Imperial College London
Department of Electrical and Electronic Engineering

Signal Processing and Deep Learning Methods for Robust and Accurate Intracortical Brain-Machine Interfaces

Nur Ahmadi

Supervised by:
Dr. Timothy G. Constandinou
Dr. Christos-Savvas Bouganis

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Electrical and Electronic Engineering of Imperial College London and
the Diploma of Imperial College London

July 2020

Declaration

I herewith certify that the work presented in this thesis is my original own. All material in this thesis which is not my own work has been properly acknowledged.

Nur Ahmadi

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial-No Derivatives 4.0 International Licence (CC BY-NC-ND). Under this licence, you may copy and redistribute the material in any medium or format on the condition that; you credit the author, do not use it for commercial purposes and do not distribute modified versions of the work. When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

To my beloved parents, my wife, and two little kids: Naila and Syaffa...

Knowledge is not merely what is memorised. Knowledge is what benefits.

— Imam Shafi'i

Abstract

Brain-machine interfaces (BMIs) have emerged as a promising assistive technology for restoring lost motor functions in individuals with neurological disorders (e.g. spinal cord injury, amyotrophic lateral sclerosis, stroke) by allowing them to control external devices directly with their thoughts. The last two decades have seen impressive progress in research and pilot clinical trials of BMIs using intracortical microelectrode arrays. Most high-performance BMIs use well-isolated neuronal action potentials (spikes) also known as single unit activity (SUA) as the input signal. These SUA-based BMIs have been shown to suffer from signal instability and longevity, therefore hampering their translation into widespread clinical use.

To mitigate this problem, numerous studies have proposed two alternative neural signal inputs: multiunit activity (MUA) and local field potential (LFP). MUA represents the aggregate spikes from an ensemble of neurons, whereas LFP is thought to mainly reflect summed synaptic activity from a local population of neurons around the recording electrode. Despite offering better signal stability and longevity, MUA- and LFP- based BMIs have been shown by several studies to yield lower decoding accuracy than that of SUA-based BMIs. Thus, developing robust and accurate BMIs remains a major challenge.

This thesis addresses the aforementioned challenge from neural signal processing and decoding points of view and is composed of three parts. The first part focuses on our efforts to push the performance of MUA-based BMIs forward. Specifically, we propose an adaptive firing rate estimation method as feature extraction and evaluate its impact on the decoding performance. The second part describes our attempts to improve the performance of LFP-based BMIs further. Particularly, we develop decoding algorithms based on two variants of deep learning methods, namely long short-term memory (LSTM) and temporal convolutional network (TCN), and assess their decoding performance. In the last part of this thesis, we explore another alternative neural signal referred to as entire spiking activity (ESA) and different deep learning architectures to improve both the robustness and accuracy of decoding. We then compare the decoding performance of ESA-based BMIs with that of SUA-, MUA-, and LFP-based BMIs described in the previous parts. Furthermore, we investigate the relationship between ESA and LFP by examining whether ESA can be solely inferred from LFP with high accuracy. Overall, this thesis offers a new approach to advancing BMI research and gives insights into signal processing and decoding design consideration towards clinically viable translation of BMIs.

Acknowledgements

First of all, alhamdulillah, all the praises and thanks be to Allah. Next, I would like to thank my supervisors, Dr. Timothy Constandinou and Dr. Christos Bouganis, for the support, guidance, and encouragement. I am very grateful for the opportunity, optimism, and trust they have given me. I see them as role models and inspiration whom I have learnt a lot from.

I would also like to thank Iza and Wiesia for all their support throughout my PhD journey. I also extend my thanks to all the members of the Centre for Bio-Inspired Technology (CBIT) and Next Generation Neural Interfaces (NGNI) Lab for all the memories we shared together. This place is so special and full of inspiration and motivation. I have been so fortunate to be part of such a lovely and amazing research group.

I would like to thank Dr. Dan Goodman and Dr. Kianoush Nazarpour for their constructive feedback and insightful comments on my thesis. I also want to thank all my neighbours in Nansen Village and my fellow Indonesian students/families in London, specifically at Imperial. Thanks for the friendship and unforgettable memories.

Last but not least, I would like to express my deepest gratitude to my family, especially my mother for her continuous support and prayers, my wife for always being beside me and continuously supporting and praying for me, my two little kids who have always been the source of my joy and happiness. Finally, I would like to thank LPDP for their financial support.

Contents

Abstract **ii**

Acknowledgements **iv**

1 Introduction **1**

1.1 Motivation 1

1.2 Research objectives 3

1.3 Thesis outline 5

2 Brain-Machine Interface: Background and State of the Art **8**

2.1 BMI history and overview 8

2.2 BMI components 10

2.3 BMI input signals 13

2.4 BMI decoding 16

2.4.1 Decoding paradigms 17

2.4.2 Decoding algorithms 17

I Improving Decoding Accuracy of MUA-based BMIs **21**

3 Firing Rate Estimation Using Bayesian Adaptive Kernel Smoother (BAKS) **22**

3.1 Introduction 22

3.2 Methods 25

3.2.1 Kernel-based firing rate estimation 25

3.2.2 Bayesian adaptive kernel smoother (BAKS) 26

3.2.3 Spike train generation model 31

3.3 Results . . . 34
3.3.1 Synthetic datasets . . . 34
3.3.2 Tuning the shape parameter of prior distribution . . . 37
3.3.3 Comparison with established methods . . . 38
3.3.4 Comparison across different values of intensity and frequency . . . 40
3.3.5 Comparison across different values of shape parameter . . . 42
3.3.6 Comparison across different numbers of trials . . . 43
3.3.7 Comparison across different underlying rate functions . . . 44
3.3.8 Comparison of computational complexity . . . 46
3.3.9 Application to real neural data . . . 48
3.4 Discussion . . . 52
3.5 Conclusion . . . 57

4 Enhanced MUA-based BMIs with BAKS and Deep Learning 58

4.1 Introduction . . . 58
4.2 Methods . . . 59
4.2.1 Neural recordings . . . 59
4.2.2 Behavioural task . . . 60
4.2.3 Spike signal processing . . . 61
4.2.4 Firing rate estimation . . . 61
4.2.5 Decoding algorithm . . . 62
4.2.6 Performance evaluation and metrics . . . 66
4.3 Results . . . 67
4.3.1 Performance comparison across firing rate estimation methods . . . 67
4.3.2 Impact of window width on decoding performance . . . 70
4.3.3 Performance comparison across decoding algorithms . . . 72
4.3.4 Decoding performance comparison between SUA and MUA . . . 73
4.4 Discussion . . . 75
4.5 Conclusion . . . 76

