Feature Selection and Decoder Design for Closed-loop Neural Interface Learning

Si Jia Li

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2026

Reading Committee:

Amy L. Orsborn, Chair

Azadeh Yazdan-Shahmorad

Sam A. Burden

Program Authorized to Offer Degree:
Bioengineering

University of Washington

Abstract

Feature Selection and Decoder Design for Closed-loop Neural Interface Learning

Si Jia Li

Chair of the Supervisory Committee:
Amy L. Orsborn
Bioengineering

Neural interfaces, ranging from non-invasive scalp electroencephalography (EEG) and peripheral electromyography (EMG) to invasive intracortical arrays, hold transformative potential for restoring motor and communicative independence to individuals with neurological impairments. However, the translation of these systems from controlled laboratory settings to robust, everyday clinical use is hindered by a universal scalability bottleneck. Across all modalities, advances in recording hardware now permit the simultaneous monitoring of hundreds to thousands of channels, creating a "data deluge" that overwhelms current decoding frameworks. This expansion in feature space presents a critical challenge: clinical systems lack the computational efficiency to process high-dimensional feature sets within the strict power and latency constraints of portable or implantable hardware. Furthermore, the inherent non-stationarity of biological signals—whether cortical or peripheral—necessitates adaptive frameworks that can sustain performance over long durations without burdensome recalibration. This dissertation establishes principled computational frameworks to optimize feature selection and decoder design, effectively bridging the gap between expanding sensor capabilities and the requirements of real-time, robust control.

First, to motivate the need for dimensionality reduction, we characterized the distribution of task information across biophysical, spatial, and spectral scales. Using simultaneous recordings across multiple physiological scales, we demonstrated that neural feature variance is highly redundant and spatially fractured. We identified "hub" electrodes that, while

strongly correlated with broad population dynamics, often encode minimal task-relevant information. These findings challenge the standard practice of indiscriminate feature inclusion and provide the physiological justification for disentangling feature selection from task decoding.

Second, we developed a novel framework for **Adaptive Feature Selection** to address the instability of high-dimensional inputs in closed-loop settings. We demonstrated that applying standard, static feature selection methods directly to online data frequently leads to performance degradation due to statistical volatility. To overcome this, we introduced a dynamic selection algorithm governed by temporal continuity constraints. This approach autonomously identifies and tracks the informative subspace in real-time, stabilizing control performance while significantly reducing the computational load required for decoding.

Finally, we addressed the behavioral complexity of real-world use by developing a closed-loop decoder for **Hybrid Multitasking**. Moving beyond single-degree-of-freedom control, we designed a system that enables users to simultaneously perform continuous tracking and discrete classification. Our results indicate that participants can achieve efficient control over these complex, interfering task demands, and that the rate of user learning is strictly governed by specific decoder adaptation parameters.

Collectively, this thesis advances the critical transition of neural interface technology toward scalable, "plug-and-play" architectures. By solving the dual challenges of efficient online feature selection and robust multitasking control, this work contributes essential methodologies for the development of high-bandwidth clinical systems capable of restoring complex human agency.

TABLE OF CONTENTS

Page

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor, Amy Orsborn. Thank you for your unwavering support, your visionary guidance, and for creating a lab environment that is both intellectually rigorous and incredibly supportive. Your mentorship has been invaluable in shaping not just this thesis, but my entire approach to science and research.

I am also grateful to my committee members—Sam Burden, Azadeh Yazdan-Shahmorad, Matt Golub, Eric Shea-Brown, and Bing Brunton—for their insightful feedback, challenging questions, and the time they dedicated to guiding my doctoral journey. Sam, thank you for your specific contributions and collaboration on the EMG project; your perspective was essential to its success.

To the members of the Orsborn Lab, thank you for being my second family. A special thank you to Pavithra Rajeswaran and Leo Scholl. You have been the backbone of my PhD experience—from teaching me the ropes of the recording rig to the countless walks and pep talks when things got tough. I genuinely could not have done this without your technical wisdom and constant friendship.

To Lauren Peterson, thank you for being such a fantastic collaborator on the feature selection project. It was a joy to brainstorm, debug, and write with you.

To the "EMG team"—Maneeshika Madduri, Momona Yamagami, Sasha Burckhardt, Gus Millevolte, and Amber Chou—thank you for being the best partners in experimentation I could ask for. Thank you for your tireless efforts in running human-subject experiments with me, for the fun brainstorming sessions, and for making the long hours of data collection feel like hanging out with friends.

Finally, thank you to my family and friends for their love and encouragement throughout this process.

Chapter 1

INTRODUCTION

1.1 Development of brain-computer interfaces

Brain-Computer Interfaces (BCIs) have emerged as a transformative clinical technology, offering a direct neural bridge to restore independence for individuals suffering from severe neurological impairments, including spinal cord injury, amyotrophic lateral sclerosis (ALS), and stroke [41, 97]. The therapeutic scope of BCIs has expanded rapidly to encompass the restoration of complex motor functions—driving robotic limbs [16, 2] or reanimating paralyzed muscles via functional electrical stimulation [3, 10]—and the re-enablement of communication through high-speed speech and handwriting decoding [93, 54]. Beyond physical rehabilitation, the field is now traversing new frontiers in cognitive neuroprosthetics, with emerging applications designed to modulate mood disorders and restore memory function [75, 38].

Fundamentally, BCIs restore volitional control by bridging the disconnected central nervous system and periphery through a sequential signal processing pipeline. This process begins with **signal acquisition**, where sensors transduce extracellular ionic currents—ranging from single units to mesoscopic field potentials—into digitized voltage traces. However, transforming these raw signals into precise control commands presents a significant computational challenge. The **decoding** stage must navigate a massive, noisy state space, necessitating robust **feature selection** to isolate the informative neural subspace and accurate **decoder estimation** to model the complex mapping to the intended kinematic state. Finally, this estimated state drives the **actuation** of an external effector to restore patient agency. Realizing the full clinical potential of this pipeline, however, requires more than static components; it demands a synergistic evolution across both recording hardware and decoding architectures, as detailed in the following section.

1.2 Evolution of Hardware and Decoding Algorithms

The advancement of robust, high-performance BCIs necessitates a synergistic evolution across signal acquisition and decoding architecture. On the hardware front, the field is transitioning from focal, low-channel count recordings to massive, distributed interfaces—such as Neuropixels and Argo—that capture thousands of channels across multiple cortical areas [46, 85]. This "data deluge" is further compounded by a combinatorial explosion in the feature space, where modern protocols extract multiple signal modalities—ranging from threshold-crossing spikes to spectral power in distinct local field potential (LFP) bands—from each individual site, effectively multiplying the input dimensionality by an order of magnitude.

Paralleling this expansion, decoding algorithms are evolving from interpretable linear filters to high-capacity deep learning architectures and Transformer-based foundation models capable of modeling non-linear latent dynamics [64, 94]. However, this explosion in data availability presents a critical bottleneck for clinical translation: current decoding frameworks lack the computational efficiency to process high-dimensional feature sets in real-time. This limitation is particularly acute for the development of "everyday-use" BCIs, where algorithms must operate continuously within the strict power, thermal, and latency constraints of fully implantable, mobile hardware [12, 35]. Consequently, bridging the gap between high-channel-count interfaces and clinical viability requires the development of intelligent feature selection algorithms that can dynamically isolate the informative feature space to maximize task reward rates and decoder estimation stability while minimizing the computational burden

1.3 Tradeoff of neural signals in terms of signal quality and invasiveness

To achieve robust closed-loop decoding, the selection of a neural interface must balance the useful signal information density against the long-term stability of the recording. Contemporary BCI technology is defined by a fundamental design trade-off between signal fidelity and surgical risk, driving innovation across three distinct tiers of invasiveness. At the highest level of performance, **invasive systems** such as the Utah Array and Neuralink

physically penetrate the cortex to capture the single-unit resolution required for complex, high-dimensional motor control [53, 57]. The evolution of these interfaces has followed a "Moore's Law" for neuroscience, where the capacity of simultaneous neural recording doubles approximately every seven years [85]. This trajectory has rapidly scaled the field from the 100-channel standard established by the Utah Array to modern probes like Neuropixels 2.0, which now support stable, long-term recording across thousands of channels in freely moving subjects [84].

In contrast, **non-invasive systems** like scalp electroencephalography (EEG) prioritize patient safety and accessibility but face inherent performance limitations. The "skull barrier" acts as a strict low-pass filter that smears high-frequency neural information, restricting decoding capabilities to low-dimensional tasks [1]. To bridge this gap, emerging **minimally invasive solutions**—such as Synchron's endovascular Stentrode and high-density μECoG arrays—offer a compelling middle ground. These systems allow for superior signal stability and spatial resolution without penetrating the brain parenchyma, with modern grids scaling to 244 electrodes to resolve fine-scale cortical maps [89]. Furthermore, modern BCI architectures increasingly integrate **peripheral muscle recordings (EMG)**, fusing cortical intent with residual motor output to enhance decoding robustness. However, effectively leveraging these diverse signal sources requires a rigorous quantitative understanding of how task-relevant information is distributed across biophysical scales and spatiotemporal axes. Simultaneously, ensuring consistent control fidelity necessitates adaptive frameworks capable of mitigating the inherent non-stationarity of these biological signals, which often degrade performance over time. Ultimately, this spectrum of invasiveness dictates the quality and stability of the features available to the decoding pipeline, presenting a fundamental question of whether high-resolution, unstable signals or lower-resolution, stable potentials provide a more effective substrate for user learning—a trade-off I evaluate through the multiscale analysis in Chapter 1 to establish the informative subspace required for the adaptive frameworks developed in Chapters 2 and 3.

1.3.1 Spectral, spatial, and temporal distribution of task information

To maximize closed-loop decoding performance, the decoder must effectively navigate the hierarchical distribution of motor information, which is stratified across distinct biophysical, spectral, spatial, and temporal axes. At the biophysical level, single-unit spikes provide the highest kinematic fidelity but suffer from chronic instability due to tissue reaction [53]. In contrast, mesoscopic signals like Local Field Potentials (LFP) and Electro-corticography (ECoG) offer a stable "middle ground" for chronic decoding [81]. Within these field potentials, information is functionally compartmentalized: High-gamma (> 70 Hz) activity proxies local firing for execution, while low-frequency oscillations (Delta/Theta) distinctly encode trajectory and planning [70, 6].

Spatially, motor representations are not distinct but "fractured" and intermixed, with significant neural variance driven by uninstructed, task-irrelevant movements rather than the target behavior itself [76, 56]. This complexity is compounded by temporal dynamics, where information evolves through a high-dimensional state space over the course of a trial. Neural activity flows from "preparatory subspaces" in frontal regions into "execution subspaces" in the motor cortex, a coordination often mediated by cross-scale interactions between macroscale oscillations and local processing [14, 63]. Decoding these intermixed representations requires computational frameworks—such as the multiscale and adaptive methodologies presented in this thesis—capable of isolating task-relevant features from this high-dimensional, non-stationary signal landscape.

1.3.2 Neural signal is non-stationary

To maintain high-quality closed-loop decoding over long periods, a BCI must be able to handle 'signal non-stationarity'—the fact that brain signals are constantly changing, a phenomenon that violates the fundamental static assumptions of standard decoding algorithms. This instability arises from a confluence of abiotic factors, such as electrode micromotion and reactive gliosis, which progressively alter the interface impedance and signal quality over time [7]. Simultaneously, the system is subject to biotic variability, specifically "representational drift" and plasticity, wherein the neural population code for identical motor outputs

evolves over days or weeks, even when behavior remains stable [22, 62]. This inevitable distributional shift precipitates rapid performance degradation in static decoders, necessitating frequent and burdensome recalibration protocols that hinder the viability of "plug-and-play" clinical systems [45, 67]. Overcoming this degradation requires the transition from static models to the more flexible, adaptive feature selection and decoding frameworks that form the core of my research in Chapters 2 and 3.

1.4 Decoders are becoming more complex and more adaptive

Traditional linear decoding architectures, such as Kalman Filters, have long served as the benchmark for clinical BCIs due to their interpretability, millisecond latency, and data efficiency—attributes that are critical for robust operation in low-power, fully implantable systems [42]. However, these models inherently underfit the rich, non-linear spatiotemporal dynamics of the motor cortex, imposing a strict ceiling on decoding performance [58]. To overcome this limitation, the field has shifted toward high-capacity deep learning models, such as Recurrent Neural Networks (RNNs) and LSTMs, which achieve state-of-the-art accuracy by modeling latent population dynamics and learning features end-to-end from raw neural data [64]. Yet, this performance comes at a cost: the "data hungry" nature and opacity of these "black box" architectures pose significant risks of overfitting and regulatory challenges in clinical contexts characterized by sparse, non-stationary data [32, 74].

1.4.1 Adaptive decoder outperform fixed decoders

To overcome the limitations of static models, the field has increasingly adopted Closed-Loop Decoder Adaptation (CLDA), a paradigm that consistently outperforms fixed decoders by updating parameters in real-time to match the user's evolving neural state [31, 62]. This approach is critical for bridging the "open-loop gap," as neural statistics often shift significantly once the user actively engages in a task; by training on actual closed-loop data, adaptive algorithms correct for these discrepancies to yield significantly faster and more accurate control [31, 100]. Beyond immediate performance gains, CLDA is essential for mitigating non-stationarity: while fixed models degrade rapidly due to biological plasticity or electrode drift, adaptive algorithms (e.g., self-calibrating Kalman filters) autonomously

track these fluctuations to maintain high performance over weeks without daily retraining [44, 12]. Ultimately, these frameworks resolve the "two-learner" problem by facilitating principled co-adaptation, where the decoder actively assists the user by stabilizing the control manifold, thereby accelerating the motor learning curve and reducing cognitive load [62, 79].

1.4.2 Limitations of Existing Adaptive Frameworks

Despite the success of Closed-Loop Decoder Adaptation, significant barriers remain in scaling these frameworks to next-generation interfaces and complex behaviors. First, a **Scalability Gap** exists because standard adaptive decoders (e.g., SmoothBatch) rely on a preselected, fixed set of neural features. As recording technologies scale to thousands of channels (e.g., Argo, Neuropixels), continually updating weights for the entire high-dimensional feature set becomes computationally prohibitive and statistically unstable. This "curse of dimensionality" creates a critical need for **Adaptive Feature Selection** methods capable of dynamically isolating the informative subspace in real-time, discarding noise to maintain efficiency without sacrificing control [20, 48].

Simultaneously, a **Complexity Gap** limits current protocols, which typically focus on single degrees of freedom, such as simple cursor control. These systems fail to account for "hybrid" scenarios—such as simultaneous continuous tracking and discrete classification—where the neural representations are often entangled. In these contexts, learning one task can disrupt the representation of the other, necessitating the development of closed-loop coadapation strategies designed to explicitly minimize such interference and enable robust multitasking [68].

1.5 Critical challenges

The "Curse of Dimensionality." The shift toward high-density interfaces like Neuropixels has created a "data deluge," enabling the recording of thousands of channels but introducing the "Curse of Dimensionality" [85]. Decoders trained on these massive feature sets with limited clinical data are prone to overfitting [18]. Furthermore, processing

this volume of data at millisecond resolution exceeds the bandwidth of current wireless architectures, creating an urgent need for efficient online subspace isolation [29].

Neural Non-Stationarity. Compounding this scale is the inherent instability of the neural interface. Signals drift over time due to reactive gliosis, neuron turnover, and electrode micromotion, which destabilize the relationship between neural activity and motor output [67]. Consequently, static decoders degrade rapidly. To maintain long-term viability without daily recalibration, clinical systems require adaptive algorithms capable of autonomously tracking these fluctuations in real-time [45, 62].

Real-Time Clinical Constraints. Finally, the transition to fully implantable devices imposes severe efficiency limits. Closed-loop systems must process signals within < 100 ms to preserve user agency [17], while implantable ASICs must operate within strict thermal budgets ($< 1°C$) to prevent tissue damage [35]. These constraints prohibit the use of computationally heavy deep learning models on-chip, necessitating highly efficient algorithms to enable fully wireless, long-term use [39].

1.6 Thesis Organization

My doctoral research addresses the fundamental computational bottlenecks hindering the translation of high-bandwidth neural interfaces into robust clinical systems. Specifically, I focus on establishing efficient algorithmic frameworks—spanning feature characterization, adaptive selection, and hybrid architecture design—to bridge the gap between the explosive growth in sensor data and the strict efficiency requirements of real-time control (Figure 1.1).

First, to navigate the trade-off between massive signal density and actual information content, I conducted a quantitative multiscale analysis of simultaneous intracortical and electrocorticographic (ECoG) recordings. This work rigorously characterizes the spatiotemporal distribution of task-relevant variance, identifying the physiological boundaries of useful information (Chapter 1).

Second, to resolve the conflict between high-dimensional feature sets and the latency constraints of implantable hardware, I developed an Adaptive Feature Selection algorithm. This framework enables the real-time isolation of the informative neural subspace within non-stationary environments, ensuring stable decoding without exceeding computational

budgets (Chapter 2).

Finally, to extend BCI capabilities beyond simple degrees of freedom, I designed a Hybrid Decoder architecture explicitly engineered to manage task interference. This system demonstrates how structured decoding paradigms can enable robust performance in complex multitasking scenarios that require simultaneous continuous tracking and discrete classification (Chapter 3).

Chapter 2

MULTISCALE ANALYSIS OF NEURAL FEATURES

2.1 Introduction

Neural signals are multiscale, requiring methods that compare activity across different physiological scales. This comparison is critical because neural information is increasingly multidimensional. It is also very important to compare these measurements because ECoG is minimally invasive, whereas intracortical recordings are invasive. A key scientific question regarding these scales is depth-dependence. For instance, in the prefrontal cortex, local field potentials (LFPs) in shallow layers (1 mm) contain considerably greater information than those in deeper layers or on the cortical surface [52]. Conversely, deeper layers have been shown to contain more information in high frequencies (> 300 Hz) [65]. Despite these findings, we still do not know how surface signals quantitatively compare to signals at different intracortical depths.

Beyond spatial comparisons, it is also important to examine temporal variability and spectral structure in neural features. [15] showed that neural variability quenches at stimulus onset. While this phenomenon is well-established in spiking activity, it remains unclear if this reduction in variability translates to the mesoscopic scales of LFP and ECoG. Additionally, [37] proposed that low-frequency activity (1–4 Hz) arises from intrinsic rhythms facilitating synchronization, suggesting that low frequencies coordinate widespread networks while high frequencies are more focal. However, a critical gap remains: we lack a comprehensive understanding of how this neural variability and spectral dynamics manifest across multiple recording scales.

Understanding how signals correlate across scales requires examining the mechanisms of network connectivity on different scales. Neural communication is thought to be gated by Communication Through Coherence (CTC), where synchronized excitability windows route information through specific frequency channels—typically gamma for feedforward

and beta for feedback signals [27, 8]. This spectrally mediated signaling naturally converges onto functional "hubs," establishing a small-world topology optimized for integrating diverse computations [83, 91]. A major open question is how this frequency-specific architecture and hub organization translates across the disparate spatial scales of simultaneous surface and intracortical recordings.

To answer these questions regarding depth, variability, and connectivity, we leveraged a multiscale setup and recorded data using the task paradigm of [62]. We demonstrate that task-relevant information concentrates in shallow cortical depths, confirming that ECoG primarily reflects superficial activity. Intracortical SC32-LFP signals exhibit higher dimensionality than surface ECoG, with this dimensionality dynamically changing around movement onset, making it a potential BCI biomarker. Crucially, sparsely connected "Non-hub" electrodes were found to be more critical for high-accuracy movement decoding than highly-connected "Coherence Hubs," which carry more redundant, common-mode signals, providing key insights for optimal BCI electrode placement and feature selection.

2.2 Methods

2.2.1 Neural recording hardware and signal processing pipeline

We leveraged a custom-designed multiscale recording interface capable of simultaneous surface and intracortical monitoring, as detailed in [61]. The assembly consisted of a semi-chronic microdrive system integrated with a custom recording chamber and an artificial dura (Figure 2.1A). This setup housed a high-density electrocorticography (ECoG) array with 211 contacts to provide broad surface coverage. Co-registered with this surface grid were 32 intracortical microelectrodes, which could be manually lowered via the microdrive to target specific cortical depths. The arrays were implanted to span key sensorimotor areas, including the dorsal premotor (PMd), primary motor (M1), and primary somatosensory (S1) cortices, positioned relative to the central (CS) and arcuate (ArS) sulci (Figure 2.1B).

Raw LFP voltage traces were band-pass filtered (0.3–500 Hz) and downsampled to 1 kHz. Time-resolved power spectral density (PSD) was estimated using the multitaper method [55] with a 300 ms moving window, 50 ms step size, time-bandwidth product $NW = 3$, and

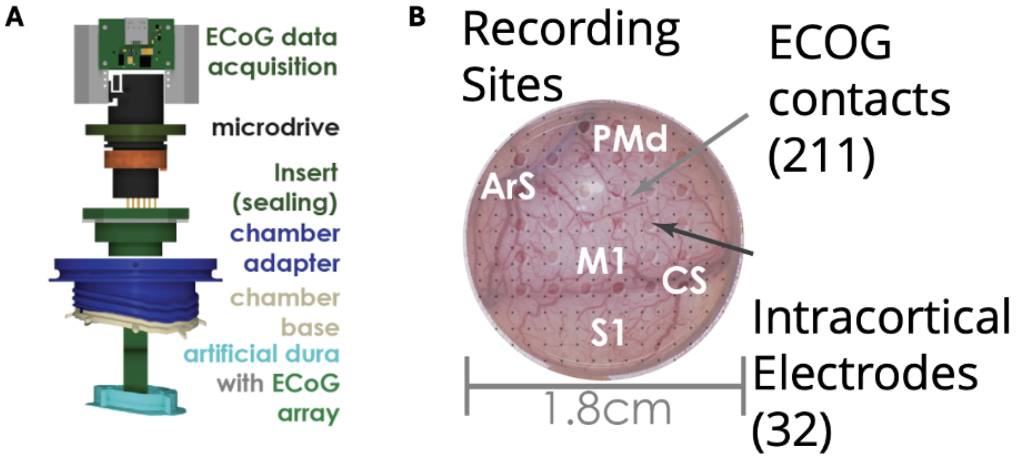

Figure 2.1: **Multiscale Recording Interface and Electrode Placement. A:** A schematic of the custom recording assembly, showing the data acquisition interface, microdrive, sealing components, and the artificial dura housing the ECoG array. **B:** Cortical map showing the coverage of the 211 ECoG contacts (surface) and 32 intracortical electrodes relative to anatomical landmarks. The recording footprint spans the dorsal premotor (PMd), primary motor (M1), and primary somatosensory (S1) cortices. CS: Central Sulcus; ArS: Arcuate Sulcus. (Modified from Orsborn et al., 2014).

$K = 5$ tapers. The resulting PSDs were integrated into six canonical frequency bands: Delta (0.5–4 Hz), Theta (4–8 Hz), Alpha/Mu (8–12 Hz), Beta (12–30 Hz), Low Gamma (30–80 Hz), and High Gamma (80–150 Hz) [6]. To normalize, band power was log-transformed and z-scored relative to a 500 ms baseline period preceding the target cue.

2.2.2 The tasks

To assess signal stability, we employed a visual stimulation task where the subject viewed a full-field white flash to elicit visually evoked potentials (VEPs); as a fundamental sensory response, these signals were not expected to change with learning. We then implemented a center-out reaching task [62]. Trials were initiated when the cursor (0.5 cm radius) acquired and held a central target (2 cm radius) for 200 ms. Upon a successful hold, one of eight peripheral targets (2 cm radius) appeared at 45° intervals, 10 cm from the center. Successful acquisition required reaching the target within 10 s and holding for 200 ms. Trials were separated by a 500 ms inter-trial interval, with the cursor bounded within a 20 cm square and not reset between trials.

2.2.3 Measuring dimensionality and connectivity

To characterize the neural state space, we performed Principal Component Analysis (PCA) on the covariance matrix to obtain the full eigenvalue spectrum $(\lambda_1, \ldots, \lambda_N)$. Rather than applying an arbitrary variance threshold, we quantified the effective dimensionality (D_{PR}) using the Participation Ratio, a threshold-free metric defined as:

$$D_{PR} = \frac{(\sum_{i=1}^{N} \lambda_i)^2}{\sum_{i=1}^{N} \lambda_i^2} \tag{2.1}$$

where λ_i are the eigenvalues of the covariance matrix [72].

For functional coupling, we computed intra-modal (ECoG-ECoG, LFP-LFP) and inter-modal Magnitude-Squared Coherence (MSC) between all unique channel pairs using Welch's method with 300 ms moving windows, yielding a normalized connectivity measure (0–1) for each frequency band.

2.2.4 Decoding task information via Linear Discriminant Analysis (LDA)

To quantify target-specific information, we predicted the intended reach target (one of K locations) using trial-segmented neural features. Feature vectors consisted of either raw LFP voltage traces or z-scored spectral power averaged within a task-relevant epoch (e.g., 200–600 ms post-cue) to capture planning and early execution signatures. Classification was performed using Linear Discriminant Analysis (LDA), a generative classifier that models class-conditional densities as multivariate Gaussians with a shared covariance matrix. This method was selected for its robustness in BCI applications with limited trial counts, as it is less prone to overfitting than complex non-linear models [9].

2.3 Results

2.3.1 Flash trials show broad depth-dependent connectivity and signal stability

To ensure the reliability of our multiscale analysis, we first assessed the stability of neural signals over the duration of the experiment. We analyzed Visual Evoked Potentials (VEPs) recorded during flash trials across five separate days. The mean VEPs for surface ECoG electrodes exhibited qualitatively similar waveforms across all sessions, preserving a consistent morphology throughout the recording period (Figure 2.2A, top). Similarly, intracortical LFP signals recorded by the SC32 probe showed consistent evoked responses across days (Figure 2.2A, bottom). This consistency confirmed that the neural preparation remained stable, allowing for reliable longitudinal comparisons of signal properties.

Leveraging this stability, we examined the relationship between surface and intracortical signals as a function of cortical depth. We calculated the correlation coefficient between simultaneously recorded ECoG and SC32-LFP signals. We observed a distinct depth-dependent trend in signal similarity: intracortical LFPs recorded at shallower depths (closer to the cortical surface) exhibited significantly higher correlation with the ECoG signals (Figure 2.2B). As the microdrive electrodes descended into deeper cortical layers, this correlation progressively diminished. This result quantifies the spatial reach of the surface recordings, confirming that ECoG signals preferentially reflect neural activity from superficial cortical layers.

