

visual feedback of the neural curvature. We predicted that if the observed dynamics were arbitrary to the activity, the animals would be able to straighten the cursor path. This straightening behavior occurs during visual motor rotation (VMR) experiments for both human and non-human primate (NHP) subjects [66–72].

Second, we required the cursors to follow an instructed path distinct from the flow. Specifically, we designed paths that were time-reversed sequences of the observed cursor trajectories. Since we know that the animals can produced these activity patterns, we are able to focus on if the temporal order of activity patterns matter. If the observed time course is not inherent to the circuit, then the animals can reverse of the order of activity to follow the path.

In both cases we found that the neural activities time course remained unchanged. These results show first that the activity's time course is not arbitrary and second that volition was not sufficient to change the time course.

5.3 Methods

All experiments were collected using the rotated BCI mapping (see below, section 5.3.1) unless otherwise specified. The rotated mapping experiments were always collected after a block of either conditional grid or two target task using intuitive BCI mapping. These intuitive mapping trials were always used to help identify the rotated mapping. All rotated mapping experiments also included at least one 100-trial block of the two target task using rotated BCI control.

We collected 168 total sessions of the two target rotated mapping experiment (Monkey E - 86 sessions, Monkey D - 61, and Monkey Q - 21). Given discrepancies in experimental methods, we defined methodical criteria that sessions had to match in order to be included the population analysis for this chapter. We therefore included 111 sessions (Monkey E - 50 sessions, Monkey D - 40, and Monkey Q - 21).

We collected 57 total sessions of the instructed path experiment (Monkey E - 35 sessions, Monkey Q - 13 sessions, and Monkey D - 9 sessions).

Note that for all monkey E sessions and 20 rotated mapping experiments for monkey D, we used different GPFA models (ie low dimensional neural spaces) with the intuitive and rotated mappings. These models were fit to the same set of trials used to find the mapping parameters. In the case of the intuitive mapping this involved updating the GPFA model parameters for each intuitive mapping update (see section 2.5.2). For monkeys D and Q, the same GPFA model fit during intuitive mapping calibration was used with the rotated mapping.

5.3.1 Defining the BCI Rotated Mapping

The rotated BCI mapping was designed to highlight projections of neural activity that maximized our desired projection features. Our three criterion features were separation between target pair, separation between trajectory midpoints, and variance at the midpoints (Fig 5.1B). We used an optimization-based procedure to minimized an objective function parameterizing these features [21, 73, 74]. Specifically, we sought to find an orthonormal set of vectors $P_{rot} = [\mathbf{p}_1, \mathbf{p}_2] \in \mathcal{R}^{p\times 2}$ which minimized the objective function:

$$J = -w_{mid}\mathbf{p}_1^\top(\mu_{AB} - \mu_{BA}) + w_{var}\mathbf{p}_1^\top(\Sigma_{AB} + \Sigma_{BA})\mathbf{p}_1 - w_{start}\mathbf{p}_2^\top(\mu_A - \mu_B) \tag{5.1}$$

where μ_A and $\mu_B \in \mathcal{R}^{p\times 1}$ are the average starting locations for the A→B and B→A trajectories during the two-target task, μ_{AB} and $\mu_{BA} \in \mathcal{R}^{p\times 1}$ are the average midpoints of the A→B and B→A trajectories, and Σ_{AB} and $\Sigma_{BA} \in \mathcal{R}^{p\times p}$ is the covariance of the neural trajectories at the midpoint of the A→B and B→A trajectories (See Fig 5.1C for a visualization).

Weighting factors w_{mid}, w_{var}, and w_{start} were used to weight the influence of the mid point, covariance, and starting point terms on the overall objective function value. For monkey E, each of these terms was set to 1, while for monkeys D and Q, the weighting factors were set such that the corresponding terms in the objective function had the same range when evaluated over 10000 different random orthonormal projections P_{rot}; we found that this resulted in more consistent asymmetries for these monkeys. Objective function minimization proceeded as outlined by Cunningham and Ghahramani [73], with a convergence criteria of

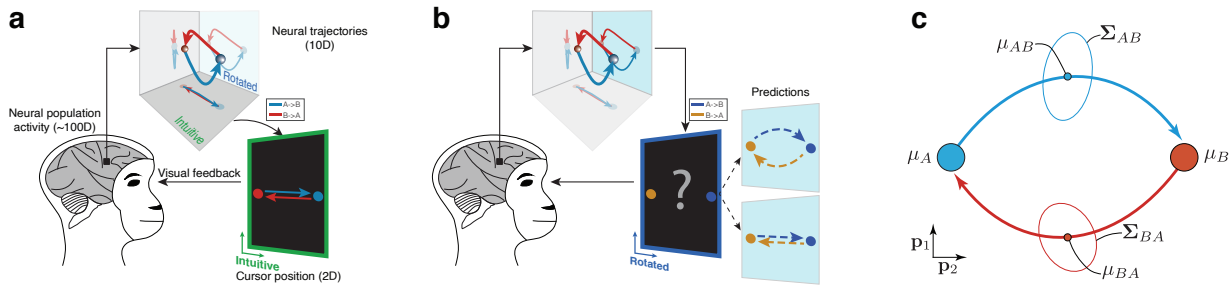

Figure 5.1: Conceptual illustration depicting different BCI mappings. *a)* Animals move the BCI cursor between two targets using the intuitive mapping, i.e. the gray projection of the multi-dimensional neural trajectories. *b)* Animals control the BCI cursor using the rotated mapping, i.e. projection with greatest trajectory asymmetry (blue). The existence of flow-field-like dynamics would predict that the trajectory asymmetry should persist (top prediction). *c)* Conceptual illustration of the features that define our objective function to identify trajectory asymmetry.

$\Delta J = 10^{-10}$, a maximum of 1000 gradient iterations, and a line search step size of 0.1. All values were calculated using the ~ 100 target pair trials found in the conditional grid task.

The resultant set of projection vectors $[\mathbf{p}_1, \mathbf{p}_2]$ defined the 2D subspace in which the trajectory asymmetry was largest. To determine the final set of rotated mapping parameters (W_{rot} and $\mathbf{c}_{rot}$), this space was aligned with the animals' workspace such that the starting point of the A→B and B→A trajectories in the rotated mapping were at the same position as targets A and B in the cursor workspace.

$$A = OR_\theta S \tag{5.2}$$

$$W_{rot} = AP_{rot}^\top \tag{5.3}$$

$$\mathbf{c}_{rot} = -A\mathbf{z}_c \tag{5.4}$$

where $R_\theta \in \mathcal{R}^{2\times 2}$ is a matrix rotating P_{rot} by θ, $S = sI \in \mathcal{R}^{2\times 2}$ is a diagonal matrix scaling the axes of P_{rot}, $O \in \mathcal{R}^{2\times 2}$ is a matrix that could be used to reflect the axes of W_{rot} as desired, and $\mathbf{z}_c$ is the average of μ_A and μ_B. Matrix S was used to scale P_{rot} such that the distance between average latent states μ_A and μ_B was equal to the locations of targets A and B in the workspace. Matrix O was used to make the rotated mapping as intuitive for the animal to control as possible, specifically by changing the sign of $\mathbf{p}_1$. The decision to change the sign of $\mathbf{p}_1$ was based on visual inspection of neural trajectories to the perpendicular targets during the conditional grid task. The neural trajectories were projected into the rotated mapping and the form of O was chosen such that the endpoint of these trajectories was closest to the actual target location (Fig. 5.2). While unnecessary for the visual feedback experiment, this design choice did reduce the cognitive burden imposed on animals during the instructed path experiments.

5.3.2 Fixed Timescale Decoders

To create the fixed timescale decoders, we redefined the covariance matrix described in Eqn. 2.2 from

$$K_i(t_1, t_2) = (1 - \sigma_i)e^{\frac{-\gamma_i}{2T_{diff}^2}} + \sigma_i * \delta \tag{5.5}$$

to

$$\begin{aligned} K_i(t_1, t_2) &= (1 - \sigma_i)e^{\frac{-\overline{\gamma}}{2T_{diff}^2}} + \sigma_i * \delta && (5.6) \\ \overline{\gamma} &= (\frac{45}{\overline{\tau}}) && (5.7) \end{aligned}$$

where $\sigma_i \in \mathcal{R}_+$ is the signal variance, T_{diff} is the timing difference, and $\overline{\tau}$ is the average τ for all latents [58]. This change would propagate to the smoothing matrix, M.

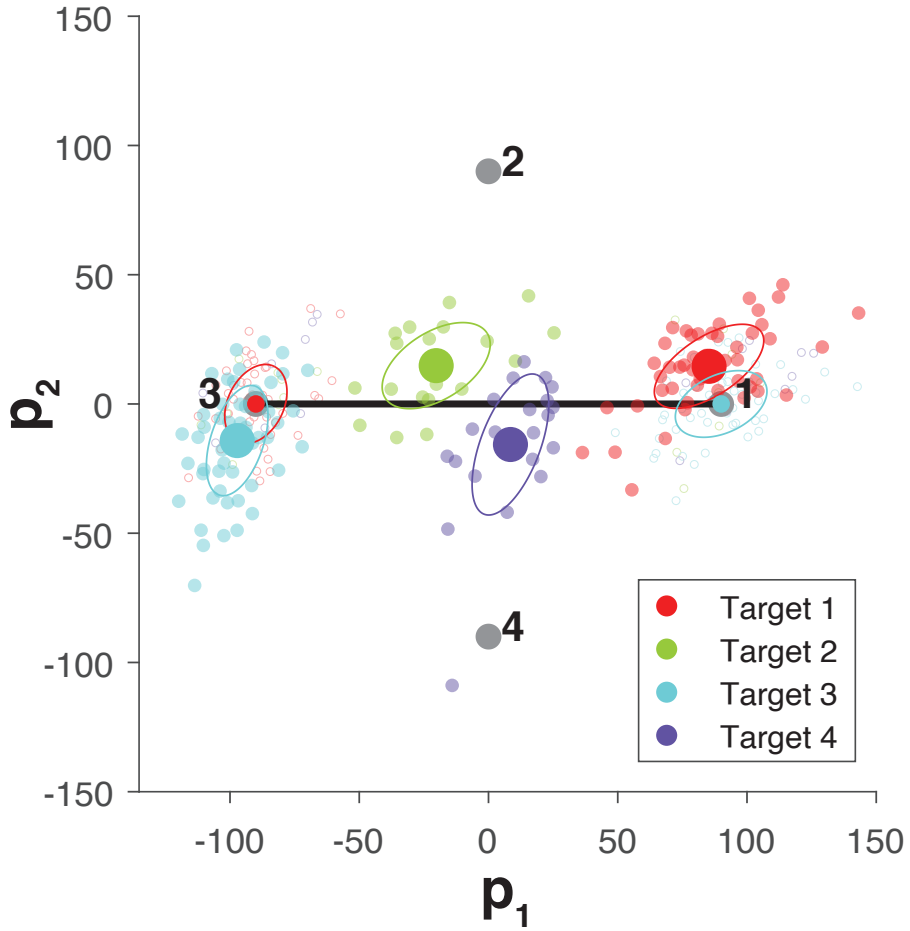

Figure 5.2: Predicted rotated trajectory endpoints for a conditional grid session. The intuitive mapping trials from session E54 are passed through the rotated BCI mapping. The first and last timepoint for each trajectory is then plotted in the workspace colored according to the end point. The small hollow dots near targets 1 and 3 are the first timepoints of individual trials and the larger markers are the last timepoints. The mean and covariance ellipse are also plotted. For this example session the neural activity for target 2 and target 4 are closer to their respective workspace location. This means that we did not flip the sign of $\mathbf{p}_1$.

5.3.3 Behavioral Task - Instructed Path

See figure 2.6A and B for visual

The purpose of this task was to test if the animals were able to overcome the temporal time course of their behavior when required to do so by a visual boundary. The task was a two-step target acquisition task performed under the rotated mapping BCI control. The first step was to acquire one target of the target pair. The second step was to acquire a target, called the intermediate target (IT), placed along the axis orthogonal to the target pair axis. The IT location was selected each day so that the animal could 1) acquire it from both start target and 2) acquire it with a straight ballistic trajectory from one of the start targets.

Following a 100 trials with no visual restraint and the intermediate target fixed, a visually enclosed tube was displayed on the screen and enforced a boundary on the animal's behavior [47]. The tube size is initially selected based on the task performance with the rotated mapping then reduced over the block. There were two different protocols to reduce the tube boundary.

In the first protocol, the tube constraint was reduced based on performance, with only a single IT was tested. The boundary's radius was reduced in 10mm increments if the animal was over 75% successful (first in 25-trial blocks, then in 50-trial blocks once the success rate failed to reach the threshold for a given block). For half of the Monkey Q sessions, we varied the radius reduction so that there was a predicted 25% reduction in the animal's performance with the new tube. Only one start target direction was tested at a time for 500 trials (Fig 5.10).

5.3.4 Calculating the Initial Angle

See purple trace in figure 5.11B for visual Our initial angle metric measured the angle difference between the target axis (i.e. start target, ST, to end target, IT) and the initial cursor trajectory (Fig 5.11B - purple metric). To calculate the value, we identified the unit vector between the start target (ST) and the intermediate target (IT) and the unit vector between the first and fourth time point of the average cursor trajectory (x_c). We then calculated the angle between the two vectors.

$$\vec{v} = < ST_x - IT_x, ST_y - IT_y > \tag{5.8}$$

$$\vec{x_c} = < x_{x,4} - x_{x,1}, x_{y,4} - x_{y,1} > \tag{5.9}$$

$$\theta = arccos(\frac{\vec{v} \cdot \vec{x_c}}{|\vec{v}||\vec{x_c}|}) \tag{5.10}$$

This resulted in a single value for condition in a session, with θ calculated in degree. To compare the angle change across experiments, we normalized the initial angle metric θ:

$$\hat{\theta} = \frac{\theta_{unres} - \theta_{res}}{\theta_{unres}} \tag{5.11}$$

where θ_{unres} and θ_{res} are the initial angles for the unrestricted trials and tightest restricted trials. A normalized initial angle value of 0 would mean that there is no change the initial angle. Less than 0 would means the initial angle increased with the tube. A value of 1 would mean a complete straightening of the trajectory to the target axis.

5.3.5 Calculating the Trajectory Bowing

See green trace in figure 5.11B for visual We defined trajectory bowing as the orthogonal distance of the cursor trajectory from the axis between the start target (ST) and the end target (IT). To calculate the amount of trajectory bowing, we identify the unity vector ($(\vec{u})$) and the perpendicular vector ($\vec{u_\perp}$)) of the target axis . We then found the orthogonal projection for each average cursor trajectory time point (x_c) onto that axis.

$$\vec{u} = \frac{\vec{v}}{|\vec{v}|} \tag{5.12}$$

$$\vec{u_\perp} = \vec{u}^T * \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{5.13}$$

$$x_{c\perp} = \vec{u_\perp} x_c \tag{5.14}$$

where $x_{cursor} \in \mathcal{R}^{1\times N}$, and N is the number of time points. This resulted in a distribution of bowing distance. For comparison, we used the max of this distribution, since we care about the excursion from the optimal, i.e. straight path.

To compare bowing across experiments, we normalized the bowing metric b:

$$\hat{b} = \frac{b_{unres} - b_{res}}{b_{unres}} \tag{5.15}$$

where b_{unres} and b_{res} are the maximum trajectory bowing values for the unrestricted trials and tightest restricted trials. A normalized bowing value of 0 or less than 0 would mean that there is no change in the bowing or the bowing got worse with the tube. A value of 1 would mean a complete straightening of the trajectory to the target axis.