II Improving Decoding Accuracy of LFP-based BMIs 77

5 Decoding Hand Kinematics from LFPs Using Deep Learning 78

5.1 Introduction . . . 78

5.2 Methods . . . 79

5.2.1 Neural recording and behavioural task . . . 80

5.2.2 Neural signal processing . . . 80

5.2.3 Feature extraction . . . 81

5.2.4 Decoding algorithm . . . 82

5.2.5 Performance evaluation and metrics . . . 84

5.3 Results . . . 85

5.3.1 Performance comparison across different LFP features . . . 85

5.3.2 Impact of window width on decoding performance . . . 86

5.3.3 Performance comparison across input signals and decoders . . . 87

5.3.4 Cross-correlation between learnt and hand-crafted LFP features . . . 89

5.3.5 Inference time and power consumption . . . 90

5.4 Discussion . . . 90

5.5 Conclusion . . . 92

6 Impact of Referencing Scheme on Decoding Performance of LFP-based BMI 93

6.1 Introduction . . . 93

6.2 Methods . . . 94

6.2.1 Neural recordings and Behavioural task . . . 94

6.2.2 Signal referencing . . . 95

6.2.3 Signal processing and feature extraction . . . 97

6.2.4 Decoding algorithm . . . 97

6.2.5 Performance evaluation and metrics . . . 99

6.3 Results . . . 99

6.3.1 LFP feature comparison and selection . . . 99

6.3.2 Impact of interelectrode distance in bipolar reference . . . 101

6.3.3 Impact of referencing scheme on decoding performance . . . 103
6.3.4 Impact of referencing scheme on interchannel correlation . . . 106
6.4 Discussion . . . 107
6.5 Conclusion . . . 111

III Developing Robust and Accurate BMIs 112

7 Robust and Accurate Decoding of Hand Kinematics from Entire Spiking Activity Using Deep Learning 113
7.1 Introduction . . . 113
7.2 Methods . . . 114
7.2.1 Neural recordings and behavioural task . . . 115
7.2.2 Signal processing and feature extraction . . . 115
7.2.3 Decoding methods . . . 117
7.2.4 Hyperparameter optimisation . . . 120
7.2.5 Decoder training schemes . . . 120
7.2.6 Performance evaluation and metrics . . . 121
7.3 Results . . . 122
7.3.1 Characteristic comparison across neural signals . . . 122
7.3.2 Decoding performance comparison across decoders . . . 126
7.3.3 Decoding performance comparison across neural signals . . . 127
7.3.4 Impact of different training schemes and the amount of training data . . . 131
7.4 Discussion . . . 132
7.5 Conclusion . . . 138

8 Inferring Entire Spiking Activity from LFPs with Deep Learning 139
8.1 Introduction . . . 139
8.2 Methods . . . 140
8.2.1 Neural recording and behavioural task . . . 140
8.2.2 Signal processing and feature extraction . . . 142

8.2.3 Long short-term memory (LSTM) . . . 144
8.2.4 LFP channel importance/contribution . . . 144
8.2.5 Effect of number of LFP channels on inference accuracy . . . 145
8.2.6 Performance evaluation and metrics . . . 145
8.3 Results . . . 146
8.3.1 LFP feature informativeness for ESA inference . . . 146
8.3.2 Comparison of inference performance: SUA vs MUA vs ESA . . . 147
8.3.3 LFP channel importance/contribution . . . 151
8.3.4 Effect of number of LFP channels on inference accuracy . . . 151
8.3.5 Performance valuation on different subject and task . . . 153
8.4 Discussion . . . 155
8.5 Conclusion . . . 158

9 Conclusions and Future Directions **159**
9.1 Original contributions . . . 159
9.2 Future directions . . . 161
9.3 Concluding remarks . . . 162

Appendices

A Mathematical Derivation of BAKS **164**
A.1 Posterior distribution of bandwidth . . . 164
A.2 Adaptive bandwidth estimate . . . 166

B Supplementary Figures and Tables **167**

C List of Publications **180**

Bibliography **182**

List of Figures

Figure 2.1 Timeline of major advances in the field of BMIs from 1970s to the present . 10

Figure 2.2 Block diagram of a typical BMI system 11

Figure 2.3 Different types of neural signal input for BMIs 14

Figure 2.4 Comparison of properties between different types of signal inputs 16

Figure 3.1 Graphical representation of Bayesian adaptive kernel smoother (BAKS) . . 30

Figure 3.2 Illustration of the underlying rate functions used in this chapter 35

Figure 3.3 MISE comparison under different values of prior parameter (α) 38

Figure 3.4 MISE comparison of BAKS with other methods 40

Figure 3.5 Comparison of firing rate estimates across all methods 41

Figure 3.6 MISE comparison across different values of intensity and frequency for IG model 42

Figure 3.7 MISE comparison across various values of shape parameter (γ) 43

Figure 3.8 MISE comparison across different numbers of trials 44

Figure 3.9 MISE comparison across different underlying rate functions for IG model . . 45

Figure 3.10 Firing rate estimate comparison for IG model with Gaussian-damped sinusoidal rate function 46

Figure 3.11 WMISE comparison across all firing rate estimation methods on real neural data 51

Figure 3.12 Firing rate estimates comparison for real neural data 52

Figure 4.1 Block diagram of spike-based BMI 60

Figure 4.2 Comparison of decoding performance of MUA-driven KF decoder across different firing rate estimation methods 68

Figure 4.3 Comparison of decoding performance of MUA-driven WF decoder across different firing rate estimation methods 69
Figure 4.4 Comparison of decoding performance of MUA-driven LSTM decoder across different firing rate estimation methods 71
Figure 4.5 Decoding performance comparison across different firing rate estimation methods with varying window widths 72
Figure 4.6 Comparison of decoding performance of MUA-driven decoders with firing rates computed by BAKS 73
Figure 4.7 Comparison of decoding performance between SUA- and MUA-driven decoders with firing rates computed by BAKS 74

Figure 5.1 Block diagram of LFP-based BMI using LSTM decoder 80
Figure 5.2 Schematic overview of TCN decoder 81
Figure 5.3 Decoding performance comparison across different LFP features 85
Figure 5.4 Decoding performance comparison of SUA-, MUA-, and LFP-driven LSTM decoders across different window widths. Red stars indicate window widths that yielded the best decoding performances on validation set from session I. 86
Figure 5.5 Decoding performance comparison across different decoders driven by SUA, MUA, and LFP 87
Figure 5.6 Decoding performance comparison across different decoders relative to SUA-KF 88
Figure 5.7 Examples of actual and decoded velocities from different decoders in x- and y-directions 89
Figure 5.8 Cross-correlation between TCN-learnt and hand-crafted LFP features 90

Figure 6.1 Block diagram of LFP-based BMI system with different referencing schemes 95
Figure 6.2 Decoding performance comparison across different LFP features in unipolar reference 100
Figure 6.3 Decoding performance comparison across different interelectrode distances in bipolar reference with LMP feature 101

Figure 6.4 Decoding performance comparison across different interelectrode distance in bipolar reference with gamma feature 102
Figure 6.5 Decoding performance comparison across different referencing schemes using LMP-driven LSTM decoder 104
Figure 6.6 Decoding performance comparison across different referencing schemes using gamma-driven LSTM decoder 106
Figure 6.7 Decoding performance comparison across different referencing schemes using LMP-driven GRU decoder 107
Figure 6.8 Comparison of interchannel correlation of LFPs across different referencing schemes 108

Figure 7.1 Schematic diagram of the brain-machine interface (BMI) system 115
Figure 7.2 Long-term stability of different types of neural signals 123
Figure 7.3 Inter-channel/unit correlation of neural signals over time 124
Figure 7.4 Long-term decoding performance comparison across different decoders driven by ESA signal 125
Figure 7.5 Long-term decoding performance comparison across QRNN decoders driven by different neural signals 128
Figure 7.6 Decoding performance comparison across different types of neural signals . . 130
Figure 7.7 Decoding performance comparison across decoders under different training schemes and amount of training data 132