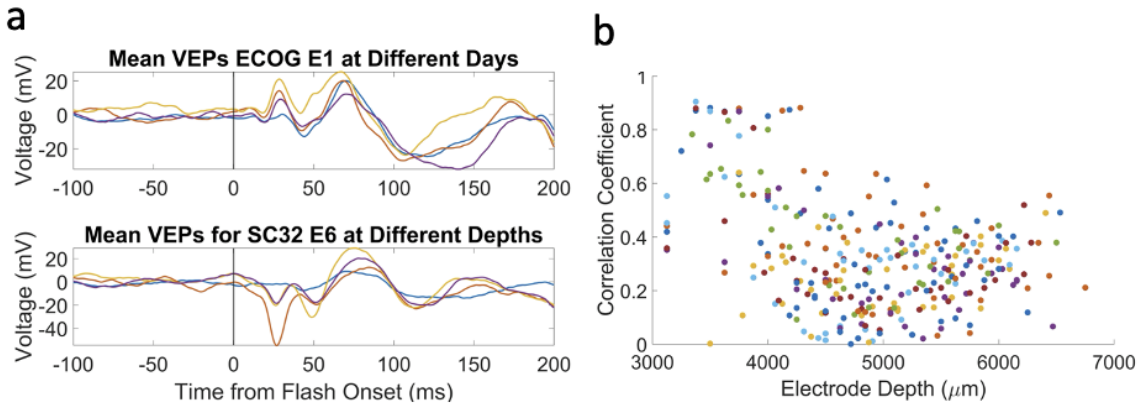

Figure 2.2: **Stability and Depth-Dependence of Visual Evoked Potentials (VEPs). A:** Plot of mean VEP waveforms aligned to flash onset (vertical line, $t = 0$) for a representative ECoG electrode (top) and SC32-LFP electrode (bottom) across five recording sessions (indicated by color). The example electrode was lowered to different depths at different days in comparison to the top plot. **B:** Correlation between simultaneous ECoG and SC32-LFP signals plotted against intracortical electrode depth. Each data point represents the correlation coefficient for a single electrode pair on a given day.

2.3.2 Dimensionality and Signal Complexity Across Scales

To characterize the information content and complexity of the neural signals, we performed Principal Component Analysis (PCA) on the broadband signals from both ECoG and intracortical SC32 recordings. We first examined the cumulative variance explained by the principal components (Figure 2.3A). Despite the significantly smaller number of recording sites for the SC32 probe compared to the ECoG grid (14 vs. 208 contacts), the SC32 signals exhibited higher dimensionality; the cumulative variance curve for SC32 rose more slowly, indicating that a greater number of components were required to capture the signal variance compared to the ECoG. This suggests that intracortical recordings capture a more diverse, less redundant state space than the surface recordings.

We next investigated whether this dimensionality was frequency-dependent. We calculated the effective dimensionality across canonical frequency bands (Figure 2.3B). Consistent with the hypothesis that higher frequencies represent more localized processing, we observed that effective dimensionality increased with frequency. Gamma (30–80 Hz) and high-gamma

(80–150 Hz) bands displayed significantly higher dimensionality compared to low-frequency bands. In these high-frequency regimes, the SC32 recordings consistently maintained higher dimensionality than the ECoG, reinforcing the notion that intracortical probes capture localized neural processes that are spatially smoothed in the surface aggregate.

Finally, we assessed how signal dimensionality evolves dynamically during task engagement. In visual stimulation (flash) trials, the effective dimensionality remained relatively stable throughout the trial duration for both modalities, showing no significant modulation by the onset of the visual stimulus (Figure 2.3C). In contrast, during the center-out reaching task, the dimensionality exhibited distinct temporal dynamics (Figure 2.3D). Qualitatively, the SC32 signal dimensionality expanded noticeably around the time of movement onset ($t = 0$). This expansion suggests that the local motor cortex engages a larger neural state space to execute complex movement, a feature that was more pronounced in the intracortical depth recordings than in the surface ECoG.

2.3.3 Single electrode analyses show non-uniform distribution of task information

To quantify the task-relevant information available at different scales, we performed trial-by-trial decoding of the reach target direction using Linear Discriminant Analysis (LDA). We first characterized the temporal dynamics of information availability by varying the decoding window's start time and duration (Figure 2.4A, B). For both intracortical (SC32) and surface (ECoG) signals, classification accuracy peaked when the decoding window started approximately 200 ms after trial initiation. This indicates that directional motor intent is most robustly represented in the network shortly after the movement cue/onset, with both scales capturing this temporal "sweet spot."

We then examined how this information is distributed across cortical depth. By plotting the maximum decoding accuracy against the recording depth of the SC32 electrodes (Figure 2.4C), we observed a depth-dependent trend: classification accuracy exhibited a subtle decrease as electrode depth increased. This suggests that task-relevant directional information is most concentrated in the superficial layers of the cortex, consistent with the higher correlation to ECoG signals observed previously.

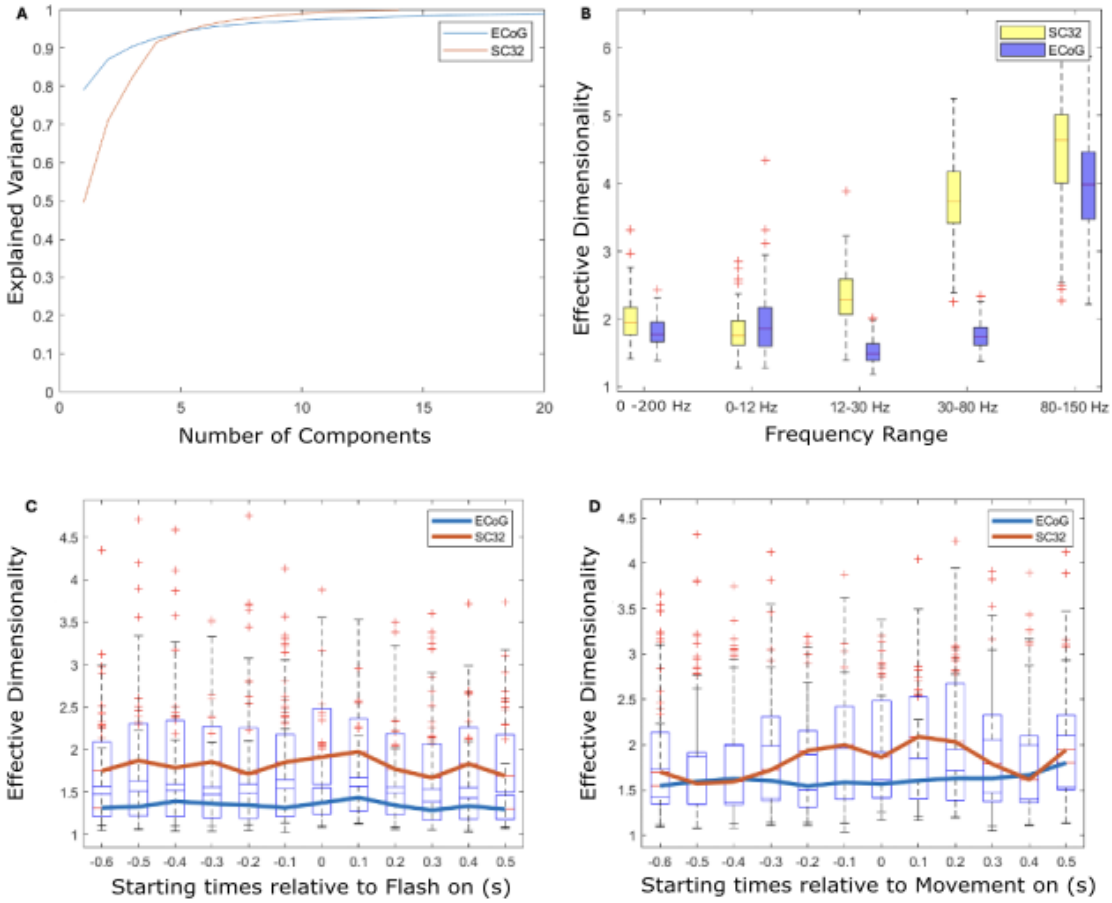

Figure 2.3: **Cumulative variance and effective dimensionality. A:** Cumulative variance with increasing number of components for ECoG and SC32. Data taken from -0.6 s to 0.6 s across the electrodes (208 electrodes for ECoG and 14 lowered electrodes for SC32) from one example trial. **B:** Effective dimensionality at different frequency bands. Data from 121 trials aligned to 1.2 seconds around movement start. **C:** Boxplot of effective dimensionality around the flash on with only filtered ECoG (208 contacts) and SC32 (14 contacts) and 121 trials. **D:** Boxplot comparison of effective dimensionality at different behavior time points. The lines connect the medians at the time points.

Finally, we mapped the spatial topography of decoding performance (Figure 2.4D). We found that high-accuracy electrodes ("decoding hubs") were not clustered in a single location but were distributed across the recorded area. Notably, high-performance SC32 electrodes (colored dots) often spatially co-occurred with high-performance ECoG electrodes (large red circles), further linking the local and macro-scale representations of the motor task.

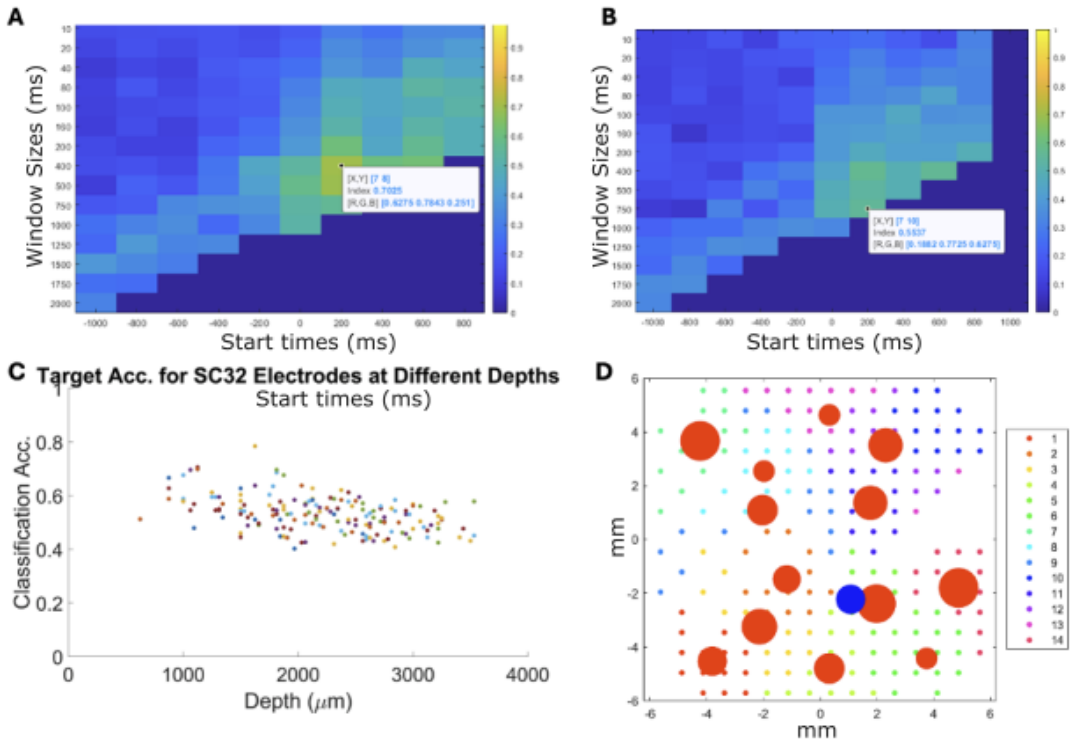

Figure 2.4: **Spatiotemporal Dynamics of Task Decoding. A:** Classification accuracy heatmap for a representative SC32 electrode as a function of temporal parameters. The color indicates LDA decoding accuracy for different window start times (x-axis) and window durations (y-axis). The inset highlights the optimal window (max accuracy) occurring around 200 ms post-start. **B:** Corresponding temporal optimization heatmap for a representative ECoG electrode. **C:** Plot of classification accuracy vs. cortical depth. Each dot represents a single SC32 electrode's max accuracy; a trend of decreasing accuracy is visible with increasing depth. **D:** Scatter plot of decoding accuracy for ECoG electrodes (red) and SC32 electrodes (blue). The size of each marker is proportional to its normalized classification accuracy.

2.3.4 Spectral Connectivity and the Informational Role of Hubs

To determine if the identified network topology reflects functional information content, we first validated the spectral properties of our signals during the motor task. Both surface ECoG and intracortical SC32-LFP electrodes exhibited distinct spectral modulation during the movement phase. Specifically, we observed a characteristic increase in high-gamma power synchronized with movement onset ($t = 0$) in both modalities (Figure 2.5A, B). This shared spectral activation profile confirms that the high-frequency bands used for our connectivity analysis are physiologically relevant to the motor execution.

Using this high-gamma activity, we constructed a functional network by calculating the spectral coherence between every pair of ECoG and SC32 electrodes. By applying a strict coherence threshold (> 0.8), we generated a binary connectivity map (Figure 2.5C). This analysis revealed a non-uniform topology: a specific subset of SC32 electrodes (yellow dots) exhibited dense connectivity to the ECoG grid, identifying them as "Coherence Hubs." In contrast, other "Non-hub" intracortical electrodes showed sparse or no significant coherence with the surface, despite being located within the same cortical area.

We then tested the hypothesis that these topological hubs act as centers of task-relevant information. We compared the target decoding accuracy of the identified Hub electrodes against randomly sampled subsets of Non-hub electrodes using a "neuron drop" analysis (Figure 2.5D). Contrary to the "rich club" hypothesis, the Hub electrodes (blue dashed line) yielded relatively low decoding accuracy ($\approx 26\%$). Strikingly, random subsets of Non-hub electrodes outperformed the Hubs with as few as three electrodes ($N = 3$). As more Non-hubs were added to the decoder, accuracy continued to rise significantly (up to $\sim 70\%$). This suggests that while Hubs are strongly coupled to the global field, they likely carry redundant "common mode" signals. Conversely, Non-hub electrodes appear to capture unique, localized, and high-dimensional information that is critical for decoding specific movement vectors.

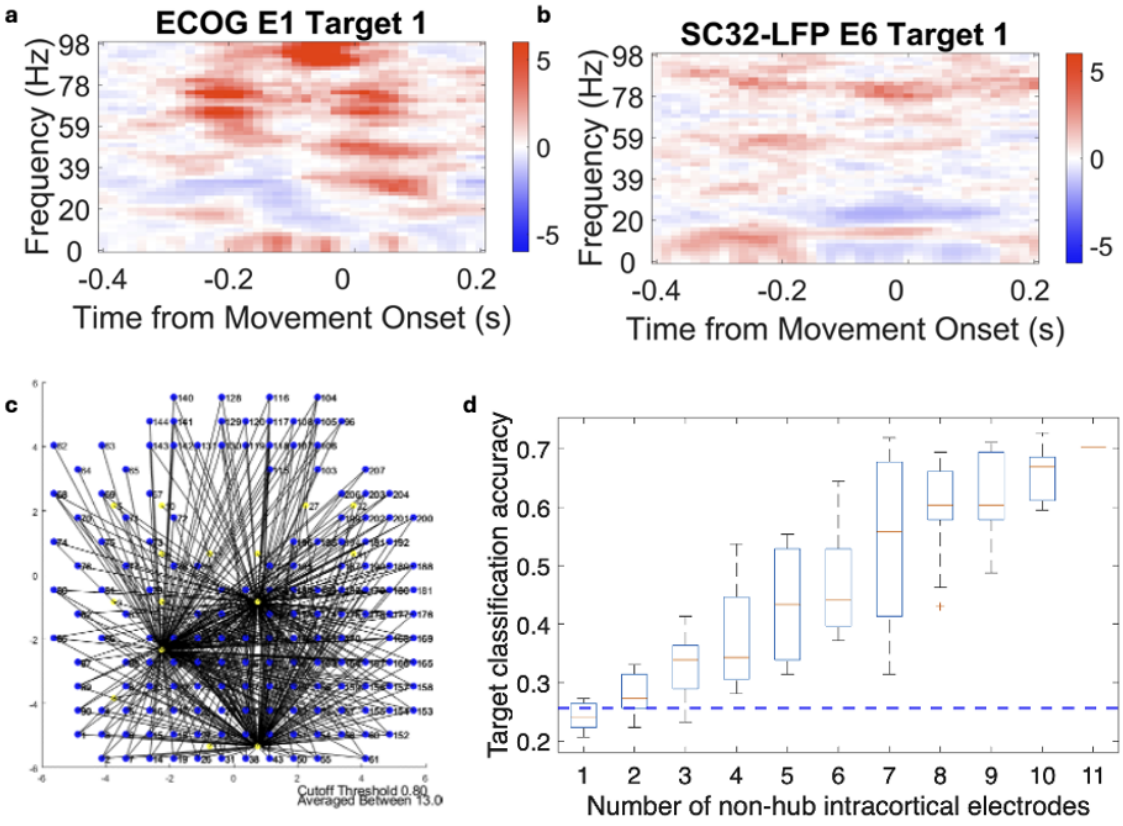

Figure 2.5: **Spectral correlation between ECoG and LFP electrodes. A:** Normalized spectrum for an example ECoG electrode as the monkey was moving the cursor towards a target. **B:** Normalized spectrum for an example SC32-LFP electrode. **C:** Plot of connectivity map calculated from thresholding spectral coherence between every pair of ECoG-SC32 LFP electrodes. Blue and yellow dots are ECoG and SC32 electrodes, respectively. An edge is drawn if the spectral coherence exceeds 0.8. **D:** Neuron drop test for the multiscale feature selection. The blue horizontal line represents accuracy calculated using the hub electrodes from C.

2.4 Discussions and conclusions

In this study, we characterized the multiscale dynamics of motor cortical activity by simultaneously recording from surface ECoG and variable-depth intracortical electrodes. Our results reveal three key principles of signal organization relevant to BCI design. First, we found that task-relevant LFP information is not uniformly distributed but concentrates in shallow cortical layers, confirming that surface signals efficiently capture superficial dynamics. Second, we observed distinct dimensional dynamics between scales, where intracortical variability "quenches" at movement onset to serve as a robust biomarker for motor preparation. Finally, we identified a functional trade-off in network topology: while highly connected "Coherence Hubs" integrate global activity, it is the sparsely connected "Non-hub" electrodes that encode the unique, high-dimensional features essential for accurate movement decoding.

Our analyses offer insights for BCI electrode placement strategies. The observation that task information concentrates in shallow cortical layers suggests that for direct task decoding, targeting electrodes at the surface level (approximately 2000 μm depth) may be advantageous. However, a limitation is that we restricted our analysis to LFP frequencies up to 150 Hz. We did not analyze single-unit action potentials (SUA) or signals above 300 Hz, which have been shown to contain greater information in deeper layers [65]. Consequently, future work is needed to characterize SUA at these depths. While shallow layers appear sufficient for LFP-based decoding, electrodes in deeper layers might instead serve as information hubs, potentially functioning as monitors of "network health" to track status changes that influence decoding stability.

Our analyses demonstrate that variability in SC32-LFP signals, as measured by dimensionality, shrinks at movement onset and subsequently expands during movement execution. This dynamic appears behaviorally meaningful: Churchland et al. (2006) showed that firing rate variability in the dorsal premotor cortex (PMd) correlates with reaction time and reach speed, suggesting that neural activity must align within a specific subspace before proper behavioral execution [14]. Consequently, this dimensionality shift offers a valuable biomarker for naturalistic BCI control, which requires robust detection of the transition

from idle to execution modes. By leveraging this biomarker to identify movement intention, a BCI could trigger a switch to a dedicated movement decoder.

Finally, our network analysis revealed a striking trade-off between connectivity and information content. We identified a subset of "Coherence Hubs"—electrodes with dense spectral connectivity to the global field—that paradoxically yielded poor decoding performance. In contrast, sparsely connected "Non-hub" electrodes were far more effective at encoding specific movement vectors. This distinction echoes recent findings at the single-neuron level, where Finkelstein et al. (2025) identified a similar architecture in the murine motor cortex: sparse "hub neurons" were found to be weakly tuned to task variables (such as target location) yet strongly influenced neighboring cells, whereas precise feature encoding was driven by locally connected, non-hub populations [26]. This convergence across scales suggests a fundamental principle of motor cortical organization: highly connected elements likely serve as "common mode" coordinators or homeostatic regulators—broadcasting state changes like movement onset—while low-degree nodes maintain the high-dimensional, diverse activity patterns necessary for precise kinematic representation. For BCI design, this implies a counter-intuitive strategy: rather than targeting the most active or connected sites, feature selection should prioritize sparsely connected nodes that exhibit high independence from the global signal.

Chapter 3

ONLINE SMOOTH FEATURE SELECTION VIA CONVEX SMOOTH OPTIMIZATION

Si Jia Li, **Lauren N. Peterson**, **Pavithra Rajeswaran**, **Leo Scholl**, and **Amy L. Orsborn**

Abstract

With the advancements in neural recording technologies, the number of recordings is increasing; and yet there are bottlenecks in how we process the increasing number of neural features. Feature selection methods select highly relevant and sparse sets of features. Conventional feature selection methods are built on the assumption of large training data sets and stationary feature distribution. These methods may not apply to online feature selection for neural interface learning where hardware bottlenecks exist to disallow simultaneous streaming of multitudes of features. We propose an online feature selection algorithm that optimizes for feature selection relevance, sparsity, and smoothness. We compared this algorithm to Lasso regression for feature selection in simulation as well as in selecting offline electrocorticography features. With an explicit goal to optimize consistent features online, the simulation studies have shown that the smooth convex algorithm resulted in smooth feature selections and improved simulation task performance.

3.1 Introduction

Brain-computer interfaces (BCIs) are a promising therapy to restore motor function, but many technical challenges limit the ability to scale existing approaches, potentially limiting performance. BCIs measure a subset of neural activity and translate these signals into control commands for external devices through a decoder. BCIs that adapt the decoder during real-time BCI use (in closed-loop with respect to the user) consistently outperform algorithms where decoders are trained offline (in open-loop with respect to the user) in both

linear and non-linear decoder algorithms [98, 31, 44, 95]. These adaptive frameworks also provide opportunities for BCIs to adapt to individual users over time [87, 62, 80, 50, 79]. Existing adaptive algorithms only adapt a subset of the full transformation from neural activity to movement because they keep the neural activity used as inputs to the decoder, or neural features, fixed. Neural recording hardware continues increase device measurement capacity (channel counts) [86, 43, 90, 92, 101], drastically expanding the number of neural features that can be used as inputs to a BCI decoder [92, 43, 11, 82]. This opens new opportunities and challenges to design efficient and scalable real-time adaptive BCI algorithms. For instance, adaptive decoding algorithms that adaptively select neural features could ultimately improve efficiency given real-world hardware constraints such as data transfer rates, network latency, and power consumption [96, 24, 47, 25, 59, 84].

Adaptively selecting the neural features used for decoding could provide advantages that complement existing closed-loop decoder parameter training. Closed-loop decoder parameter adaptation has been useful for maintaining performance as neural measurements and representations drift due to state changes [44]. Adapting neural features could address additional changes that impact BCI performance over time, such as hardware failures and degradation in neural recording quality [23]. Adaptive feature selection may also be beneficial in cases where neural representations of movement themselves adapt due to learning. Existing approaches for feature selection are primarily designed for offline applications and assume stationary data distributions [9, 28], and are therefore not well-suited to address these sources of performance variability in BCI. Offline methods are also typically designed to leverage large datasets to optimize feature selection [34], which introduces challenges in online, real-time BCI settings that operate in severely limited data regimes (e.g., minutes to hours of data) [12, 49, 77]. This limited data, combined with non-stationary neural signals, can easily lead to performance-limiting overfitting [40]. Developing online feature selection methods will ultimately enable BCIs that can adaptively and efficiently leverage high channel-count neural recording devices.

Online feature selection opens opportunities, but integrating online feature selection with online decoder weight estimation introduces new complexities to BCI algorithm training. Current BCIs use offline algorithms to predetermine a fixed set of neural features

and then use online methods to adapt the decoder weights (Figure 1A). In such systems, decoder weights are re-estimated for all features whenever the decoder is updated, treating feature selection and decoder weight estimation computations independently. Online feature selection changes this dynamic: the order of operations matter, resulting in two possible strategies. One strategy is to estimate decoder weights for all available features before selecting a subset (Figure 1B). This ensures decoder weights for all possible features remain accurate but may ultimately be computationally expensive as channel counts increase. Another strategy is to sub-select features first and then estimate weights only for the subset of features (Figure 1C). While efficient, this approach may introduce temporal history dependence, since estimated decoder weights may not match the selected feature subset, potentially compromising BCI performance.

We aimed to investigate the interactions between adaptive feature selection and adaptive decoder parameter estimation in order to develop high-performance online adaptive feature selection algorithms that offered a path to efficient scalability. We hypothesize that the history dependence issues inherent in combined online decoder estimation and feature selection could be mitigated by designing algorithms that enforce temporally consistent feature selection. Explicitly designing algorithms that parametrically control the temporal dynamics of selected features (e.g., gradual vs. abrupt changes) could provide principled computational tools to balance the requirements of feature selection and decoder weight estimation computations.

We used BCI simulations to systematically evaluate how different feature selection methods influence task performance and develop new computational frameworks for online feature selection. Using simulations with a known encoder model established a ground truth to make straightforward comparisons between methods. We demonstrated the potential advantages of online feature selection by showing it can maintain performance when neural encodings shift. We also show that existing feature selection methods didn't perform well unless all decoder weights were estimated before features were selected, a potentially inefficient approach with scaling limitations. We addressed this by developing a feature selection algorithm that uses computationally-efficient convex optimization to consider multiple factors in feature selection. We use this framework to show that constraints on the temporal

continuity of selected features can eliminate the need to estimate all decoder weights, potentially improving overall efficiency. We characterized our feature selection algorithms' performance under varying neural encoding and decoding scenarios in simulation. Finally, we tested how our algorithm performed with real electrophysiology data recorded from behaving macaque monkeys offline to assess translation to experimental settings. Together, our work demonstrates that enforcing temporal continuity constraints is the key mechanism that enables effective online feature selection without the need for computationally expensive full decoder estimation.