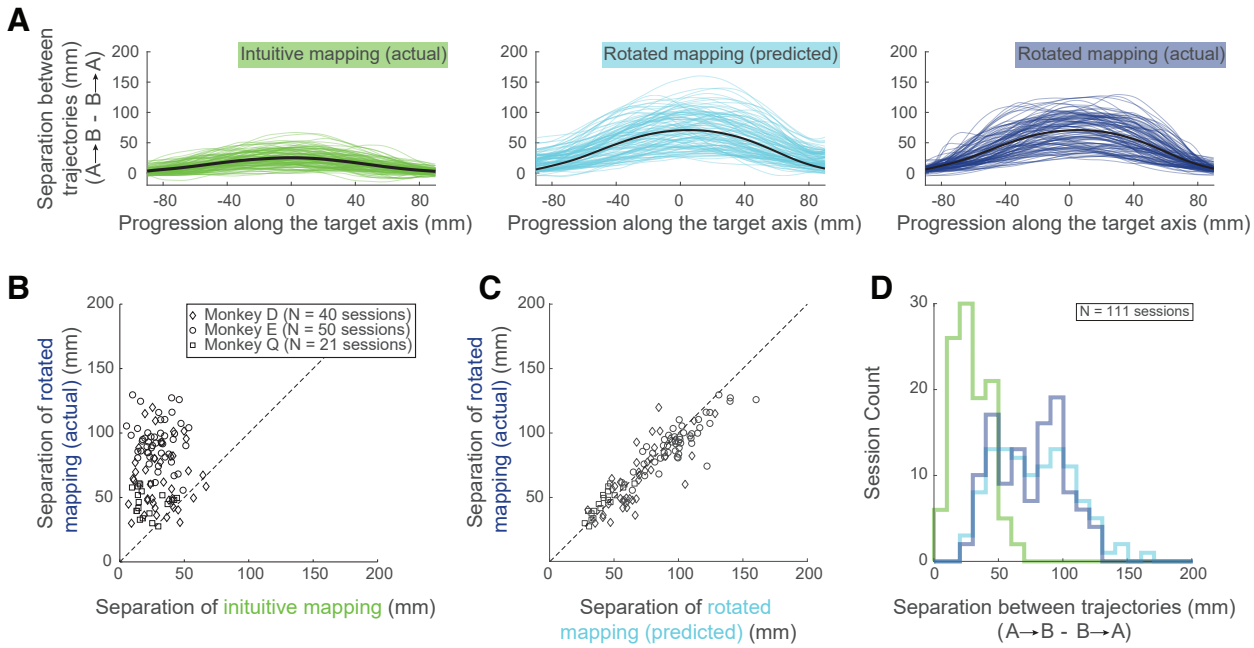

Figure 5.3: Quantifying 2D distance between trajectories. *a)* The separation between trajectories A→B and B→A for the online intuitive BCI mapping, the predicted rotated mapping, and the online rotated mapping. Single trials examples are plotted in green, light blue, or navy. Averages for each condition are plotted in black. *b)* A scatter comparison of the online rotated mapping and online intuitive mapping trials. Each symbol represents the maximum average separation for each session. The dash line is the unity line. *c)* A scatter plot of both intuitive and rotated mapping trials projected into the rotated mapping. *d)* Separation histograms for all conditions of all sessions. Online intuitive mapping histogram is plotted in green, the predicted rotated mapping sessions in light blue, and the online rotated mapping in navy.

5.4 Results

5.4.1 Distinct Neural Time Courses Persist with Visual Feedback

In chapter 3, we observed that the high-dimensional neural trajectories were distinct. That is, neural trajectories took a different path moving from target A to B than when moving from target B to A . We then sought to determine if the observed separation was meaningful to the circuit. To do this, we looked for two-dimension projections exhibiting strong trajectory asymmetries. These projections defined a rotated BCI mapping to probe the neural time courses. Specifically, the rotated mapping provides visual feedback of the asymmetries. If asymmetries are arbitrary artifacts of behavior, then they might change with altered feedback. However, if the they are products of the network connectivity then they would persist.

Using the intuitive mapping trials (IM), we identified the optimal rotated projection. We passed IM neural activity through the rotated mapping to create predicted trajectories. We observed little visual overlap of the trajectories through the mapping (Fig. 5.6A, *top row*). We also determined the separation between the two conditions for each mapping. To do this, we found the maximum distance between the two average trajectories. The separation of the conditions was significantly greater for the rotated mapping than the intuitive mapping (one-sided paired t-test: $p << 0.05$, Fig 5.3D)). These results demonstrate that the mappings are distinct projections of the neural activity with limited overlap (Fig. 5.6A).

Animals were then asked to control the BCI cursor using the rotated mapping. This meant the cursor was now driven by neural activity in the 2D subspace with the strongest trajectory asymmetry (Fig. 5.6B). Rotated mapping control performance was high and comparable to the intuitive mapping control. Acquisition times were also consistent between the two mappings (Fig. 5.4) The rotated mapping cursor trajectories exhibited strong asymmetries highly consistent with the predicted trajectories (Fig 5.3C-D).

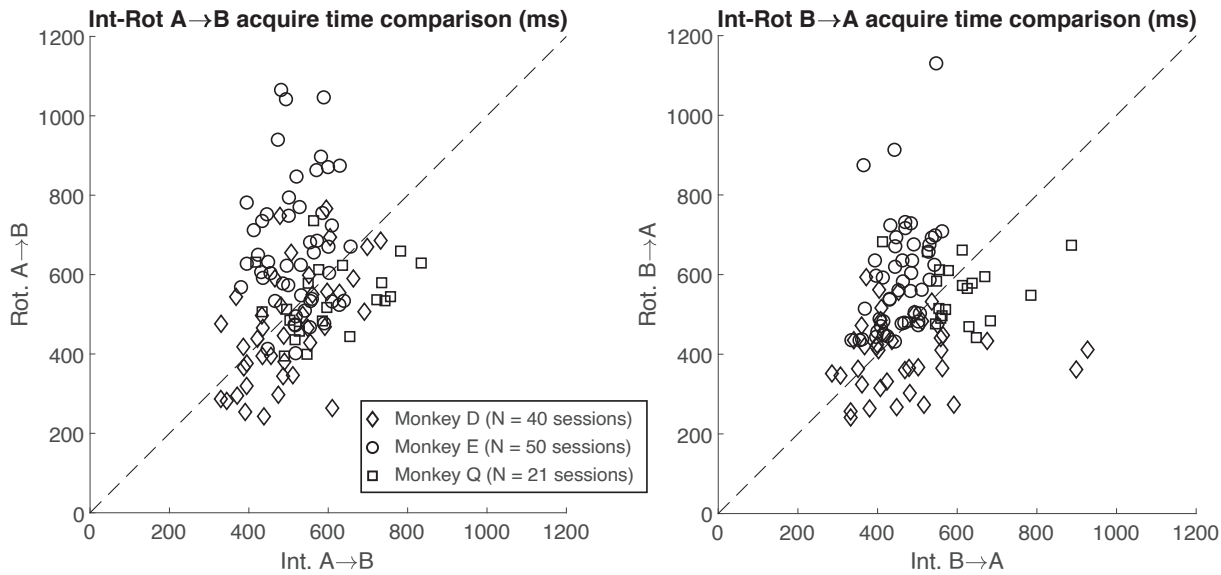

Figure 5.4: Comparison of target acquisition across mappings. *a)* A scatter plot of both intuitive and rotated mapping mean acquire times for the A→B. *b)* A scatter plot of both intuitive and rotated mapping mean acquire times for the B→A.

To quantify differences in the intuitive and rotated mapping control, we compared both distances and flow fields. For the distance metric, we found a linear relationship between the predicted and actual rotated trajectories (5.3C). The distance distributions were significantly different from the intuitive trajectories, but not each other (t-test rot. vs int.: $p << 0.05$, t-test pred. vs int.: $p << 0.05$, t-test rot. vs pred.: $p = 0.327$).

For the flow field metric, we compared the cursor velocity as a function of position for each set of cursor trajectories (see methods 2.6). The rotated-actual cursor velocities were more similar to the rotated-predicted velocities than the intuitive mapping velocities (Fig. 5.6C). Our results show high consistency within at least one two-dimensional projection (i.e. the rotated mapping).

We next asked if the conditions were consistent throughout the 10-dimensional neural space. To assess this, we compared the conditions for hundreds of random two-dimensional projections. In each random projection, flow fields were fit to both intuitive and rotated

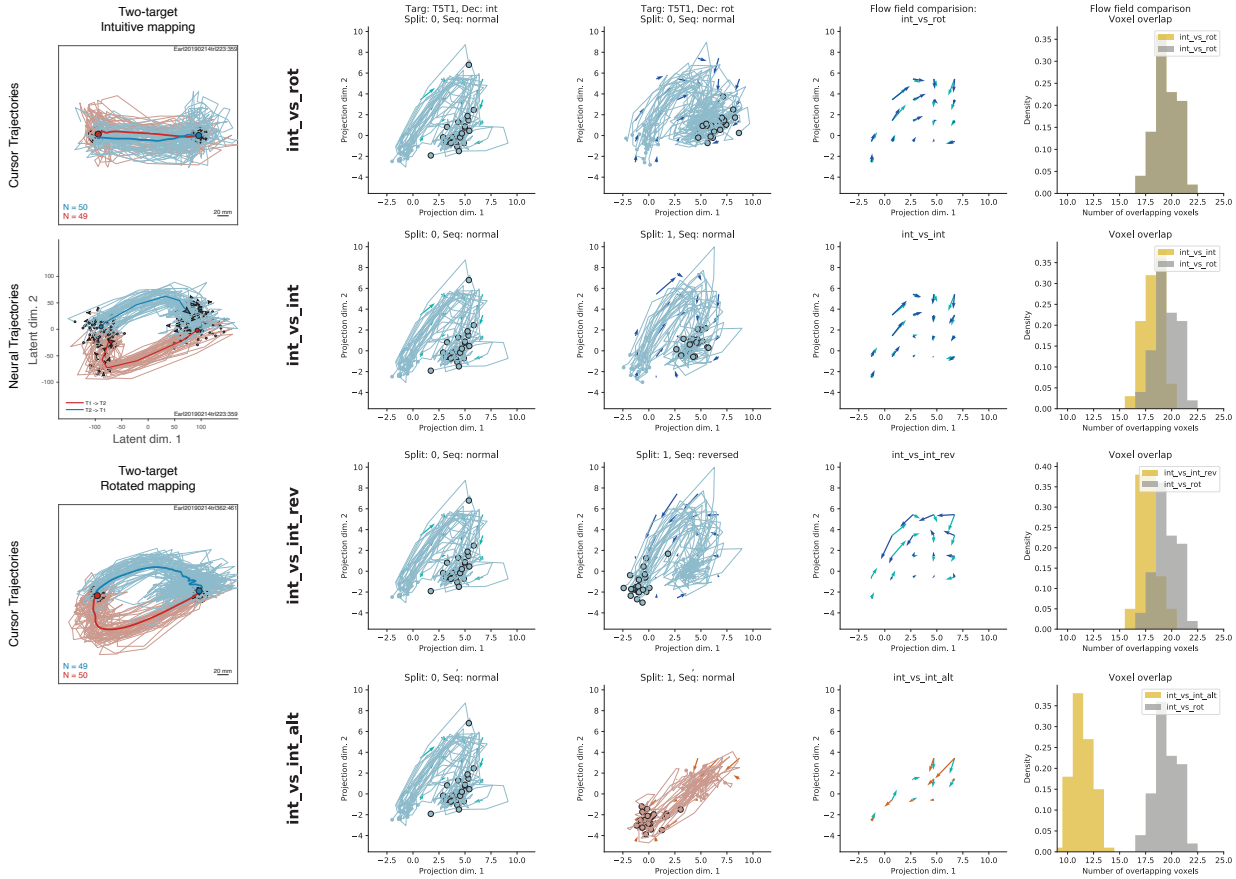

Figure 5.5: Flow field overlap comparison for a single random comparison. Each row represents a different comparison. *1st Column* The intuitive and rotated mappings for the session. *2nd/3rd Column)* Flow field for the split trials. *4th Column)* Combined flow field. *5th Column)* Distribution of voxel overlap.

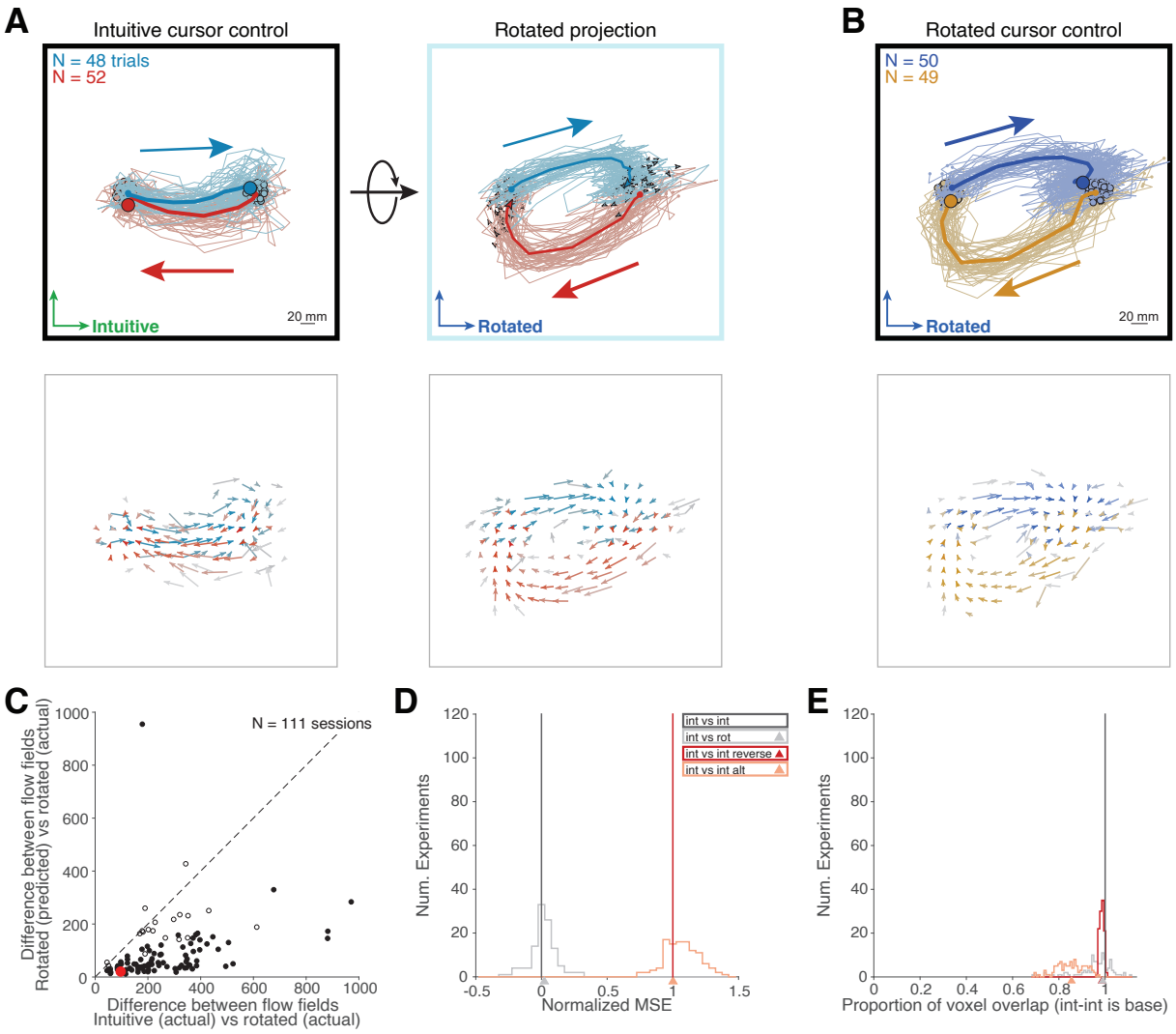

Figure 5.6: Neural trajectories are similar for different visual feedback conditions. *a)* The intuitive mapping trials for the example session from figure 2. (Left) Cursor trajectories and flow fields (*bottom*) are overlapping. (Right) The intuitive mapping neural activity projected into the rotated mapping. There is minimal overlap of the trajectory flow fields. *b)* Cursor trajectories during BCI control using the rotated mapping. *c)* Summary of cursor trajectory flow field comparisons across experiments. Filled markers are significant. *d)* Normalized flow field differences for all experiments. Normalized flow values of 0 and 1 correspond to fixed-feedback and time-reversed conditions, respectively. *e)* Normalized flow field overlap for all experiments. Overlap values are normalized relative to the fixed-feedback condition.

mapping trials. We then compared four different flow field configurations to determine similarity between the conditions (see methods 2.6.2). The first configuration compared the two feedback conditions, which we called variable-feedback. The fixed-feedback compared flow fields for the same feedback condition. The time-reversed configuration compared observed trajectories and a time-reversed version of the neural trajectories for the same feedback condition . Finally, the alternate-target compared the flow fields for two different target directions (A→B and B→A) for the same feedback condition.

For each experiment, we examined the distribution of total MSE for the four comparisons (Fig. 5.5). We found that distributions of flow differences were highly similar for the variable-feedback and fixed-feedback comparisons, and were different for the time-reversed and alternate-target comparisons. To summarize these results across experiments, we compared normalized distributions of MSE and flow field overlap (see methods 2.6.2). Normalized flow MSE was close to 0 across experiments for the variable-feedback condition (Fig. 5.6D), while normalized flow field overlap was close to 1 (Fig. 4f). These results indicate that neural trajectories were highly stereotyped in all dimensions, regardless of the visual feedback provided to the animal.