Figure 8.1 Schematic illustration of signal processing and deep learning (DL) based inference steps 141
Figure 8.2 Performance comparison of ESA inference across different LFP features . . . 146
Figure 8.3 Comparison of inference accuracy (CC) among different types of spiking activities (SUA, MUA, and ESA) 148
Figure 8.4 Comparison of inference accuracy measured with RMSE metric 150
Figure 8.5 The importance of each LFP channel and the effect of the number of LFP channels to inference accuracy 152

Figure 8.6 Performance evaluation on different subject and task 155

Figure B.1 MISE comparison under different values of intensity and frequency for IIG model 167

Figure B.2 MISE comparison under different values of shape parameter 168

Figure B.3 MISE comparison for IIG model with Gaussian-damped sinusoidal rate function 168

Figure B.4 Firing rate estimate comparison for IIG model with Gaussian-damped sinusoidal rate function 169

Figure B.5 MISE comparison for IG model with square rate function 170

Figure B.6 Firing rate estimate comparison for IG model with square rate function . . 171

Figure B.7 Graphical illustration of the deep learning decoders 171

Figure B.8 Visualisation of hyperparameter optimisation result for ESA-driven deep learning decoders 172

Figure B.9 Long-term decoding performance comparison across different decoders driven by SUA signal 173

Figure B.10 Long-term decoding performance comparison across different decoders driven by MUA signal 174

Figure B.11 Long-term decoding performance comparison across different decoders driven by LFP signal 175

Figure B.12 Long-term decoding performance comparison across SRNN decoders driven by different neural signals 176

Figure B.13 Long-term decoding performance comparison across GRU decoders driven by different neural signals 177

Figure B.14 Long-term decoding performance comparison across LSTM decoders driven by different neural signals 178

Figure B.15 Decoding performance comparison across different types of neural signals . . 179

List of Tables

Table 2.1 Summary of decoding algorithms used in BMI studies 20

Table 3.1 Model settings for the synthetic dataset generation 37

Table 3.2 Average runtime comparison of BAKS with other methods (in second) 48

Table 3.3 Comparison summary of BAKS against other methods 56

Table 4.1 Hyperparameter configuration of SUA- and MUA-driven LSTM decoders across firing rate estimation methods 66

Table 5.1 Hyperparameter search space for TCN and LSTM decoders 84

Table 5.2 Hyperparameter configuration of TCN and LSTM decoders 84

Table 6.1 Hyperparameter configuration of LSTM decoders across LFP features. 98

Table 6.2 Hyperparameter configuration of LSTM decoders across referencing schemes 99

Table 7.1 Hyperparameter configurations of deep learning decoders driven by ESA and QRNN decoders driven by other neural signals. 122

Table 8.1 Hyperparameter setting of LSTM models for inferring different types of spike signals 144

Table B.1 Hyperparameter search spaces during the optimisation of deep learning decoders. 169

Table B.2 Hyperparameter configurations for SUA-, MUA-, and LFP-driven DL decoders.169

List of Abbreviations

AI	Artificial intelligence
ALS	Amyotrophic lateral sclerosis
ANN	Artificial neural network
BAKS	Bayesian adaptive kernel smoother
BARS	Bayesian adaptive regression splines
BMI	Brain-machine interface
CAR	Common average reference
CC	Pearson's correlation coefficient
CNN	Convolutional neural network
CSD	Current source density
DBS	Deep brain stimulation
DL	Deep learning
ECoG	Electrocorticography
EEG	Electroencephalography
ESA	Entire spiking activity
GRU	Gated recurrent unit
IG	Inhomogeneous Gamma
IIG	Inhomogeneous inverse Gaussian
IP	Inhomogeneous Poisson
ISI	Interspike interval

KF	Kalman filter
LFP	Local field potential
LMP	Local motor potential
Locfit	Local polynomial fit
LSTM	Long short-term memory
MEA	Multielectrode array
MISE	Mean integrated squared error
MUA	Multiunit activity
NHP	Nonhuman primate
OKS	Optimised kernel smoother
OLE	Optimal linear estimator
PVA	Population vector algorithm
QRNN	Quasi-recurrent neural network
RMSE	Root mean squared error
RNN	Recurrent neural network
SCI	Spinal cord injury
SRNN	Standard recurrent neural network
STFT	Short-time Fourier transform
SUA	Single-unit activity
TCN	Temporal convolutional network
UKF	Unscented Kalman filter
VKS	Variable kernel smoother
WCF	Wiener cascade filter
WF	Wiener filter

1 | Introduction

1.1 Motivation

Millions of people around the world live with partial or complete paralysis due to neurological disorders such as spinal cord injury (SCI), amyotrophic lateral sclerosis (ALS), and stroke [1]. Paralysis conditions can severely impact the quality of life of the patients and their families. Besides, the immediate and long-term care of the patients can represent a substantial economic burden for societies as well as public and private health care services. For example, there are approximately 1,270 to 2,500 new SCI patients every year in the UK, corresponding to lifetime direct and indirect costs (including initial hospitalisation) of around £ 1.43 billion to £ 3.7 billion [2], [3]. Combined with other neurological disorders, the total lifetime costs become dramatically higher. Therefore, it is of great importance to address the psychological, societal, and economic issues associated with paralysed patients.

Brain-machine interfaces (BMIs) have emerged as a potential solution to address the issues mentioned above by restoring lost motor functions in paralysed patients, providing them with some degree of independence without the need for specialist care services. According to surveys on paralysed patients [4]–[6], hand movement and grasp are the highest priority of functions that they desire to be restored. Towards this direction, substantial progress has been made in the field of BMIs over the past two decades. Research and early clinical studies have demonstrated that nonhuman primate (NHP) and human subjects were able to control external devices such as a computer cursor [7]–[13] and robotic arm [14]–[18] to accomplish certain tasks. These achievements are enabled partly by the significant advances in intracortical microelectrode technologies that allow simultaneous recordings of action potential (spike) activity from large numbers of single neurons. This spiking activity, also known as single-unit activity

(SUA), has been predominantly used as the input signal for a decoder, the essential element in a BMI system that translates neural activity into motor commands. However, despite compelling results from animal experiments and pilot human clinical trials, SUA suffers from short- and long-term instability which hampers the clinical translation of SUA-based BMIs [19]–[23]. The quality and number of recorded spikes gradually decline over time, which can degrade the decoding performance. These adverse effects are thought to be primarily caused by the foreign body response inducing glial scarring around the electrodes, micromotion of the electrodes, and degradation of insulation properties [19], [23]–[25]. Moreover, due to high sampling rate (at least 16 kHz) and complexity of spike sorting —a process of classifying spikes into distinct putative single units— on-chip spike processing or wireless transmission of the raw neural signals for off-chip spike processing requires high power consumption [25]. This can, in turn, generate more heat to the neural tissue and limit the device's battery lifespan.