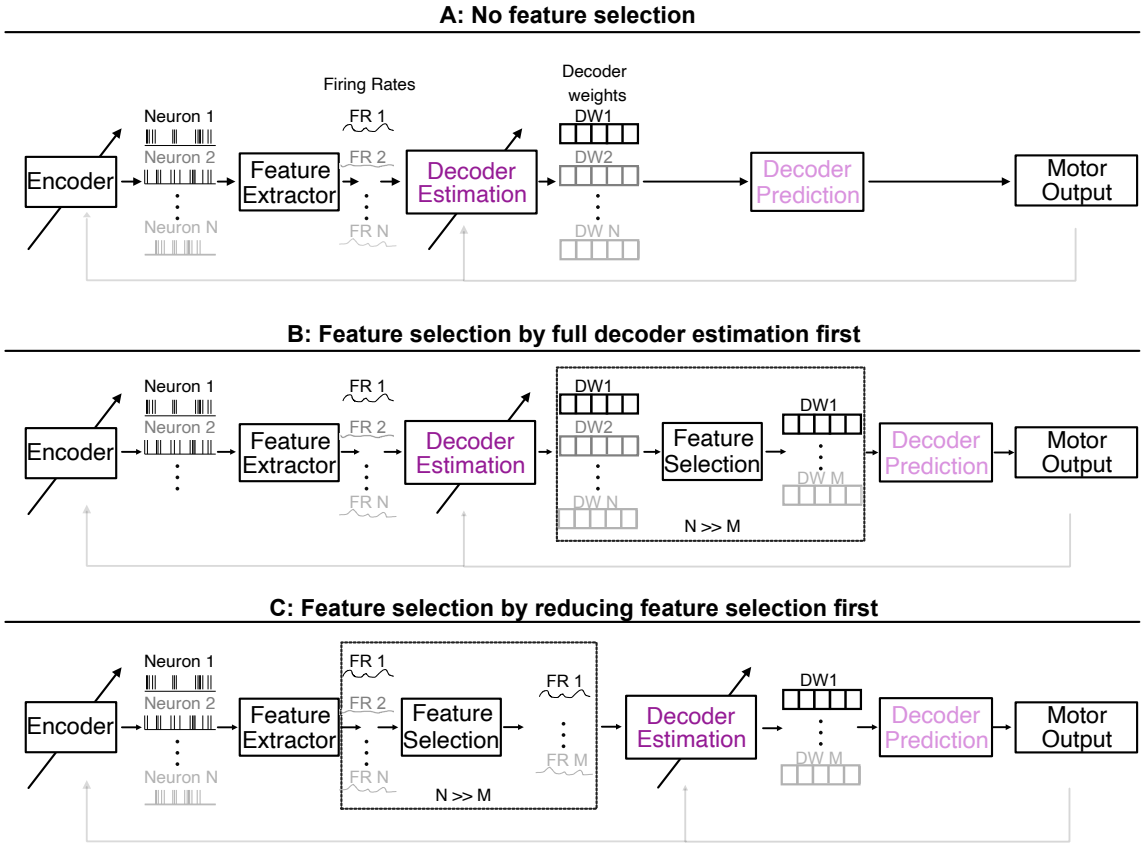

Figure 3.1 Schematics of online feature selection approaches. A: Existing online BCI models. The brain (encoder) generates neural activity (spikes) based on task demands. The feature extractor extracts a wealth of neural features (Firing Rates from N neurons), which are used to estimate decoder weights (DW) along with task information. The esti-

mated decoder is used to make motor predictions. Diagonal arrows through the encoder and decoder represent updates that can occur throughout the task. The features used for neural decoding are static. **B:** Schematic of online feature selection by reducing the decoder size. All features are extracted and decoder parameters are estimated for all features. Feature selection is then performed to drastically downsize the number of decoder weights (dotted box $M \ll N$) for predicting motor output. **C:** Schematic of online feature selection by reducing the feature size. After all features are extracted, feature selection is first performed to greatly reduce the number of features (dotted box). Decoder weights are then estimated for the selected features.

3.2 Methods

3.2.1 Closed-loop brain-computer interface simulations

We used computer simulations to systematically evaluate different feature selection approaches (Figure 2). Simulations allowed us to define the properties of neural encoders and features, and also provided a ground truth to evaluate algorithm performance in selecting relevant features.

Simulated center-out movement task and performance metrics

We simulated a center-out reaching task commonly used in BCI control studies (e.g., [62]). At the beginning of each trial, a central target (2 cm radius) appeared. Trials were initiated by moving a cursor (0.5 cm radius) into the central target. After the central target was acquired and held for 200 ms, one of eight peripheral targets appeared. Peripheral targets (2 cm radius) were arranged in a circle around the center target, equally spaced at 45-degree intervals 10 cm away. A target was successfully acquired if the cursor reached the target position within 10 seconds of target appearance and stayed within the target for 200 ms. After each successful or unsuccessful trial, there was an inter-trial period of 500 ms, after which the center target re-appeared to begin the next trial. The cursor was not reset in between trials but was bounded within a 20 cm square. Each simulated BCI session lasted 20 minutes. We repeated each simulation with 10 different random seeds to test robustness

against stochastic elements such as target presentation order (Target position in T_x , T_y in Figure 2A), and simulated encoder noise (equation 3.1).

We quantified task performance using common BCI metrics of reward rates and cursor error. Reward rate was calculated as the number of successful trials per minute, and captures the overall control efficiency, as a higher rate requires both high accuracy and speed [60, 69]. Cursor path error was calculated to quantify the directness of movement trajectories ([41, 36]). For each successful trial, we defined the ideal path as a straight line from the center of the workspace to the center of the presented target and computed the error as the average perpendicular distance from the cursor's actual trajectory to this ideal path.

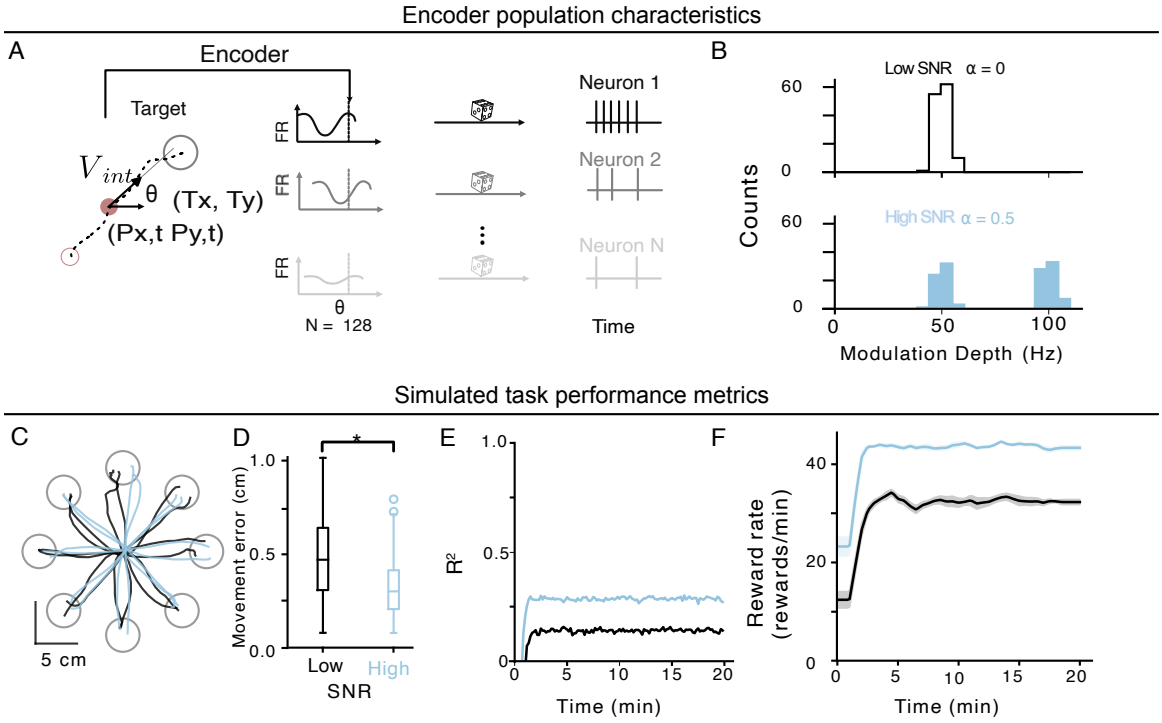

Figure 3.2 Simulated Encoder. A: Schematic of the simulated encoder, which maps intended velocity into neuron firing rates. Intended velocity angle (θ) was calculated using the difference between the current cursor position (P_x, P_y) and target position (T_x, T_y) (left), which then determined neuron firing rates using a cosine tuning model (middle). Spike trains were generated using a noisy Poisson process (right). **B:** Histograms of cosine

tuning modulation depths for high (blue) and low (black) signal-to-noise (SNR) encoder populations. **C:** Comparison of simulated trajectories with low (black) and high (light blue) SNR neuron populations. **D:** Boxplot of movement error for the last 10 trajectories for each reach direction (Wilcoxon test, * $p < 0.05$, $N = 10$). The central mark indicates the median, and the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The whiskers extend to the most extreme data points not considered outliers. **E:** Comparison of coefficient of determination between neuron firing rates and intended velocity for low (black) and high SNR (blue) neuron populations. **F:** Comparison of task reward rates for low (black) and high (light blue) SNR neuron populations. For E and F, shaded area represents standard errors around means ($N = 10$).

Neural encoder model

Our computer simulation used a neural encoder model to translate movement intentions into the activity of a group of neurons (Figure 2A). The input into the encoder was the current state of the cursor and the current movement target, which was used to generate spike times for each neuron as outputs.

We modeled the simulated user as consistently aiming toward the current movement target. We used a Linear Quadratic Regulator (LQR) framework to estimate the user's intended movement goal based on the current cursor state and target state [78, 88]. The LQR model specifies a cost function that considers both how far the cursor deviates from the target (the state error) and the "effort " required to achieve the target (control effort). Control effort is linearly proportional to the current state, which is minimized for physically realistic movement control. We represented the cursor's current state as a vector $x_t = [P_{hor,t}, P_{ver,t}, V_{hor,t}, V_{ver,t}, 1]$, where P_{hor} and P_{ver} are its horizontal and vertical position coordinates, V_{hor} and V_{ver} are its two-dimensional velocity components, and the constant term 1 serves as a bias. The LQR controller calculates the cursor's next desired state (x_{t+1}) by minimizing the overall cost function (see section 3.6.1).

We generated the neural activity using the velocity components of the state vector ($N_{\text{states}} = 5$), as primary motor cortex activity is most strongly correlated with movement

velocity rather than absolute position [30]. The neural activity was generated using a linear encoding model [13], where the mean firing rates λ_t were related to the 3-D velocity subspace $v_{t+1} = [V_{hor,t+1}, V_{ver,t+1}, 1]^T$ of the 5-D state vector x_{t+1}.

$$\lambda_t = Ev_{t+1} + w \tag{3.1}$$

where $\lambda_t \in \mathbb{R}^{N_{\text{neurons}}}$ are mean firing rates, $E \in \mathbb{R}^{N_{\text{neurons}} \times 3}$ is the encoder matrix, v_{t+1} is the velocity-components of the state vector x_{t+1} and w is independent Gaussian noise $w \sim \mathcal{N}(0, \text{diag}(\sigma)^2)$. The encoder matrix E can be further parameterized by the 2-D direction and speed that results from a neuron's firing, often described as the "preferred direction" and "modulation depth", respectively [30]. Given a desired preferred direction (θ_i) and modulation depth (MD_i), the corresponding row in E for neuron i corresponds to $[MD_i \sin(\theta_i), MD_i \cos(\theta_i), b_i]$. Neuronal firing rates, λ_t, were finally used to generate discrete spike trains for each neuron i with a Poisson probability mass density function:

$$P_{t,i}(X = k) = \frac{e^{-\lambda_{t,i}} \lambda_{t,i}^k}{k!}$$

where k represents the spike counts.

We simulated 2-D cursor control using populations of 128 neurons. We restricted the state space to 3 dimensions ($N_{\text{states}} = 3$) by using only the velocity components and the baseline firing rate $(V_{hor}, V_{ver}, 1)$ of the state vector x_t . We created neural encoders with varying signal-to-noise ratio (SNR) distributions. We held the firing rate noise (σ^2) fixed to 9 Hz and similarly fixed the distribution of preferred direction angles (θ_i) fixed to a uniform distribution from [0 360] degrees. SNR was manipulated, then, by changing the modulation depths (MD), which were sampled from bi-modal Gaussian distributions: one sub-population of neurons with large MD (center frequency 100 Hz, standard deviation 3 Hz) and a second sub-population with comparatively low MD (center frequency 50 Hz, standard deviation 3 Hz). We tuned the relative proportions of the low and high MD subpopulations with a parameter α (Figure 2B):

$$N_{\text{high MD}} = \alpha N_{\text{neurons}}$$

$$N_{\text{low MD}} = (1 - \alpha) N_{\text{neurons}}$$

where increasing α included more high SNR neurons in the population.

We used an adaptive Kalman Filter decoder (see section 3.2) to compute cursor trajectories resulting from simulated neural activity. We then confirmed that our SNR manipulations in the encoder influenced decoding performance as expected by comparing task performance when using encoders with high and low SNR populations (Figure 2D–F). Trajectories from the high modulation depth population were significantly straighter (Figure 2D, Wilcoxon test, $p < 0.05$, $N = 10$). We also quantified the task relevance of the neural populations by computing the coefficient of determination (R^2) between the intended velocity and firing rates, which confirmed that the high SNR population carried more information about the intended directions (Figure 2E). The high SNR population also achieved higher reward rates (Figure 2F).

We used our ability to manipulate encoder properties to systematically evaluate different feature selection algorithms in different encoder scenarios. We first tested a stationary encoder, where the encoder matrix (E) remained the same throughout the simulation. We then evaluated how well algorithms tracked encoder changes using simulations with a non-stationary encoder. BCI experiments involve encoder changes on many timescales with different underlying causes (e.g., learning that may occur slowly, versus hardware-related changes in neural measurement that can occur abruptly). As an extreme challenge for our algorithms, we focused on testing an edge case of a discrete, sudden, large encoder change. We started with one encoder (containing a mix of high and low modulation depth neurons as described above). After 5 minutes, we then changed the encoder by randomly shuffling the rows of the encoder matrix. This manipulation allowed us to control for the overall distribution of modulation depths, and therefore encoding information within the ensemble, while disrupting the relationship between the decoder and the encoder.

Neural Decoding

We decoded Poisson neuronal spike trains using a linear Kalman Filter (KF) trained on neural firing rates, a widely used approach for intracortical BCIs [99]. Discrete spike trains P_t were translated into estimated firing rates ($y_t \in \mathbb{R}^{N_{\text{feats}} \times 1}$), by counting spike events

within non-overlapping time intervals (100 ms). These neural features were input to the KF to predict the cursor state, x_t. We adopted a position-velocity KF model [60, 80], which models the position and velocity of the cursor (in 2 dimensions): $x_t = [p_x, p_y, v_x, v_y, 1]^T \in \mathbb{R}^{N_{\text{state}} \times 1}$, where p_x and p_y are the horizontal and vertical components of the cursor position, respectively, and v_x and v_y are the horizontal and vertical components of the cursor velocity, respectively.

The KF makes predictions by combining inferences from two linear models. One model describes how states evolve over time (state transitions):

$$x_t = Ax_{t-1} + w, \quad w \sim \mathcal{N}(0, W)$$

where $A \in \mathbb{R}^{N_{\text{state}} \times N_{\text{state}}}$ and $w \in \mathbb{R}^{N_{\text{state}}}$. The matrix A was designed so that the cursor velocity was integrated into the cursor position, and the velocity decayed by 0.8 (without neural input) at each timestep, as in prior work [62, 20].

The second equation of the KF model describes how observed neural activity relates to the states:

$$y_t = Cx_t + q, \quad q \sim \mathcal{N}(0, Q)$$

where $C \in \mathbb{R}^{N_{\text{feats}} \times N_{\text{state}}}$ defines the relationship between the cursor state and neural features, and $Q \in \mathbb{R}^{N_{\text{feats}} \times N_{\text{feats}}}$ is the neural feature covariance matrix. At each time point, both the neural feature measurements ($y_t \in \mathbb{R}^{N_{\text{feats}} \times 1}$) and the prior cursor state were used to update the current cursor state using standard KF methods[99].

The KF's state transition matrices (A and W), representing cursor dynamics, were fixed [60]. The observation model matrices (C and Q) were initialized randomly and re-estimated online using the data from each training batch. Each batch consisted of 10 seconds of the intended velocity and concurrent neural measurements, sampled at 10 Hz for a total of 100 data points ($N_L = 100$). We chose this batch size to evaluate performance in a rapid adaptation regime with limited data availability [12, 19, 78].

For each data batch, we updated the observation model parameters (C and Q) by first regressing the neural spike rates ($Y \in \mathbb{R}^{N_{\text{feats}} \times N_L}$) against the intended velocities using

maximum likelihood estimation:

$$\widetilde{C} = YX^T(XX^T)^{-1}$$
$$\widetilde{Q} = \frac{1}{N_L}(Y - \widetilde{C}X)(Y - \widetilde{C}X)^T$$

After we obtained batch-estimated parameters, we updated the KF observation model using an exponential weighting of the new estimates of $\widetilde{C}$ and $\widetilde{Q}$ and the previous estimates (C_{i-1} and Q_{i-1}), following SmoothBatch closed-loop decoder adaptation [60, 19].

$$C_i = (1 - \rho)C_{i-1} + \rho\widetilde{C}$$
$$Q_i = (1 - \rho)Q_{i-1} + \rho\widetilde{Q}$$

Where ρ controls the fraction of previous batch-estimated KF observation parameters carried over to the new decoder parameters. It can be considered the half-life of the old decoder parameters [20].

3.2.2 Online feature selection methods

Adding online feature selection to existing BCI systems requires considering how feature selection coordinates with decoder weight estimation because it introduces dependencies between the two computations (Figure 1). We illustrate the details of these dependencies by showing the relationships between the encoder, selected features, decoder weight estimation, and cursor prediction over time (Figure 3A) for four different adaptation algorithm schemes (Figure 3B–E). The encoder in this toy example consists of three high SNR neurons and one low SNR neuron (circles). We illustrate online update computations for feature selection (F, black squares) and/or decoder weight estimation (D_e, dark pink squares) using algorithms that follow temporal rules consistent with those used in our study, like SmoothBatch [60, 62, 20, 80]. F and/or D_e computations are based on batches of data (10 s duration) and occur after each batch. This produces a time-series of features and decoder weight matrices for each batch B_n, F_n and $D_{e,n}$. The feature weight matrix and decoder weight matrix at batch n (F_n and $D_{e,n}$, respectively) are estimated based on the data (intended velocity and firing rates) in the prior data batch, B_{n-1}. The final decoder weight estimates (e.g., calculated

using 3.2) are used to generate the cursor prediction, D_n (light pink squares). Within each batch, the neural decoder predicts a new velocity every 100ms (10 Hz). The cursor position is updated every 16ms (60 Hz) by integrating the most recent velocity prediction from D_n

In the absence of online feature selection, all features are selected, and all decoder weights are estimated at each batch update (Figure 3B). Without feature selection, the size of the decoder, as well as the relationship between estimated decoder weights and features, does not change over time. This scenario removes any temporal history dependence and any dependence between the feature selection and decoder weight estimation calculations. Algorithms that perform feature selection after decoder weight estimation have similarly simple relationships between features and decoder parameters (Figure 3C). To distinguish this from other approaches, discussed below, we term this approach "Full decoder estimation with Feature Subsets" (Full D for short). While this approach simplifies feature–decoder relationships, this approach may ultimately limit the ability to leverage computational efficiency opportunities provided by feature selection as feature sizes increase.

Introducing feature selection before decoder weight estimation may ultimately provide opportunities for improved computational efficiency, but complicates the relationship between features and decoder parameter estimates. We call this approach "subset decoder estimation" (Sub D), illustrated with two example algorithms where two of the three high SNR features in the encoder are selected (Figure 3D, E). Both approaches first select features and then estimate decoder weights for this subset of features to compute $D_{e,n}$. At the first batch update B_n, the feature selection algorithm may pick the first two high SNR features (F_1 and F_2), which are then used for decoder estimation and cursor prediction. In the next batch, the first algorithm (Figure 3D) optimizes only for selection of high SNR features and picks a different set of two features (e.g. F_1 and F_3). In this scenario, the newly selected feature (F_3) was not updated in the previous batch of data. The associated decoder weight is therefore "out of date,".

We hypothesized that "out of date" decoder weights would compromise cursor predictions, and therefore developed an algorithm for online feature selection that explicitly considers the temporal history of selected features over time (Figure 3E). This algorithm incorporates an explicit objective to select consistent features across batches alongside the

objective to maximize the predictive power of selected features that is used for offline feature selection. This multi-objective cost function provides the potential to balance trade-offs between decoder estimation accuracy, which benefits from stationary feature selection, with the need to dynamically select features when encoders change. For example, the algorithm could be tuned to minimize the likelihood of selecting features with "out of date" decoder weights unless the feature priority changes significantly. We termed this objective the "temporal continuity constraint" (TCC).

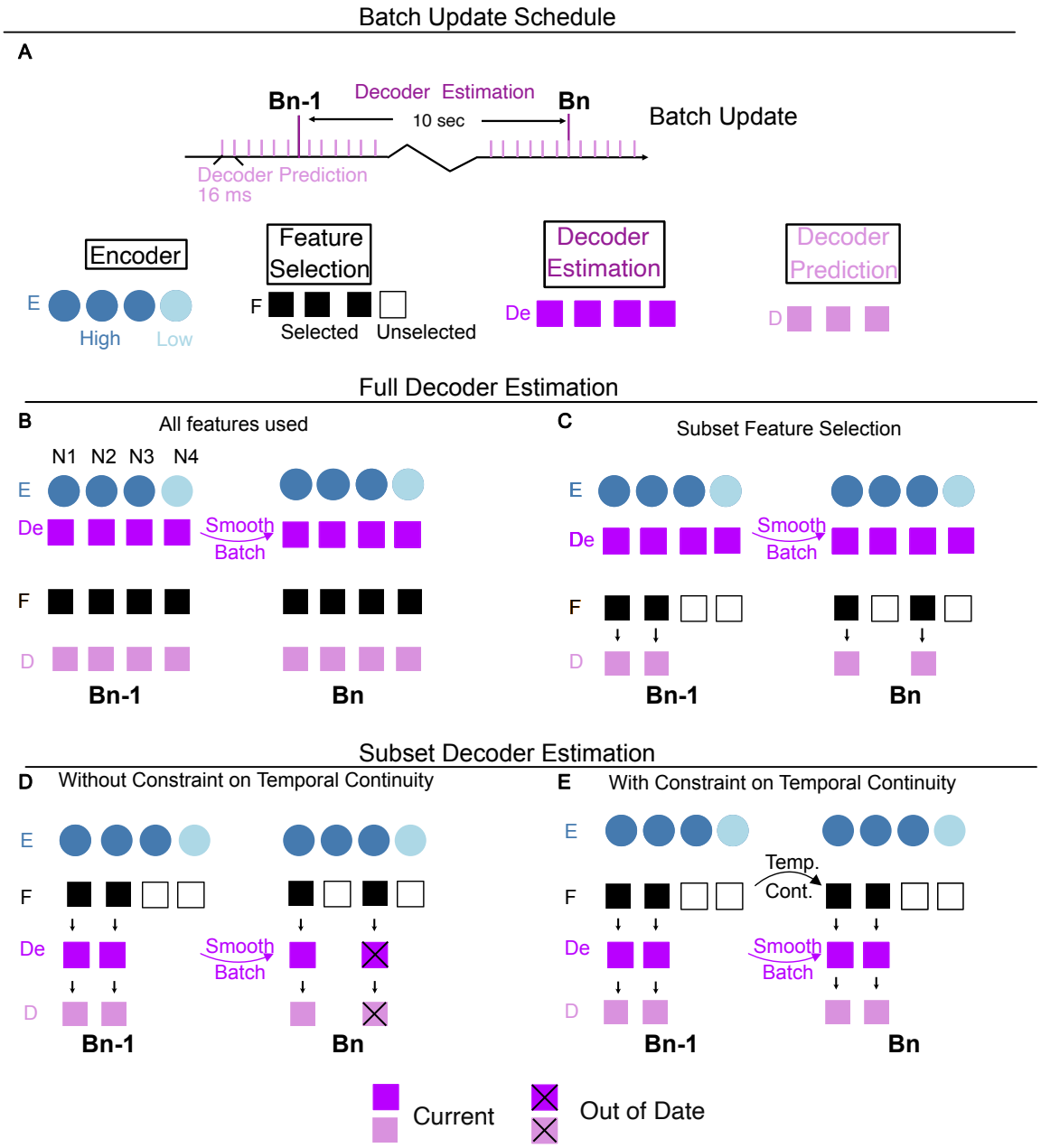

Figure 3.3 Coordination between feature selection and decoder estimation. Top row A: Schematic of batch update schedules and illustration of feature selection and decoder computations. B_{n-1} and B_n are batch updates at steps $n-1$ and n, respectively. The encoder (E) has 3 high and 1 low SNR features. At each batch update, decoder weights

are estimated at a rate of 0.1 Hz (every 10 seconds) (De, decoder estimation). The decoder predicts movement output at a rate of 60 Hz (every 16 ms) (D, decoder prediction). In this example, three neural features were selected for decoding. **B and C:** Schematics of feature selection and decoder computations for algorithms that perform full decoder estimation. **B:** Schematic of no feature selection. Parameters are estimated for all features (4 features in the encoder E, blue circles; 4 estimated decoder parameters De, purple squares), all features are "selected" (F, black squares), and used for cursor prediction (D, pink squares) at each time-step, and decoder parameters update between batches (B_{n-1} and B_n). **C:** Schematic for full decoder weight tracking, followed by feature selection. All decoder weights are estimated at every time-step but only a subset of features are used to predict cursor prediction. Small downward arrows indicate positional caching of previously used features ("decoder caching"). **D and E:** Schematics of feature selection and decoder computations for algorithms that perform subset decoder estimation. Features are selected first before decoder weights are estimated and before decoder prediction. **D:** Schematic of subset decoder estimation without any constraint on feature temporal continuity. **E:** Schematic of subset decoder estimation with a feature temporal continuity, represented by an arrow between the selected features on two batch update steps.