5.4.2 Observed Time Courses Persisted in Control Experiments

We next assessed if the neural time courses were an artifact of behavior preferences or experimental design. For a set of experiments, we tested both orientations of the rotated BCI mapping, i.e flipping Matrix O (See 5.3.1). We observed that the orientation of the cursor trajectories always matched the orientation of the mapping. We compare the neural activity of the two mapping orientations (rotated and rotated-flipped) using the flow field analysis. Specifically, we passed the neural activity from the rotated-flipped mapping through the rotated mapping and calculated the flow fields (Fig. 5.7A) For all sessions, we found the flow fields of rotated and rotated-flipped to be highly similar (Fig. 5.7B). This indicated that the trajectory asymmetries were not due to the animals' visual preference.

It is possible that the observed asymmetries are an artifact of using GPFA to define our mappings. In GPFA, temporal smoothing is applied to each latent independently. This

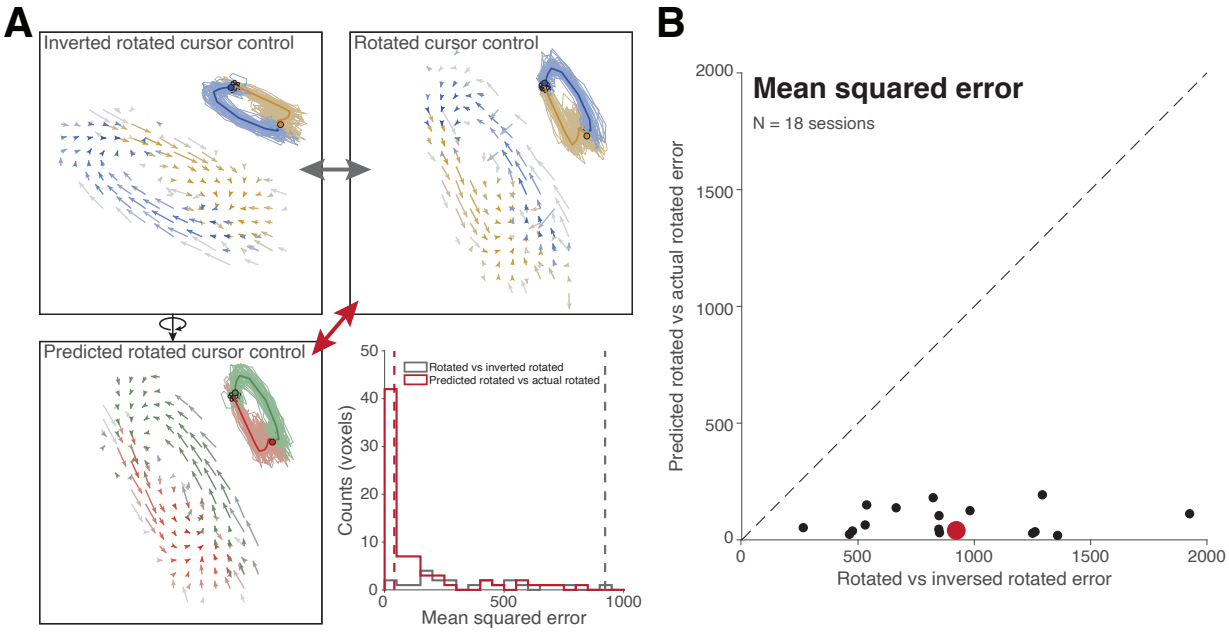

Figure 5.7: Observed asymmetries are unaffected by the mapping orientation. *a)* An example session where we tested both orientations of rotated mapping. The cursor flow fields are plotted with the cursor trajectories shown as insets. *top row)* The online cursor trajectories for each orientation. *bottom left)* One of set of trials passed through the other orientation mapping. *bottom right* The flow difference for the online control (gray) and the match-mapping condition (red). The means for each are plotted in matched dashed lines. *b)* A scatter plot of flow difference for all sessions where both orientations were tested. Online control on x-axis, predicted on the y-axis. The example from *a* is plotted in red.

means that latents will have different time constants, τ, which shape how activity traverses the neural space. To account for this, we ran experiments with Monkey E and Q where we held τ constant across for all latents (see methods 5.3.2). We then ran the two target task with the rotated mapping. For all target pairs, we observed asymmetries comparable to sessions with an unfixed τ (Fig 5.8). Their normalized 10D distances were also consistent (Fig 5.9A). As a post-hoc quantification, we identified the fixed and unfixed τ neural spaces for every two target session. We then projected the neural activity into both spaces and calculated the 10D distance (Fig. 5.9B-C). Running a paired t-test, we found that the distances were consistent between the two space. This demonstrates that the trajectory asymmetries are not an artifact of GPFA smoothing.

5.4.3 Time Course Metrics were Consistent across Target Pairs

We separated the sessions by target pair and ran the same analysis as describes in section 3.4.1. The distance and flow field metrics were similar across target pairs.The consistency of these results suggests that the observed trajectories are not idiosyncratic features of the high-dimensional activity. Nor are they due to some preference of the animal. Rather these trajectories reflect underlying network mechanisms.

5.4.4 Animals Could Not Easily Violate the Observed Time Course of Their Neural Activity

We next examined whether the animals could violate their dynamics if given incentive. To do this, we used an instructed path task to challenged animals to generate time-reversed sequences of their activity (Fig. 5.10A). If the time course of activity is reflective of underlying network constraints, then the animals should not be able to reverse the order.

First, we had the animals move the cursor from one starting target to a target positioned along the other start target's cursor path. We did not apply any restriction to how the animals acquired the target. Yet we still observed trajectory curvature consistent with the two target task asymmetry (Fig. 5.10B). Seeking to reduce the curvature, we applied a boundary around the path from start to IT [47]. This instructed path was displayed as

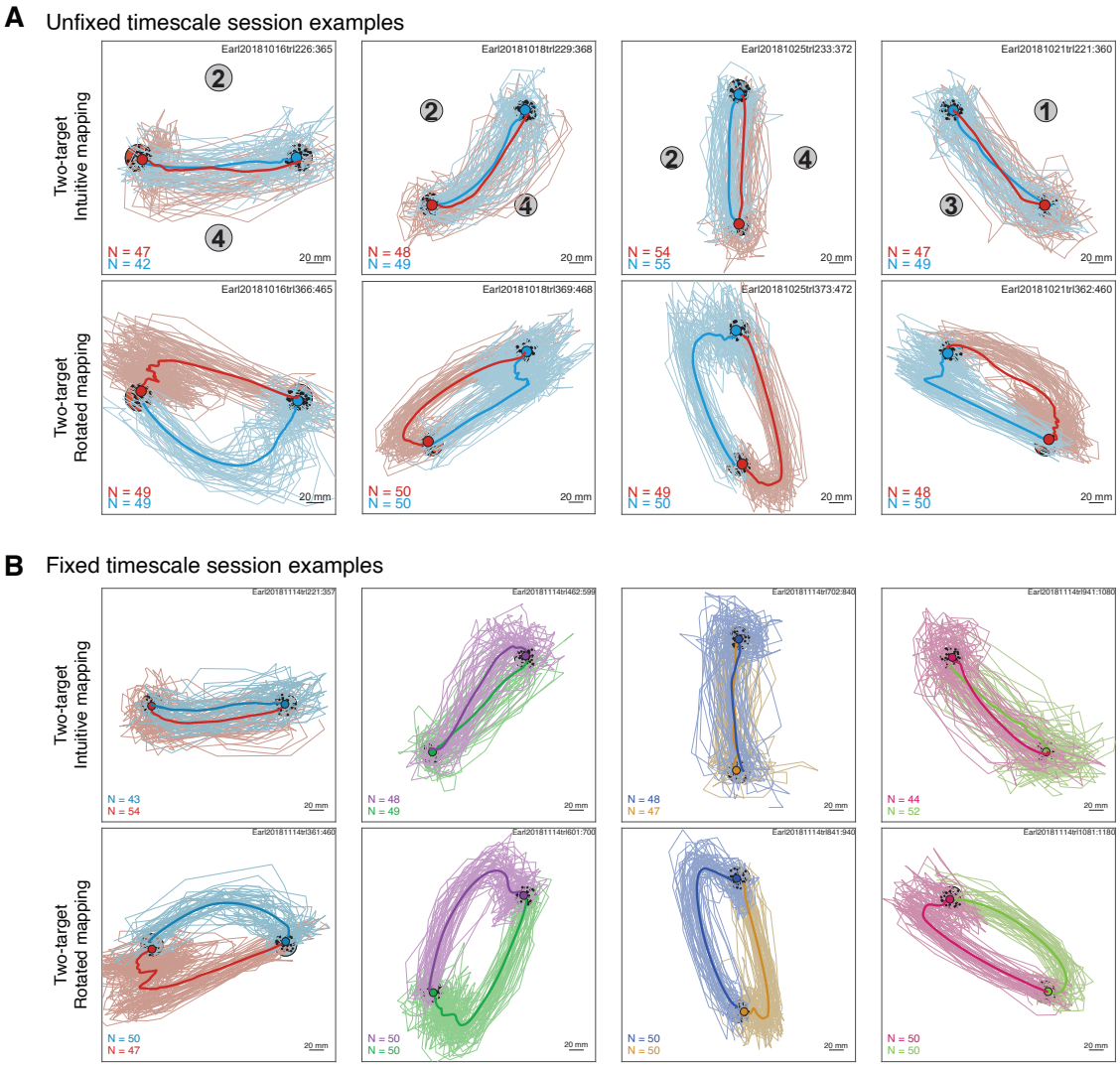

Figure 5.8: Trajectories for unfixed and constant τ values. The trajectory asymmetry persists even when the time constant, τ, is fixed. *a)* Four example sessions for the standard decoder method, i.e. τ is not constant across latents. (Top row) The intuitive mapping cursor trajectories for each target pair. (bottom row) The respective rotated mapping cursor trajectories. *b)* Four example sessions using a constant τ decoder. Top and bottom row ordering is consistent with *a*.

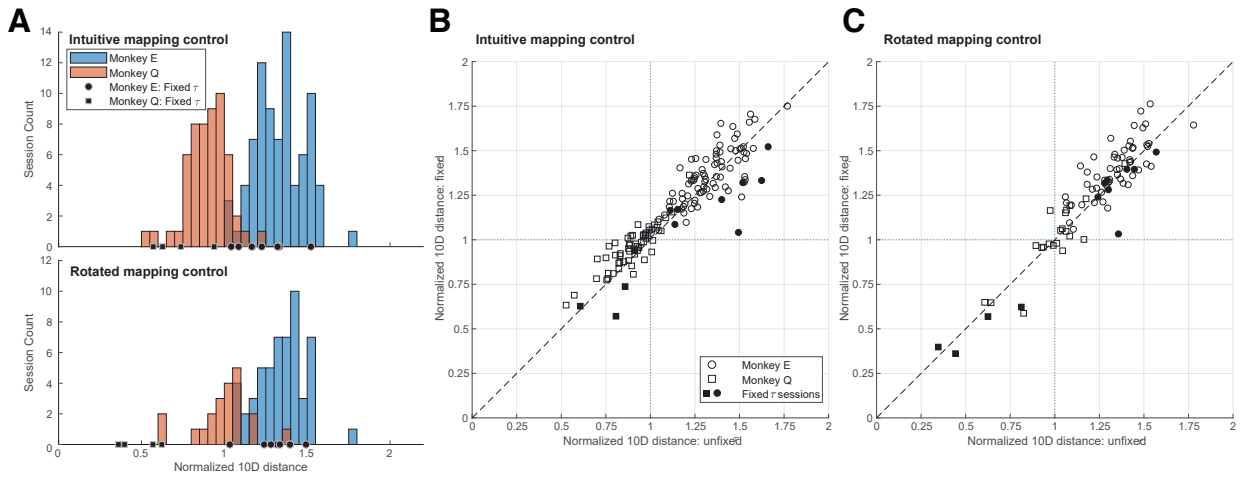

Figure 5.9: Post-hoc analysis of fixing τ. *a)* Distribution of the normalized 10D distance for the unfixed τ intuitive mapping (*top*) and the rotated mapping (*bottom*) experiments. The distances for the constant τ experiments are plotted along the x-axis. *b)* The normalized 10D distance of the intuitive mapping control sessions projected into both unfixed and constant τ neural spaces. Filled markers are for sessions that used constant τ decoder online. Hollow markers are for unfixed τ decoder sessions. *c)* The same plot as *b*, but for the rotated mapping experiments.

a fully enclosed tube. To successfully acquire the end target, the animals had to stay in bounds.

To encourage the animals to modify their activity, we gradually decreased the boundary's diameter over the course of each experiment. We linked tube diameter reductions to a success rate threshold (see methods 5.3.3). Animals did better at the instructed tube than predicted from the unrestricted trials (Fig 5.10C). This demonstrates that animals understood and responded to the boundary.

The curvature of the instructed path trajectories was compared to the unrestricted trials. In general, the cursor trajectories showed a small decrease in curvature in response to the boundary. While the extent of reduction did vary across experiments (Fig 5.11A), we never observed a time-reverse of the cursor trajectory. To quantify the observed change, we

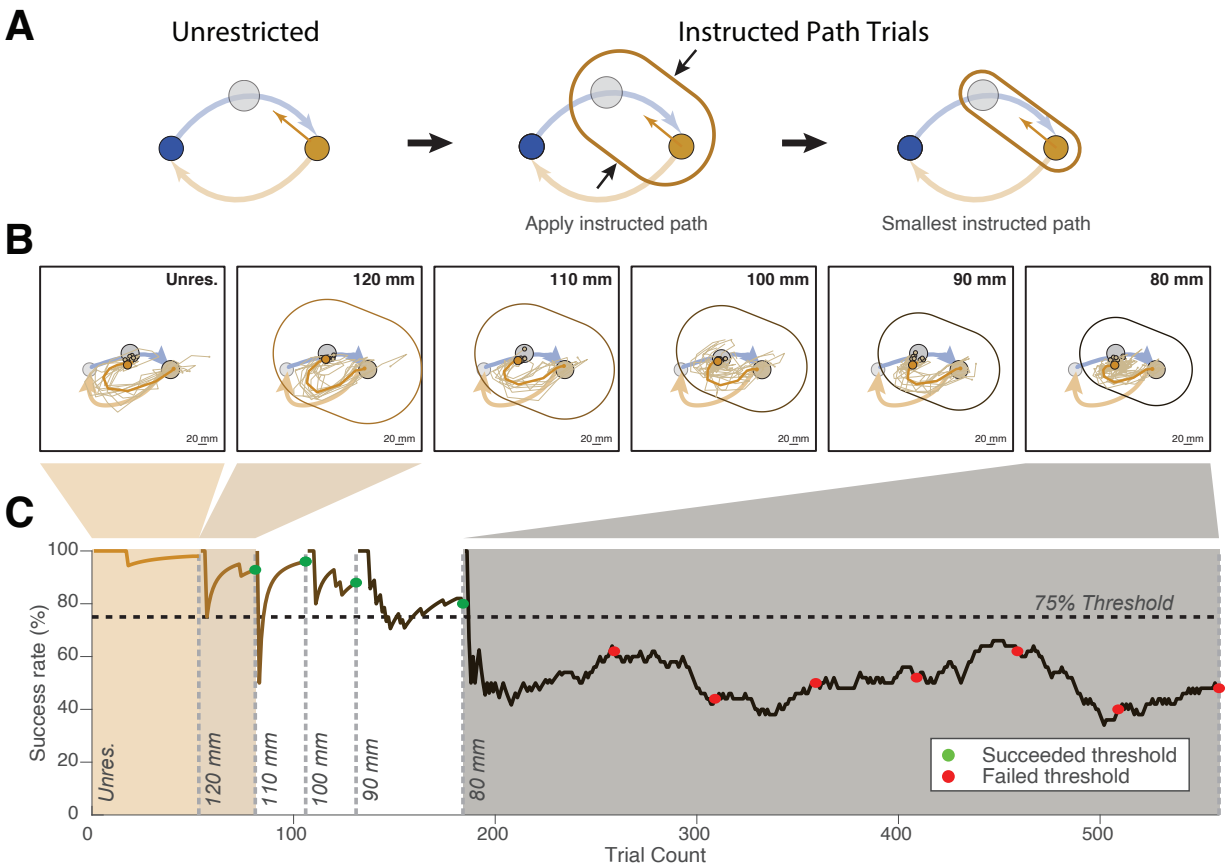

Figure 5.10: Experimental design for the instructed path task. *a)* Conceptual illustration of the instructed path task used to probe the dynamics. *b)* An example session of the instructed path task. The blocks are plotted in order with 15 randomly selected trials and the overall average plotted. The observed rotated mapping asymmetry is also plotted in the background. *c)* A moving average of the success rate. The line is a moving average of 50 trials, reset each time the tube is reduced. The markers show the success rate measured to test whether to reduce the tube size.

calculated two metrics to assess early and late stage modifications. If the trajectories were highly malleable than we would expect both early and late stage modifications (Fig 5.11B).