To mitigate the signal instability problem of SUA, multiunit activity (MUA) has been proposed in the literature as an alternative input signal. MUA represents the aggregate spiking activity from a local ensemble of neurons and can be obtained by spike detection (through threshold crossing technique) without requiring spike sorting. It provides better signal stability while offering simpler signal processing than SUA [19], [26], [27]. An alternative approach is to utilise the local field potential (LFP) signals, thought to mainly reflect the summed synaptic activity from a population of neurons near the recording electrodes [28]–[30]. It has been postulated and demonstrated through experimental studies that LFPs exhibit long-term signal stability [31]–[34] and retain movement-related information even when the spikes are lost [22], [35]–[37]. Furthermore, LFPs can be extracted with a simple low pass filter and sampled at lower rates, hence reducing power consumption. While there are appealing advantages of MUA and LFP, a considerable number of published papers have reported lower decoding accuracy of MUA [22], [26], [27], [38], [39] and LFP [33], [35], [39]–[41] compared to that of SUA. Thus, it is critical to develop a BMI system that provides high decoding accuracy and can maintain its accuracy over a long period of time (i.e. robust).

Decoding accuracy is heavily influenced by the choice of the decoding algorithm. A vast majority of BMI studies employ linear decoders, such as Wiener filter (WF) [7], [8], [11],

[14], [15], [42], [43] and Kalman filter (KF) [12], [13], [44]–[48], which may not be optimal in the face of nonlinear and nonstationary neural signals [49]. Despite the fact that there exist nonlinear extensions of both algorithms, e.g. Wiener cascade filter (WCF) [35], [50]–[53] and unscented Kalman filter (UKF) [54]–[57], the optimality of WCF and UKF are based on the assumptions of stationary (additive) and Gaussian noises, respectively. Any violation of these a priori assumptions would lead to degraded decoding performance. With the re-emergence of artificial intelligence (AI) in recent years, particularly deep learning, we have seen its massive adoption in a wide variety of applications (e.g. speech recognition, language translation, and medical diagnostics) with state-of-the-art performances [58], [59]. So far, however, there have been few studies in the field of BMIs that utilise deep learning as the decoding algorithm [60]–[62]. This opens new avenues for the development of a deep learning based decoder to improve the decoding accuracy of BMIs. In other domains such as image and speech recognition, the data have a fixed number of channel input, pixel array size (image) or sampling rate (speech). In the field of BMIs, however, there is a lack of standardisation in neural dataset. Each dataset is different since not only the underlying tissue but also the electrode array location/proximity is different. It is thus challenging to leverage on big data, a key enabler of AI methods, for BMI applications.

1.2 Research objectives

This thesis aims to develop neural signal processing and decoding methods for robust and accurate BMIs. We use an existing, freely-available intracortical neural data chronically recorded from the primary motor cortex area of a monkey while he was performing self-paced reaching tasks. We evaluate our proposed methods for offline decoding of hand kinematics, specifically the fingertip velocity. Decoding accuracy is quantified in terms of shape similarity and error magnitude computed between the actual and decoded hand kinematics. Robustness is defined as the sustained decoding accuracy over a long period of time. To achieve this ultimate aim, we set three main objectives as follows:

- *To improve decoding accuracy of MUA-based BMIs.* Although many studies have suggested that MUA is more robust than SUA, the decoding accuracy of MUA is usually

lower than that of SUA. The decoding accuracy is affected significantly by feature extraction and decoding algorithms. Most MUA-based BMIs employ a simple binning method to extract firing rate estimates, that is, by counting the number of spikes within a predefined window size. Regarding the decoding algorithm, they still primarily use linear methods, such as Wiener filter and Kalman filter. These linear decoders could yield suboptimal performance since real neural signals often exhibit nonlinear, nonstationary, and non-Gaussian noise properties, violating the decoders' inherent assumptions. Therefore, it is necessary to develop a new feature extraction and decoding algorithm to improve the decoding accuracy of MUA-based BMIs.

- *To improve decoding accuracy of LFP-based BMIs.* There are a plethora of studies demonstrating the robustness of LFPs; unfortunately, these have not frequently been accompanied by high decoding accuracy. Since LFPs are often contaminated by common (correlated) noise, we need to perform noise removal in the preprocessing step. Before extracting the features, LFPs are typically divided into several distinct frequency bands; for example, delta, theta, alpha, beta, and gamma bands. Power spectra computed from these bands can then be used as features. However, there is no consensus on how the frequency range for each band is specified and which band carries the highest movement-related information. It is thus crucial to investigate which feature is most informative (i.e. predictive) for decoding tasks. Additionally, it is of great interest to combine feature extraction and decoding algorithm into a single unified approach. This is aimed at eliminating the need for a tedious process of manual feature extraction while at the same time improving the decoding accuracy of LFP-based BMIs.

- *To explore another alternative input signal and decoding algorithm for robust and accurate BMIs.* We further attempt to address the issues of robustness and accuracy by simultaneously exploring another alternative input signal and decoding algorithm. The new input signal and decoding algorithm will be benchmarked against the existing methods, including our previously proposed methods associated with the two objectives above. In addition, it is beneficial to examine the relationship between the new input signal and other commonly used input signals (SUA, MUA, and LFP).

1.3 Thesis outline

This thesis is composed of three parts corresponding to the three aforementioned objectives. Each part comprises two technical chapters. The overall structure this thesis is organised as follows:

- Chapter 2 | Brain-Machine Interface: Background and State of the Art

  This chapter provides the necessary background on brain-machine interfaces (BMIs) focusing especially on neural signal processing and decoding methods. It starts with a brief history of BMIs, followed by an overview of BMI components. It continues with a description of different types of BMI input signals along with their advantages and challenges. Lastly, it reviews different decoding approaches and algorithms with a highlight on the current state of the art.

Part I Improving Decoding Accuracy of MUA-based BMIs

- Chapter 3 | Firing Rate Estimation Using Bayesian Adaptive Kernel Smoother (BAKS)

  This chapter addresses the challenges in estimating firing rate from single-trial spike trains as commonly performed during feature extraction step in spike-based BMIs. We present a new method based on kernel smoothing technique, referred to as Bayesian adaptive kernel smoother (BAKS), which considers the bandwidth parameter as a random variable with a prior distribution that is adaptively updated under an empirical Bayesian framework. We evaluate the performance of BAKS on both synthetic and real spike trains and demonstrate improved performance compared to other methods.

- Chapter 4 | Enhanced MUA-based BMIs with BAKS and Deep Learning

  This chapter approaches the decoding accuracy issue of MUA-based BMIs from both feature extraction and decoding algorithm perspectives. Many studies employ a simple binning method for the former and linear decoder for the latter. Here, we apply BAKS (introduced in Chapter 3) for extracting features (i.e. firing rate) and propose a deep learning decoder based on long short-term memory (LSTM). We show that the proposed methods yield better decoding accuracy than the conventional methods.

Part II Improving Decoding Accuracy of LFP-based BMIs

- Chapter 5 | Decoding Hand Kinematics from LFPs Using Deep Learning

  This chapter proposes decoding algorithms based on two different deep learning techniques, namely recurrent neural network (RNN) and convolutional neural network (CNN). Particularly, we utilise an RNN with LSTM architecture (described in Chapter 4) driven by features that are extracted manually. Next, we exploit a special type of CNNs called temporal convolutional network (TCN) to learn the features automatically and decode hand kinematics from LFPs in a single unified approach. By doing so, we can eliminate the time-consuming process of manual feature extraction and learn features that are optimised for the decoding tasks. These proposed methods are shown to achieve better decoding accuracy than a commonly used Kalman filter. We further analyse features automatically learnt from TCN and features extracted from manual process.