Applying existing offline feature selection algorithms online

We explored using existing feature selection algorithms commonly used in offline applications for online adaptive feature selection. We used LASSO as an example, and implemented it for both Full D (LASSO Full D, Figure 3C) and Sub D approaches (LASSO Sub D, Figure 3D). The LASSO feature selection algorithm solves for a feature weight matrix $W^i_{\text{LASSO}} \in \mathbb{R}^{N_{\text{feats}} \times N_{\text{states}}}$ which trades off the objectives of selecting relevant vs. sparse features. The feature relevance objective is to minimize the mean-squared error between observed firing rates ($Y \in \mathbb{R}^{N_L \times N_{\text{feats}}}$, where $N_L \in \mathbb{R}$ is the number of samples in a batch) and model-estimated firing rates (the product of intended velocity, $X^i \in \mathbb{R}^{N_L \times N_{\text{states}}}$, and the transposed feature weight matrix, W) at each batch. The sparse features objective is to minimize the size of the feature weight matrix calculated with an L1 norm. The weighting

of these two objectives is controlled by a hyperparameter ($\alpha_l \in \mathbb{R}$):

$$W^i_{\text{LASSO}} = \underset{W}{\text{argmin}} \left\{ \frac{1}{2N_L} \left\| Y^i - X^i W^T \right\|_2^2 + \alpha_l \left\| W \right\|_1 \right\}$$

After computing the optimal W for this cost function, discrete feature selection is typically implemented by applying a threshold, Θ_l, to the row norms of the feature weight matrix, $\left\| W_{(\text{LASSO},i,:)} \right\|_2$, producing a binary feature vector (θ_l). A larger LASSO regularization parameter (α_l) increases feature sparsity. While this can improve generalization across online data batches by selecting only robust features, it may also reduce the decoder's overall predictive power.

Determining the optimal value for the sparsity parameter (α_l) is a key challenge in LASSO [40]. In offline feature selection, data is abundant and $N_L \gg N_{\text{feats}}$, and α_1 is typically determined by standard cross-validation procedures. In the online case, $N_L \ll N_{\text{feats}}$ and the optimal α_1 parameter is not generally unknown *a prior*. This uncertainty complicates cross-algorithm comparisons. We therefore set α_l based on the number of selected features for a fixed threshold value Θ_l. Specifically, we set $\alpha_l = 10$ and $\Theta_l = 1$, which corresponded to roughly 50% of all features being selected on each batch. We ranked the features based on their decoder weight magnitude, then selected the top $N_{\text{select}} = 64$ features.

We implemented LASSO Full D by using all features to estimate decoder weights at each batch update. In the feature selection step, relevant subsets of features were selected using LASSO. Once features were selected, the relevant weight estimates were extracted from the fully-estimated decoder matrix to generate cursor predictions.

We implemented LASSO Sub D by using LASSO to select a subset of features from the population at the beginning of each batch. Instead of estimating all decoder weights, we simply reused previously estimated (cached) weights on newly selected features in each batch. If there was no previous weight, new feature weights were estimated and used to generate cursor predictions. For a summary of the LASSO Sub feature selection algorithm, see section 3.6.2.

Online feature selection by regularizing feature set temporal history (TCC algorithm)

The LASSO subset decoder estimation (LASSO Sub D) algorithm only optimizes objectives related to feature relevance and sparsity without considering the temporal evolution of selected features. This could produce variation in selected features over batches due to estimation imprecision, particularly in regimes where data batches are small to facilitate rapid adaptation. If feature selection and decoder estimation are performed uncoupled, variability in which features are selected across batches will reduce the currency of decoder weight estimation, which we hypothesized could impact overall BCI performance. We therefore propose incorporating additional objectives that allow a feature selection algorithm to consider the features selected across batches — the feature set temporal history. We developed a novel algorithm that uses a dual-objective cost function to flexibly optimize both task-relevance of features and temporal continuity across selected feature sets, which we call the Temporal Continuity Constraint algorithm (TCC). This flexible framework provides parametric control of factors that are relevant for online feature selection, which may prove useful in ultimately optimizing online BCIs with adaptive neural features.

One way to add a temporal constraint would be to modify the cost function used to compute the LASSO weight matrix (denoted as W_{LASSO} in Equation 1) to consider an additional term. This approach, however, somewhat undermines the goal of selecting discrete features, as it still requires down-stream thresholding (Θ_l). We therefore chose to directly optimize the binary feature selection vector $\theta \in \mathbb{R}^{N_{\text{feats}} \times 1}$, $0 \leq \theta_{i,m} \leq 1$.

We constructed a multi-term cost function to optimize the feature selection vector. The cost function term optimizes for task relevance by maximizing the negative log-determinant of the weighted Fisher Information Matrix: $-\log\det(C_i^T \tilde{Q}_i^{-1} \Theta C_i)$. The weighted Fisher Information Matrix quantifies the task information captured by neural features (selected via $\Theta = \text{diag}(\theta_i)$) while accounting for the measurement model (C_i) and noise precision ($\tilde{Q}_i^{-1}$). (Note that minimizing the negative log-determinant ultimately aims to maximize task information.) We impose our new feature temporal continuity constraint with a second term that captures the similarity between successive feature selection vectors, θ_i and θ_{i+1}, formulated as: $(\theta_i^T \cdot \theta_{i-1})$. The total cost function weights the temporal constraint, relative

to the task relevance constraint, by a scalar μ. We further apply a cardinality constraint, $B \in \mathbb{R}$, on the number of features to enable discrete selection of a desired number of features. Together, the TCC algorithm cost function is given by:

$$\begin{aligned} \min_{\theta_i} \quad & -\log\det(C_i^T \tilde{Q}_i^{-1} \Theta C_i) - \mu \theta_i^T \theta_{i-1} \\ \text{s.t.} \quad & 0 \leq \theta_{i,m} \leq 1 \ \textbf{AND} \ \theta_i^T \mathbf{1} \leq B \end{aligned} \tag{3.2}$$

This algorithmic formulation provides both interpretability and computational tractability. The (μ) parameter can be shown to define the length of time a batch update remains influential for feature updates (the half-life), akin to similar temporal considerations in adaptive decoder parameter update algorithms [20]. The overall objective function is also convex in θ, rendering the optimization problem computationally tractable and well-suited for real-time applications. For a full derivation, see section 3.6.3.

The objective function allows flexibility to incorporate additional temporal constraints, such as consistency across multiple successive batches. We implemented this by developing an extended formulation that compares the current feature vector, θ_i with k prior feature vectors, $\theta_{i-1}, ..., \theta_{i-k}$, where k defines the temporal window being considered. We incorporated exponential weighting, ρ, to discount scores as temporal recency decreases. This produces an extended cost function formulation:

$$\begin{aligned} \min_{\theta_i} \quad & -\log\det(C_i^T \tilde{Q}^{-1} \Theta_i C_i) - \mu \theta_i^T \begin{bmatrix} \theta_{i-1} & \cdots & \theta_{i-k} \end{bmatrix} \begin{bmatrix} \rho \\ \vdots \\ \rho^k \end{bmatrix} \\ \text{s.t.} \quad & 0 \leq \theta_{i,m} \leq 1 \ \textbf{AND} \ \theta_i^T \mathbf{1} \leq B \end{aligned} \tag{3.3}$$

Similar to LASSO, the TCC algorithm has a hyper-parameter – the temporal continuity constraint, μ – whose optimal value might be unknown *a priori*. Based on adaptive decoder parameter estimation work, adjusting this parameter may also be needed depending on specific experimental goals and application use-cases. We leveraged our simulations to systematically explore how TCC parameters impact algorithm performance (see Results). To focus our exploration, we restricted our testing to settings where the number of features remains constant over time and where features are significantly pruned ($N_{\text{select}} \ll N_{\text{total}}$).

Decoder weight caching

A significant bookkeeping challenge arises when the set of selected features changes across batches, since features must ultimately be matched with downstream decoder parameters (see Figure 3). Across all algorithms we test, we addressed this by maintaining a persistent mapping between each neural feature and its decoder parameters over time. We refer to this process as "decoder weight caching". This procedure ultimately requires storing feature identities and decoder weights in consistent columns, as depicted in Figure 3. When the feature set is updated, weights for re-selected (intersecting) features are reused, weights for newly selected features are initialized, and weights for deselected features are preserved for potential future use.

Decoder caching notably simplifies the feature selection and decoder parameter estimation problem in adaptive decoding by preventing mis-matches between features and parameters. Empirically, we found that this caching was critical for applications like those explored here, where feature selection and decoder estimation algorithms were initialized *de novo* (i.e., with no initial training data). Consequently, we include caching in all our approaches to provide a realistic baseline and fairly highlight the strengths and weaknesses of each algorithm. We recognize, however, that caching has important implications for overall system design and computational efficiency (see Discussion, section 3.4).

3.2.3 Metrics to quantify online feature selection performance

Our primary objective was to develop online feature selection algorithms that enhance BCI performance when initializing a system *de novo*. We therefore compared our algorithms using a variety of metrics to quantify task performance and decoder parameter estimation accuracy. Motivated by our hypothesis that temporal continuity of selected features influences performance, we also compared algorithms using metrics to capture feature selection similarity over time.

Feature selection smoothness (Jaccard Similarity)

We quantified the similarity of selected features across batches using the Jaccard similarity, a discrete analogue to the correlation coefficient for continuous variables. For two adjacent feature selection sets S_{i-1} and S_i, the Jaccard similarity coefficient is calculated as the size of the intersection (i.e., common features) divided by the size of the union (i.e., total features across the two batches):

$$J_i(S_i, S_{i-1}) = \frac{|S_i \cap S_{i-1}|}{|S_i \cup S_{i-1}|}$$

and using the feature selection vector from above,

$$J(\theta_i, \theta_{i-1}) = \frac{\theta_i^T \theta_{i-1}}{||\theta_i||_1 + ||\theta_{i-1}||_1 - \theta_i^T \theta_{i-1}} \tag{3.4}$$

Encoder-decoder error

We quantified decoder parameter estimation error by computing the difference between the (known) neural encoder parameters and the corresponding estimated decoder parameters. We specifically calculated the angular error via the cosine similarity between corresponding rows of two matrices (i.e. the decoder matrix and the encoder matrix E in equation 3.1 and C in equation 3.2).

3.2.4 Offline validation using electrocorticography (ECoG) features

We aimed to compare the algorithms' performance using real neural data. We used neural recordings from an array of micro-electrocorticography (μECoG) electrodes implanted in a non-human primate to validate our simulation results. We trained a non-human primate on a delayed center-out-reaching task as we recorded neural data [63]. As the animal performed the task, the position of its right hand was tracked using a marker-based camera system (NaturalPoint, Inc., Corvallis) and used to control a cursor during the task. The monkey's horizontal and vertical hand positions were mapped to control the horizontal and vertical cursor position on the screen, respectively. The monkey was rewarded when the cursor entered and successfully held the at the peripheral target for 200 ms. We aimed to select

neural features offline that encoded the monkey's intentions to drive the cursor using his arm movements.

We recorded field potential signals from a μECoG array with 244 electrodes implanted sub-durally covering portions of three pre-frontal motor cortical regions (primary motor cortex, M1; dorsal premotor cortex, PMd, and dorsolateral prefrontal cortex, DLPFC). We extracted neural features from five frequency bands (δ: 0-4 Hz, θ: 4-10 Hz, β: 12-30 Hz, γ: 30-80 Hz, γ_h: 80-150 Hz) at a rate of 10 Hz. We extracted features from 236 out of 244 total electrodes based on signal quality, resulting in 1,180 features. Unlike the neuron firing rates in the simulation, the μECoG features were normalized individually for each feature before use in feature selection and decoding algorithms, to account for amplitude differences across frequency bands. Raw neural and behavioral data was temporally aligned with cursor positions during data acquisition and cursor positions were downsampled (to 10 Hz) offline to maintain alignment with neural features. Full details on task training, electrode placement, and recording hardware can be found elsewhere [63].

3.3 Results

3.3.1 Online feature selection improves task performance with non-stationary encoder

We first used our BCI simulator, where the neural encoder properties were known and could be manipulated arbitrarily, to build intuitions for the potential benefits of online feature selection in the presence of non-stationary neural encoders. We created a simulated neural encoding with a sub-set of relevant neural features and included a large, discrete encoding change (see methods, Neural Encoder Model) (B). We then decoded the simulated population with an adaptive Kalman filter equipped with two different feature selection algorithms that reflect a worst case and best case. Specifically, we compared a worst case constant feature selection (Figure 4C) with a best case oracle feature selection algorithm that had perfect knowledge of the most informative neural features (Figure 4D). As expected, the oracle adaptive feature selection algorithm yielded higher task performance (measured by reward rates) in the presence of non-stationary neural encoding (Figure 4E). Improved task performance also corresponded to increased feature relevance after the neural encoding

changed (Figure 4F). Finally, we measured the stability of feature selection across batches using the Jaccard similarity metric (see Methods). The constant feature selection method yielded a Jaccard similarity of 1, as its feature set was fixed. The Oracle method's feature set also remained stable when the neural encoder was stable, with a transient change, producing a corresponding decrease in the Jaccard similarity, that tracked the shift in the neural encoder (Figure 4G). These best-case (Oracle) and worst-case (Constant) algorithms, thus, demonstrated the potential benefits for adaptive feature selection when neural encodings changed and provide an upper bound on achievable performance for adaptive feature selection algorithms.

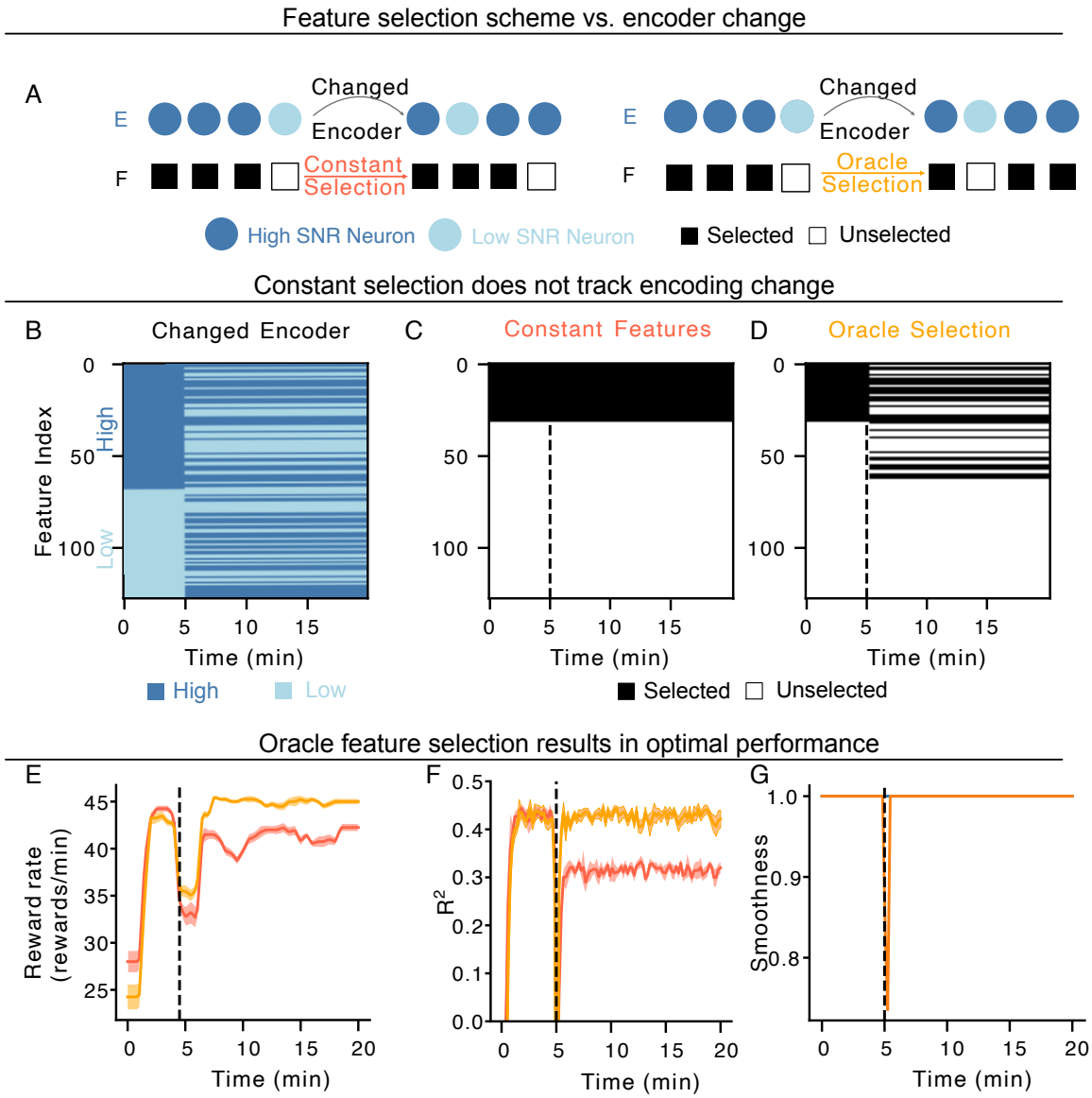

Figure 3.4: Online feature selection with perfect neural encoding knowledge. A: Cartoon of encoder (E) shift with constant feature selection (left) and oracle feature selection (right). The neural encoder changed at 5 minutes. With constant feature selection, a fixed feature set is used, whereas the Oracle feature selection chooses the optimal features based on encoder knowledge. **B:** Example heatmap of the changing encoder. Features ranked by their initial SNR for visualization. The labels 'High' and 'Low' denote the **current** SNR of each

neuron. **C:** Example heatmap of constant feature selection. The first 32 neurons were always selected. **D:** Example heatmap of the Oracle feature selection scheme. **E:** Comparison of reward rate over training time for constant (dark orange) vs. oracle (light orange) feature selection schemes. **F:** Comparison of feature selection relevance for constant (dark orange) vs. oracle (light orange) feature selection schemes. **G:** Comparison of smoothness for constant (dark orange) vs. oracle (light orange) feature selection schemes. For panels E, F, and G, shaded areas represent the standard error around the mean, calculated from simulations using 10 random seeds.

3.3.2 Full decoder weight estimation improved subset feature selection

We next evaluated how existing feature selection algorithms performed in online settings. We used the Lasso feature selection Algorithm (3.6.2) as an example algorithm. We applied Lasso feature selection in small batches to adapt features in real-time. We then performed BCI simulations with either a stationary or non-stationary encoder (Figure 5, left and right columns, respectively). We first applied Lasso feature selection before estimating decoder features (Lasso Sub D). Using this approach resulted in notably worse task performance compared to an adaptive decoder with Oracle feature selection, regardless of encoder dynamics (Figure 5 B and F, respectively). Importantly, we found that, when applied online, the Lasso algorithm selected highly variable features over time regardless of whether the encoder was stationary or dynamic, yielding smoothness values far below 1 (Wilcoxon Signed Rank Test, 6e-18 for Lasso Sub D) (Figure 5C and G, respectively). Variability in the selected features, in principle, should not lead to reduced decoding performance if decoder parameters are estimated accurately for each feature. We hypothesized that the suboptimal performance we observed in Lasso was due to poor decoder estimation, as opposed to a failure of feature selection. Comparing the encoder and decoder parameters, indeed, highlights that Lasso Sub D led to a large estimation error (Figure 5D and H, respectively). Moreover, Lasso Sub D also failed to mitigate neural encoding non-stationarity since decoder weights for previously unselected features were inaccurate (see Figure 5H). These findings highlight that Lasso Sub D failed to extend to online settings due, in part, to coupling between the fea-

ture selection and decoder estimation computations. We therefore explored whether Lasso feature selection algorithms could succeed when the order of computations changed to better de-couple feature selection and decoder estimation (Full decoder estimation approach, Figure 1B). Subset feature selection with full decoder estimation (Lasso Full D) resulted in better performance compared to Lasso Sub D, on par with performance with "Oracle" feature selection (Figure 5 B and F). Even with full decoder estimation, the lasso algorithm selected variable features from batch to batch (Figure 5 C and G). However, changing the order of feature selection and decoder estimation allowed for more accurate decoder estimation (Independent two-sample t-test, p ¡ 0.05, N = 10, Figure 5D and H). Together, these results suggest that the interactions between decoder estimation and feature selection can lead to suboptimal performance when applying established algorithms designed for offline applications in an online setting. We further showed that these shortcomings can be offset through algorithms that serve to de-couple the computations (Lasso Full D), which present potential computational trade-offs.

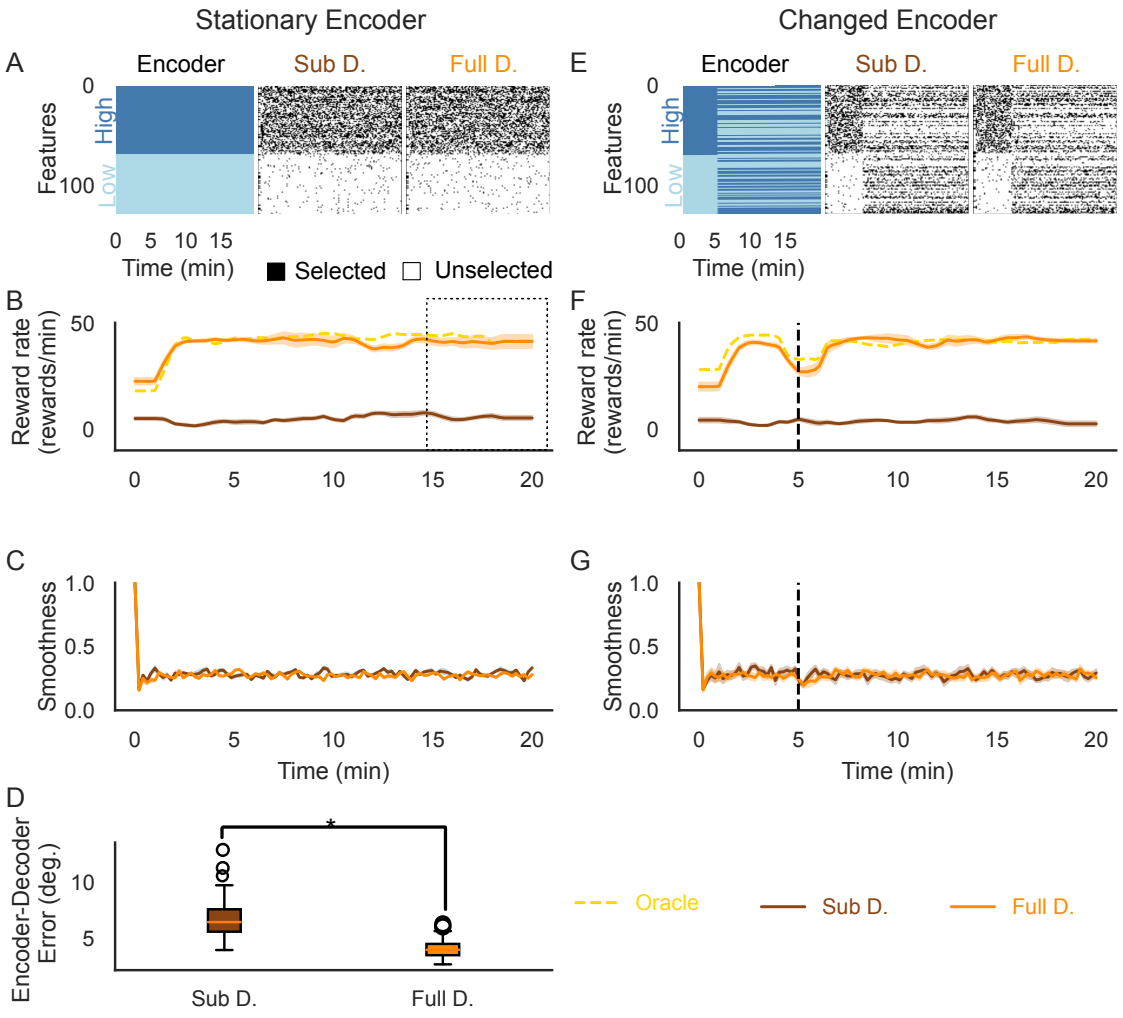

Figure 3.5 Feature selection performance for subset decoder tracking (Sub D) and full decoder tracking (Full D). Panels A–D: Stationary Encoder. **A:** Example simulation with a stationary encoder. The weight distribution of the stationary encoder (left) remains the same across time. The subset decoder (middle, brown) and full decoder (right, orange) algorithms select neurons on every batch. **B:** Comparison of simulated center-out task reward rate for Sub. D. (brown), Full D. (golden), and Oracle (purple) feature selection schemes. Dotted box indicates the data window used for the encoder-decoder error calculation in D. **C:** Comparison of feature selection smoothness for Oracle (yellow), Sub. D (brown), and Full D (orange) feature selection schemes. **D:** Boxplot of

Encoder-Decoder error. The central line indicates the median, the box edges represent the 25th and 75th percentiles, and the whiskers extend to 1.5× the interquartile range. Average of all features and batches after feature selection started in the last 5 minutes (dotted box in panel B, Independent two-sample t-test, $p < 0.05$, $N = 10$). **Panels E–H:** Changed encoder, formatted the same as in A–D. For **F** and **G**, dashed line boxes show the time during and after encoder change. **H:** For panels B, C, F, and G, shaded areas represent standard errors around the mean, calculated from 10 simulations with random seeds. Boxplot of Encoder-Decoder error during (left) and after (right) the encoder change.

3.3.3 Temporally constraining feature selection improves task performance

Full decoder estimation (Lasso Full D) performed better than estimating decoder estimation for subsets of features (Lasso Sub. D), likely due to the inherent coupling between decoder estimation and feature selection in online feature selection. However, estimating all decoder weights may be a less efficient algorithmic approach that scales poorly as the number of possible features grows. We hypothesized that algorithms designed to address the relationship between decoder estimation and feature selection by explicitly considering the temporal evolution of selected features could provide improved performance and alternative computational trade-offs. We formulated a convex feature selection algorithm with an explicit temporal continuity constraint (TCC; equations 3.2 and 3.3) and tested the algorithm on the same stationary and non-stationary encoder scenarios explored with Lasso.

We first applied the TCC subset decoder estimation feature selection (TCC Sub D) algorithm to the stationary encoder (Figure 6A-D). The algorithm's objective function allowed us to parametrically tune the weight (i.e., prioritization) of feature selection temporal continuity, relative to maximizing the information of selected features, via the hyper-parameter μ (Larger values of μ prioritizes selecting smooth features). Increasing μ led to higher reward rates (Figure 6B). As expected, a larger μ also resulted in the algorithm selecting more consistent features from batch to batch (Figure 6C). We further found that increasing μ brought the feature decoder weights significantly closer to their corresponding encoder weights (Page trend test, $p < 0.05$ for TCC $\mu = 0, 0.125$, and 0.25 Figure 6D). This sug-

gests that the temporal constraint on features improved task performance via improving the ability to estimate decoder weights.

While the TCC Sub. D algorithm provides avenues to adapt selected features independently from decoder parameter estimation, temporal constraints on the selected features may introduce trade-offs in the algorithm's ability to optimize selected features. For example, our algorithm initialized the selected features randomly. At sufficiently large values of μ, the algorithm would ultimately maintain those randomly chosen features, likely leading to sub-optimal performance. We indeed see evidence of this trade-off: the reward rate and decoder estimation accuracy showed non-monotonic relationships with μ, increasing until $\mu = 0.25$ and then decreasing for $\mu = 0.25$ (Figure 6B and D).