First, we determined the initial angle between the average cursor trajectory and the IT target. This would give us insight into if animals change their behavior prior to feedback. We calculated the vector from the first 4 time bins of the average cursor trajectory. This was then compared with line between the start and end target to determine the initial angle. We found that initial angle for unrestricted and restricted trials were comparable (Fig. 5.11C). We calculated the normalized difference using equation 5.11. We found that the initial angle remained unchanged with the path boundary (Fig 5.11D).

Second, we assessed how the trajectory bowing reduced with the tube. For all sessions, we observed a hook-like bowing feature for all instructed path. Because we used straight instructed paths, this bowing behavior was the exact feature we were challenging. Even with fast control, the animals were able to response to feedback of their behavior. Furthermore, this feature was the most likely point of task failure. We calculated the maximum bowing value for both unrestricted and tightest restricted trials. We observed that bowing distances were slightly smaller for the restricted trials (Fig 5.11E) . This result held with the normalized metric.

Altogether, we found that trajectory malleability was low across experiments. Animals showed only a modest ability to modify either their trajectories (Fig 5.11D/F). This suggests that these asymmetries may reflect strong constraints on the time course of neural activity.

5.5 Discussion

In summary, we used a BCI paradigm to probe the time course of neural population activity. We designed rotated BCI mappings that presented visual feedback of M1 dynamics. We then tested if the animals would change the dynamics naturally or when challenged to do so.

In the first experiment, we observed that visual feedback alone was not enough to change neural trajectories. But the absence of change does not immediately confirm neural con-

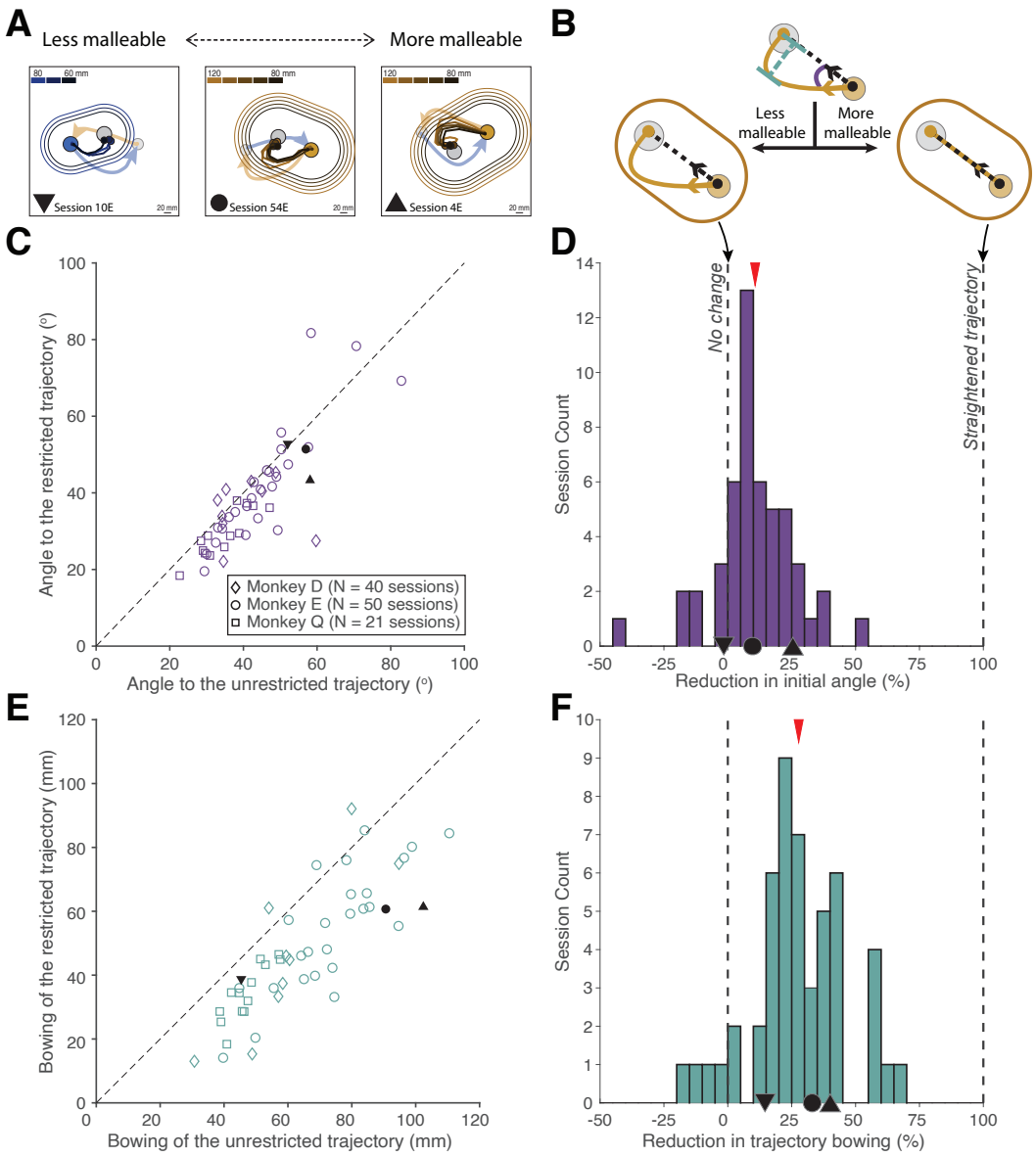

Figure 5.11: Quantifying the instructed path trajectory changes. *a)* Variable malleability examples for the instructed path task. *b)* Conceptual drawing of the two features we are quantifying, initial angle (purple) and trajectory bowing (green). Cartoons of low and high malleability label the goal posts. *c)* Scatter of the initial angle. The dashed line is the unity line. *d)* Histogram of the reduction in initial angle. Example session values are plotted along the x-axis. The distribution mean is plotted in red. *e-f)* Same figure format as *c-d*, but plotting the bowing of the trajectory.

straints. Though we designed the rotated BCI mapping to highlight asymmetry, the animals were not required to overcome the asymmetry. This meant that the animals may have lacked the incentive to change their behavior. We do have a few single trial exceptions where we observed some straightening of the trajectories. While we need to analyze them more thoroughly, the initial results still support the presence of neural constraints. First, the acquire times were longer than the intuitive control acquire times. Second, the trajectories appear to take more effort. They were generally observed early in the block before the animals would shift back to the predicted trajectory path. Finally, we observed stereotyped behavior during error correction. We aligned our workspace and rotated BCI mapping using the start position of the trajectories. This meant that the end trajectory activity did not exactly map to the end target, so the animals had to adapt. For both overshoot and undershoot trials, we often observe a retracing, or looping, behavior from the animals.

Animals still showed little ability to modify their trajectories even when challenged. The instructed path experiment directly challenged the time course of the neural activity. We assessed both early and late stages of the neural trajectories and found both showed low malleability. Though we did observed more malleability in the late stage of the neural trajectory. Finally, we observed a weak correlation between the early and late stage changes.These results suggest that the observed dynamics are inherent to the M1 circuit.

There were a few instructed path sessions that demonstrated greater straightening behavior. There are a few possible explanations that would still be consistent with the presence of inherent dynamics. First, the sessions may have been testing weaker more malleable dynamics. As a first-pass assessment, we tested if there was a correlation between separability (d' from chapter 3) and our malleability metrics. There was not a significant correlation (stats). While we assumed that stronger dynamics exhibit greater separation, this may not be the defining feature. Additional experiments probing different features of the neural space will be required.

Second, these "malleable" sessions may actually be the result of changing the initial start position. Monkey E and D each demonstrated such behavior. These experiments could be evidence to support the hypothesis that both network mechanisms and inputs form temporal constraints. By shifting the initial state, it would alter how the local network

dynamics shaped the neural trajectories. Thus the local network dynamics could stay rigid while the upstream inputs modulate behavior. Further analysis will need to done as well as experiments where we can probe both M1 inputs and network.

Finally, our controls and metrics may be inappropriate for assessing malleability. We tried to create control experiments where the animals' behavior was fully malleable. Unfortunately, neither our straight path and perturbation experiments are true apple-to-apple comparisons. For the straight path task, we tested the animal during hand control, making it difficult to compare across control modes. For perturbation control, following the instructed path was more difficult than expect. This was because the perturbation function amplified the noisy behavior of the cursor. This resulted in highly variable and noisy control for the animal.

Our findings support the view that neural trajectories during motor control reflect underlying network mechanisms. More studies will be needed to isolate the role of inputs and network mechanisms in shaping these dynamics.

6.0 Conclusion

We used a BCI paradigm to probe constraints on the time course of neural population activity in the motor cortex of rhesus monkeys. We observed that neural trajectories during BCI control exhibited rich temporal structure suggestive of network constraints. This temporal structure persisted when animals were given visual feedback of their neural activity in "rotated" BCI mappings designed to highlight neural trajectory asymmetries. Furthermore, animals showed only a modest ability to modify the time courses of their neural activity when challenged to do so. Finally, we found that neural trajectories in all dimensions were largely invariant to the visual feedback provided to the animals. Taken together, these results suggest that the time course of neural activity in motor cortex is subject to strong constraints.

Our findings add to a growing body of work showing the importance of dynamics in neural computation. It is often theorized that the neural population dynamics observed during different cortical processes reflect the biological instantiation of theoretical network constructs such as line attractors, point attractors, etc. [6, 8–11, 13, 15, 18, 19, 24–26, 28, 30–34, 36, 38, 65, 75–77]. A key prediction made by these models is that naturally-occurring neural trajectories are driven by an underlying dynamics model. Using a BCI, we are able to provide causal evidence that volitional modification of these naturally-occurring trajectories is difficult.

The neural trajectory asymmetries we observed during BCI control invite comparisons to findings of rotational dynamics in motor cortex [8, 20–22]. This rotational structure is believed to enable the autonomous generation of overt movements, and has been found to exist in 'untangled' representations [20–22]. Our results build on these previous findings by showing that the strong temporal structure observed in motor cortical activity cannot be modified volitionally. Furthermore, our findings indicate that the time course of neural population activity is highly constrained even in the absence of overt movements, suggesting that they are unlikely to be solely an artifact of sensory feedback or the control of an effector with dynamics.

An open question is how biological network structure might give rise to temporal constraints. A common hypothesis is that such dynamical structure is the result of local network structure. Recent work has also shown that movement-related neural dynamics are heavily dependent on input from the thalamus cite [41]. Our results do not distinguish between the role of local connectivity and inputs in establishing constraints on the time course of neural activity. In our experiments, animals were free to modify the inputs to motor cortex in whatever way possible to attempt to modify the dynamics of neural activity recorded in M1. It is possible that the low malleability of neural trajectories in M1 are due to constraints which exist upstream of motor cortex. Further work is needed to fully disassociate the roles of local connectivity and inputs on the dynamics of neural population activity in motor cortex.

It is worth considering how our results are related to work investigating the relationship between the low-dimensional structure in neural population activity and learning. Such structure (sometimes termed the "intrinsic manifold") has been shown to be predictive of learning [47, 78], with the formation of new neural activity patterns requiring extended periods of training possibly consistent with network modification [60]. In the present study, animals were not required to generate new neural activity patterns, but to generate previously-observed patterns in a new temporal sequence. The inability of animals to do this within a single experimental session implies that neural population activity is more highly constrained than previously thought, consistent with observations of coupling of activity across different subspaces [79]. Probing the ability of animals to modify the time course of their neural activity through extended training (similar to [49, 51, 60]) may allow us to gain further insight into the origin of the observed constraints.

In conclusion, our results provide important causal evidence for the existence of constraints on the generation of time-varying activity in motor cortex. While this is an important step in the development of a more complete understanding of the role of neural dynamics in computation, additional work is needed to fully understand the roles of local connectivity, external inputs, network architecture, and more in the generation of time-varying neural activity. Combining population recording technologies, advanced network manipulation tools (e.g., optogenetics), and brain-computer interfaces will enable the investigation of neural computation in unprecedented detail.

Appendix A

Animal SOP Training

A.1 Monkey D's BCI Training

The purpose of the document is to design a consistent strategy for working monkey D towards the same level of BCI control that monkey E has. The four main areas that we will be working on improving during this process are:

1. **Number of targets:** Currently, we use four targets, but would like to use eight.
2. **Kinematic scale factor:** Reduce the scale factor from 2 (the current value) to 1 (no additional scaling of the workspace targets).
3. **Target size:** Monkey D uses a target size of 30mm when using a decoder without assistance. We would like this to match the value that monkey E uses which is 15mm.
4. **Motivation:** Monkey D has not been a very motivated monkey for reach tasks, he typically works for about 20-25 minutes or about 100mL of water. We will be slowly incrementing the total amount of water monkey D receives during an experiment and also slowly reducing the reward size per trial.

The first three items are task related changes. For any given experiment, we will at most change one task related item. For these changes, we want to see significant improvement and high performance before moving to the next step of the training plan. **This means that we want overall performance to be above 85% prior to a task change.**

For the motivation items, we will slowly increment them throughout the training process. This will be very small increments so that monkey D doesn't notice the change and will continue to work well.

A.1.1 Number of Targets

Once monkey D is over 90% successful with the current paradigm (See A.1.5), we will switch from the 4 target design to the 8 target task. We want to complete this switch before the making any other modifications to the experiment. This is because monkey D could struggle to acquire the new targets and we don't want to reintroduce crutches after we have removed or reduced them in the experiment.

A.1.2 Kinematic Scale Factor

Currently, we use a kinematic scale factor of 2. We will want to reduce this value by 0.2 each time we change the value. If monkey D's overall performance falls below 60% for all targets, the plan should be to increase the kinematic scale factor by 0.1 and recalibrate the decoder.

A.1.3 Target Size

To match the task that monkey E was using, we need to use a target size of 10 during calibration and then a target size of 15 with the full decoder. To accomplish this, we will first reduce the target size of the tapering and center out blocks by 2mm each step. Once the task target size is 20 mm, we will reduce the calibration target size once by 2mm so that the calibration target size is 16mm. We will then reduce the calibration target size to 14mm (down by 2mm) and the target size to 19mm (down by 1mm) within the same experiment. Finally, we will then reduce both targets by 1 mm until the calibration target size is 10 mm and the task target size is 15 mm. The advantage of the final step is that we will maintain the size relationship between the calibration target and the task target.

A.1.4 Motivation

A.1.4.1 Water Motivation: We will try to increase the average amount of total water monkey D receives per session by 5 mL each week. While we don't know what monkey D's exact limit will be for consistent daily training, I suspect that it will likely be around 130 mL.

A.1.4.2 Time Motivation: To increase the amount of time and number of trials that monkey D works for we will very slowly decrease the reward size for trials. The decreases should be about 5 to 10 ms. This decrease will occur at most once every 3 days (ie 1-2 reductions per week).

A.1.5 Current Task Design

Here is the current task design that I use for monkey D's experiments

1. **Observation** 16 trials of observation trials where the automonkey cursor moves to the four targets. These trials are used to calibrate the first decoder.
   a. *Target Size:* 18 (Kept small during calibration)
   b. *Kinematic Scale Factor:* 2
2. **Gradual Training - Assist On** Recalibrate the decoder every 32 trials (Total number of trials 128).
   a. *Target Size:* 18 (Kept small during calibration)
   b. *Kinematic Scale Factor:* 2
   c. *Assist Value:* 0.001
3. **Gradual Training - Tapering Assist** We decrease the decoder assistance every 25 trials that passes a 90% threshold value.
   a. *Target Size:* 30
   b. *Assist Values:* [0.001 0.1 0.25 0.5 0.75 1]
4. **Center Out Trials** Once monkey D has gotten full control of the decoder, let him work for the remainder of his trials on center out.

5. **Recalibration - Only when necessary** If monkey D is ever especially struggling to acquire 1 or more targets, recalibrate the decoder. Reset the decoder assist to 0.001 and run 16 trials of the decoder. Recalibrate every 16 trials. After 80 trials, start to tapering the assist level again.

A.1.6 Training Schedule

Below is a tentative schedule assuming that we are able to make a task change every day based off of expected performance.