- Chapter 6 | Impact of Referencing Scheme on Decoding Performance of LFP-based BMI

  This chapter considers the problem of common noise that often contaminates LFPs when recorded with respect to a unipolar reference. Several referencing schemes have been proposed to remove common noise, such as bipolar reference, current source density (CSD), and common average reference (CAR). We investigate the impact of these referencing schemes on decoding performance of LFP-based BMI. We also examine the impact of interelectrode distance and interchannel correlation on decoding performance. The performance evaluation is undertaken across long-term recording sessions using different features and deep learning decoders.

Part III Developing Robust and Accurate BMIs

- Chapter 7 | Robust and Accurate Decoding of Hand Kinematics from Entire Spiking Activity Using Deep Learning

  This chapter aims to improve both robustness and accuracy of BMIs by exploring a new, alternative input signal and decoding algorithm. We propose entire spiking activity (ESA) as the input signal and quasi-recurrent neural network (QRNN) as the decoder. The proposed ESA-driven QRNN decoder is compared against SUA-, MUA-, and LFP-driven

decoders developed in the previous chapters. We also examine the stability characteristics of each input signal over time. Moreover, we assess the impact of different decoder's training schemes and the amount of training data on decoding performance.

- Chapter 8 | Inferring Entire Spiking Activity from LFPs with Deep Learning

  This chapter attempts to shed light on the relationship between ESA and LFP by investigating whether we can infer ESA solely from simultaneously recorded LFPs. We employ an LSTM based deep learning as an inference method and evaluate it on neural recordings from monkeys' motor cortex during point-to-point and instructed delay reach-to-grasp tasks. We analyse which LFP features are most informative about ESA. Next, we compare the inference accuracy of ESA with that of SUA and MUA. Finally, we examine the contribution of each LFP channel and the effect of the number of LFP channels on the inference accuracy.

- Chapter 9 | Conclusions and Future Directions

  This chapter concludes the works presented in this thesis along with a highlight of original contributions. Lastly, it discusses potential directions for future work.

2 | Brain-Machine Interface: Background and State of the Art

2.1 BMI history and overview

Brain-machine interfaces (BMIs), also referred to as brain-computer interfaces (BCIs), are systems that enable users to interact with the environment using assistive devices controlled directly by their brain activity alone. BMIs hold great promise for restoring lost motor functions of patients suffering from neurological disorders such as spinal cord injury (SCI), amyotrophic lateral sclerosis (ALS), and stroke. The foundations of BMI research can be traced back to a series of experiments carried out in the late 1960s. Evarts showed that single-neuron firing rates recorded from the motor cortex of monkeys were strongly correlated with the force of flexion-extension movements of the wrist [63]. In 1969, Fetz performed a pioneering study on operant conditioning, also known as neurofeedback or biofeedback, using an invasive approach [64]. He demonstrated that monkeys could volitionally modulate the firing rates of individual neurons from precentral motor cortex using visual biofeedback and food reinforcement. A year later, Humphrey *et al.* demonstrated for the first time that an ensemble of individual neurons in the motor cortex of unanesthetised monkey could be used to predict the kinetic and kinematic parameters of the arm movements [65]. Inspired by these studies, Vidal published a paper in 1973, introducing the idea of BMI for communicating with computers or controlling prosthetic devices. This idea first came into practice a few years later when he conducted an experiment to control a cursor-like graphical object displayed on a computer monitor using electroencephalographic (EEG) signals [66]. In the 1980s, Georgopoulos and colleagues proposed the concept of "preferred direction" where each neuron fires more frequently towards a

particular direction than towards other directions of arm movements [67]. They also proposed a population vector algorithm (PVA) that predicts the direction of arm movement by summing the preferred directions of a neuronal population, weighted by the associated firing rates [68]. These early works established the basis for further BMI research and clinical trials.

BMI research experienced rapid growth around the mid-1990s thanks to the advances in intracortical microelectrode technologies, such as silicon-based Michigan probe [69] (initially designed in 1970s [70]) and silicon-based Utah array [71], which allow simultaneous recording of extracellular activity from many individual neurons. The first human trial of invasive BMIs (with a neurotrophic electrode) was reported by Kennedy and Bakay [72], suggesting that patients with ALS could restore their paralysed muscles through muscle stimulators controlled by their neural signals. More sophisticated intracortical BMIs were then developed in animals to control a robotic arm [7], [14], [73] or a computer cursor [8], [9], [42]. To advance BMI research and translate them into pilot human clinical trials, the BrainGate project was launched in 2004 [74]. This project has resulted in several significant advancements in the field of BMIs, including the demonstrations of paralysed patients controlling a computer cursor [15], a robotic arm [75], or functional electrical stimulation systems [76]. Other research groups reported that a patient with tetraplegia was able to continuously control a prosthetic arm with many degree-of-freedoms (DOFs), representing the current state-of-the-art performance [16], [17]. See Lebedev and Nicolelis' paper for a comprehensive review on BMIs from the fundamental science to the clinical applications [77] and Brandman *et al.*'s paper for a thorough review on intracortical BMIs in humans [78].

The past two decades have seen significant progress in several important aspects of BMIs. For example, local field potentials (LFPs) have received increasing attention as an alternative neural signal input for BMIs [31], [79] and have been demonstrated to be more robust than spikes over long-term periods [22], [52]. With regard to BMI decoding, a number of improved algorithms over PVA have been proposed, such as Wiener filter [7], Kalman filter [44], and unscented Kalman filter [54]. With the growing popularity of artificial intelligence (AI), particularly deep learning (DL), and its state-of-the-art performance across various tasks, researchers in the field of BMIs have started to adopt DL algorithms as a BMI decoder [60]. As

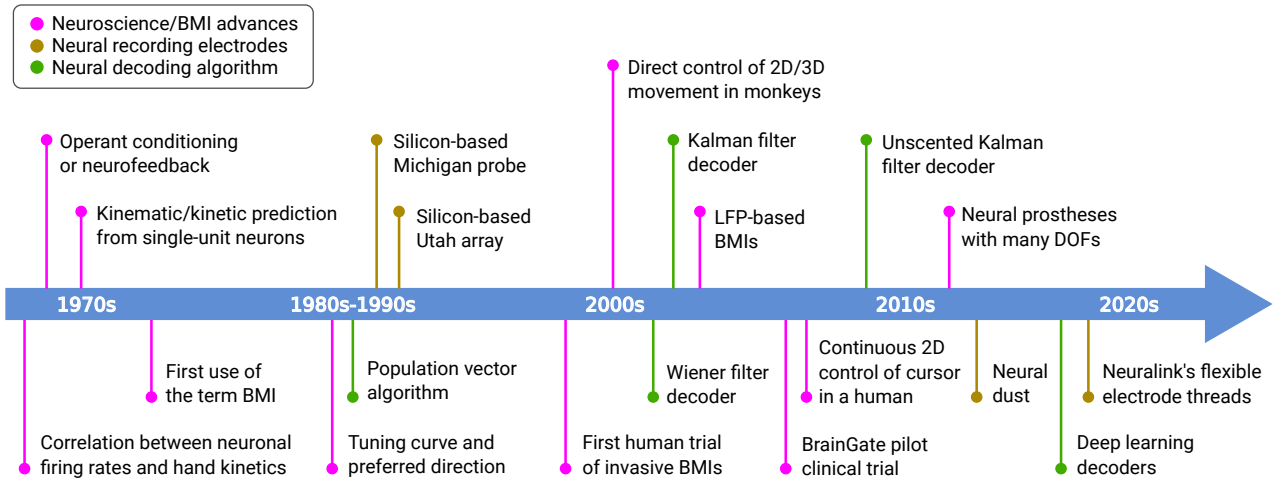


Figure 2.1 Timeline of major advances in BMIs from 1970s to the present. This timeline highlights the advances in several aspects including neuroscience/BMI research, recording electrode technology, and decoding algorithm.

for the recording technology, new microelectrodes have been developed to increase the number of channels within a single, highly complex implant (e.g. Neuropixels [80] and Neuralink's threads [81]) or distributed into multiple, simpler, and smaller implants (e.g. UC Berkeley's neural dust [82] and Imperial College's ENGINI probe [83]). For an in-depth review on recording electrode technology, refer to [84], [85]. A timeline summary of BMI major advances is presented in Figure 2.1.