Trade-offs between selecting optimal versus consistent features may particularly limit the algorithm's ability to track encoder changes, which is a key motivation for adaptive feature selection. We used our changing encoder simulation to evaluate the TCC Sub. D algorithm's ability to track dynamic encoders. We observed a similar pattern overall, where increasing temporal constraints improved reward rates and feature smoothness (Figure 6F and G, respectively). Importantly, this pattern was present both before and after the encoder change. Immediately after the encoder change, performance and smoothness show transient decreases that gradually return to approximately the pre-change values. The decoder estimation accuracy (similarity between encoder and decoder weights), both before and after the encoder change, showed a non-monotonic relationship with μ (Figure 6H), similar to our observations for the static encoder simulations. Together, these results suggest that TCC Sub D can track discrete encoder changes, and that the μ hyper-parameter provides a means to balance trade-offs between potentially competing goals in adaptive feature selection and decoding architectures.

The TCC Sub D algorithm also provides the means to parametrically control the number of selected features (in equation 2). This may be particularly useful because the optimal number of selected features may not be known a priori. We used our algorithm to systematically explore the relationship between the number of selected features and task performance. We created a simulation with a stationary encoder where half of the features had high SNR (64 of 128 total neurons) and then varied the number of selected features and the temporal

continuity constraint (μ) used in TCC Sub D. As expected, task performance improves as more features are used for decoding (Figure 7A). However, the relationship between task performance and selected features varies with the temporal constraint parameter μ. For all but the lowest μ, performance began to plateau at or below half of all features, which corresponded to the proportion of features that are high SNR. As observed before, increasing μ led to smoother selected features (Figure 7B). This analysis also highlights interactions between the temporal constraint and the number of selected features. For instance, temporal constraints have less influence as the algorithms select more features because the possible variability of selected features decreases towards zero as you select more features. Pooling across all simulations, we observed a positive correlation between task reward rate and feature selection smoothness (Pearson correlation test, $R^2 = 0.94$, $N = 32$, Figure 7C). These findings highlight the potential benefits of adaptive feature selection algorithms with parametrically tunable temporal constraints for BCI applications where significant feature reduction may be needed.

The above analyses highlight that the fraction of useful neural features among all possible measured features may also influence the performance of our adaptive algorithms. This fraction would similarly be unknown a priori in BCI applications. We ran an alternate variation of our simulation where we systematically varied the number of high SNR neurons within the population while controlling the total population size, thus manipulating the fraction of useful neural features (Figure 7D). In all cases, the TCC Sub D algorithm was set to select 25% of the total neural features while we varied the temporal constraint parameter μ. As expected, smoothness of selected features increased as μ increased, but was not significantly influenced by neural encoding properties (fraction of high SNR neurons) (Figure 7E). Reward rates across the simulations similarly showed the expected relationship between task performance and μ, but also revealed additional interactions with algorithm parameters (Figure 7F). Potentially counter to intuitions, the presence of more relevant features in the population did not always lead to better task performance. Instead, task performance peaked or began to reach a plateau when the number of high SNR features was the same as the number of features selected by the algorithm (25%). This phenomenon likely reflects the inherent interaction between feature selection consistency and decoder

estimation, because it was most prominently observed at intermediate values of μ (e.g., 0.125). Indeed, this demonstrates a scenario where having more relevant features within a neural population diminished task performance because it increased the variability of selected features over time. Increasing the temporal constraints within the algorithm, however, was able to mitigate the sensitivity of feature selection algorithms to the underlying neural population properties, leading to improved performance as the fraction of high SNR neurons increased.

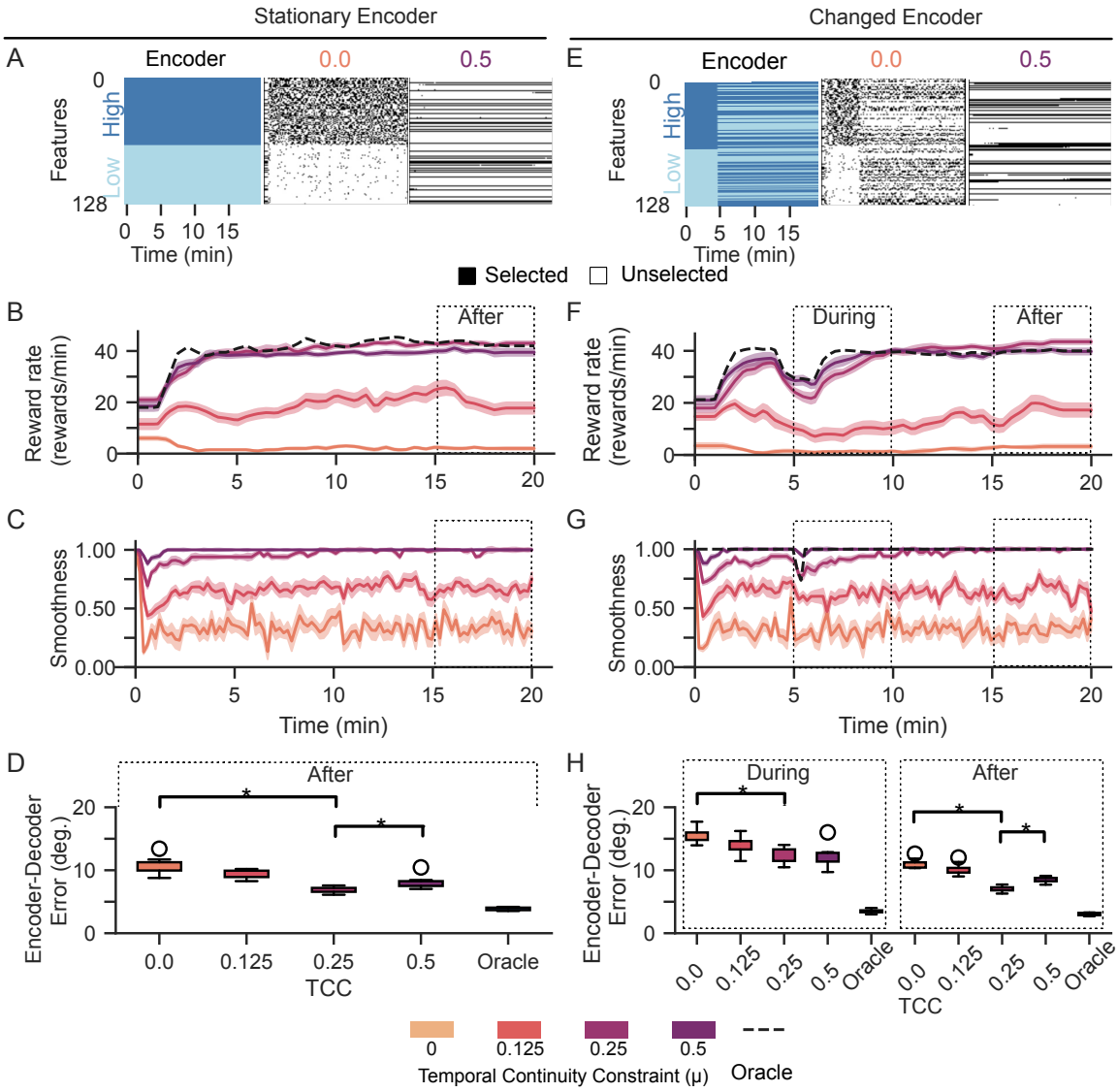

Figure 3.6: Performance of the temporal continuity constraint algorithm (TCC). **Panels A–D:** Stationary encoder. **A:** Example simulation under a stationary encoder. The weight distribution of the stationary encoder (left) remains the same across time. "High" and "Low" represent high and low SNR neurons respectively. Two coefficients of temporal continuity (μ) resulted in different dynamics of feature selection over time (middle and right). **B:** Comparison of task reward rates at four temporal continuity coefficients (μ increases as shading gets darker, see legend). Dotted box indicates the data window used for the encoder-decoder error calculation in D. **C:** Comparison of feature selection smoothness for four temporal continuity coefficients. **D:** Boxplot of Encoder-Decoder error (degrees) after the encoder change for varying Temporal Continuity Constraint (TCC) weights and the Oracle baseline. The central line indicates the median, box edges represent the 25th and 75th percentiles, and whiskers extend to 1.5× the interquartile range. The encoder-decoder errors were averaged over the 10 simulations for each smoothness coefficient in the last 5 minutes of each simulation trial (dotted box in B). Page trend test, $p < 0.05$, $N = 300$. **Panels E–H:** Changed Encoder simulations, formatted the same as in **A**–**D**. For **F** and **G**, the "During" and "After" rectangles represent 5 minute time windows during and after encoder change. For panels **B**, **C**, **F**, and **G**, shaded areas represent standard error from 10 simulations with random seeds. **H:** Boxplot of Encoder-Decoder error (degrees) during (left) and after (right) the encoder change.

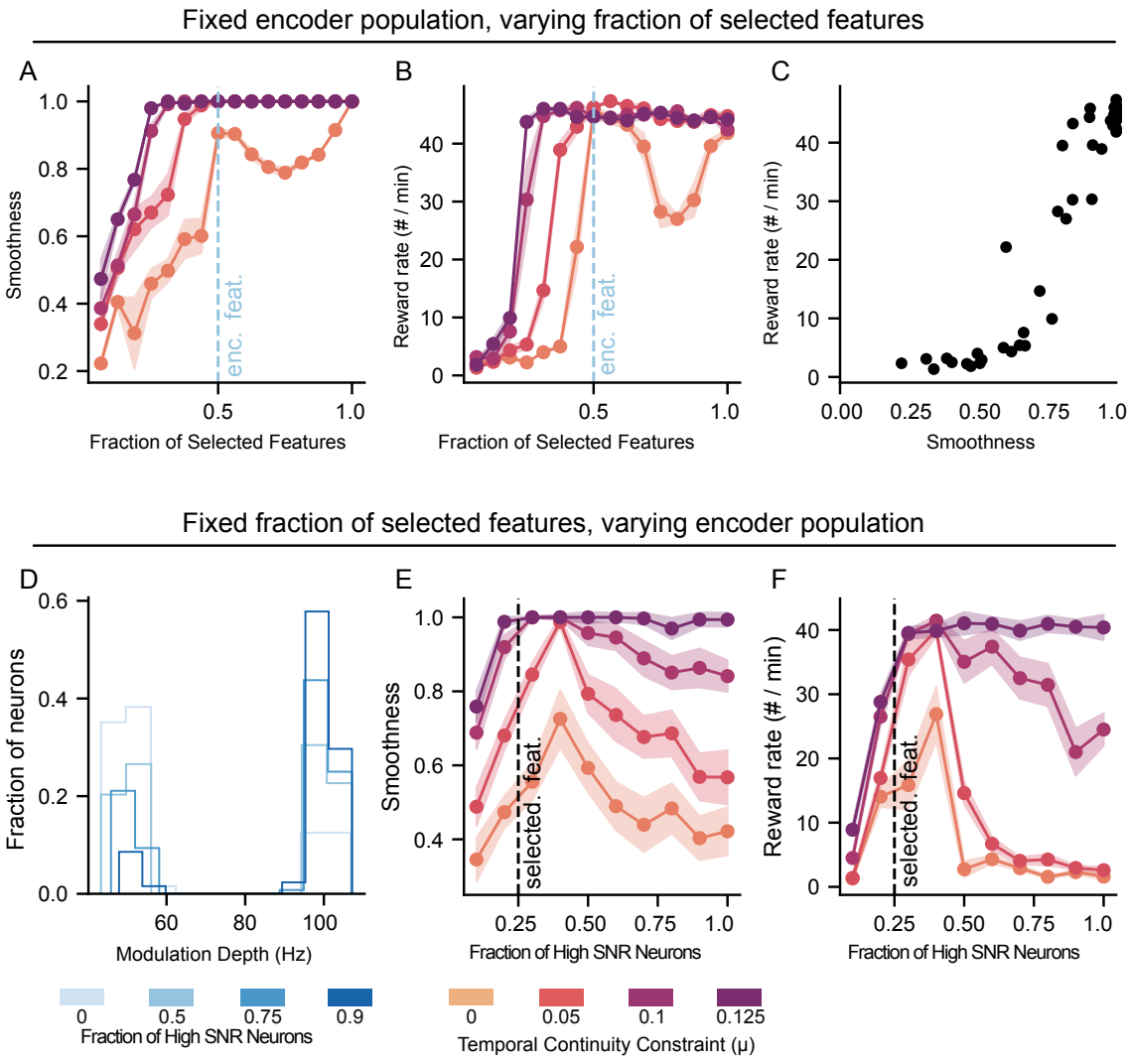

Figure 3.7: Feature selection outcomes as the number of selected features and informative features in the encoder vary. A–C: Comparison of algorithm performance as a function of the number of selected features. **A:** Smoothness of the selected features as a function of the desired fraction of selected features at increasing values of the temporal continuity constraint (μ increases as shading gets darker; see legend). The dotted line indicates the fraction of high SNR encoder neurons. **B:** Reward rate as a function of the desired fraction of features; formatted as in A. **C:** Feature selection smoothness compared with reward rate. Each data point corresponds to a data point in a reward rate in A and a smoothness value in B. Pearson correlation test, $R^2 = 0.96$, $p = 1.4 \times 10^{-19}$. **D–F:**

Comparison of algorithm performance as a function of fraction of informative neurons in the encoder. **D:** Histograms of the encoder population with increasing fractions of high SNR neurons. **E:** Feature selection smoothness as a function of fraction of high SNR neurons. The dotted line indicates the desired fraction of selected features. **F:** Reward rate as a function of the fraction of high SNR neurons in the last six batches. For A, B, E, and F, shaded areas represent standard error from ten simulations with random seeds.

3.3.4 Temporal constraint feature selection applied to micro-electrocorticography data

Simulations with known and manipulable ground truth neural encoders allowed us to develop and evaluate algorithms for online feature selection. We now explore applying these algorithms to neurophysiological data to further evaluate adaptive feature selection approaches for BCIs. Due to the absence of ground truth knowledge of the encoder in these data, we focus on evaluating whether offline analyses with real neural data are consistent with our observations and intuitions gained via our simulation studies.

We collected subdural micro-electrocorticography (μECoG) data from motor cortical and prefrontal areas while a monkey controlled a computer cursor using arm movements (Figure 8A). A well-trained monkey performed the same delayed center out reaching task used in our simulations while we recorded μECoG signals. Field potential signals like ECoG are typically analyzed in the frequency domain and generate a large space of potentially informative features. We used a relatively modest five frequency band feature basis (δ: 0–4 Hz; θ: 4–10 Hz; β: 12–30 Hz; γ: 30–80 Hz; High γ: 80–180 Hz), which results in 1,180 extracted neural features for our array with 236 electrodes (Figure 8B). We used this large number of neural features to explore feature selection algorithms.

We first sought to demonstrate that the feature selection algorithms could identify physiologically relevant features offline without considering adapting selected features over time (i.e., with large batches of data). Previous analysis of these data suggested that information about arm movement direction was distributed across rostral and caudal motor cortices, primarily in the delta and high-gamma frequency bands [63]. Consistent with these findings, example implementations of the Lasso Sub. D and TCC Sub D algorithms selected features

from ECoG electrodes in the caudal and rostral regions of the motor cortex when trained on 5 minutes of data offline and restricted to use 128 features (10.8% of the total extracted features) (Figure 8C). Moreover, the majority of the selected features were within the gamma or high gamma bands (12.5% and 56% of features selected in the gamma and high gamma bands, respectively, using the TCC Sub D algorithm; 30% and 56% of selected features in the gamma and high gamma bands, respectively, using Lasso Sub. D algorithm), which have been shown to carry hand-kinematic information both in our dataset [63] and many other studies [71, 73, 5]. Both algorithms outperformed a random feature selector in predicting arm movements, especially when the desired number of features was much smaller than the complete feature set (Figure 8D). On this single dataset, TCC Sub D was on par with the Lasso Sub D algorithm, consistent with our observations that both approaches successfully identify high-information features and extend this to local field potential data.

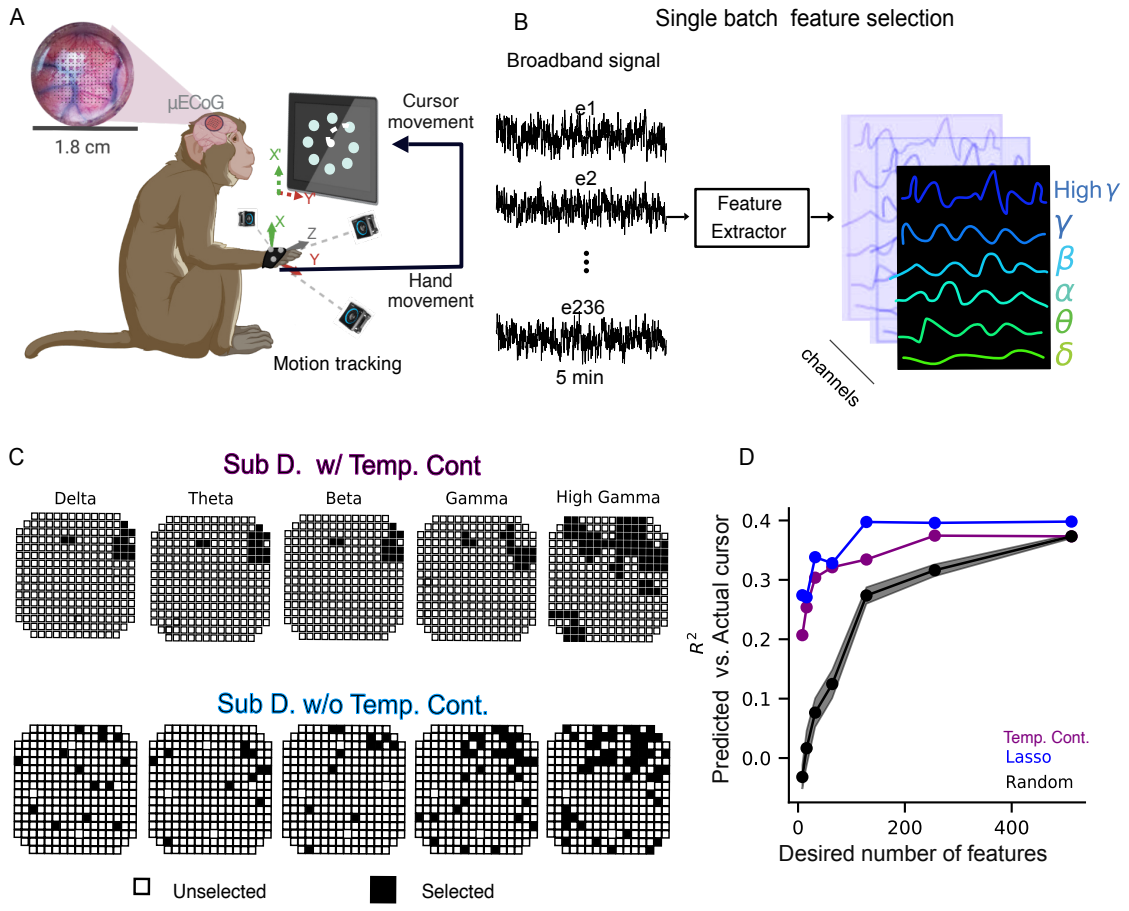

Figure 3.8: Offline feature selection using Electrocorticography (ECoG) data. A: Schematic of the reaching task. The inset shows a schematic overlay of the μECoG array's recording locations on the cortical surface, spanning frontal motor cortices. μECoG signals were recorded as monkeys performed a center-out reaching task. The monkey performed the task by controlling the movement of a cursor on a visual display with hand movements tracked in real-time with motion capture cameras. **B:** Illustration of feature extraction for μECoG features. Broadband signals from each recording electrode (right) were band-pass filtered ("feature extractor" box, middle) to extract five neural features per electrode (right; δ: 0–4 Hz; θ: 4–10 Hz; β: 12–30 Hz; γ: 30–80 Hz; High γ: 80–180 Hz). **C:** Maps of electrode locations where feature selection algorithms identified 128 informative features, shown for each of the five frequency bands (columns) for algorithms with and

without temporal continuity constraints (top row: TCC Sub D; bottom row: Lasso Sub D). **D:** Coefficients of determination (R^2) for predicted versus actual cursor movements as a function of the desired number of features. Random performance plotted as the mean (markers) and standard errors (shaded region) estimated via 10 random feature selections.

We next explored the algorithms' performance in an 'online' feature selection scenario (Figure 9A; see Methods). We selected a batch size of 2 minutes based on past adaptive decoding studies [60, 19]. As in the online simulations, μECoG features and recorded offline cursor movements were used to select features and update the decoder at each batch. The selected features and decoder weights were then used to predict the cursor movements in the subsequent batch (Figure 9A, "Predicted Movements"). Note that this inherently offline analysis does not fully capture potentially relevant user-algorithm interactions in closed-loop BCIs.

We first examined spatial and temporal patterns in the features selected by the algorithms across batches for an example time segment of data. Similar to the non-adaptive scenario, Lasso Sub. D selected electrodes that were mostly in the expected regions of motor cortex, though it also selected electrodes elsewhere intermittently across batches (Figure 9B). The TCC algorithm also tended to select similar features between the adaptive and non-adaptive scenarios, while selecting more consistent features over time (0.99 ± 0.01 and 0.2 ± 0.1 for TCC and Lasso algorithm, respectively, Figure 9B, Sub D. w/ TCC).

We then quantified differences in predictive power for movement kinematics between feature selection algorithms. Across multiple datasets, the TCC algorithm was significantly more predictive for the subsequent batch than Lasso Sub. D. (Wilcoxon signed rank test, $p < 0.05$, $N = 39$; Figure 9C). These results paralleled our simulations with a stationary encoder in which TCC increased smoothness of feature selection and resulted in better task performance (see Figure 6A). However, Lasso Full D yielding better predictions than Lasso Sub D, despite variability in the selected features (Wilcoxon signed rank test, $p > 0.05$, $N = 39$; Figure 9C), similar to our findings in simulation (see Figure 5B and F). Similarly, Lasso Full D performed similarly to TCC Sub. D (Wilcoxon signed rank test, $p > 0.05$, $N = 39$) as expected from simulations.

Finally, we tested how the temporal continuity coefficient μ affected feature selection

in this physiological dataset. We confirmed that the temporal constraint coefficient in the convex smooth objective function (equation 2) parametrically controlled the smoothness of selected features from μECoG data (Figure 9D). Increasing the temporal constraint coefficient led to increases in the mean cursor movement prediction power until $\mu = 0.008$ ($p < 0.05$, Page trend test, $N = 39$). Further increases in μ ($\mu = 0.016$), however, led to a decrease in the predictive power, similar to our simulations using spike data (Figure 6B and F). We also confirmed that increasing the TCC weight coefficients led to higher smooth features at a smaller number of desired features (Page trend test, P = 9e-6, N=39; Figure 9E). Together, these results corroborated the findings in our simulation (Figure 7).

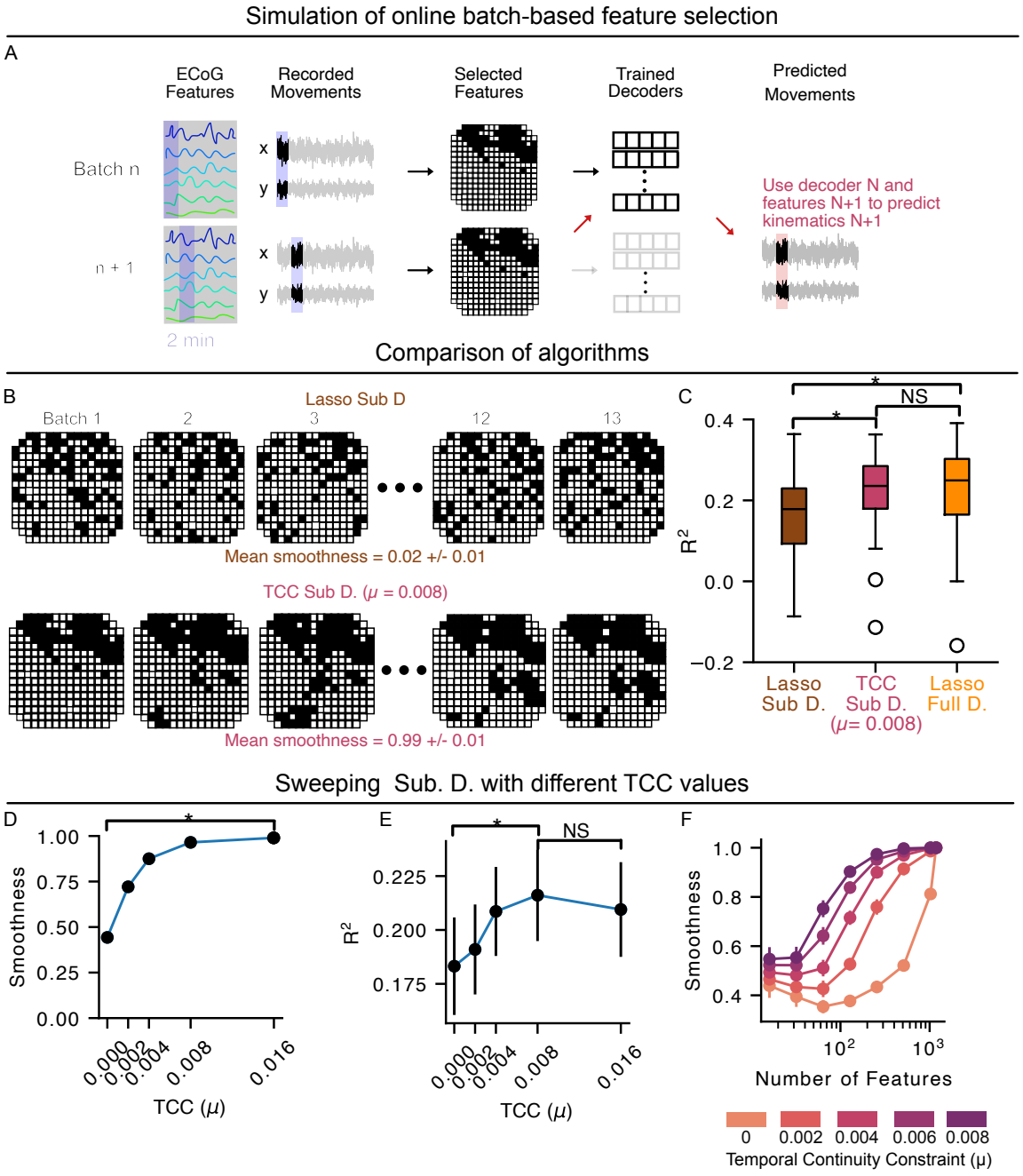

Figure 3.9: Simulation of online batch-based μECoG feature selection. A: Schematic of the online batch-based feature selection. Top row illustrates the calculations performed in batch n; bottom row illustrates the calculations performed in batch $n+1$. Vertical blue rectangles represent the 2-minute batch window for μECoG features

and recorded cursor data (left) that were used for feature selection and decoder parameter estimation calculations (middle). The decoder parameters estimated in batch n were used to predict cursor movement in the batch $n+1$ using selected features from batch $n+1$ (right; red arrows). **B:** Example maps of the selected electrodes for the high-gamma band features calculated with the Lasso Sub. D (top) and TCC Sub. D (bottom) algorithms over 13 batches (columns proceeding left to right). The mean feature selection smoothness is 0.2 ± 0.1 and 0.99 ± 0.01, respectively, for Lasso Sub. D. and TCC Sub. D. **C:** Comparison of coefficient of determination (R^2) between the offline decoded and actual cursor velocity. Data pooled across batches from three recording sessions ($N = 39$). *: Wilcoxon Ranked Test, $P < 0.05$. NS: Not significant $P > 0.05$. **D:** Mean Feature selection smoothness over batches for the TCC algorithm as a function of the temporal continuity constraint μ. Data points indicate mean; error bars denote the standard error of the mean (*: Page trend test, $P < 0.05$, $N = 39$ data batches pooled across 3 recording sessions). **E:** Average cursor velocity prediction accuracy (R^2) as a function of the temporal continuity constraint μ. Format as in panel D. (*: Page trend test, $P = 9 \times 10^{-6}$, $N = 39$). **F:** Smoothness as a function of the algorithm's number of selected features for different values of the temporal continuity constraint μ (μ increases as shading gets darker, see legend). Format as in panel F.