1. **2019.07.16** 95% Successful
   - *Task Change:* None
   - *Reward Size:* 145 ms
   - *Total Water:* 120 mL (Start to pay attention to motivation/behavior drops for water at 110 mL.)
2. **2019.07.17** 95% Successful
   - *Task Change:* Switch to the 8 target task
   - *Reward Size:* 145 ms
   - *Total Water:* 120 mL (Start to pay attention to motivation/behavior drops for water at 110 mL.)
3. **2019.07.18** 95% Successful
   - *Task Change:* Reduce the kinematic scale from 2 to 1.8
   - *Reward Size:* 140 ms
   - *Total Water:* 125 mL (Start to pay attention to motivation/behavior drops for water at 115 mL.)
4. **2019.07.19** 95% Successful
   - *Task Change:* Reduce the kinematic scale from 1.8 to 1.6
   - *Reward Size:* 140 ms
   - *Total Water:* 125 mL (Start to pay attention to motivation/behavior drops for water at 115 mL.)
5. **2019.07.22** 95% Successful

- *Task Change:* None, Monday's can be more difficult for monkey D.
- *Reward Size:* 140 ms
- *Total Water:* 120 mL (Start to pay attention to motivation/behavior drops for water at 115 mL.)

6. **2019.07.23** 95% Successful
   - *Task Change:* Reduce the kinematic scale from 1.6 to 1.4
   - *Reward Size:* 135 ms
   - *Total Water:* 125 mL (Start to pay attention to motivation/behavior drops for water at 115 mL.)
7. **2019.07.24** 95% Successful
   - *Task Change:* Reduce the kinematic scale from 1.4 to 1.2
   - *Reward Size:* 135 ms
   - *Total Water:* 125 mL (Start to pay attention to motivation/behavior drops for water at 115 mL.)
8. **2019.07.25** 95% Successful
   - *Task Change:* Reduce the kinematic scale from 1.2 to 1.0
   - *Reward Size:* 135 ms
   - *Total Water:* 130 mL (Start to pay attention to motivation/behavior drops for water at 115 mL.)
9. **2019.07.26** 95% Successful
   - *Task Change:* No change, observe performance another day with the kinematic scale at 1.
   - *Reward Size:* 130 ms
   - *Total Water:* 130 mL (Start to pay attention to motivation/behavior drops for water at 115 mL.)
10. **2019.07.29** 95% Successful
    - *Task Change:* Reduce the target size from 30 to 28.
    - *Reward Size:* 130 ms
    - *Total Water:* 130 mL (Start to pay attention to motivation/behavior drops for water at 115 mL.)

11. **2019.07.30** 95% Successful
    - *Task Change:* Reduce the target size from 28 to 26.
    - *Reward Size:* 130 ms
    - *Total Water:* 130 mL (Start to pay attention to motivation/behavior drops for water at 115 mL.)
12. **2019.07.31** 95% Successful
    - *Task Change:* Reduce the target size from 26 to 24.
    - *Reward Size:* 125 ms
    - *Total Water:* 130 mL (Start to pay attention to motivation/behavior drops for water at 115 mL.)
13. **2019.08.01** 95% Successful
    - *Task Change:* Reduce the target size from 24 to 22.
    - *Reward Size:* 125 ms
    - *Total Water:* 130 mL (Start to pay attention to motivation/behavior drops for water at 115 mL.)
14. **2019.08.02** 95% Successful
    - *Task Change:* Reduce the target size from 22 to 20.
    - *Reward Size:* 125 ms
    - *Total Water:* 130 mL (Start to pay attention to motivation/behavior drops for water at 115 mL.)
15. **2019.08.05** 95% Successful
    - *Task Change:* Reduce the calibration target size from 18 to 16.
    - *Reward Size:* 120 ms
    - *Total Water:* 130 mL (Start to pay attention to motivation/behavior drops for water at 115 mL.)
16. **2019.08.06** 95% Successful
    - *Task Change:* Reduce the calibration target size from 16 to 14. Reduce the target size from 20 to 19.
    - *Reward Size:* 120 ms

- *Total Water:* 130 mL (Start to pay attention to motivation/behavior drops for water at 115 mL.)

17. **2019.08.07** 95% Successful
    - *Task Change:* Reduce the calibration target size from 14 to 13. Reduce the target size from 19 to 18.
    - *Reward Size:* 120 ms
    - *Total Water:* 130 mL (Start to pay attention to motivation/behavior drops for water at 115 mL.)
18. **2019.08.08** 95% Successful
    - *Task Change:* Reduce the calibration target size from 13 to 12. Reduce the target size from 18 to 17.
    - *Reward Size:* 115 ms
    - *Total Water:* 130 mL (Start to pay attention to motivation/behavior drops for water at 115 mL.)
19. **2019.08.09** 95% Successful
    - *Task Change:* Reduce the calibration target size from 12 to 11. Reduce the target size from 17 to 16.
    - *Reward Size:* 115 ms
    - *Total Water:* 130 mL (Start to pay attention to motivation/behavior drops for water at 115 mL.)
20. **2019.08.12** 95% Successful
    - *Task Change:* No change
    - *Reward Size:* 115 ms
    - *Total Water:* 130 mL (Start to pay attention to motivation/behavior drops for water at 115 mL.)
21. **2019.08.13** 95% Successful
    - *Task Change:* Reduce the calibration target size from 11 to 10. Reduce the target size from 16 to 15.
    - *Reward Size:* 115 ms

- *Total Water:* 130 mL (Start to pay attention to motivation/behavior drops for water at 115 mL.)

A.2 Memory Guided Training SOP

The purpose of the document is to layout the experimental changes to take an animal from a center out reach task to a memory guided reach task. The design and proposed timeline are based on the training steps we used with monkey Q from October 31st, 2019 to December 1st, 2019. The four main task changes to teach during this process are:

1. **Incorporate the go-cue into the center out task:** This is to introduce the go cue to the animal. I would recommend using a go cue where additional information appears, like a target ring, rather than the center target being extinguished.
2. **Memory Guided Center Out - Step 1:** Introduces the target flash during a center hold task. The animal has to maintain a center target hold during and after the target flash.
3. **Memory Guided Center Out - Step 2:** Add in the reach to the task. Still have learning crutches to help connect the dots of the task and to limit the animal's frustration.
4. **Memory Guided Center Out - Step 3:** Removes the learning crutch of the looping states for failing the center hold.
5. **Memory Guided Center Out - Step 4:** Removes the learning crutch of the target reminder flash during the reach state. (*Note: I have not incorporated this step in monkey Q's training as it didn't appear to be necessary. After the first week, he rarely if ever used the target flash reminder.*)

Quick notes: I have listed all parameter files used during training, but the most recent version for each step should have all the necessary state tables. Additionally, the go cue I used with monkey Q was a target ring with gray target. You may want to lighten the color since I noticed that my target colors were slightly darker following the Windows 10 update.

A.2.1 Incorporate the Go-cue into the Center Out Task

A.2.1.1 Intermediate Steps: Include the target ring (or other go cue) in only a small percentage of the trials of center out. Increase the percentage of trials every few 100 trials if the animal's performance crossed a threshold of 85%. Once the animal is always working

with the task that incorporates the go cue, keep the animal there for a few sessions without any task changes.

A.2.1.2 Parameter Files Used

1. `Quincy_WITHmarker__8Target_20191031.prm`
2. `Quincy__8TargetCOandTargetRing_20191031.prm`
3. `Quincy__8TargetCOandTargetRing_20191103.prm`

A.2.2 Memory Guided Center Out - Step 1

A.2.2.1 Intermediate Steps Introduces the target flash. The animal has to maintain a center target hold during and after the target flash. Remember trials that have the target flash at this point will not have an active reach.

85% of trials will be the center out reach task with the target ring and 15% of center hold task with a target flash task. This design has two advantages. First, it introduces the target flash while still require the animal to reach. Second, it helps distinguish the target cue (ie target flash) from the go cue (ie target ring).

Start with short holds following the target flash and gradually increase those times. I found it helpful to loop the target flash state and center hold 2 state back to the beginning of the trial if the animal preemptively leaves the center target.

Similarly start with short target flashes and gradually increase. The steps I took with monkey Q were 30, 50, and 100 ms.

Once the animal is able to maintain the hold for several hundred ms after the target flash move on to step 2.

A.2.2.2 Parameter Files Used

1. `Quincy__8TargetCOandTargetRing_20191104.prm`
2. `Quincy__8TargetMG_TargetRing_step1_20191107.prm`

A.2.3 Memory Guided Center Out - Step 2

A.2.3.1 Intermediate Steps Run a memory guided task with learning crutches. There are three learning crutches within the initial design. First, the looping center hold states used in step 2. Second, the target appears, or turns on, if the animal reach in the correct direction. Finally, there is a target flash reminder during the reach period if the animal doesn't acquire the target within a certain amount of time.

For the target appear, I start will a really large tolerance window around the reach target and then I gradually decrease it. Once the tolerance window is about 85% of the way to the target, remove the target appear state.

For the target flash, I typically set it to occur 2 seconds after the reach period is initiated. Once it flashes the task will loop back to the reach period. I apply a global time out to the reminder state so that it will only repeat 3 or 4. This applies a light pressure on the animals while also preventing an infinite task loop/the experimenter having to go into the room to complete the task.

Finally, I also set a percentage of trials to be the step 2 task, ie the center target hold with a target flash. I find this helpful to prevent jumping the gun with teaching longer delay periods.

A.2.3.2 Parameter Files Used

1. `Quincy__8TargetMG_TargetRing_step2_20191109.prm`
2. `Quincy__8TargetMG_TargetRing_step2_20191113.prm`
3. `Quincy__8TargetMG_TargetRing_step2_20191119.prm`
4. `Quincy__8TargetMG_TargetRing_step2_20191121.prm`
5. `Quincy__8TargetMG_TargetRing_step2_20191123.prm`

A.2.4 Memory Guided Center Out - Step 3

A.2.4.1 Intermediate Steps Removes the learning crutch of the looping states for failing the center hold. There will still be the learning crutch of the target reminder flash if the animal reaches to the wrong target. Use this step to continue pushing the center hold times and possibly target distance.

A.2.4.2 Parameter Files Used

1. `Quincy__8TargetMG_TargetRing_step3_20191125.prm`
2. `Quincy__8TargetMG_TargetRing_step3_20191203.prm`
3. `Quincy__8TargetMG_TargetRing_step3_20191206.prm`
4. `Quincy__8TargetMG_TargetRing_step3_20191211.prm`
5. `Quincy__8TargetMG_TargetRing_step3_20191218.prm`

A.2.5 Memory Guided Center Out - Step 4

A.2.5.1 Intermediate Steps Removes the learning crutch of the target flash reminder during the reaching portion of the task. This is the full memory guided task without any crutches. If the animal is highly dependent on the target flash reminder I would recommend using an intermediate step where you reduce the reward size by half each time that the reminder happens.

A.2.6 Monkey Q's Training Schedule

Below is the schedule we used for monkey Q when training the memory guided task. This can be useful to estimate the amount of training time required.

1. **2019.10.31** Center Out with Target Ring
   - *Task Changes:* Presentation of the target ring
   - *Percentage Changes:* [30%, 40%, 50%, 75%]
2. **2019.11.01** Center Out with Target Ring
   - *Task Changes:* Presentation of the target ring

- *Percentage Changes:* [75%, 100%]

3. **2019.11.02** Center Out with Target Ring
   - *Task Changes:* n/a
   - *Percentage Changes:* [100%]
4. **2019.11.03** Center Out with Target Ring
   - *Task Changes:* Added in the center hold loop and center hold failure states.
   - *Center Hold Total:* [300 400 500 550 600] ms
   - *Center Hold Loop:* 250 ms
   - *Target Hold:* 300 ms
5. **2019.11.04** Center Out with Target Ring
   - *Task Changes:* Unsuccessful introduction to the target flash (10ms).
   - *Center Hold Total:* [300 400 500 550 600] ms
   - *Center Hold Loop:* 200 ms
   - *Target Hold:* 300 ms
6. **2019.11.06** Memory Guided Center Out - Step 1
   - *Task Breakdown:* 85% of trials had the target ring, 15% center hold with target flash
   - *Task Changes:* Use the center hold target flash task with a loop state. Target flash was 30 ms.
   - *Center Hold Total:* [300 400 500 550 600] ms
   - *Center Hold Loop:* 200 ms
   - *Center Hold 1:* [50 100 150] ms
   - *Target Flash:* 30 ms
   - *Center Hold 2:* [400 500 600 650 700] ms **Length of time Quncy held after the target flash**
   - *Target Hold:* 300 ms
7. **2019.11.07** Memory Guided Center Out - Step 1
   - *Task Breakdown:* 85% of trials had the target ring, 15% center hold with target flash
   - *Task Changes:* Increased the target flash and the hold after the flash.
   - *Center Hold Total:* [300 400 500 550 600] ms
   - *Center Hold Loop:* 200 ms

- *Target Flash:* 50 ms
- *Center Hold 2:* [350 450 550 600 650] ms **Gradual increase to this.**

8. **2019.11.09** Memory Guided Center Out - Step 2
   - *Task Breakdown:* 85% of trials memory guided reach, 15% center hold with target flash **Assume this layout unless otherwise stated**
   - *Task Changes:* Memory Guided task, target appears
   - *Center Hold Total:* [300 400 500 550 600] ms
   - *Center Hold Loop:* 200 ms
   - *Target Flash/Target Flash Reminder:* 200 ms
   - *Center Hold 2:* [50 100 150] ms
   - *Target Appear Radius:* 55 mm
9. **2019.11.10** Memory Guided Center Out - Step 2
   - *Task Changes:* Decrease target appear
   - *Target Appear Radius:* [55 50 45] mm
10. **2019.11.11** Memory Guided Center Out - Step 2
    - *Task Changes:* Decrease target appear radius, increased center hold 2
    - *Center Hold 2:* [50 100 150 200 250] ms
    - *Target Appear Radius:* [45 35 25] mm
11. **2019.11.12** Memory Guided Center Out - Step 2
    - *Task Changes:* Increased center hold 2
    - *Center Hold 2:* [50 100 150 200 250 300 350 400 450] ms
    - *Target Appear Radius:* [25] mm
12. **2019.11.17** Memory Guided Center Out - Step 2
    - *Task Changes:* Reduced reward and added 400 ms hold.
    - *Center Hold 2:* [50 100 150 200 250 300 350 400] ms
13. **2019.11.19** Memory Guided Center Out - Step 2
    - *Task Changes:* Decreased the target appear radius to 15 mm.
    - *Target Appear Radius:* [25 15] mm
14. **2019.11.21** Memory Guided Center Out - Step 2

- *Task Changes:* Switched to a version of the task that didn't use the target appear state. That's to say that the target didn't turn on until after the animal was fully enclosed within the target.
- *Center Hold 2:* [50 100 150 200 250 300 350 400] ms

15. **2019.11.23** Memory Guided Center Out - Step 3
    - *Task Changes:* Removed the looping state crutch for the target flash and center holds (1 and 2). Added a failure state for those states.
16. **2019.11.24** Memory Guided Center Out - Step 3
    - *Task Changes:* Increased target distance from center of the workspace and added a new hold time.
    - *Center Hold 2:* [50 100 150 200 250 300 350 400 500] ms

Appendix B

TDT vs UDP Spiking Data Comparison

B.0.1 Problem:

We observed that there were differences between the spiking data saved by the TDT system and the data transmitted to host via UDP. We want to fully understand all the sources of variability causing these discrepancies and account for them in our data. Fig. B4 and Fig. B5 shows the differences between TDT spikes/sec and UDP spikes/sec for each channel trial for the 20180824 recording session.

B.0.2 Possible Sources of Variability

Below is a list of the possible sources of variability that we have identified. There are also brief descriptions of how the sources will affect the spiking data. We ordered the list according to the sources' hypothesized contributions. Fig. B1 shows a block diagram of the flow of the spiking data through the system. While Figs. B2 and B3 show how the different sources might affect UDP spike trains.