2.2 BMI components

Figure 2.2 shows the block diagram of a BMI system, which typically consists of four main components, as follows:

- **Signal acquisition.** This component acquires, amplifies, and converts the neural activity into digital format data, termed as the raw neural signals, for subsequent processing. There are different types of electrodes that can be used to acquire the raw neural signals; for instance, electroencephalography (EEG), electrocorticography (ECoG), deep brain stimulation (DBS), and intracortical microelectrode array (MEA) as illustrated in Figure 2.3. The works presented in this thesis use intracortical MEA, which can record both action potentials (spikes) and LFPs. A more detailed description of these signals is provided in Section 2.3.

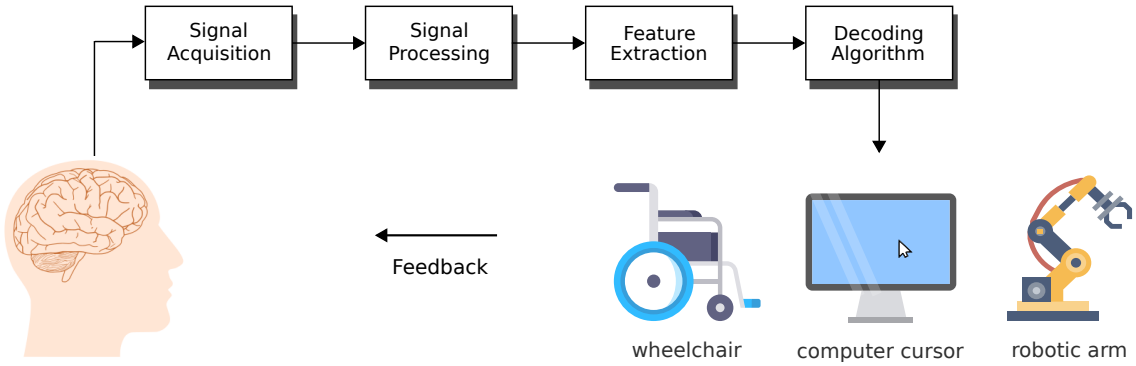


Figure 2.2 Block diagram of a typical BMI system

- **Signal processing.** To obtain spikes or LFPs, the raw neural signals have to be appropriately processed. Spikes are detected by band-pass filtering (typically between 300 Hz and 3000 Hz [86]) the raw neural signals and followed by a threshold-crossing technique, that is, by checking whether the filtered signals exceed a particular threshold value. The value of the threshold can be determined manually or automatically based on the estimate of the (filtered) signal's standard deviation multiplied by a constant (usually between 3 and 5). The detected spikes are hereinafter defined as multiunit activity (MUA), representing aggregate spiking activity from an ensemble of neurons in the vicinity of the recording electrode tip. It is common practice to classify further the detected spikes into distinct putative single neurons called single-unit activity (SUA). This process is commonly referred to as spike sorting. One of the most common and simplest approaches is by selecting templates of spike waveforms (i.e. shape) and classifying the detected spikes through template matching based on predefined criteria. A more advanced spike sorting approach extracts the informative features from spike waveforms (e.g. via principal component analysis (PCA) or wavelet decomposition) and classifies/clusters them into distinct single-neurons (e.g. via support vector machines (SVM) or super-paramagnetic clustering (SPC) algorithms) [87], [88]. For more recent progress on spike sorting methods, see [86], [89]. Unlike spike sorting which requires relatively high computational complexity, LFP processing utilises a simple low pass filter with a cut-off frequency of less than ∼300 Hz. Spikes and LFPs can be used separately as a BMI input signal or simultaneously (i.e. hybrid) to combine the benefit of each signal [37], [90].

- **Feature extraction.** This component extracts meaningful representation (i.e. features) from spikes or LFPs. The extracted features are then fed to the decoding algorithm. For spike signals, the features are commonly represented in the form of firing rates, that is, the rate or frequency at which a neuron fires spikes. The simplest and most frequently used method to estimate the firing rates is by counting the number of spikes within a predefined bin (i.e. window). There exist several more sophisticated methods to improve the estimate of firing rates, which include adaptive kernel-based smoothing techniques, Bayesian adaptive regression splines (BARS), and Gaussian process firing rates (GPFR) [91]. However, due to partly by their high complexity, these methods have not been widely adopted for BMI applications.

  As for LFP signals, the features can be extracted from time and/or frequency domain. The time-domain features include the amplitude of low-frequency LFPs [31], [33] and average amplitude extracted by a moving average filter, also known as local motor potential (LMP) [35], [41]. The frequency-domain features usually make use of average power spectra computed from narrow-band (e.g. delta, theta, alpha, beta, and gamma) or wide-band LFPs using time-frequency transformation techniques (e.g. short-time Fourier transform (STFT) and Wavelet transform) [92], [93]. One of the challenging tasks in the LFP feature extraction process is how the frequency range within each narrow-band is determined and how to select the most informative (relevant) features. The frequency range is usually determined based on the experimenter's intuition, visual inspection of the data, or information from previous studies [94]. The most informative features can be determined, for instance, through a greedy-selection procedure [33], correlation-based ranking [52] or decoding-based ranking [37].

- **Decoding.** A decoder acts as a "translator" from the extracted features into control signals for guiding assistive devices such as a robotic arm, computer cursor, or wheelchair. According to the characteristic of the estimated parameters of interest (i.e. states), BMI decoder can be categorised into discrete and continuous decoders [42], [95], [96]. Discrete decoder estimates a state into a finite number of possible choices; for example, decoding an intended direction from eight radial targets in centre-out reaching tasks [35], [42].