3.4 Discussion

Adaptive neural decoding methods are widely used to improve and maintain BCI performance. Existing adaptive decoding algorithms focus on re-estimating decoder weights over time with small sets of predetermined features [60, 31, 20, 12]. We aimed to expand frameworks for adaptive decoding to enable dynamic and adaptive selection of subsets of informative neural features from a large pool of candidate features. We used simulations to show that naive use of offline feature selection algorithms, such as Lasso, performed poorly in an adaptive setting due to inherent coupling between feature selection and decoder weight estimation algorithms (Figure 3.1). Decoupling these two operations however, requires integrating the two adaptive computations. We present two options for engineering solutions to this integration which provide different potential trade-offs. The Lasso Full D

algorithm prioritizes parameter estimation accuracy by tracking parameters for all features at the expense of additional computations. Alternately, the TCC algorithm enables potentially more efficient parameter estimation for selected subsets of features by prioritizing the temporal continuity of selected features, at the potential expense of compromising feature optimization. Our analyses with simulated spiking data and μECoG data characterized the performance of these algorithms, laying theoretical foundations for adopting adaptive feature selection in real world BCIs.

We are motivated by the overarching goal of designing adaptive and computationally efficient algorithms that can scale with increasing channel counts. Since prior work has not extensively explored feature selection as a process distinct from decoder parameter estimation, our primary focus has been to establish the fundamental computational principles governing their interaction. We acknowledge, however, that we have not yet optimized these algorithms for physical hardware constraints; our current framework assumes data is streamed from all recording channels prior to the selection step. To fully consider the implementation of these algorithms in resource-constrained systems, future work must extend these principles to embedded hardware architectures where feature selection directly gates data transmission, ensuring that only the relevant subsets are streamed to realize true power and bandwidth savings.

One critical observation from our work is that offline feature selection algorithms did not perform well when applied to online settings. Lasso feature selection without additional feature selection objectives led to poor BCI performance (simulations; Figure 3.8) and movement predictions (μECoG data, Figure 3.6). Our analyses revealed that this poor performance stemmed from the fact that estimating decoder parameters after selecting features often led to using "out of date", poorly estimated decoder parameters (Figure 3.3). This observation motivated us to develop feature selection algorithms designed for online settings by restructuring how feature selection interacts with feature weight estimation. We presented multiple possible ways to select features online and estimate their decoder weights, which vary in their efficiency and suitability for a particular BCI goal. Therefore, successful online feature selection requires algorithms that explicitly decouple these steps or account for the 'out-of-date' decoder weights.

The properties of BCI data may make it particularly important to consider interactions between feature selection and parameter estimation. Neural data related to movement is known to be high-dimensional and redundant [4, 80, 63]. Our simulated spiking data demonstrated this inherent redundancy, evidenced by the fact that offline feature selection methods applied to different batches of data, generated with the same neural encoding, selected different but equally high-performing subsets of neural features (Figure 3.3). When coupled with downstream decoder parameter estimation, the inherent redundancy of neural data exacerbates the "out of date" decoder estimation problem (Figure 3.3 conceptual diagram). We developed an algorithm that addresses this problem by directly regularizing the temporal history of selected features (Figure 3.6). Our results show that imposing a temporal constraint helped select more consistent subsets of informative neural features, thereby reducing the impact of "out of date" decoder parameters. Importantly, over-prioritizing temporal continuity can limit the benefits of adaptive feature selection, for instance hindering the ability to track neural encoder changes and compromising performance optimization overall (Figure 3.7). However, our simulations demonstrate the promise of our computational framework, which allows for flexible and parametric control of algorithm priorities via cost function optimization. Our framework could be extended to optimize additional properties of selected features, such as redundancy, which has been successful in offline settings [66, 21].

Our work highlights that feature selection algorithms will need to be fine-tuned to match properties of the neural data, including how task information is distributed across neural features and the timescales of encoder change. For example, we used our simulation to characterize how our TCC algorithm performed as we varied both the model hyperparameters and the neural encoder (Figure 3.7). This revealed that the optimal value of the temporal constraint parameter μ depends, in part, on the fraction of neural features within the population that have significant task information. Consistent with this, we found (via empirical testing) that we had to use notably different ranges of μ in our simulated data and the μECoG data. This demonstrates our algorithm's ability to adapt to data with different properties via hyperparameter tuning, but also highlights the potential challenge of setting these hyperparameters. In practical applications, one straightforward approach

is to estimate hyperparameter values from prior offline data and potentially fine-tune them online. Allowing algorithm parameters to adapt themselves may also be beneficial, particularly given the multiple sources of non-stationarity in BCIs. Decoder parameter adaptation research, for instance, shows potential benefits for letting update rates vary dynamically [20, 50]. Future work could explore a similar strategy for the TCC coefficient. Fully optimizing feature selection algorithms will ultimately require developing principled frameworks and methods to guide parameter tuning.

Critically, developing online feature selection requires benchmarking the algorithms in online closed-loop settings. In this study, we evaluated our online feature selection algorithms by developing a novel simulation framework that leveraged micro-electrocorticography (μECoG) data to approximate a closed-loop BCI scenario. By segmenting μECoG recordings from a well-learned task into sequential batches, we could rigorously test our methods on neural data with realistic, stationary statistical properties. This approach confirmed that, in an offline context using a single large data batch, our feature selection identified a task-relevant subset of features, consistent with established methods. However, these simulations have inherent limitations. While using μECoG data effectively captures the stationary statistics of neural encoding, this offline-replay model does not account for the dynamic, non-stationary changes that arise from user learning and other true closed-loop interactions. Our simulations, similarly, did not incorporate neural encoder changes based on closed-loop dynamics. Ultimately, the definitive validation of our algorithm will require its implementation and testing within a live, adaptive BCI system.

3.5 Conclusion

In conclusion, the increasing number of neural recordings from advancements in neural recording technologies presents a bottleneck in processing the growing number of neural features. To address this issue, feature selection methods have been proposed to select highly relevant and sparse sets of features. However, conventional feature selection methods may not be suitable for online feature selection in neural interface learning, where hardware bottlenecks limit the simultaneous streaming of multitudes of features. To overcome this challenge, we propose an online feature selection algorithm that optimizes for

feature selection relevance, sparsity, and smoothness. Our simulation studies and offline electrocorticography feature selection comparisons show that the smooth convex algorithm improved smooth feature selections and simulation task performance. Our proposed algorithm provides a promising solution for online feature selection with explicit optimization for consistent feature selection.

3.6 Supplementary

3.6.1 Mathematical encoder setup

Our simulation used a neural encoder model to translate the simulated user's intended movement direction into a neural activity of the population of neurons (Figure 2A). We assumed the user always intended to drive the cursor into the current target. User intention was modeled using a Linear Quadratic Regulator (LQR) framework [33]. The LQR framework models the brain's strategy as minimizing a cost function that balances state error (deviation from the target state) and control effort. In this model, the current state of the cursor is represented as $x_t = [p_{hor,t}, p_{hor,t}, v_{hor,t}, v_{ver,t}, 1]$, where $p_{hor,t}$ and $p_{ver,t}$ are the coordinates of the horizontal and vertical position, $v_{hor,t}$ and $v_{ver,t}$ are the velocity components, and the constant term 1 represents a bias term. The target state is defined as $[T_{hor}, T_{ver}, 0, 0, 1]$, where T_{hor} and T_{ver} specify the respective coordinates of target position, and the zero velocity ensures the cursor reaches the target and stops. The cost function is given by:

$$J = \sum_{t=0}^{\infty} (x_t^T Q x_t + u_t^T R u_t)$$

where $Q = \text{diag}([1, 1, 0, 0, 0])$ penalizes deviations in position while allowing more flexibility in velocity, and $R = 10000 \cdot \text{diag}([1, 1, 1])$ heavily penalizes excessive neural control inputs. Using this cost function, the next state is computed as:

$$x_{t+1} = Ax_t + BK(x_{\text{target}} - x_t)$$

where K is the optimal feedback gain matrix derived from the LQR optimization. Finally, the brain's neural encoder encodes the next state x_{t+1} into neural firing rates using equation 3.1 in the main text, enabling continuous control of the cursor toward the target.

3.6.2 Lasso Sub D algorithm

Algorithm 1 Batch-based Feature Selection with LASSO Regression

Input: X^i, Y^i, N_{select}

Output: θ^i

Step 1: Solve the LASSO regression problem with a low regularization parameter:

$$C^i_{LASSO} = \underset{C}{\operatorname{argmin}} \frac{1}{2n} ||Y^i - X^i C||_2^2 + \alpha_l ||C||_1 \quad (\alpha_l \text{ low})$$

Step 2: Rank-order the features: Rank the features based on the magnitude of their coefficients $\|C^i_{LASSO,m,:}\|_2$. **Step 3: Select the top N_{select} features:** Pick the top N_{select} features based on the rank-ordering.

Step 4: Return the feature selection vector:

$$\theta^i_m = \begin{cases} 1 & \text{if } m \text{ is among the top } N_{select} \text{ rank-ordered features} \\ 0 & \text{otherwise} \end{cases} \qquad m = 0, 1, 2, ..., N_{feats}$$

3.6.3 Full derivation of the relaxed sensor selection problem

The maximum likelihood estimation of x is

$$\hat{x} = \left(\sum_{i=1}^{m} c_i^T c_i \right)^{-1} \sum_{i=1}^{m} y_i c_i^T$$

The error of estimation $x - \hat{x}$ has zero mean and covariance of $\Sigma = \left(\sum_{i=1}^{m} c_i^T c_i \right)^{-1}$. A scalar indicator of the estimation's accuracy is the volume of the η-confidence ellipsoid around $x - \hat{x}$ given by

$$\operatorname{vol}(\mathcal{E}_\alpha) = \frac{(\alpha\pi)^{n/2}}{\Gamma\left(\frac{n}{2+1}\right)} \det \Sigma^{1/2} \tag{3.5}$$

where $\alpha = F^{-1}_{\chi^2_n}(\eta)$ ($F^{-1}_{\chi^2_n}$:is the cumulative distribution function of a χ-squared random variable with n degrees of freedom). By minimizing the log volume of the confidence

ellipsoid, Joshi and Boyed arrived at the canonical sensor selection problem expressed as an optimization problem:

$$\begin{aligned} \textbf{maximize} \quad & \textbf{logdet}\left(\sum_{i\in S} a_i a_i^T\right) \\ \text{s.t.} \quad & |S| = k \end{aligned} \tag{3.6}$$

where $|*|$ denotes the cardinality of the feature selection set. The above problem can be rewritten as

$$\begin{aligned} \text{maximize} \quad & \log\det\left(\textstyle\sum_{i=1}^m z_i a_i a_i^T\right) \\ \text{subject to} \quad & \mathbf{1}^T z = k \\ & z_i \in \{0,1\}, \quad i = 1,\ldots,m \end{aligned} \tag{3.7}$$

Joshi and Boyed then proposed to relax the binary constraint on the selection vector z. The convex relaxed problem is then,

$$\begin{aligned} \text{maximize} \quad & \log\det\left(\textstyle\sum_{i=1}^m z_i c_i^T c_i\right) \\ \text{subject to} \quad & \mathbf{1}^T z = k \\ & 0 \le z_i \le 1, \quad i = 1,\ldots,m \end{aligned} \tag{3.8}$$

where z_i indicates if a feature is being selected and k is the constraint on the total number of features.

3.6.4 Algorithms for selecting decoder updates

Inputs: feature selection vector $\theta^i \in R^N$, full decoder weight tracked $C \in R^{N\times N_s}$ **Output:** **decoder matrix:** C^i

Construct the feature selection mask $M \in R^{N\times N}$:

$$M_{m,:} = \theta_m^i$$

Select decoding features:

$$C^i = C \odot M^i$$

Yields C^i with non-zero elements

When the selected features differ from those currently used in BCI decoding, we expand the decoder observation matrix with zero rows to match the required dimensions. To simplify notation, we assume a known maximum number of neural features, which in practice does not create memory constraints due to the use of sets and maps. We define $\tilde{C}'$, the expanded matrix of dimension $n \times n_s$, as follows:

$$\tilde{C}'_{m,:} = \begin{cases} C_{\text{index}(m),:}, & \text{if } \theta^i_m = 1 \\ 0, & \text{otherwise} \end{cases} \tag{3.9}$$

Here, index(m) represents the position of the m-th selected feature in the smaller matrix. This function maps the indices of selected features from the feature vector to their corresponding rows, filling in the unselected positions with zeros. In the updated observation matrix, the newly selected features will be initialized with weights of zeros. Similarly, we expand the observation covariance matrix to have the dimension of $n \times n_s$ of the form,

$$\tilde{Q}'_{m,n} = \begin{cases} Q_{\text{index}(m),\text{index}(n)}, & \text{if } \theta^i_m = 1 \text{ and } \theta^i_n = 1 \\ \infty, & \text{otherwise} \end{cases} \tag{3.10}$$

Then the updated decoder parameters at i-th batch with newly selected batches becomes:

$$C^i = M^C \odot \tilde{C} \text{ and } Q^i = M^Q \odot \tilde{Q}$$

After removing the corresponding rows and columns, the observation matrix and its covariance matrix are ready to be updated and used for decoder update. The limitation of the earlier approach is that when a feature is deselected and later reselected, its weights are reset as if it were never used. We can enhance the previous method by preserving the feature weights. Instead of reinitializing, we track and store the weights of previously selected features, using weight-tracking matrices to identify selections from prior batches.

Inputs: feature selection vectors: θ^i and θ^{i-1} decoder matrices: $C^{i-1} \in R^{n_{i-1} \times N_s}$, $C \in R^{n \times R_s}$ **Output** decoder matrix: C^i

Construct the feature selection masks:

$$M_{mn}^{i-1,C} = \theta_m^{i-1}, U_{mn}^{i-1,C} = 1 - \theta_m^{i-1}$$

$$M_{mn}^{i,C} = \theta_m^{i}, U_{mn}^{i,C} = 1 - \theta_m^{i}$$

Expand the C^{i-1} to have the full dimension of $\tilde{C}^{i-1}$ using the equation 3.9 and 3.10

Store the previous observation matrix:

$$C \leftarrow M^{i-1,C} \odot \tilde{C}^{i-1} + U^{i-1,C} \odot C$$

$$Q \leftarrow M^{i-1,Q} \odot \tilde{Q}^{i-1} + U^{i-1,Q} \odot Q$$

Perform decoder parameter selection:

$$C^i = M^C \odot \tilde{C} \text{ and } Q^i = M^Q \odot \tilde{Q}$$

Then, the rows of zeros are removed to yield the new observation matrix and

Chapter 4

DESIGN OF A MYOELECTRIC CO-ADAPTIVE DECODER FOR RAPID AND CONCURRENT ACQUISITION OF A HYBRID TASK

Abstract

This study addresses the problem of EMG interface users struggling to control multiple tasks at once due to difficulties with calibration, performance variability, and unintuitive multitasking. We developed a novel hybrid EMG task that involves a continuous tracking task and a binary color classification task, and implemented a co-adaptive approach to the hybrid task decoder that takes into account the user's learning. The hybrid decoder is continuously adapted online and optimized using the SmoothBatch algorithm. The study's preliminary results suggest that participants can efficiently control the hybrid tasks, and that decoder adaptation parameters strongly influence the ability to do so. This study is one of the first to demonstrate that EMG users can simultaneously acquire two tasks with the help of principled EMG decoder design.

4.1 Introduction

Natural arm and hand movements enable multitasking, allowing for smooth transitions between tasks. Although recent electromyographic (EMG) interfaces have shown promise for controlling simple tasks with robotic arms and computer cursors, existing EMG interfaces have problems with ease of calibration, performance variability, and unintuitive multitasking. Most EMG multitask control schemes focus either on classification or continuous control. Therefore, it is unclear if EMG users can concurrently learn control of distinct EMG interface tasks, and unclear how to design EMG decoders that accommodate simultaneous task acquisition.

To address this problem, we developed a novel hybrid EMG task where the user is instructed to simultaneously perform a continuous tracking task and a binary color classifi-

cation task. To help the user learn the hybrid task, we consider the user and the decoder as a whole to optimize their learning at the same time, as opposed to optimizing each individual learner separately and potentially resulting in globally non-optimal decoder solutions. We leverage our game-theoretic framing of two-learner systems [51], which allows us to explicitly define objectives of the system that may improve performance, such as a crosstalk term to minimize interference between tasks. To accommodate user learning, we take into account the user's learning and apply a co-adaptive approach to the hybrid task decoder where the decoder is continuously adapted online [62]. In this work, we aim to test if a decoder with an explicit goal to minimize task interference may help participants improve performance on the hybrid task.

4.2 Methods

4.2.1 The hybrid task

The hybrid task consisted of a one-dimensional tracking task similar to prior work [51] and a two-color classification task. The hybrid decoder is adapted in real-time as users control the task with EMG signals measured from a 64-channel surface EMG system (Bioelettronica). The hybrid decoder consisted of a velocity-based Wiener Filter for continuous control and a linear classifier for color switching. EMG activity from all channels is used as input to both tasks, and the continuous decoder and the classifier are coupled by a cost term to minimize channel crosstalk. The hybrid decoder was recursively exponentially optimized online using the SmoothBatch algorithm [60].

4.2.2 Participants' demographics and experience with EMG control

We recruited eight participants from the University of Washington Seattle campus. Participant diversity depended on voluntary enrollment in the study. Among the participants, there were six male and two female participants. The participants had an average age of 28 years (SD = 6 years).

Participants were naive to the hybrid task and EMG control. Four out of the eight participants reported that they played racing games growing up. Six out of the eight

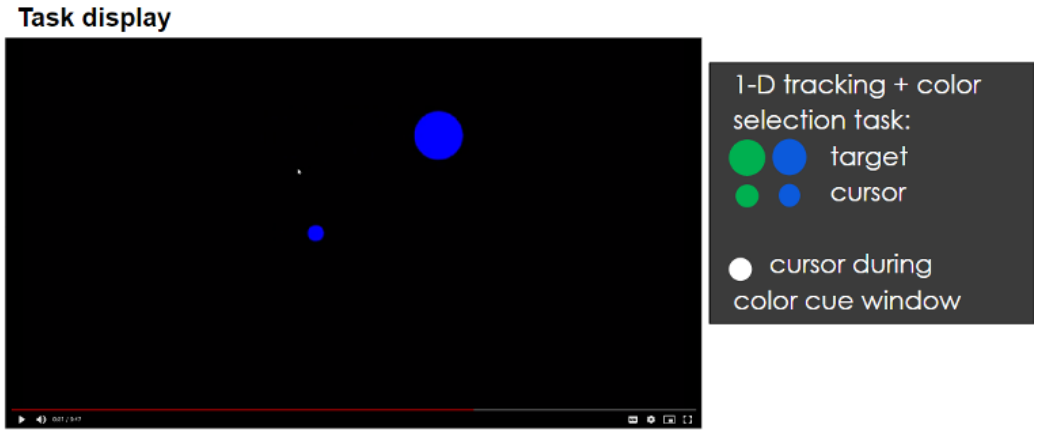

Figure 4.1: **Overview of the hybrid task (top) and the hybrid decoder scheme (bottom).** The decoder scheme includes: (0) velocity-based decoder for both tasks (60 Hz), (1) Integration for continuous position and classification for discrete color, (2) Asynchronous decoder updates, and (3) SmoothBatch updates.

participants self-reported as being casual or novice gamers, and one reported as a non-professional expert. All eight participants had no prior experience with EMG control.

4.2.3 Study design and data collection

The goal of this study was to examine whether co-adaptive decoders influence participants' learning of hybrid control. For naive participants, however, we found that the results may be confounded by participants learning the task structure and EMG control. We began by not suggesting possible approaches to gesture control. Unfortunately, for two of the first three participants, the EMG recording setups were not effective in picking up the user's selected gestures. As an example, one participant transferred learning from mouse control to gesture control by tapping their index finger for one color and middle finger for the other color (Figure 4.2A).

We therefore modified our experimental protocol for the remaining 6 participants with a goal of guiding participants' EMG control while still allowing them to explore a wide range of gestures (Figure 4.2B, C). Participants reported identifying strategies that gave them color switching success, such as a palm open strategy for one participant (Figure 4.2C, D).

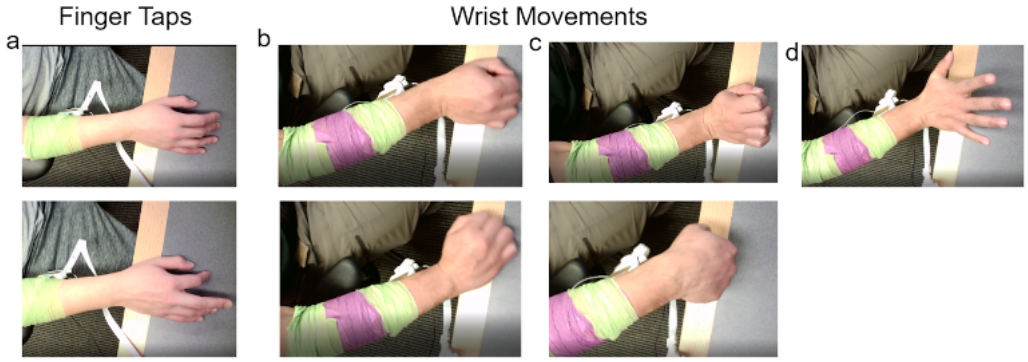

Figure 4.2: **Examples of participant's gestural control strategies. A:** Finger Taps were ineffective. **B:** Left and right wrist movements were effective for tracking. **C:** Up and down wrist movements were an example for color switching. **D:** Participants in later trials learned a combined wrist up and palm open strategy for color switching.

Session structure: Participant data collection sessions included the following steps and instructions:

1. **Preparing the participants for the task and EMG control:** Participants were instructed to perform the hybrid task with a mouse cursor. The participants were then allowed to explore wrist/hand control with a radar task where the EMG amplitudes from the 64 channels were plotted in polar coordinates. Participants were reminded to keep their forearm still, as forearm movements would not produce distinct EMG control patterns.

2. **Help the participants co-learn with the decoder:** We suggested that participants keep the same gestures for at least one minute before trying another gesture control strategy.

3. **Participants perform the tasks with four decoder conditions in two blocks:** Each trial for a decoder condition was 6 minutes long to provide sufficient time for learning. Participants were given a break in between the two blocks and reminded of

the EMG control tips.

4. **Participants perform two trials of individual tasks:** One trial of stationary color switching task only, and one trial of one-dimensional tracking task only.

5. **Interview:** Participants were interviewed and their demo gestures were recorded with consent.

4.2.4 Subject data inclusion/exclusion criterion

We collected data from eight participants in total. Due to technical challenges, we were only able to collect one block of data from the first participant. The next two participants did not seem to acquire effective control gestures: one was stuck with trying finger taps for the discrete color task; the other participant used a hand swirl about their forearm to acquire continuous tracking control. After this experience, we altered our subject instruction protocols (see above) to improve the likelihood of users identifying successful strategies.

We will first report the hybrid performance from the seven subjects who finished the entire data collection ($N = 7$). Because we refined our experimental protocols, we excluded the first two subjects for the analyses reported below on the effect of crosstalk penalty on the hybrid task performance ($N = 5$).

4.2.5 Measure of decoder orthogonality

Because we explicitly penalized decoder similarity in the objective function with a dot product, we can use the decoder dot product to directly measure decoder similarity.

4.2.6 Performance metrics

We measured each participant's color classification accuracy in rolling 8 trials. The chance level of randomly choosing the correct color was 0.5 (50%). We also measured participants' continuous tracking performance by tracking error smoothed with a sliding 1-minute window.

4.2.7 Encoder estimation

Users initially had no knowledge of what they were controlling. User encoders were estimated as linear with two different models, with user input as either the target trajectory or velocity. EMG activity was z-scored and user input was normalized before processing. One encoder was estimated for each trial. For initial analyses, we only computed the encoder for the continuous task.

MODEL 1 (Position Control):

$$S = W \cdot P_{target} + \epsilon \tag{4.1}$$

Where S is EMG activity $(64 \times t)$, W is the user encoder (64×1), P_{target} is the target position $(1 \times t)$, and ϵ is the offset $(64 \times t)$.

MODEL 2 (Velocity Control):

$$S = W_v \cdot v_{int} + \epsilon \tag{4.2}$$

Where W_v is the user encoder (64×1), and v_{int} is the intended velocity (estimated as the velocity vector between the cursor and target).

4.3 Results and Discussion

4.3.1 Effects of decoder objectives

To determine the relative weights in the decoder formulation, we simulated the decoder change as a pilot subject ($N = 1$, outside of the experimental group) was performing the task without the decoder objectives. For both the continuous and discrete decoders, increasing the size penalty decreases the decoder sizes (Figure 4.3, top row). Furthermore, when there was no decoder constraint (Figure 4.3, bottom, $\lambda = 0$), a higher decoder crossover penalty successfully separated the decoder as measured by the cosine angle. However, increasing the size penalty weight made it hard for the decoders to be separated apart (Figure 4.3, bottom).