1. **Single Spike per 2ms bin:** RZ2 transmits packets that are 8 channels of 32-bit Words to the RT computer. Based on the RZ documentation, this means that RZ sends a packet every 2 ms. Given that we must send other information to the RT computer, that means that we transmit one bit per recording channel every 2 ms. A channel's given bit will be set low if no spiking occurred and set high if any spiking occurred. Thus, it is possible that high firing rate channels may have higher spike counts in the TDT data than in the UDP data. Since we use threshold crossing activity, we will have more high firing channels. *We expect that high firing rates will account for the majority of the spiking discrepancies.*

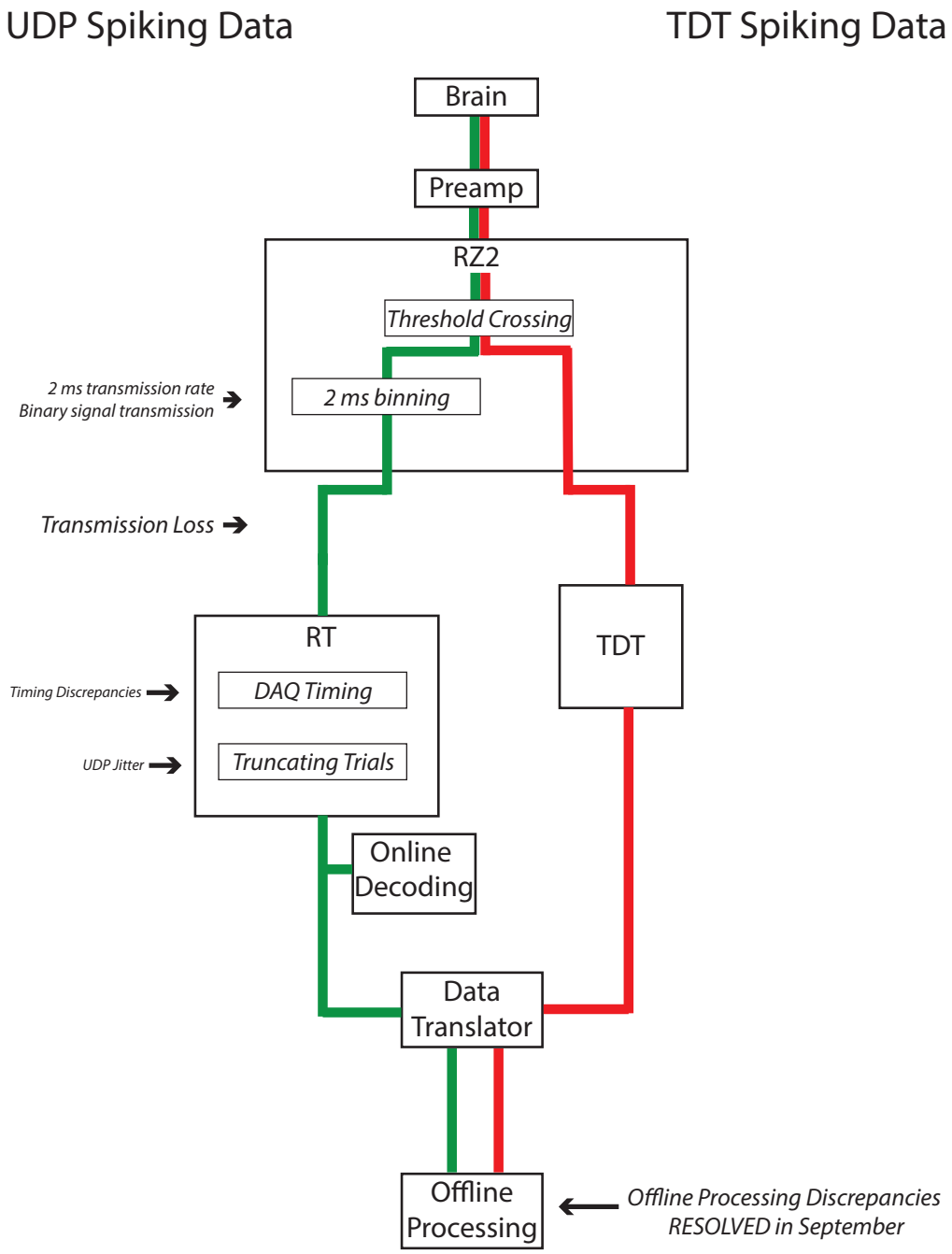

Figure B1: Block Diagram of the flow of the TDT and UDP spiking data.

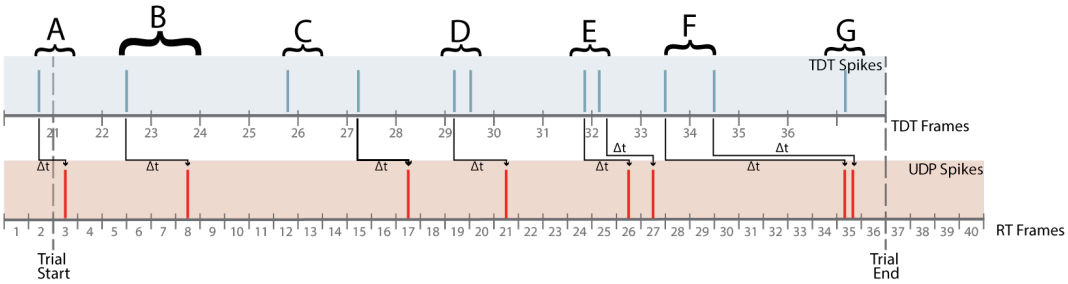

Figure B2: A theoretical TDT spike train for a single channel transmitted into an UDP spike train. This figure demonstrates the different sources of variability. *A)* A TDT spike is unaccounted for because of how the trial was truncated. *B)* Demonstrates how the priority level of reading the UDP packets varies the delay between the TDT and UDP spike train. The delay here is much longer than it was for the previous spike (A). *C)* Demonstrates a UDP packet loss, TDT records the spike, but there was no data transmitted to the RT computer. *D)* Demonstrates the lost of spikes for high firing channels. Two spikes occurred within the same TDT frame, but only one spike was recorded in the UDP spike train. *E)* Presents an example where both spikes firing within 2 ms of each other are both accounted for in the UDP spike train because the spikes fell within different TDT frames. *F)* An example of RT timing discrepancies. Here the RT frame duration is much longer than 1 ms, so several UDP packets Several UDP packet were given the timestamp of 35. *G)* Another example of variability caused by trial truncation. In this case, the UDP spike is lost.

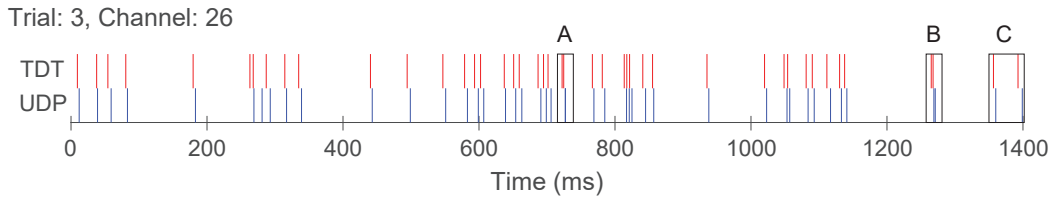

Figure B3: A real TDT spike train for a single channel transmitted into an UDP spike train. This figure demonstrates the different sources of variability. *A)* Shows the lost of spikes when there is high firing activity. Two spikes occurred within the same TDT frame, but only one spike was recorded in the UDP spike train.*B)* Presents an example where both spikes firing within 2 ms of each other are both accounted for in the UDP spike train because the spikes fell within different TDT frames. *C)* Demonstrates how the priority level of reading the UDP packets varies the delay between the TDT and UDP spike train.

2. **UDP trial truncating delay:** When translating the experiment's data into a MAT structure, we separated and save the TDT and UDP data into trials. Because of the transmission jitter/delay for UDP data it is possible that we might have mismatches in spike counts between the two structure. This would result in fewer TDT spikes at the beginning of the trial and fewer UDP spikes at the end of the trial. *Overall, we expect that this will account for a small portion of our spiking data discrepancies.*
3. **Timing Discrepancies between TDT and RT:** The TDT data is clock based on the TDT computer clock and synced to the trial by a TTL output pulse from the RT computer. The UDP is clocked according to the frame number of the RT, which we relate to the DAQ timing. We assume that the DAQ will sample at 1 ms intervals, however, when there is high demand on RT, such as large computations or complex state transitions, this amount of processing time will extend beyond 1 ms. This could lead to further misalignment between the TDT and UDP data, causing there to be a mismatch in spike counts. *This should not affect the spike counts that we have for either TDT or UDP, but it is possible that this will cause random contraction and expansion effects on the UDP spike time stamps that we will have to be conscious of while completing this*

analysis. Additionally, because of the contraction and expansion effects on the UDP spike times, it is possible that the UDP packets will be time-stamped with the wrong time.

4. **UDP transmission loss:** When transmitting information from RZ2 to the RT computer, it is possible that the information may get lost or delayed. This will result in the UDP data having fewer spikes than the TDT data. This is an inherent problem to using UDP transmission as there is no log, or 'handshake', of the data being transmitted and received. *We expect this problem to be rare.*

B.0.3 Solution

We determined that the primary source of variability was the single spike per 2 ms. The max sampling rate of the TDT system was 24414.06 Hz, so we actually send UDP packets every 1.9661 ms. Since we send one bit per recording channel in each packet, we would only know if a channel had spiked or not. The RT computer would interpret this binary signal as a single spike. By accounting for this bin width discrepancy and the packet alignment, we could exactly recreate the UDP data from the TDT spiking data (Fig. B12). We were able to do this for several sessions. While UDP packet transmission loss is still possible, we did not observe a single instance of UDP packet loss in the eight datasets tested.

B.0.4 Evaluation of the Variability

Fig. B4 shows the difference in spike counts between the TDT and UDP data across all channels for each trial in an intuitive two target task. The positive values in the figure, shaded yellow through red, mean that there were more TDT spikes collected for the trial than UDP spikes. Negative values, shaded green, mean that there were more UDP spikes than TDT spikes. Areas shaded grey mean that there was a match between the TDT spike count and the UDP spike count. Fig. B5 shows the same information presented in a histogram form. Finally, table B1 list the total percentage of trials with more UDP spikes, more TDT spikes, and matched UDP/TDT data. This table also gives an estimated mean number of additional spikes across all channels.

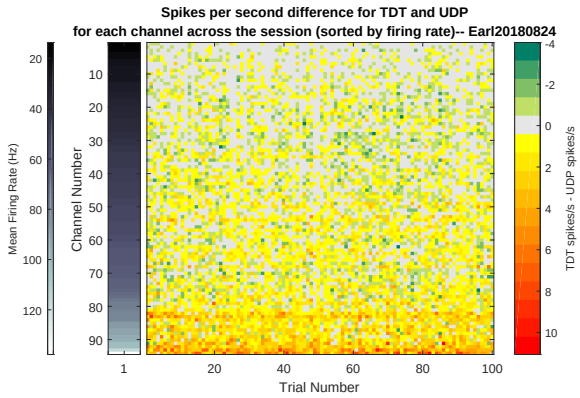

Figure B4: TDT spike counts - UDP spike counts for 100 trials of the intuitive control two target task. The spike counts were taken over 45 ms bins, starting at time equal 0 until the end of the behavioral task (ie trial duration). These 45 ms bins were then summed together over the entire time of the trial in order to determine the spike counts of TDT and UDP spikes. These spike counts were then normalized by the trial duration before subtracting the UDP spikes from the TDT spikes. The channels have been ordered according to the average firing rate of the channel.

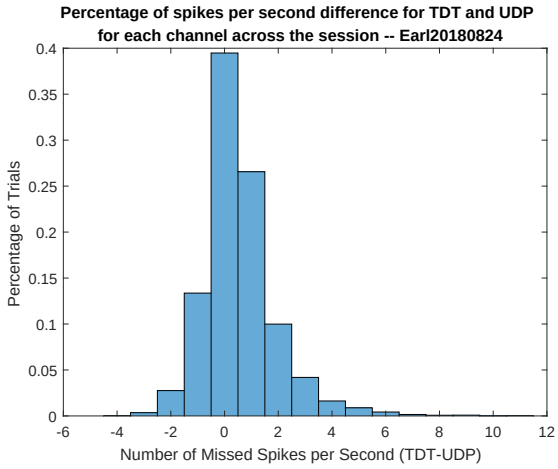

Figure B5: Histogram of the spiking difference. This is a histogram presentation of the data in Fig. B4. As can be observed there is a large portion of trials where the UDP spike count does not match the TDT spike count.

Table B1: The Percentage breakdown of more UDP spikes, more TDT spikes, and matched data across all channels and trials.

	More UDP Spike	Matched Data	More TDT Spikes
Percentage of All Channels	17.85%	33.03%	49.10%
Mean Number of Additional Spikes	0.98 (spks/s)	N/A	1.44 (spks/s)

As we observe in Fig. B4 there are a greater number of unaccounted TDT spikes for channels with high firing rates. For low firing channels, there are more instances of higher UDP spike counts than TDT spike counts. It is likely that these instances are caused by the UDP trial truncating delay, where UDP spikes at the very beginning of the trial weren't matched to TDT spikes. In order for us to access the root cause of each observed spiking discrepancy we will need to correctly identify which TDT spikes were missing and which spikes have matched UDP spikes. We can accomplish this by recreating the UDP spike train from the TDT spike train. We found the estimated UDP spike train by optimizing the parameters listed below for all channels simultaneously (Fig. B6 shows the steps involved).

1. **TDT initial bin edge:** This value is necessary in order to determine the bin edge between one UDP packet and another UDP packet.
2. **TDT bin width:** This is the amount of time each UDP packet represent. Based on the documentation of the TDT system, this value should be around 2 ms. We will want to sweep this value in order to get a more realistic measurement.
3. **UDP Transmission ID number (i.e. frame number):** There is some delay between when a UDP packet was sent and when it was read. We will therefore want to align the spike trains according to when the UDP packet was sent. Aligning the data to this counter instead of the RT's clock, will remove timing delay discrepancies and allow us to view the UDP packets independently.

Fig. B8 shows the spike error values found running our optimization function for the first trial of the session. The gray bar in the left panel represents where there is no error between

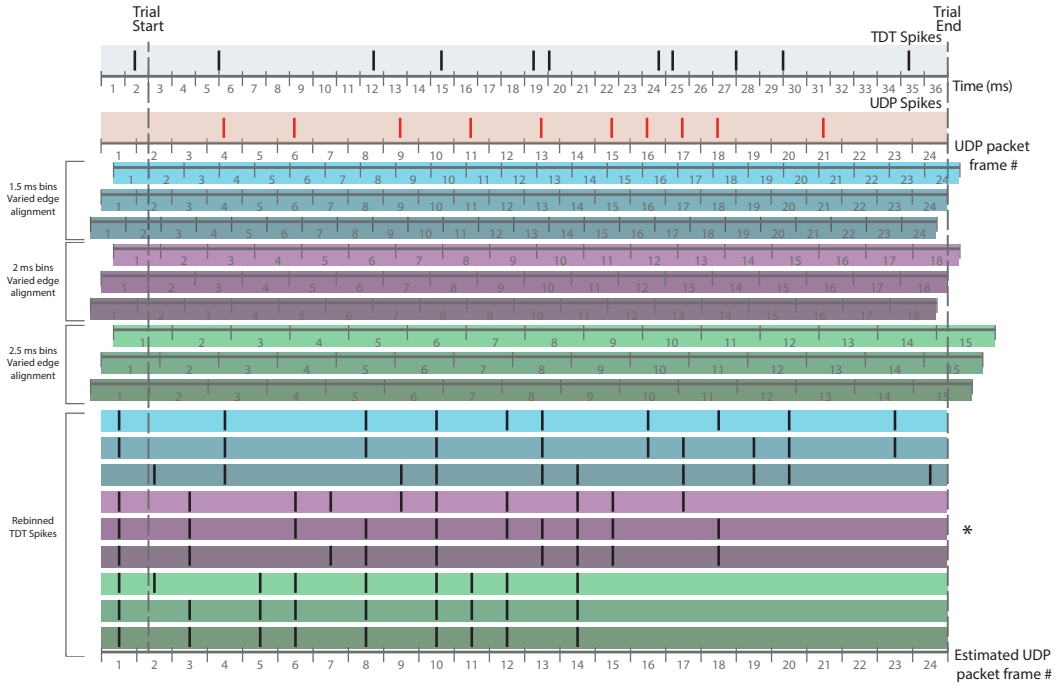

Figure B6: Aligning the TDT and UDP spiking data for a single channel by sweeping the parameters initial bin edge and bin width. The top two rows shows the raw TDT spiking data of a single channel and its corresponding UDP spike train. The different colors represent different sampling bin widths (Blue = 1.5ms, purple = 2 ms, and green = 2.5 ms). We then shift the initial bin edge and bin the TDT spike train at several different parameters. The row labeled with an asterisk is the closest match of the UDP spike train.

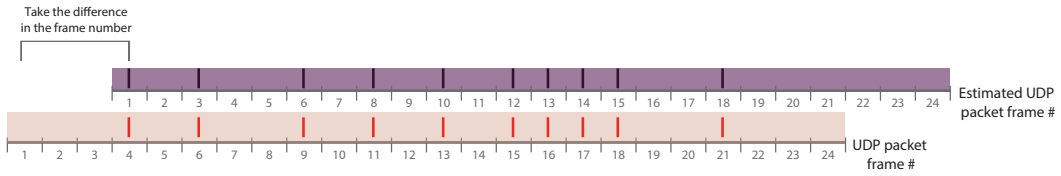

Figure B7: Aligning the estimated UDP spike train with the true UDP spike train. The estimated UDP spike train is the starred line in Fig. B6 the TDT and UDP spiking data for a single channel.

the two spike trains. When we plot the difference between the raw UDP spike train and the estimated UDP spike train using these parameters, we observe that there is a perfect match between the data (Fig. B9). If we were to shift the frame alignment in either direction we would observe that there was still a one to one match between the UDP spikes, but that the spike trains would be offset from each other (Figs. B10 and B11). We ran this optimization over all trials in the session and exactly recreated the UDP data from the TDT spiking data (Fig. B12). Figs. B13, B14, and B15 show the distribution of the alignment parameters across all trials.