Continuous decoder, on the other hand, estimates a state into a continuous range of values; for instance, decoding a trajectory of hand movement [35], [42]. It is also possible to combine both discrete and continuous decoders to address a particular task such as typing through a point-and-click cursor control [97]–[103]. Kao [102] and Pandarinath [103] employed combined discrete-continuous decoders for typing tasks in non-human primate and paralysed human subjects, respectively. The continuous decoder was used to control the movement of a cursor whereas the discrete counterpart was used to select (i.e. click) the target during hold time. A brief review of decoding paradigms and algorithms are described in Section 2.4

2.3 BMI input signals

According to the level of invasiveness, BMIs can be classified into two categories: invasive and non-invasive BMIs. Invasive BMIs require surgical procedure of opening up scalp and skull and implanting the recording electrodes. This procedure poses a risk of tissue damage and/or infection. Examples of recording electrodes from this category are intracortical MEA [77], [78], DBS [104]–[106], and ECoG [107], [108]. Intracortical MEA and DBS penetrates the cortex, whereas ECoG resides above (epidural) or under (subdural) the dura mater [77], [109]. Even though intracortical MEA carries a higher risk than ECoG, it offers better spatiotemporal resolution and enables the recording of action potential (i.e. spike) activity from individual neurons. Unlike intracortical MEA, DBS electrodes have a small number of contacts (channels) and cannot record individual spikes (usually record only LFPs). While non-invasive BMIs do not carry a risk as those of the invasive counterparts, the recorded signals have a relatively low spatiotemporal resolution. One of the most widely used non-invasive BMIs is based on EEG electrodes placed on the scalp surface. An illustration of different types of recording electrodes and their recorded signals is presented in Figure 2.3.

Despite being simple, non-invasive, and useful for many simple tasks [110]–[116], EEG-based BMIs exhibit a poor spatiotemporal resolution, posing a great challenge to the development of a fast, accurate, and robust decoding performance, especially for complicated tasks (e.g. high degree-of-freedom (DOF) movement). On the other hand, intracortical MEA can record

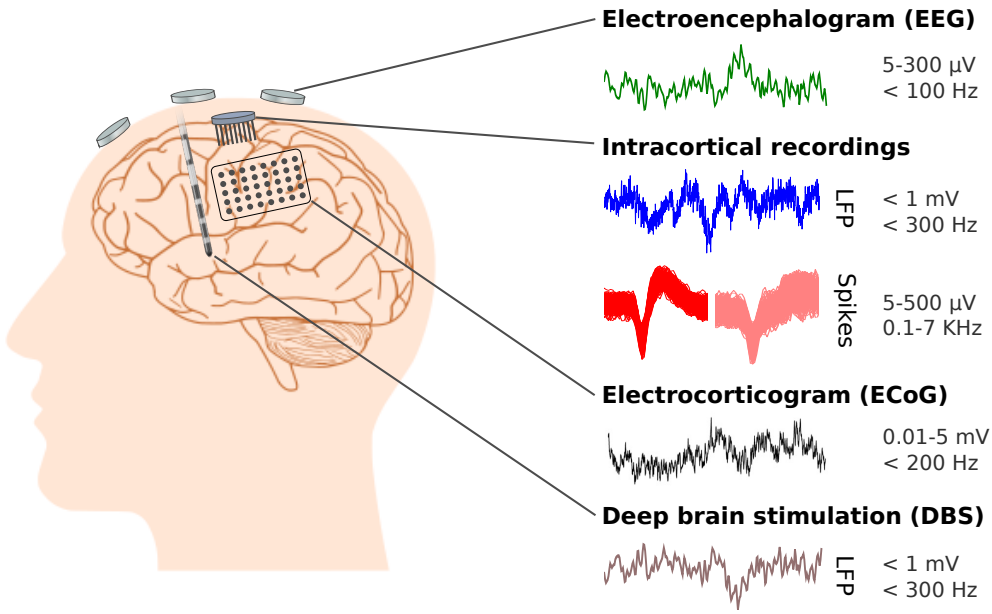


Figure 2.3 Different types of neural signal input for BMIs. EEGs are recorded noninvasively from the scalp; ECoGs are recorded semi-invasively with electrodes placed above (epidural) or below (subdural) the dura mater; LFPs and action potentials (spikes) are recorded invasively using intracortical electrodes. LFPs can also be recorded invasively using DBS electrodes placed deep in the brain.

different types of signals with high movement-related information (e.g. spikes and LFPs). This is not only beneficial for BMI studies, but also for basic neuroscience research seeking to understand how the brain works. Moreover, intracortical BMIs have been successfully demonstrated for high DOF prosthetic arm control tasks [16], [17].

Intracortical BMIs, which become the focus of this thesis, typically utilise three types of signal inputs, namely single-unit activity (SUA), multiunit activity (MUA), and local field potential (LFP). The processing steps to obtain each of these signals are described in Section 2.2. SUA represents action potentials or spiking activity of individual neurons in the vicinity of the electrode tip, whereas MUA represents aggregate spiking activity from an ensemble of neurons within a radius of around ∼140-300 μm from the recording electrode [117]–[120]. LFP is the low-frequency part of the neural signal and is thought to mainly reflect summed synaptic activity from a local neuronal population within a radius of at least a few hundred micrometres up to a few millimetres around the recording electrode [28]–[30], [121].

A vast majority of intracortical BMI studies use SUA as the signal input. This is driven by a view that individual neurons act as the functional unit in the brain for the transmission

and processing of electrophysiological signals. SUA has been the gold standard in neuroscience research due to its clear origin and solid theoretical foundation [122]. In addition, early development of BMIs was intimately linked to neurophysiological research for investigating physiological properties of individual neurons [77]. SUA has been shown to have a strong correlation with different movement parameters [67], [123], [124] and been demonstrated to yield high decoding performance in animals [8], [9], [14], [44], [125] and humans [11], [15]–[18]. Nevertheless, the quality and number of recorded spikes progressively decline over time, mainly due to the neural tissue responses inducing glial scar formation around the electrodes, micromotion of the electrodes, and insulation degradation and mechanical breakage [19], [23]–[25]. This instability poses a major challenge in tracking the same SUA over time, leading to degraded decoding performance. Another challenge in SUA-based BMIs is the spike sorting process, which requires a high sampling rate and demanding computational power.

The instability and high computational complexity of SUA have motivated researchers to propose MUA as alternative signal input. MUA is less sensitive to changes in spike amplitude/waveform and uses much simpler processing than SUA. A number of studies have demonstrated long-term stability of MUA with little performance loss or even comparable performance to that of SUA [19], [22], [26], [27]. Despite these advantages, determining the optimal threshold value adaptively over time remains a challenging problem.

Another alternative solution is to utilise LFP, which is a low-frequency extracellular voltage thought to be mainly generated from postsynaptic currents and reflects the activity of a population of neurons in the vicinity of the electrodes. Due to larger 'listening sphere' and simple low pass filtering, LFP is believed to be less sensitive to the issues encountered by spikes, hence providing more stability than the spike counterparts. The long-term stability of LFPs has been demonstrated through a multitude of experimental studies [31]–[34]. LFPs could retain movement-related information even when the spikes can no longer be recorded (lost) [22], [35]–[37].

While it is generally agreed that LFP and MUA have better long-term signal stability than SUA, there is disagreement about which signal carries the highest movement-related information. A considerable number of published papers have reported that SUA yielded higher

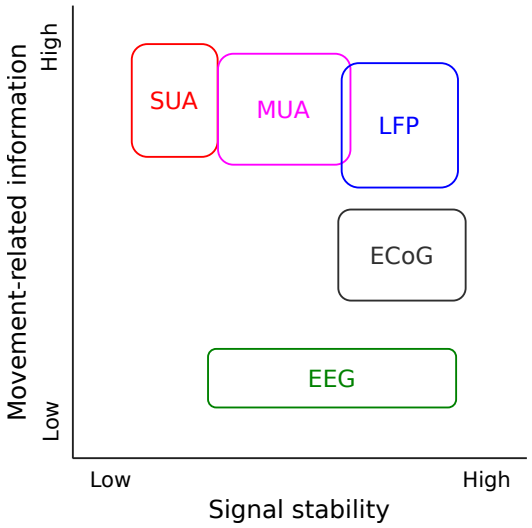


Figure 2.4 Comparison of properties between different types of signal inputs

decoding accuracy than MUA [22], [26], [27], [38], [39] and LFP [33], [35], [39]–[41] (but see [31]). This suggests that SUA carries the highest movement-related information, followed by MUA and LFP, respectively. The comparison of the properties between SUA, MUA, and LFP is summarised in Figure 2.4. For a more comprehensive review of each signal's properties, see Slutzky and Flint's review paper [122].