Based on the simulation, we collected data by varying the decoder cost function parameters as follows:

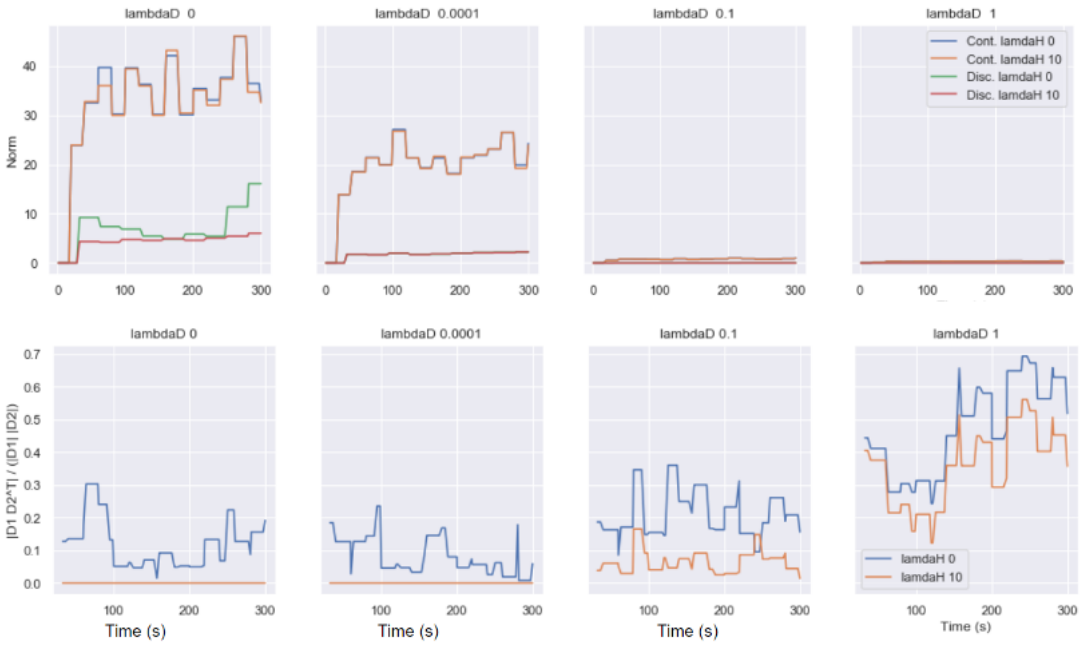

Figure 4.3: **Effects of decoder penalties.** (Top) Effects of increasing decoder norm weight on the norms of the continuous and discrete decoders. (Bottom) Effects of increasing norm weight on the separability of the decoders with large and small orthogonality objective weights.

- With or without decoder crosstalk penalty: $\mu \in [0, 10]$
- With or without decoder effort penalty: $\lambda_1 = \lambda_2 \in [0, 1]$

We studied how these decoder parameters influenced both performances on the task and the user's encoder.

4.3.2 Comparing hybrid task performance from all seven participants

For the color switching task (Figure 4.4, top), the participants on average can perform only slightly above chance level. As the curves with and without the decoder orthogonality constraint are interleaving, the decoder orthogonality constraint did not appear to influence task performance. Participants did well on the individual color switching task, approaching an accuracy of 1.0. This highlights that the color switching task alone is both understood by the subject and can be performed by the decoder. This highlights the difficulty of the hybrid task for both the participant and the decoder.

For the tracking task (Figure 4.4, bottom), tracking errors do not decrease for decoders without decoder orthogonality (yellow curve), but do decrease for decoders with the decoder orthogonality constraint (blue curve). Interestingly, the tracking errors for tracking task alone rapidly decreased but then increased again around halfway through the task (150 seconds), suggesting likely user fatigue that could contribute to modest performance gains observed across all conditions.

4.3.3 The effects of orthogonality penalty on participants' performance

As described in the Methods section, we focus the analyses on the five participants who acquired effective EMG control gestures.

For the color switching task (Figure 4.5, top), participants were able to learn the task beyond chance level for both with and without the decoder orthogonality penalty ($\mu = 0$ or $\mu = 10$) but not approaching the success rate with the color task only. However, the effect of orthogonality was not clear from the experiment because the success rates in both curves increased at the beginning and then intertwined as the learning went on.

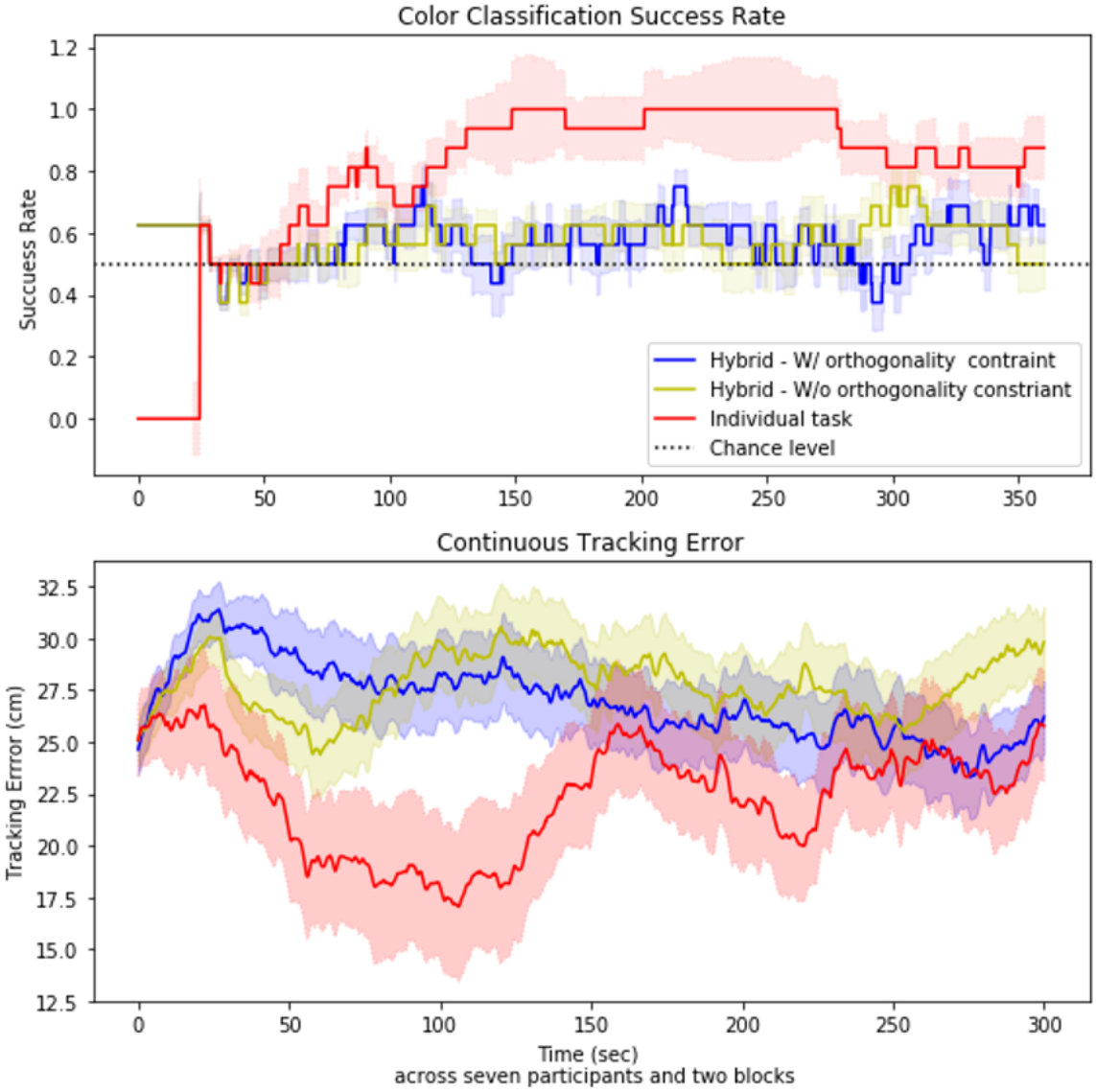

Figure 4.4: **Comparison of participants' performance on the color switching task (top) and the tracking task (bottom).** Data is shown with and without decoder orthogonality penalty, alongside individual task performance.

For the tracking task (Figure 4.5, bottom), the orthogonality penalty helped to decrease the tracking error compared to the decoder condition without the orthogonality penalty. Even with the orthogonality penalty, the participants did not decrease the error as quickly as when they did the tracking task alone, highlighting the difficulty of the hybrid task.

It is interesting to note that even for participants doing the individual tasks, their performance started to fluctuate around the middle of the trial (Figure 4.5 top, around 280 seconds; Figure 4.5 bottom, individual task at around 125 seconds). This effect could be due to participants trying to explore different strategies or experiencing fatigue.

4.3.4 The cross-talk penalty term only helped all participants improve the continuous task

We found that decoder effort penalties can potentially limit the decoders to be orthogonal. Here these results are confirmed but in the continuous tracking performance. Compared to decoders with the decoder effort penalty, decoders without the decoder effort penalty had greater color switching success and lower tracking errors (Dotted lines vs. Solid lines on Figure 4.6).

The orthogonality penalty without decoder effort penalty helped the participants converge to smaller tracking errors (Figure 4.6, right, comparing yellow and blue solid lines). This was not the case for the color switching task (Figure 4.6, left, comparing yellow and blue solid lines). The improvement for the decoder with the orthogonality penalty was significant (Figure 4.7).

4.3.5 User encoder estimation across decoder conditions

We next aim to quantify user learning in the hybrid task by quantifying user encoders. We first compared goodness of fit for the position control model (MODEL 1) and velocity control model (MODEL 2). Both velocity and position control models performed better on actual user data than randomly shuffled data, suggesting that the models captured some users' control strategies. Across the five participants and for both mean and max R^2, the velocity control model was the better fit (Figure 4.8, left and right), which matched the velocity-control decoder by design.

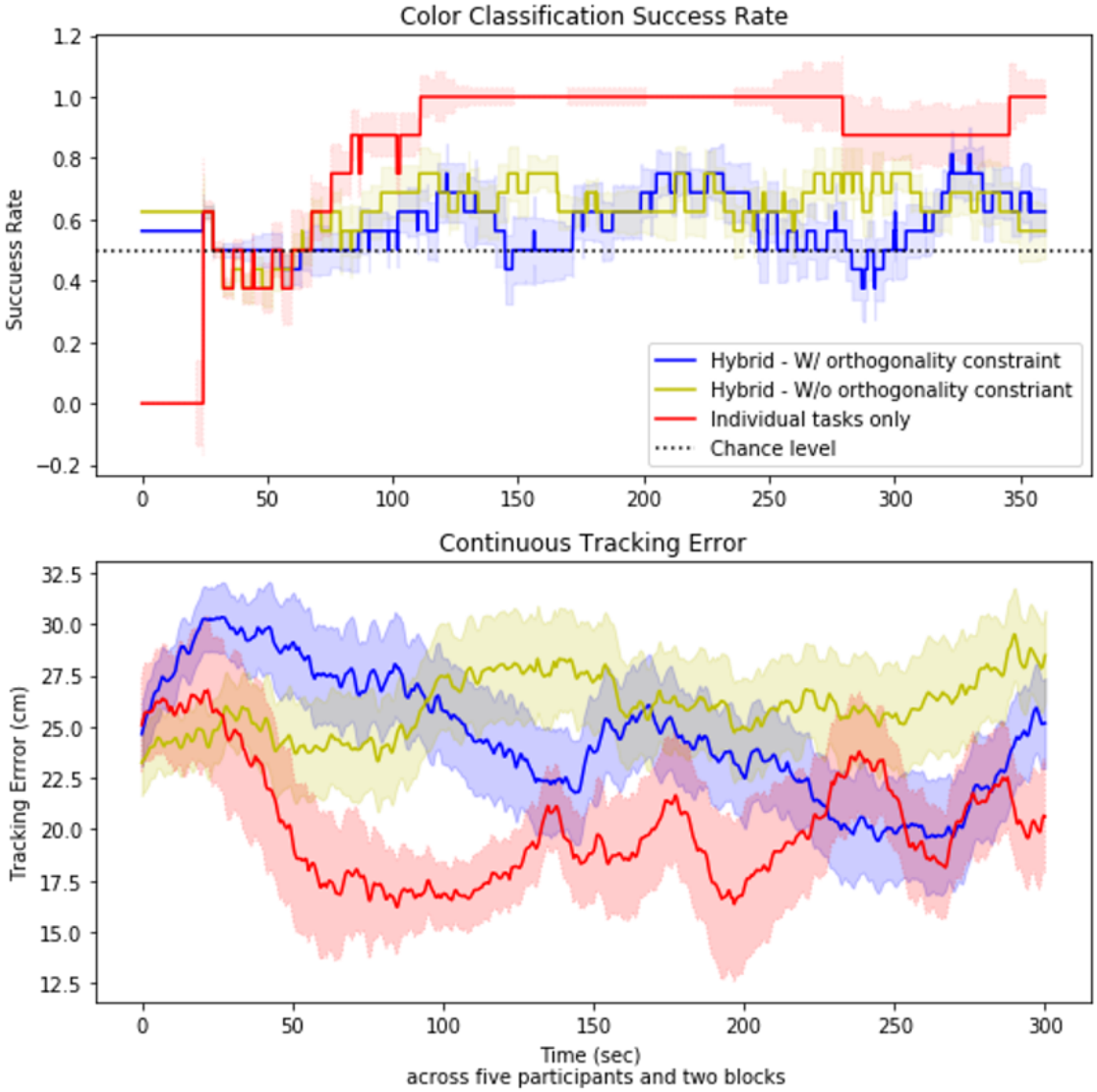

Figure 4.5: **Comparison of participants' performance on the color switching task (top) and the tracking task (bottom).** Comparison shown with and without decoder orthogonality penalty and individual tasks.

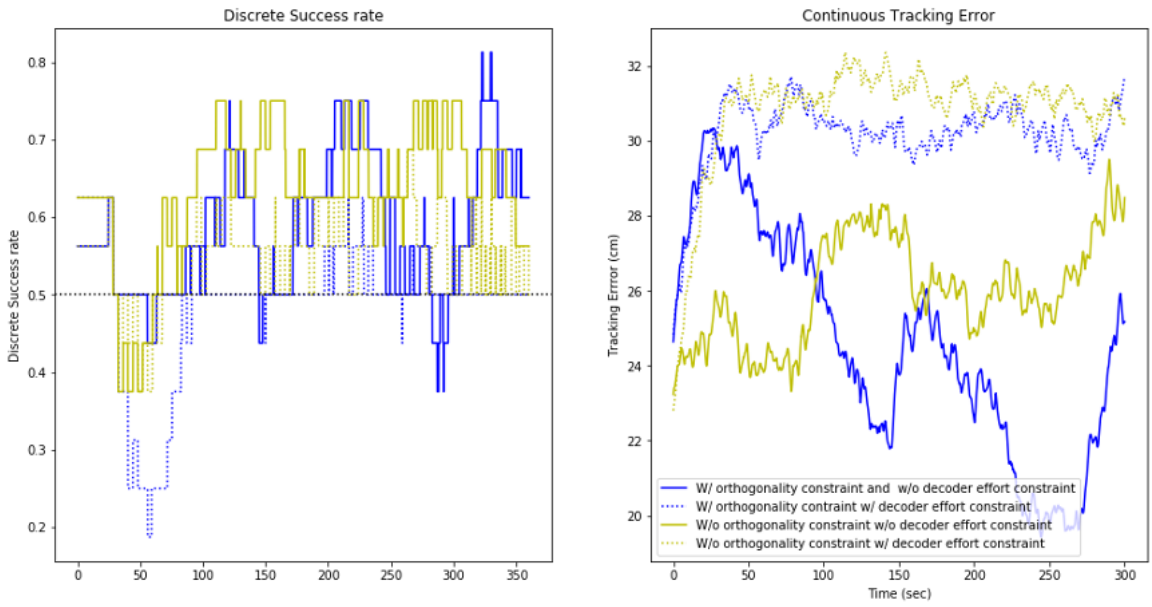

Figure 4.6: **Comparison of participants' performance for a breakdown of orthogonality and effort penalties.** (Left) Color switching success rate. (Right) Mean continuous tracking error.

We then used the velocity encoder model to measure the user's encoder change for the continuous task. We used the average decoder from the last three decoder updates and reconstructed the velocities. If the user's encoder changes over the course of a trial, the error between reconstructed/fixed velocity and the intended velocity will become smaller towards the last few decoder updates.

The errors for trials without the orthogonality penalty were oscillatory and appeared to be non-decreasing (Figure 4.9, blue curve). However, the trials with the orthogonality penalty appeared to be decreasing on average (Figure 4.9, orange curve), though with some additional oscillations. This suggests that with the orthogonality penalty, users may begin learning a more consistent strategy, consistent with the observed performance improvements. However, the oscillations observed in both conditions suggest that the demands of the hybrid task may limit the user's ability to identify and maintain a consistent strategy throughout the trial. Furthermore, it is interesting to note that in all cases, the difference in the velocity error increased significantly after 300 seconds, which may be a result of encoder change from

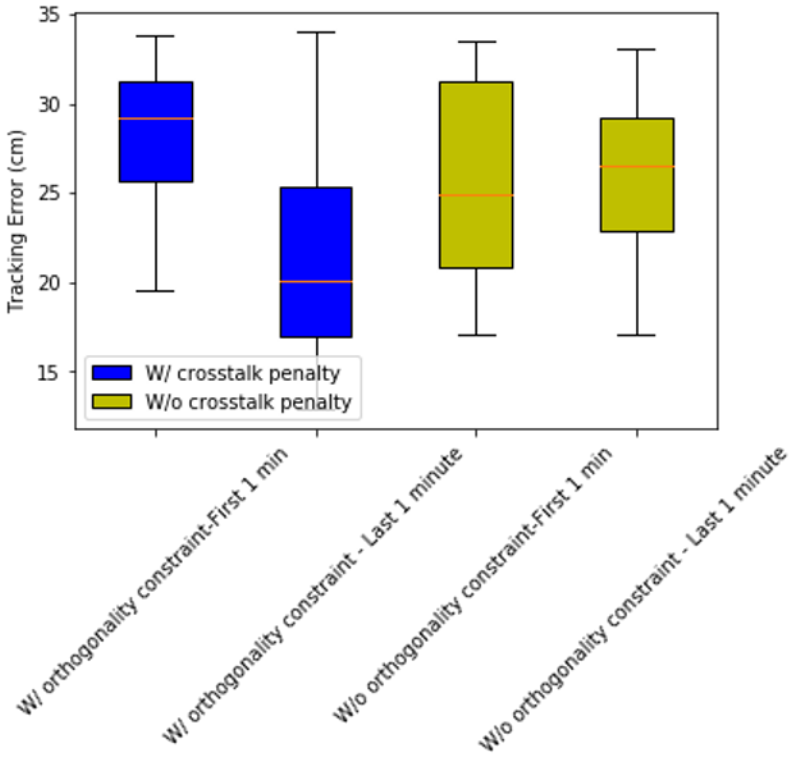

Figure 4.7: **Comparison of participant's performance before and after with and without decoder orthogonality penalties.**

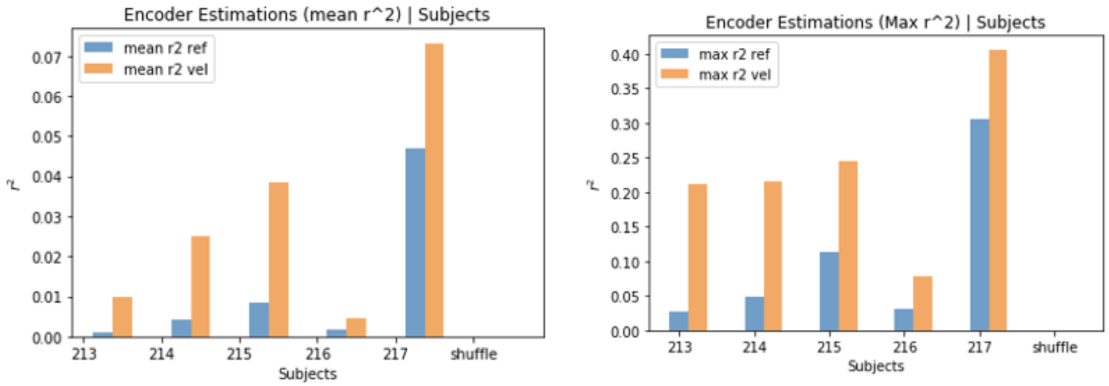

Figure 4.8: **Comparison of goodness of fits (R^2) for the five subjects.** (Left) The mean goodness of fit. (Right) The maximum goodness of fit. Note that vertical scales differ between plots.

participant fatigue.

In addition, we applied the encoder estimation analyses to trials with and without the decoder effort constraint. Without the decoder effort constraint, the difference between the intended and fixed/reconstructed velocity appears to be oscillatory and non-decreasing, suggesting that the user's encoder was not converging. This observation accords well with the observation that with the decoder effort constraint, the user's performance was poor and did not improve. For the trials without the decoder effort constraint, the differences in velocities were decreasing with occasional oscillations, and a sharp increase in the last 50 seconds of the trial, consistent with our findings above that users begin to converge on a strategy but may shift near the end of trials due to fatigue.

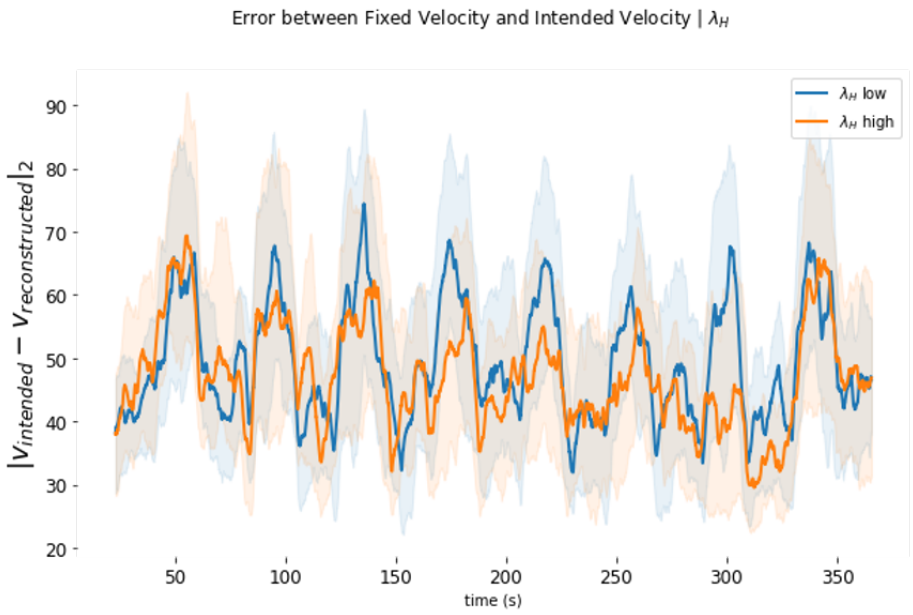

Figure 4.9: **Comparison of error between fixed velocity and intended velocity with and without decoder orthogonality constraint.**

4.4 Conclusions and Future Directions

Our preliminary results suggest that participants can obtain efficient control of the hybrid tasks, and that this ability is strongly influenced by decoder adaptation parameters. Sub-

ject learning rates varied significantly, suggesting that customized decoder adaptation may facilitate additional improvements. To the best of our knowledge, this study is one of the first studies to demonstrate that the EMG user can simultaneously acquire two tasks with the help of principled EMG decoder design.

In sum, these analyses suggest that when user performance improves in the continuous task, it is paired with users learning some more consistent encoder. Encoder analyses further suggest that users may vary their strategies for the continuous task as they try to balance the color switching task demands, and that fatigue may be a significant factor. Future directions will explore providing users with adequate time to learn the task, but incorporating frequent breaks to mitigate fatigue. Based on clear and consistent observations about the contributions of penalty terms, these experiments will focus on a limited range of decoder parameters (removing the decoder effort penalty parameter).

Chapter 5

CONCLUSION

Clinical BCI translation requires the rigorous integration of high-density recording hardware, decoding algorithms, and user learning protocols. This thesis provides computational frameworks that optimize closed-loop BCI decoding by addressing distinct yet complementary aspects of signal processing and control. We establish methodologies that improve decoding by characterizing multiscale signal properties (Chapter 1), stabilizing online feature selection (Chapter 2), and enabling robust multitasking control in closed-loop systems (Chapter 3).

Standard decoding pipelines typically treat neural features uniformly, ignoring inherent signal non-stationarity. In Chapter 1, our multiscale analysis challenges this approach, demonstrating that feature variance is distinct across spatial, temporal, and physiological scales. Specifically, we identified intracortical channels that correlate strongly with mesoscopic surface potentials (ECoG) yet encode minimal task-relevant information. These findings indicate that feature abundance does not equate to information density, necessitating the decoupling of feature selection from the decoding objective to effectively isolate the informative subspace. Future work should evaluate whether targeted modulation of these hub electrodes can improve decoder performance by providing a more stable, mesoscopic representation of the underlying neural manifold. By prioritizing these informative hubs, we can develop decoding architectures that are less susceptible to the high-frequency noise and non-stationarity of individual intracortical channels, ultimately leading to more robust and consistent real-time BCI control.

In Chapter 2, we addressed the coordination of feature selection and decoder training in online regimes. We demonstrated that standard feature selection methods fail when applied directly to closed-loop data due to statistical instability. By explicitly modeling the temporal evolution of feature relevance, our proposed framework stabilizes online perfor-

mance. This approach lays the technical groundwork for efficient dimensionality reduction, enabling high-bandwidth interfaces to operate within the computational budgets of clinical hardware. While Chapter 2 established the theoretical framework for the Temporal Continuity Constraint (TCC) to decouple feature selection from weight estimation, future work must optimize this for resource-constrained systems. Specifically, implementing the convex optimization directly on embedded hardware would allow for 'on-chip' selection, where the algorithm gates data transmission at the sensor level to realize actual power and bandwidth savings. Additionally, future research should explore automated, online methods for tuning the temporal constraint parameter (μ) to account for varying rates of neural non-stationarity, such as the abrupt signal loss seen in hardware degradation versus the gradual shifts of user learning.