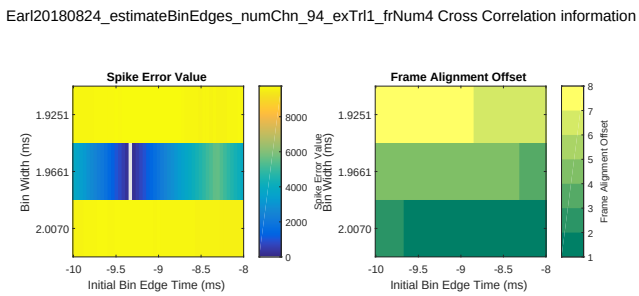

Figure B8: Spike Count Error Estimate for Different TDT and UDP alignments. *Left Panel)* Minimum spike error plot for bin width and initial bin edge time. *Right Panel)* The frame alignment related to the minimum spike error value in the left panel.

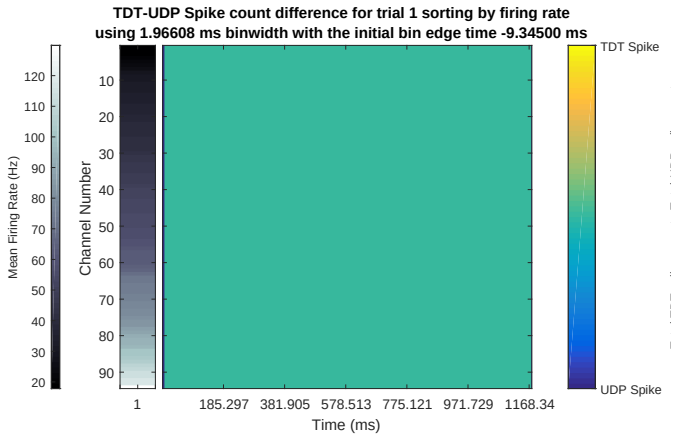

Figure B9: Spike count error with frame alignment 4. The alignment is correct so we see no spiking discrepancy after the first few ms of the trial.

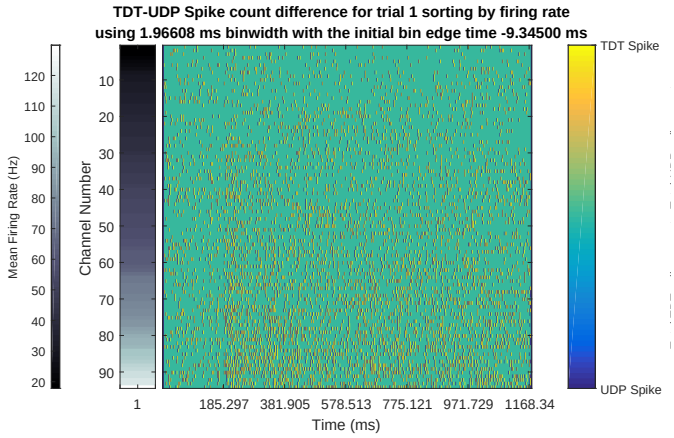

Figure B10: Spike count error with frame alignment 3. The alignment is off by 1 frame and we observe that there is often a UDP spike closely followed by the TDT spike.

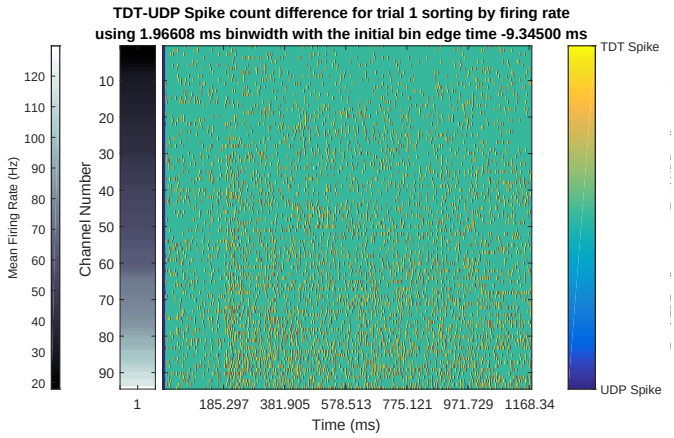

Figure B11: Spike count error with frame alignment 5. The alignment is off by 1 frame in the opposition direction of Fig. B10. This means that we now observe that there is often a TDT spike closely followed by the UDP spike.

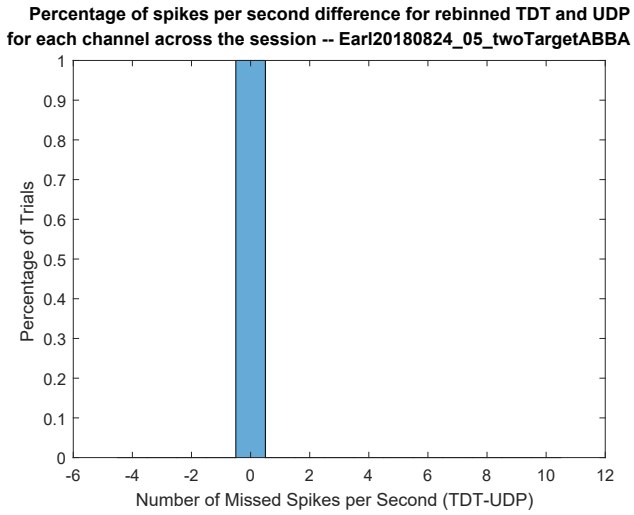

Figure B12: Histogram of the spiking difference after rebinning the TDT spiking data according to the alignment algorithm. As can be seen, rebinning the TDT data lead to a perfect spike count match across all channels for all trials.

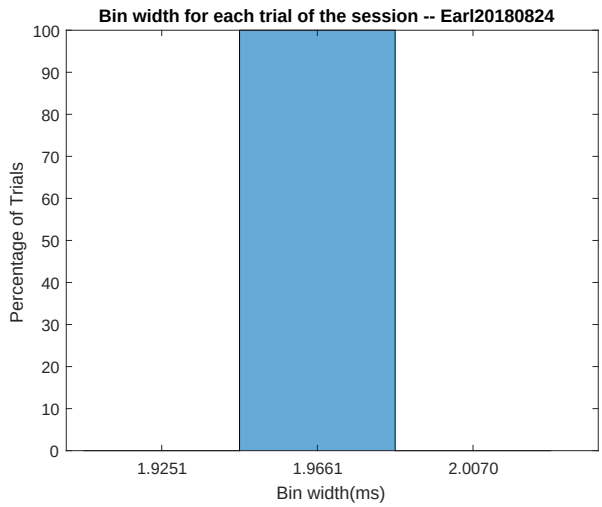

Figure B13: Histogram of the alignment bin width for all trials. All trials required a bin width size of 1.96608 ms. This is expected since the TDT system has a 24414.06 Hz sampling rate, meaning the system is sampling data about every 41 μs. Therefore, the system send a UDP packet every 48 clock interval or 1.96608 ms.

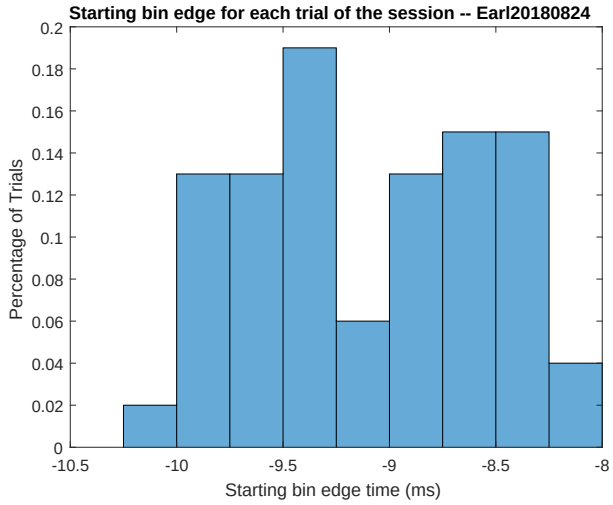

Figure B14: Histogram of the initial bin edge for all trials. It takes about 10 ms to set up each trial, before any visual displays are presented. During this time we are still collect TDT data and sending UDP packets. The wide spread of initial alignments is expected since the binning and transmitting of the UDP packets is constant throughout the session, but how trials align to those bin edges are not.

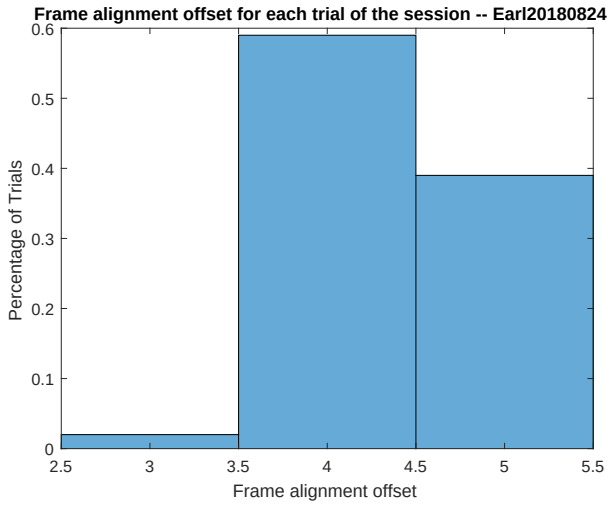

Figure B15: Histogram of the frame number alignment. The spread of frame number alignments is expected since the delay between RT and TDT is variable.

Appendix C

Collected datasets

Table C1: Summary of experiment datasets for monkeys E, Q, and D. The experiments at listed alphabetically in order of BCI tasks, e.g. BC, and then hand control, e.g. HC. Decoder information has also been included.

Control type	Monkey	Task	Decoder calibration	Bins	Grid Type	Datasets
bc	D	cg int	UPD: co	7		35
bc	E	cg int	UPD: co	7		6
bc	D	co int	UPD: co	7		15
bc	E	co int	UPD: co	7		2
bc	Q	coc sweep int	UPD: co	7	cart 8	1
bc	E	isoforce 1d	UPD: co	7		7
bc	E	itip int	TDT: co	22	cond 4	3
bc	E	itip int	UPD: co	7	cond 4	1
bc	E	itip rot fast	TDT: co	22	cond 33	22
bc	E	itip rot fast	TDT: co	22	cond hex	6
bc	D	itip rot slow	UPD: co, rmCIV	7		9
bc	E	itip rot slow	TDT: co	22,7	cond 4	7
bc	E	itip rot slow	UPD: co	7	cond 4	28
bc	Q	itip rot slow	UPD: co	7	cart 8	13
bc	E	itip rot sweep	TDT: co	22	cond 33	5
bc	E	itip rot sweep	UPD: co	7	cond 4	6

Table C1 (continued)						
Control type	Monkey	Task	Decoder calibration	Bins	Grid Type	Datasets
bc	Q	pertip slow	UPD: co	7	cart 8	9
bc	E	tt int	UPD: co	7		29
bc	Q	tt int	UPD: co	7	cart 8	34
bc	D	tt int fine	UDP: tt 2 step mag	7		1
bc	Q	tt int fine	UPD: co	7	cart 8	1
bc	D	tt rot wo	tt udp mag2	7	cond 4	4
bc	E	tt rot wo	UPD: co	7	cond 4	1
bc	D	tt-atp int	UPD: alternative methods	7	cond 4	9
bc	E	tt-atp int	TDT: co	22	cond hex	1
bc	E	tt-atp int	UPD: co	7	tt atp	1
bc	Q	tt-atp int	UPD: co	7	cart 8	36
bc	Q	ttip int slow	UPD: co	7	cart 8	7
bc	E	ttip rot slow	UPD: co	7	cond 4	8
bc	D	tt-ir	UPD: co	7	cond 4	24
bc	D	tt-ir	UPD: alternative methods	7	cond 4	17
bc	D	tt-ir	UPD: co, rmCIV	7	cart 8	8
bc	D	tt-ir	UPD: co, rmCIV	7	cond 4	7
bc	E	tt-ir	UPD: co	7	tt atp	4
bc	E	tt-ir	UPD: co	7	tt atp	4
bc	E	tt-ir	TDT: co	22	cond 33	16
bc	E	tt-ir	TDT: co	22	cond 4	6
bc	E	tt-ir	TDT: co	22	cond hex	3
bc	E	tt-ir	TDT: co	7	cond 4	1

Table C1 (continued)						
Control type	Monkey	Task	Decoder calibration	Bins	Grid Type	Datasets
bc	E	tt-ir	UPD: co	7	cond 4	51
bc	Q	tt-ir	UPD: co	7	cart 8	21
bc	E	tt-ir fix	UPD: co	7	cond 4	8
bc	Q	tt-ir fix	UPD: co	7	cart 8	4
bc	D	tt-ir inv	UPD: co init push	7	cond 4	1
bc	D	tt-ir inv	UPD: co, rmCIV	7		6
bc	E	tt-ir inv	TDT: co	22	cond 4	1
bc	E	tt-ir inv	UPD: co	7		16
bc	Q	tt-ir inv	UPD: co	7	cart 8	2
hc	E	co	UPD: co	7	tt atp	1
hc	Q	mg	UPD: co	7	cart 8	5
hc	Q	pertit amp				1
hc	Q	pertit sweep				2
hc	Q	pertitip slow				15
hc	E	tt				1

Bibliography

[1] J. Gordon, M. F. Ghilardi, and C. Ghez, "Impairments of reaching movements in patients without proprioception. I. Spatial errors," *Journal of Neurophysiology*, vol. 73, no. 1, pp. 347–360, 1995.

[2] J. M. Kenzie, J. A. Semrau, S. E. Findlater, T. M. Herter, M. D. Hill, S. H. Scott, and S. P. Dukelow, "Anatomical correlates of proprioceptive impairments following acute stroke: A case series," *Journal of the Neurological Sciences*, vol. 342, no. 1-2, pp. 52–61, 2014.

[3] B. H. Dobkin, "Strategies for stroke rehabilitation," *The Lancet Neurology*, vol. 3, no. 9, pp. 528–536, 2004.

[4] B. M. Yu, J. P. Cunningham, G. Santhanam, S. I. Ryu, K. V. Shenoy, and M. Sahani, "Gaussian-Process Factor Analysis for Low-Dimensional Single-Trial Analysis of Neural Population Activity," *Journal of Neurophysiology*, vol. 102, no. 1, pp. 614–635, 2009.

[5] G. Santhanam, B. M. Yu, V. Gilja, S. I. Ryu, A. Afshar, M. Sahani, and K. V. Shenoy, "Factor-Analysis Methods for Higher-Performance Neural Prostheses," *Journal of Neurophysiology*, vol. 102, no. 2, pp. 1315–1330, 2009.

[6] M. M. Churchland, J. P. Cunningham, M. T. Kaufman, S. I. Ryu, and K. V. Shenoy, "Cortical Preparatory Activity: Representation of Movement or First Cog in a Dynamical Machine?," *Neuron*, vol. 68, no. 3, pp. 387–400, 2010.

[7] A. Afshar, G. Santhanam, B. Yu, S. Ryu, M. Sahani, and K. Shenoy, "Single-Trial Neural Correlates of Arm Movement Preparation," *Neuron*, vol. 71, no. 3, pp. 555–564, 2011.

[8] M. M. Churchland, J. P. Cunningham, M. T. Kaufman, J. D. Foster, P. Nuyujukian, S. I. Ryu, and K. V. Shenoy, "Neural population dynamics during reaching," *Nature*, vol. 487, no. 7405, pp. 51–56, 2012.

[9] V. Mante, D. Sussillo, K. V. Shenoy, and W. T. Newsome, "Context-dependent computation by recurrent dynamics in prefrontal cortex," *Nature*, vol. 503, no. 7474, pp. 78–84, 2013.

[10] D. Sussillo, "Neural circuits as computational dynamical systems," *Current Opinion in Neurobiology*, vol. 25, pp. 156–163, 2014. First read, primarily focused on the introduction and conclusion. Nice quick review of RNN to study dynamics.

[11] K. Ames, S. Ryu, and K. Shenoy, "Neural Dynamics of Reaching following Incorrect or Absent Motor Preparation," *Neuron*, vol. 81, no. 2, pp. 438–451, 2014.

[12] J. C. Kao, P. Nuyujukian, S. I. Ryu, M. M. Churchland, J. P. Cunningham, and K. V. Shenoy, "Single-trial dynamics of motor cortex and their applications to brain-machine interfaces," *Nature Communications*, vol. 6, no. 1, p. 7759, 2015.