In Chapter 3, we leveraged the closed-loop paradigm to implement a decoder for multitasking. Results indicate that participants can achieve efficient control over hybrid tasks—combining continuous tracking with discrete classification. Crucially, we found that the user's ability to acquire this control is strictly governed by specific decoder adaptation parameters, rather than the architecture alone. Building on the finding that multitasking control is strictly governed by adaptation parameters rather than architecture alone, future research must address the 'co-adaptation' problem. Outstanding questions include whether reinforcement learning agents can autonomously tune these adaptation parameters in real-time by monitoring user error signals, thereby reducing the cognitive load required for the user to acquire control over hybrid continuous-discrete tasks.

Collectively, this thesis establishes methodologies for real-time feature selection and robust multitasking control. By addressing the specific computational bottlenecks of high-dimensional non-stationary data and the complexity of hybrid behaviors, these frameworks contribute essential architectural solutions for the development of robust, everyday-use BCI systems. Moving forward, the definitive validation of these computational frameworks will require their integration within live, closed-loop clinical systems. A primary objective is the development of 'hardware-aware' architectures where multiscale feature selection—informed by the stable mesoscopic hubs identified in Chapter 1—directly gates data transmission on embedded systems using the Temporal Continuity Constraints established in Chapter 2.

Furthermore, future research must address the 'co-adaptation' problem by investigating whether autonomous agents can utilize the adaptation parameters identified in Chapter 3 to respond to real-time neural non-stationarity. By coordinating these adaptive layers, we can transition from static decoding pipelines to autonomous BCI systems that maintain high-fidelity performance across the diverse, non-stationary environments of everyday use.

BIBLIOGRAPHY

[1] Reza Abiri, Soheil Borhani, Eric W. Sellers, Yang Jiang, and Xiaopeng Zhao. A comprehensive review of EEG-based brain–computer interface paradigms. *Journal of Neural Engineering*, 16(1):011001, 2019.

[2] Tyson Aflalo, Spencer Kellis, Christian Klaes, Brian Lee, Ying Shi, Kelsie Pejsa, Kathleen Shanfield, Stephanie Hayes-Jackson, Mindy Aisen, Christi Heck, et al. Decoding motor imagery from the posterior parietal cortex of a tetraplegic human. *Science*, 348(6237):906–910, 2015.

[3] A. Bolu Ajiboye, Francis R. Willett, Daniel R. Young, William D. Memberg, Brian A. Murphy, Jonathan P. Miller, Benjamin L. Walter, Jennifer A. Sweet, Harry A. Hoyen, Michael W. Keith, P. Hunter Peckham, John D. Simeral, John P. Donoghue, Leigh R. Hochberg, and Robert F. Kirsch. Restoration of reaching and grasping movements through brain-controlled muscle stimulation in a person with tetraplegia: a proof-of-concept demonstration. *Lancet (London, England)*, 389(10081):1821–1830, May 2017.

[4] Fiorenzo Artoni, Chiara Fanciullacci, Federica Bertolucci, Alessandro Panarese, Scott Makeig, Silvestro Micera, and Carmelo Chisari. Unidirectional brain-to-muscle connectivity reveals motor cortex control of leg muscles during gait. *Neuroimage*, 159:403–416, 2017.

[5] Tonio Ball, Andreas Schulze-Bonhage, Ad Aertsen, and Carsten Mehring. Differential representation of arm movement direction in relation to cortical anatomy and function. *Journal of Neural Engineering*, 6(1):016006, January 2009.

[6] Arjun K Bansal, Wilson Truccolo, Carlos E Vargas-Irwin, and John P Donoghue. Decoding 3d reach and grasp from hybrid signals in motor and premotor cortices: spikes,

multiunit activity, and local field potentials. *Journal of Neurophysiology*, 107(5):1337–1355, 2012.

[7] James C. Barrese, Naveen Rao, Kaivon Paroo, Corey Triebwasser, Carlos Vargas-Irwin, Lachlan Franquemont, and John P. Donoghue. Failure mode analysis of silicon-based intracortical microelectrode arrays in non-human primates. *Journal of Neural Engineering*, 10(6):066014, December 2013.

[8] André M Bastos, Julien Vezoli, Conrado A Bosman, Jan-Mathijs Schoffelen, Robert Oostenveld, John R Dowdall, Peter De Weerd, Henry Kennedy, and Pascal Fries. Visual areas exert feedforward and feedback influences through distinct frequency channels. *Neuron*, 85(2):390–401, 2015.

[9] Christopher M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[10] Chad E Bouton, Ammar Shaikhouni, Nicholas V Annetta, Marcia A Bockbrader, David A Friedenberg, Dylan M Nielson, Gaurav Sharma, Per B Sederberg, Bradley C Glenn, W Jerry Mysiw, et al. Restoring cortical control of functional movement in a human with quadriplegia. *Nature*, 533(7602):247–250, 2016.

[11] Mariana P. Branco, Simon H. Geukes, Erik J. Aarnoutse, Nick F. Ramsey, and Mariska J. Vansteensel. Nine decades of electrocorticography: A comparison between epidural and subdural recordings. *European Journal of Neuroscience*, 57(8):1260–1288, 2023. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/ejn.15941.

[12] David M. Brandman, Tommy Hosman, Jad Saab, Michael C. Burkhart, Benjamin E. Shanahan, John G. Ciancibello, Anish A. Sarma, Daniel J. Milstein, Carlos E. Vargas-Irwin, Brian Franco, Jessica Kelemen, Christine Blabe, Brian A. Murphy, Daniel R. Young, Francis R. Willett, Chethan Pandarinath, Sergey D. Stavisky, Robert F. Kirsch, Benjamin L. Walter, A. Bolu Ajiboye, Sydney S. Cash, Emad N. Eskandar, Jonathan P. Miller, Jennifer A. Sweet, Krishna V. Shenoy, Jaimie M. Henderson, Beata Jarosiewicz, Matthew T. Harrison, John D. Simeral, and Leigh R. Hochberg.

Rapid calibration of an intracortical brain-computer interface for people with tetraplegia. *Journal of Neural Engineering*, 15(2):026007, April 2018.

[13] Steven M. Chase, Andrew B. Schwartz, and Robert E. Kass. Bias, optimal linear estimation, and the differences between open-loop simulation and closed-loop performance of spiking-based brain-computer interface algorithms. *Neural networks : the official journal of the International Neural Network Society*, 22(9):1203–1213, November 2009.

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

[15] Mark M Churchland, Byron M Yu, John P Cunningham, Leo P Sugrue, Marlene R Cohen, Greg S Corrado, William T Newsome, Andrew M Clark, Paymon Hosseini, Benjamin B Scott, et al. Stimulus onset quenches neural variability: a widespread cortical phenomenon. *Nature neuroscience*, 13(3):369–378, 2010.

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

[17] John P. Cunningham, Paul Nuyujukian, Vikash Gilja, Cindy A. Chestek, Stephen I. Ryu, and Krishna V. Shenoy. A closed-loop human simulator for investigating the role of feedback control in brain-machine interfaces. *Journal of Neurophysiology*, 105(4):1932–1949, April 2011.

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

[19] Siddharth Dangi, Suraj Gowda, Helene G. Moorman, Amy L. Orsborn, Kelvin So, Maryam Shanechi, and Jose M. Carmena. Continuous Closed-Loop Decoder Adaptation with a Recursive Maximum Likelihood Algorithm Allows for Rapid Perfor-

mance Acquisition in Brain-Machine Interfaces. *Neural Computation*, 26(9):1811–1839, September 2014.

[20] Siddharth Dangi, Amy L. Orsborn, Helene G. Moorman, and Jose M. Carmena. Design and analysis of closed-loop decoder adaptation algorithms for brain-machine interfaces. *Neural Computation*, 25(7):1693–1731, July 2013.

[21] Chris Ding and Hanchuan Peng. Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, 3(02):185–205, 2005.

[22] I Driscoll, DL Pettit, M Minderer, SN Chettih, and CD Harvey. Dynamic reorganization of neuronal activity patterns in irregular cortical networks. *Nature communications*, 8(1):1–16, 2017.

[23] Collin F. Dunlap, Samuel C. Colachis, Eric C. Meyers, Marcia A. Bockbrader, and David A. Friedenberg. Classifying Intracortical Brain-Machine Interface Signal Disruptions Based on System Performance and Applicable Compensatory Strategies: A Review. *Frontiers in Neurorobotics*, 14:558987, October 2020.

[24] Bradley J. Edelman, Shuailei Zhang, Gerwin Schalk, Peter Brunner, Gernot Müller-Putz, Cuntai Guan, and Bin He. Non-invasive Brain-Computer Interfaces: State of the Art and Trends. *IEEE Reviews in Biomedical Engineering*, pages 1–25, 2024. Conference Name: IEEE Reviews in Biomedical Engineering.

[25] Nir Even-Chen, Dante G. Muratore, Sergey D. Stavisky, Leigh R. Hochberg, Jaimie M. Henderson, Boris Murmann, and Krishna V. Shenoy. Power-saving design opportunities for wireless intracortical brain–computer interfaces. *Nature Biomedical Engineering*, 4(10):984–996, October 2020.

[26] Arseny Finkelstein, Kayvon Daie, Márton Rózsa, Ran Darshan, and Karel Svoboda. Connectivity underlying motor cortex activity during goal-directed behaviour. *Nature*, pages 1–7, November 2025.

[27] Pascal Fries. Rhythms for Cognition: Communication through Coherence. *Neuron*, 88(1):220–235, October 2015.

[28] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4):1–37, 2014.

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

[30] A. P. Georgopoulos, R. E. Kettner, and A. B. Schwartz. Primate motor cortex and free arm movements to visual targets in three- dimensional space. II. Coding of the direction of movement by a neuronal population. *Journal of Neuroscience*, 8(8):2928–2937, August 1988. Publisher: Society for Neuroscience Section: Articles.

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

[32] Joshua I Glaser, Ari S Benjamin, Roozbeh Farhoodi, and Konrad P Kording. Machine learning for neural decoding. *eNeuro*, 7(4), 2020.

[33] Suraj Gowda, Amy L. Orsborn, Simon A. Overduin, Helene G. Moorman, and Jose M. Carmena. Designing Dynamical Properties of Brain–Machine Interfaces to Optimize Task-Specific Performance. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(5):911–920, September 2014.

[34] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[35] Behzad A. Haghi and Maryam Shoaran. A 16-channel low-power neural connectivity extraction and phase-locked deep brain stimulation soc. *IEEE Journal of Solid-State Circuits*, 2021.

[36] Behzad A Haghi, M Taghavi, M Farivar, and Maryam Shoaran. A 16-channel low-power neural connectivity extraction and phase-locked deep brain stimulation SoC. *IEEE Journal of Solid-State Circuits*, 59(1):230–245, 2024.

[37] SD Hall et al. The relationship between spiking activity and rhythmic field potentials. *Journal of Neuroscience*, 2014.

[38] Robert E Hampson, Dong Song, Brian S Robinson, Dustin Fetterhoff, Alexander S Dakos, Brent M Roeder, Xiwei She, Robert T Wicks, Mark R Witcher, Daniel E Couture, et al. Developing a hippocampal neural prosthetic to facilitate human memory encoding and recall. *Journal of Neural Engineering*, 15(3):036014, 2018.

[39] Reid R Harrison and Cameron Charles. A low-power low-noise cmos amplifier for neural recording applications. *IEEE Journal of Solid-State Circuits*, 38(6):958–965, 2003.

[40] Trevor Hastie, Robert Tibshrani, and Jerome Friedman. *Elements of Statistical Learning: data mining, inference, and prediction. 2nd Edition.* 2009.

[41] Leigh R. Hochberg, Daniel Bacher, Beata Jarosiewicz, Nicolas Y. Masse, John D. Simeral, Joern Vogel, Sami Haddadin, Jie Liu, Sydney S. Cash, Patrick van der Smagt, and John P. Donoghue. Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature*, 485(7398):372–375, May 2012.

[42] Leigh R. Hochberg, Mijail D. Serruya, Gerhard M. Friehs, Jon A. Mukand, Maryam Saleh, Abraham H. Caplan, Almut Branner, David Chen, Richard D. Penn, and John P. Donoghue. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, 442(7099):164–171, July 2006.

[43] Guosong Hong and Charles M. Lieber. Novel electrode technologies for neural recordings. *Nature Reviews Neuroscience*, 20(6):330–345, June 2019.

[44] Beata Jarosiewicz, Nicolas Y. Masse, Daniel Bacher, Sydney S. Cash, Emad Eskandar, Gerhard Friehs, John P. Donoghue, and Leigh R. Hochberg. Advantages of closed-

loop calibration in intracortical brain–computer interfaces for people with tetraplegia. *Journal of Neural Engineering*, 10(4):046012, July 2013.

[45] Beata Jarosiewicz, Anish A Sarma, Daniel Bacher, Nicolas Y Masse, John D Simeral, Brittany Sorice, Erin M Oakley, Christine Blabe, Chethan Pandarinath, Vikash Gilja, et al. Advantages of closed-loop calibration in intracortical brain-computer interfaces for people with tetraplegia. *Science translational medicine*, 7(313):313ra179–313ra179, 2015.

[46] James J Jun, Nicholas A Steinmetz, Joshua H Siegle, Daniel J Denman, Marius Bauza, Brian Barbarits, Albert K Lee, Costas A Anastassiou, Alexandru Andrei, Çağatay Aydin, et al. Fully integrated silicon probes for high-density recording of neural activity. *Nature*, 551(7679):232–236, 2017.

[47] Xin Li, Ronald D. Blanton, Pulkit Grover, and Donald E. Thomas. Ultra-low-power biomedical circuit design and optimization: Catching the don't cares. In *2014 International Symposium on Integrated Circuits (ISIC)*, pages 115–118, December 2014.

[48] Xiao-Hui Liu, Yu Cao, Bao-Qing Ma, and Bin He. Deep learning for brain-computer interfaces: A comprehensive review. *arXiv preprint arXiv:2009.04279*, 2020.

[49] Fabien Lotte. Signal processing approaches to minimize or suppress calibration time in oscillatory activity-based brain–computer interfaces. *Proceedings of the IEEE*, 103(6):871–890, 2015.

[50] Maneeshika M. Madduri, Momona Yamagami, Si Jia Li, Sasha Burckhardt, Samuel A. Burden, and Amy L. Orsborn. Predicting and Shaping Human-Machine Interactions in Closed-loop, Co-adaptive Neural Interfaces, May 2024.

[51] Maneeshika M. Madduri, Momona Yamagami, Augusto X. T. Millevolte, Si Jia Li, Sasha N. Burckhardt, Samuel A. Burden, and Amy L. Orsborn. Co-Adaptive Myoelectric Interface for Continuous Control*. *IFAC-PapersOnLine*, 55(41):95–100, 2022.

[52] David A Markowitz, YT Wong, CM Gray, and B Pesaran. Mesoscopic cortical dynamics during continuous motor control. *Journal of Neuroscience*, 31, 2011.

[53] Edwin M. Maynard, Craig T. Nordhausen, and Richard A. Normann. The Utah intracortical electrode array: a recording structure for potential brain-computer interfaces. *Electroencephalography and Clinical Neurophysiology*, 102(3):228–239, 1997.

[54] Sean L Metzger, Kaylo T Littlejohn, Alexander B Silva, David A Moses, Margaret P Seaton, Ran Wang, Maximilian E Dougherty, Jessie R Liu, Peter Wu, Michael A Berger, et al. A high-performance neuroprosthesis for speech decoding and avatar control. *Nature*, 620(7976):1037–1046, 2023.

[55] Partha P Mitra and Bijan Pesaran. Analysis of dynamic brain imaging data. *Biophysical journal*, 76(2):691–708, 1999.

[56] Simon Musall, Matthew T. Kaufman, Ashley L. Juavinett, Steven Gluf, and Anne K. Churchland. Single-trial neural dynamics are dominated by richly varied movements. *Nature Neuroscience*, 22(10):1677–1686, October 2019.

[57] Elon Musk. An integrated brain-machine interface platform with thousands of channels. *bioRxiv*, August 2019.

[58] Stephanie N Naufel, Matthew F Glaser, Konrad P Kording, Eric J Perreault, and Lee E Miller. Muscle-like kinetics from static firing rates in motor cortex. *Journal of Neuroscience*, 39(17):3266–3283, 2019.

[59] Arto Nurmikko. Challenges for Large-Scale Cortical Interfaces. *Neuron*, 108(2):259–269, October 2020.

[60] A. L. Orsborn, S. Dangi, H. G. Moorman, and J. M. Carmena. Closed-Loop Decoder Adaptation on Intermediate Time-Scales Facilitates Rapid BMI Performance Improvements Independent of Decoder Initialization Conditions. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 20(4):468–477, July 2012.

[61] Amy L. Orsborn, Charles Wang, Ken Chiang, Michel M. Maharbiz, Jonathan Viventi, and Bijan Pesaran. Semi-chronic chamber system for simultaneous subdural electrocorticography, local field potentials, and spike recordings. In *2015 7th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 398–401, April 2015.

[62] Amy L. Orsborn, Helene G. Moorman, Simon A. Overduin, Maryam M. Shanechi, Dragan F. Dimitrov, and Jose M. Carmena. Closed-Loop Decoder Adaptation Shapes Neural Plasticity for Skillful Neuroprosthetic Control. *Neuron*, 82(6):1380–1393, June 2014.

[63] Tomohiro Ouchi, Leo R. Scholl, Pavithra Rajeswaran, Ryan A. Canfield, Lydia I. Smith, and Amy L. Orsborn. Mapping Eye, Arm, and Reward Information in Frontal Motor Cortices Using Electrocorticography in Nonhuman Primates. *Journal of Neuroscience*, 45(12), March 2025.

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

[65] H Parikh et al. Layer-dependence of high-gamma power in the human cerebral cortex. *Journal of Neuroscience*, 2009.

[66] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005.

[67] János A Perge, Mark L Homer, Wasim Q Malik, Sydney Cash, Emad Eskandar, Gerhard Friehs, John P Donoghue, and Leigh R Hochberg. Intra-day signal instabilities affect decoding performance in an intracortical neural interface system. *Journal of Neural Engineering*, 10(3):036004, June 2013.

[68] Gert Pfurtscheller, Teodoro Solis-Escalante, Rupert Ortner, P Linortner, and Gernot R Müller-Putz. The hybrid bci. *Frontiers in neuroscience*, 4:3, 2010.

[69] Tsam Kiu Pun, Thomas Hosman, Anastasia Kapitonava, John D Simeral, and Leigh R Hochberg. Months-long high-performance fixed LSTM decoder for cursor

control in human intracortical brain-computer interfaces. In *2023 11th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 1–5. IEEE, 2023.

[70] Supratim Ray, Nathan E Crone, Ernst Niebur, Piotr J Franaszczuk, and Steven S Hsiao. Neural correlates of high-gamma oscillations (60–200 hz) in macaque local field potentials and their potential implications in electrocorticography. *Journal of Neuroscience*, 28(45):11526–11536, 2008.

[71] Supratim Ray, Nathan E. Crone, Ernst Niebur, Piotr J. Franaszczuk, and Steven S. Hsiao. Neural Correlates of High-Gamma Oscillations (60–200 Hz) in Macaque Local Field Potentials and Their Potential Implications in Electrocorticography. *Journal of Neuroscience*, 28(45):11526–11536, November 2008.

[72] Stefano Recanatesi, Matthew Farrell, Guillaume Lajoie, Sophie Deneve, Mattia Rigotti, and Eric Shea-Brown. Predictive learning as a network mechanism for extracting low-dimensional latent space representations. *Nature Communications*, 12(1):1417, March 2021.

[73] Jörn Rickert, Simone Cardoso De Oliveira, Eilon Vaadia, Ad Aertsen, Stefan Rotter, and Carsten Mehring. Encoding of Movement Direction in Different Frequency Ranges of Motor Cortical Local Field Potentials. *Journal of Neuroscience*, 25(39):8815–8824, September 2005.

[74] Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller. Explainable ai: interpreting, explaining and visualizing deep learning. *Lecture Notes in Artificial Intelligence*, 11700, 2019.

[75] Katherine W Scangos, Ankit N Khambhati, Patrick M Daly, Ghassan S Makhoul, Leo P Sugrue, Hani Zamanian, Tony X Liu, Vikram R Rao, Kristin K Sellers, Heather E Dawes, et al. Closed-loop neuromodulation in an individual with treatment-resistant depression. *Nature Medicine*, 27(10):1696–1700, 2021.

[76] M. H. Schieber. Constraints on somatotopic organization in the primary motor cortex. *Journal of Neurophysiology*, 86(5):2125–2143, November 2001.

[77] Maryam M. Shanechi. Brain–Machine Interface Control Algorithms. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(10):1725–1734, October 2017.

[78] Maryam M. Shanechi, Amy L. Orsborn, and Jose M. Carmena. Robust Brain-Machine Interface Design Using Optimal Feedback Control Modeling and Adaptive Point Process Filtering. *PLOS Computational Biology*, 12(4):e1004730, April 2016.

[79] Krishna V. Shenoy and Jose M. Carmena. Combining Decoder Design and Neural Adaptation in Brain-Machine Interfaces. *Neuron*, 84(4):665–680, November 2014.

[80] Daniel B. Silversmith, Reza Abiri, Nicholas F. Hardy, Nikhilesh Natraj, Adelyn Tu-Chan, Edward F. Chang, and Karunesh Ganguly. Plug-and-play control of a brain-computer interface through neural map stabilization. *Nature Biotechnology*, 39(3):326–335, September 2020.

[81] Marc W. Slutzky and Robert D. Flint. Physiological properties of brain-machine interface input signals. *Journal of Neurophysiology*, 118(2):1329–1343, August 2017.

[82] Marc W. Slutzky and Robert D. Flint. Physiological properties of brain-machine interface input signals. *Journal of Neurophysiology*, 118(2):1329–1343, August 2017.

[83] Olaf Sporns. Brain connectivity. *Scholarpedia*, 2(10):4695, 2007.

[84] Nicholas A. Steinmetz, Cagatay Aydin, Anna Lebedeva, Michael Okun, Marius Pachitariu, Marius Bauza, Maxime Beau, Jai Bhagat, Claudia Böhm, Martijn Broux, Susu Chen, Jennifer Colonell, Richard J. Gardner, Bill Karsh, Fabian Kloosterman, Dimitar Kostadinov, Carolina Mora-Lopez, John O'Callaghan, Junchol Park, Jan Putzeys, Britton Sauerbrei, Rik J. J. van Daal, Abraham Z. Vollan, Shiwei Wang, Marleen Welkenhuysen, Zhiwen Ye, Joshua T. Dudman, Barundeb Dutta, Adam W. Hantman, Kenneth D. Harris, Albert K. Lee, Edvard I. Moser, John O'Keefe, Alfonso Renart, Karel Svoboda, Michael Häusser, Sebastian Haesler, Matteo Carandini, and Timothy D. Harris. Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings. *Science*, 372(6539):eabf4588, April 2021.

[85] Ian H Stevenson and Konrad P Kording. How advances in neural recording affect data analysis. *Nature Neuroscience*, 14(2):139–142, 2011.

[86] Ian H Stevenson and Konrad P Kording. How advances in neural recording affect data analysis. *Nature neuroscience*, 14(2):139–142, February 2011.

[87] D. M. Taylor. Direct Cortical Control of 3D Neuroprosthetic Devices. *Science*, 296(5574):1829–1832, June 2002.

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

[89] Michael Trumpis, Michele Insanally, Adnan Trakic, Jonathan Wang, Karel Svoboda, Robert Froemke, and Jonathan Viventi. Sufficient sampling for kriging prediction of cortical potential in rat, monkey, and human μECoG. *Journal of Neural Engineering*, 18(3):036012, 2021.

[90] Anne E. Urai, Brent Doiron, Andrew M. Leifer, and Anne K. Churchland. Large-scale neural recordings call for new insights to link brain and behavior. *Nature Neuroscience*, 25(1):11–19, January 2022.

[91] Martijn P Van Den Heuvel and Olaf Sporns. Rich-club organization of the human connectome. *Journal of Neuroscience*, 31(44):15775–15786, 2011.

[92] Yang Wang, Xinze Yang, Xiwen Zhang, Yijun Wang, and Weihua Pei. Implantable intracortical microelectrodes: reviewing the present with a focus on the future. *Microsystems & Nanoengineering*, 9(1):1–17, January 2023.

[93] Francis R. Willett, Donald T. Avansino, Leigh R. Hochberg, Jaimie M. Henderson, and Krishna V. Shenoy. High-performance brain-to-text communication via handwriting. *Nature*, 593(7858):249–254, May 2021.

[94] Francis R. Willett, Erin M. Kunz, Chaofei Fan, Donald T. Avansino, Guy H. Wilson, Eun Young Choi, Foram Kamdar, Matthew F. Glasser, Leigh R. Hochberg, Shaul

Druckmann, Krishna V. Shenoy, and Jaimie M. Henderson. A high-performance speech neuroprosthesis. *Nature*, 620(7976):1031–1036, 2023.

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

[96] J. Adam Wilson, Jürgen Mellinger, Gerwin Schalk, and Justin Williams. A Procedure for Measuring Latencies in Brain-Computer Interfaces. *IEEE transactions on biomedical engineering*, 57(7):1785–1797, July 2010.

[97] Jonathan R Wolpaw, Niels Birbaumer, Dennis J McFarland, Gert Pfurtscheller, and Theresa M Vaughan. Brain-computer interfaces for communication and control. *Clinical Neurophysiology*, 113(6):767–791, 2002.

[98] Shizhe Wu, Kinkini Bhadra, Anne-Lise Giraud, and Silvia Marchesotti. Adaptive LDA Classifier Enhances Real-Time Control of an EEG Brain–Computer Interface for Decoding Imagined Syllables. *Brain Sciences*, 14(3):196, March 2024. Number: 3.

[99] Wei Wu, Michael J. Black, David Mumford, Yun Gao, Elie Bienenstock, and John P. Donoghue. Neural decoding of cursor motion using a kalman filter. In *Advances in Neural Information Processing Systems*, volume 15, pages 117–124, 2003.

[100] Wei Wu, Yun Gao, Elie Bienenstock, John P. Donoghue, and Michael J. Black. Bayesian population decoding of motor cortical activity using a Kalman filter. *Neural Computation*, 18(1):80–118, January 2006.

[101] Yang Xie, Zepeng Zhang, Muyang Liu, Jinhong Guo, Miao Xu, Lei Zhou, Lei Zhang, Lihao Yao, Xiaolin Zhou, Zhengwei Hu, Liang Ma, Xiaojian Li, Yongxiang Guo, Jiaxin Lei, Yue Cao, Milin Zhang, Huachun Wang, He Ding, Xin Fu, Lan Yin, and Xing Sheng. High-Resolution Spatial Mapping of Electrocorticographic Activities

with a 4096-Channel, Multiplexed Thin-Film Transistor Array, August 2024. Pages: 2024.08.18.608446 Section: New Results.