[13] N. Li, K. Daie, K. Svoboda, and S. Druckmann, "Robust neuronal dynamics in premotor cortex during motor planning," *Nature*, vol. 532, no. 7600, pp. 459–464, 2016.

[14] G. F. Elsayed, A. H. Lara, M. T. Kaufman, M. M. Churchland, and J. P. Cunningham, "Reorganization between preparatory and movement population responses in motor cortex," *Nature Communications*, vol. 7, no. 1, p. 13239, 2016.

[15] P. Gao, E. Trautmann, B. Yu, G. Santhanam, S. Ryu, K. Shenoy, and S. Ganguli, "A theory of multineuronal dimensionality, dynamics and measurement," *bioRxiv*, p. 214262, 2017.

[16] J. A. Gallego, M. G. Perich, L. E. Miller, and S. A. Solla, "Neural Manifolds for the Control of Movement," *Neuron*, vol. 94, no. 5, pp. 978–984, 2017.

[17] G. F. Elsayed and J. P. Cunningham, "Structure in neural population recordings: an expected byproduct of simpler phenomena?," *Nature Neuroscience*, vol. 20, no. 9, pp. 1310–1318, 2017.

[18] L. N. Driscoll, N. L. Pettit, M. Minderer, S. N. Chettih, and C. D. Harvey, "Dynamic Reorganization of Neuronal Activity Patterns in Parietal Cortex," *Cell*, vol. 170, no. 5, pp. 986–999.e16, 2017.

[19] S. Vyas, N. Even-Chen, S. D. Stavisky, S. I. Ryu, P. Nuyujukian, and K. V. Shenoy, "Neural Population Dynamics Underlying Motor Learning Transfer," *Neuron*, vol. 97, no. 5, pp. 1177–1186.e3, 2018.

[20] A. A. Russo, S. R. Bittner, S. M. Perkins, J. S. Seely, B. M. London, A. H. Lara, A. Miri, N. J. Marshall, A. Kohn, T. M. Jessell, L. F. Abbott, J. P. Cunningham, and M. M. Churchland, "Motor Cortex Embeds Muscle-like Commands in an Untangled Population Response," *Neuron*, vol. 97, no. 4, pp. 953–966.e8, 2018.

[21] A. H. Lara, J. P. Cunningham, and M. M. Churchland, "Different population dynamics in the supplementary motor area and motor cortex during reaching," *Nature Communications*, vol. 9, no. 1, p. 2754, 2018.

[22] A. H. Lara, G. F. Elsayed, A. J. Zimnik, J. P. Cunningham, and M. M. Churchland, "Conservation of preparatory neural events in monkey motor cortex regardless of how movement is initiated," *eLife*, vol. 7, p. e31826, 2018.

[23] J. A. Gallego, M. G. Perich, S. N. Naufel, C. Ethier, S. A. Solla, and L. E. Miller, "Cortical population activity within a preserved neural manifold underlies multiple motor behaviors," *Nature Communications*, vol. 9, no. 1, p. 4233, 2018.

[24] J. A. Gallego, M. G. Perich, R. H. Chowdhury, S. A. Solla, and L. E. Miller, "Long-term stability of cortical population dynamics underlying consistent behavior," *Nature Neuroscience*, vol. 23, no. 2, pp. 260–270, 2020.

[25] A. Libby and T. J. Buschman, "Rotational dynamics reduce interference between sensory and memory representations," *Nature Neuroscience*, pp. 1–12, 2021.

[26] S.-i. Amari, "Dynamics of pattern formation in lateral-inhibition type neural fields," *Biological Cybernetics*, vol. 27, no. 2, pp. 77–87, 1977.

[27] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-13, no. 5, pp. 834–846, 1983.

[28] J. J. Hopfield and D. W. Tank, ""Neural" computation of decisions in optimization problems," *Biological Cybernetics*, vol. 52, no. 3, pp. 141–152, 1985. This paper would be good to use to do some additional modeling of the data.

[29] S. Vyas, M. D. Golub, D. Sussillo, and K. V. Shenoy, "Computation Through Neural Population Dynamics," *Annual Review of Neuroscience*, vol. 43, no. 1, pp. 249–275, 2020.

[30] H. R. Wilson and J. D. Cowan, "Excitatory and Inhibitory Interactions in Localized Populations of Model Neurons," *Biophysical Journal*, vol. 12, no. 1, pp. 1–24, 1972. Only read to page six.

[31] K. Wimmer, D. Q. Nykamp, C. Constantinidis, and A. Compte, "Bump attractor dynamics in prefrontal cortex explains behavioral precision in spatial working memory," *Nature Neuroscience*, vol. 17, no. 3, pp. 431–439, 2014.

[32] D. Zipser, B. Kehoe, G. Littlewort, and J. Fuster, "A spiking network model of short-term active memory," *Journal of Neuroscience*, vol. 13, no. 8, pp. 3406–3420, 1993.

[33] K. Briggman and W. K. Jr., "Multifunctional Pattern-Generating Circuits," *Annual Review of Neuroscience*, vol. 31, no. 1, pp. 271–294, 2008.

[34] K. V. Shenoy, M. Sahani, and M. M. Churchland, "Cortical Control of Arm Movements: A Dynamical Systems Perspective," *Neuroscience*, vol. 36, no. 1, pp. 337–359, 2012.

[35] J. Wang, D. Narain, E. A. Hosseini, and M. Jazayeri, "Flexible control of speed of cortical dynamics," *bioRxiv*, p. 155390, 2017.

[36] D. M. Darmohray, J. R. Jacobs, H. G. Marques, and M. R. Carey, "Spatial and Temporal Locomotor Learning in Mouse Cerebellum," *Neuron*, vol. 102, no. 1, pp. 217–231.e4, 2019.

[37] E. D. Remington, S. W. Egger, D. Narain, J. Wang, and M. Jazayeri, "A Dynamical Systems Perspective on Flexible Motor Timing," *Trends in Cognitive Sciences*, vol. 22, no. 10, pp. 938–952, 2018.

[38] A. Parthasarathy, C. Tang, R. Herikstad, L. F. Cheong, S.-C. Yen, and C. Libedinsky, "Time-invariant working memory representations in the presence of code-morphing in the lateral prefrontal cortex," *Nature Communications*, vol. 10, no. 1, p. 4995, 2019.

[39] E. D. Remington, D. Narain, E. A. Hosseini, and M. Jazayeri, "Flexible Sensorimotor Computations through Rapid Reconfiguration of Cortical Dynamics," *Neuron*, vol. 98, no. 5, pp. 1005–1019.e5, 2018.

[40] M. M. Churchland and K. V. Shenoy, "Delay of Movement Caused by Disruption of Cortical Preparatory Activity," *Journal of Neurophysiology*, vol. 97, no. 1, pp. 348–359, 2007.

[41] B. A. Sauerbrei, J.-Z. Guo, J. D. Cohen, M. Mischiati, W. Guo, M. Kabra, N. Verma, B. Mensh, K. Branson, and A. W. Hantman, "Cortical pattern generation during dexterous movement is input-driven," *Nature*, vol. 577, no. 7790, pp. 386–391, 2020.

[42] E. Donchin, K. M. Spencer, and R. Wijesinghe, "The mental prosthesis: assessing the speed of a p300-based brain-computer interface," *IEEE transactions on rehabilitation engineering*, vol. 8, no. 2, pp. 174–179, 2000.

[43] G. Santhanam, S. I. Ryu, M. Y. Byron, A. Afshar, and K. V. Shenoy, "A high-performance brain–computer interface," *nature*, vol. 442, no. 7099, pp. 195–198, 2006.

[44] M. Velliste, S. Perel, M. C. Spalding, A. S. Whitford, and A. B. Schwartz, "Cortical control of a prosthetic arm for self-feeding," *Nature*, vol. 453, no. 7198, pp. 1098–1101, 2008.

[45] J. L. Collinger, B. Wodlinger, J. E. Downey, W. Wang, E. C. Tyler-Kabara, D. J. Weber, A. J. McMorland, M. Velliste, M. L. Boninger, and A. B. Schwartz, "High-performance neuroprosthetic control by an individual with tetraplegia," *The Lancet*, vol. 381, no. 9866, pp. 557–564, 2013.

[46] B. Jarosiewicz, S. M. Chase, G. W. Fraser, M. Velliste, R. E. Kass, and A. B. Schwartz, "Functional network reorganization during learning in a brain-computer interface paradigm," *Proceedings of the National Academy of Sciences*, vol. 105, no. 49, pp. 19486–19491, 2008.

[47] P. T. Sadtler, K. M. Quick, M. D. Golub, S. M. Chase, S. I. Ryu, E. C. Tyler-Kabara, M. Y. Byron, and A. P. Batista, "Neural constraints on learning," *Nature*, vol. 512, no. 7515, pp. 423–426, 2014.

[48] M. D. Golub, M. Y. Byron, and S. M. Chase, "Internal models for interpreting neural population activity during sensorimotor control," *Elife*, vol. 4, p. e10015, 2015.

[49] K. Ganguly and J. M. Carmena, "Emergence of a stable cortical map for neuroprosthetic control," *PLoS Biol*, vol. 7, no. 7, p. e1000153, 2009.

[50] A. M. Green and J. F. Kalaska, "Learning to move machines with the mind," *Trends in neurosciences*, vol. 34, no. 2, pp. 61–75, 2011.

[51] A. L. Orsborn, H. G. Moorman, S. A. Overduin, M. M. Shanechi, D. F. Dimitrov, and J. M. Carmena, "Closed-loop decoder adaptation shapes neural plasticity for skillful neuroprosthetic control," *Neuron*, vol. 82, no. 6, pp. 1380–1393, 2014.

[52] P. T. Sadtler, S. I. Ryu, E. C. Tyler-Kabara, B. M. Yu, and A. P. Batista, "Brain–computer interface control along instructed paths," *Journal of Neural Engineering*, vol. 12, no. 1, p. 016015, 2015.

[53] E. R. Oby, S. Perel, P. T. Sadtler, D. A. Ruff, J. L. Mischel, D. F. Montez, M. R. Cohen, A. P. Batista, and S. M. Chase, "Extracellular voltage threshold settings can be tuned for optimal encoding of movement and stimulus parameters," *Journal of neural engineering*, vol. 13, no. 3, p. 036009, 2016.

[54] D. H. Perkel, G. L. Gerstein, and G. P. Moore, "Neuronal spike trains and stochastic point processes: Ii. simultaneous spike trains," *Biophysical journal*, vol. 7, no. 4, pp. 419–440, 1967.

[55] E. Zohary, M. N. Shadlen, and W. T. Newsome, "Correlated neuronal discharge rate and its implications for psychophysical performance.," *Nature*, vol. 370, no. 6485, p. 140, 1994.

[56] S. Baker, R. Spinks, A. Jackson, and R. Lemon, "Synchronization in monkey motor cortex during a precision grip task. i. task-dependent modulation in single-unit synchrony," *Journal of Neurophysiology*, vol. 85, no. 2, pp. 869–885, 2001.

[57] E. Maynard, N. Hatsopoulos, C. Ojakangas, B. Acuna, J. Sanes, R. Normann, and J. Donoghue, "Neuronal interactions improve cortical population coding of movement direction," *Journal of Neuroscience*, vol. 19, no. 18, pp. 8083–8093, 1999.

[58] M. Y. Byron, J. P. Cunningham, G. Santhanam, S. I. Ryu, K. V. Shenoy, and M. Sahani, "Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity," in *Advances in neural information processing systems*, pp. 1881–1888, 2009.

[59] J. P. Cunningham and M. Y. Byron, "Dimensionality reduction for large-scale neural recordings," *Nature neuroscience*, vol. 17, no. 11, pp. 1500–1509, 2014.

[60] E. R. Oby, M. D. Golub, J. A. Hennig, A. D. Degenhart, E. C. Tyler-Kabara, B. M. Yu, S. M. Chase, and A. P. Batista, "New neural activity patterns emerge with long-

term learning," *Proceedings of the National Academy of Sciences*, vol. 116, no. 30, pp. 15210–15215, 2019.

[61] P. Cisek and J. F. Kalaska, "Neural correlates of mental rehearsal in dorsal premotor cortex," *Nature*, vol. 431, no. 7011, pp. 993–996, 2004.

[62] D. Tkach, J. Reimer, and N. G. Hatsopoulos, "Observation-based learning for brain–machine interfaces," *Current opinion in neurobiology*, vol. 18, no. 6, pp. 589–594, 2008.

[63] M. T. Kaufman, M. M. Churchland, S. I. Ryu, and K. V. Shenoy, "Cortical activity in the null space: permitting preparation without movement," *Nature Neuroscience*, vol. 17, no. 3, pp. 440–448, 2014.

[64] E. M. Trautmann, S. D. Stavisky, S. Lahiri, K. C. Ames, M. T. Kaufman, D. J. O'Shea, S. Vyas, X. Sun, S. I. Ryu, S. Ganguli, and K. V. Shenoy, "Accurate Estimation of Neural Population Dynamics without Spike Sorting," *Neuron*, vol. 103, no. 2, pp. 292–308.e4, 2019.

[65] C. Kopec, J. Erlich, B. Brunton, K. Deisseroth, and C. Brody, "Cortical and Subcortical Contributions to Short-Term Memory for Orienting Movements," *Neuron*, vol. 88, no. 2, pp. 367–377, 2015.

[66] F. A. Kagerer, J. L. Contreras-Vidal, and G. E. Stelmach, "Adaptation to gradual as compared with sudden visuo-motor distortions," *Experimental Brain Research*, vol. 115, no. 3, pp. 557–561, 1997.

[67] J. W. Krakauer, Z. M. Pine, M.-F. Ghilardi, and C. Ghez, "Learning of Visuomotor Transformations for Vectorial Planning of Reaching Trajectories," *Journal of Neuroscience*, vol. 20, no. 23, pp. 8916–8924, 2000.

[68] R. Paz, T. Boraud, C. Natan, H. Bergman, and E. Vaadia, "Preparatory activity in motor cortex reflects learning of local visuomotor skills," *Nature Neuroscience*, vol. 6, no. 8, pp. 882–890, 2003.

[69] R. Paz, C. Nathan, T. Boraud, H. Bergman, and E. Vaadia, "Acquisition and generalization of visuomotor transformations by nonhuman primates," *Experimental Brain Research*, vol. 161, no. 2, pp. 209–219, 2005.

[70] N. Saijo and H. Gomi, "Multiple Motor Learning Strategies in Visuomotor Rotation," *PLoS ONE*, vol. 5, no. 2, p. e9399, 2010.

[71] P. Bédard and J. N. Sanes, "Basal ganglia-dependent processes in recalling learned visual-motor adaptations," *Experimental Brain Research*, vol. 209, no. 3, pp. 385–393, 2011.

[72] S. M. Chase, R. E. Kass, and A. B. Schwartz, "Behavioral and neural correlates of visuomotor adaptation observed through a brain-computer interface in primary motor cortex," *Journal of Neurophysiology*, vol. 108, no. 2, pp. 624–644, 2012.

[73] J. P. Cunningham and Z. Ghahramani, "Linear dimensionality reduction: Survey, insights, and generalizations," *The Journal of Machine Learning Research*, 2015.

[74] H. D. Tagare, *Notes on optimization on stiefel manifolds*. Yale University, 2011.

[75] T. P. Vogels, K. Rajan, and L. Abbott, "NEURAL NETWORK DYNAMICS," *Neuroscience*, vol. 28, no. 1, pp. 357–376, 2005.

[76] F. Arce, I. Novick, Y. Mandelblat-Cerf, Z. Israel, C. Ghez, and E. Vaadia, "Combined Adaptiveness of Specific Motor Cortical Ensembles Underlies Learning," *The Journal of Neuroscience*, vol. 30, no. 15, pp. 5415–5425, 2010.

[77] L. N. Driscoll, M. D. Golub, and D. Sussillo, "Computation through Cortical Dynamics," *Neuron*, vol. 98, no. 5, pp. 873–875, 2018.

[78] M. D. Golub, P. T. Sadtler, E. R. Oby, K. M. Quick, S. I. Ryu, E. C. Tyler-Kabara, A. P. Batista, S. M. Chase, and B. M. Yu, "Learning by neural reassociation," *Nature Neuroscience*, vol. 21, no. 4, pp. 607–616, 2018.

[79] J. A. Hennig, M. D. Golub, P. J. Lund, P. T. Sadtler, E. R. Oby, K. M. Quick, S. I. Ryu, E. C. Tyler-Kabara, A. P. Batista, B. M. Yu, and S. M. Chase, "Constraints on neural redundancy," *eLife*, vol. 7, p. e36774, 2018.