2.4 BMI decoding

According to whether or not feedback is involved, BMI decoding can be categorised into two types: open-loop (offline) and closed-loop (online) [78], [126]. Open-loop decoding predicts movement parameters of interest without providing feedback (e.g. visual) to the user. The performance of open-loop decoding is typically assessed using a similarity or error metrics between the predicted and actual (ground truth) movement parameters. On the other hand, closed-loop decoding predicts the movement parameters while at the same time provides feedback to the user. The prediction errors from a decoder will always be compensated by the user through feedback-assisted neural modulation. To maintain performance against the nonstationarity and variability of neural signals, a decoder needs to be recalibrated either manually before the experiment or automatically during the experiment [55], [127], [128].

2.4.1 Decoding paradigms

In BMI experiments, the user and/or decoder need to learn how to gain BMI control and predict movement parameters accurately. There are two decoding paradigms according to where the learning process primarily occurs: biomimetic and biofeedback [129]. Biomimetic decoding aims to find a mapping function between the neural activity and movement parameters. A biomimetic decoder is typically trained using a supervised fashion, requiring a 'labelled' dataset that contains a pair of known neural activity and movement parameters. This paradigm has been employed in myriad of BMI literature, either using spikes [16], [75], [130], [131] or LFPs [22], [37], [52]. Despite providing compelling results, this paradigm poses a practical issue with a paralysed subject where arm movement cannot be performed. To address this issue, a biofeedback paradigm can be used instead. Owing to the neural plasticity, the subject can learn to gain BMI control by modulating his neural activity. The biofeedback paradigm was pioneered by Fetz in 1970s [64], [132] when he showed that single-neuron firing rates could be modulated through operant conditioning. Building upon this paradigm, several works demonstrated that monkeys could control a computer cursor or muscle stimulation using spikes [133]–[135] or LFPs [36], [136], [137]. The biofeedback paradigm does not require a labelled dataset during the training phase. The decoder can be made simple and static; the subject needs to learn how to use the decoder effectively assisted via feedback (e.g. visual) and reward. However, it remains unknown how long the learning process is required to achieve BMI control and whether the learnt skill could remain for a long time. It is important to note that biomimetic and biofeedback paradigms are not mutually exclusive; they can be combined, ideally (but not necessarily) yielding improved performance [127].

2.4.2 Decoding algorithms

One of the earliest BMI decoding algorithms is population vector algorithm (PVA) proposed by Georgopoulos et al. [68] and was later employed in several BMI studies [9], [10], [138]. PVA sums the contributions of each neuron in a population termed as preferred direction and weighted by the respective firing rate. The preferred direction (also called tuning curve) is defined as the direction for which a neuron fires spikes most frequently [67]. In PVA formulation, each

neuron is assumed to contribute independently to movement parameters (e.g. direction and speed). The relationship between neuronal firing rate and movement parameters is assumed to be linear. Despite its elegance and usefulness for BMIs, PVA is suboptimal since the weights are determined intuitively without error minimisation principle. PVA may not be generalisable to different tasks as neurons tend to exhibit different tuning properties across different tasks [77]. Moreover, PVA cannot be generalised to BMIs that do not utilise spikes (e.g. LFPs). Another decoding algorithm is optimal linear estimator (OLE) that can be considered as a generalisation of PVA. As opposed to PVA, OLE uses the correlation of the neurons and cross-correlation to movement parameters. The weights of OLE are determined by minimising mean squared error. Chase et al. [139] compared open-loop and close-loop decoding performance between OLE and PVA, showing that OLE performed better than or comparable to PVA. OLE was also used in a clinical trial by an individual with tetraplegia to control a robotic arm to perform reach and grasp movements [16].

Another alternative decoding algorithm is Wiener filter (WF), which takes into accounts the contributions of input from not only the current time point but also from the previous time points (termed as lags or taps). WF estimates the movement parameters at the current time point through a weighted sum of neural features at current and previous time points. WF can be considered as a generalisation of OLE; it reduces to OLE when no previous time point is used. WF has been successfully employed in numerous BMI studies with non-human primates [7], [8], [14], [42], [43] and humans [11], [15] to a control robotic arm or a computer cursor. Flint and colleagues employed a nonlinear extension of WF called Wiener cascade filter (WCF), which combines the WF output with a static nonlinearity through a fitted polynomial function [35], [53], [140]. WCF was demonstrated to yield better decoding performance than WF [141]. Both WF and WCF do not model the dynamics of movement parameters. Since movements usually obey physical laws, e.g. smooth/continuous over time [96], incorporating a dynamical model of movement parameters can provide additional information, which may lead to improved decoding performance.

Kalman filter (KF), originally developed by Rudolf E. Kalman in 1960s [142], addresses the above issue by combining a dynamical (process) model with measurement model. It is

popular due to its simplicity and optimality under the assumption that both models are linear with Gaussian noise [47]. KFs have been employed in many BMI studies [44], [46], [55], [75], [99], [143]. KF consists of two-step computations performed in a recursive manner: prediction and update. In the prediction step, the current states are estimated along with their uncertainties. In the update step, given a new measurement, the estimated states are updated by using a weighted average (larger weight given to estimates with higher certainties). There exist variants of KF such as steady-state Kalman filter (SSKF) [47] and recalibrated feedback intention trained Kalman filter (ReFIT-KF) [131]. Since neural signals are known to exhibit nonlinear property, the linearity assumption of KF may lead to suboptimal decoding accuracy. Unscented Kalman filter (UKF) was proposed to deal with the nonlinear cases while keeping the assumption of Gaussian noise [54], [55], [57], [144]. Another alternative is particle filter (PF), which relaxes the assumptions to nonlinear and non-Gaussian, at the expense of increased computational complexity. PF uses a sampling-based technique (sequential Monte Carlo) to estimate the states [145], [146].

For spike-based BMIs, the typical features are firing rates estimated by binned spike counts within 50–100 ms bin size [25], [77], [96]. The binned spike counts are usually assumed to follow Gaussian distribution. This assumption is reasonable for a relatively large bin size (i.e. large number of spikes) but may deviate for the case of small bin size. The large bin size compared to the scale of the spikes (a few ms) could lead to a slower time-scale of BMI control. To obtain a smaller input time-scale with more appropriate distribution, point process filter (PPF) was proposed [101], [147], [148]. PPF is a class of recursive Bayesian filters where the observations (e.g. binned spike counts) are modelled by a point process. Even though PPF could enable rapid adaptation, control and feedback rates of BMIs [101], [149], [150], it is computationally demanding. In addition, it cannot be generalised to BMIs driven by input signals other than spikes.

It is desirable to have a decoding algorithm that does not require any prior assumption and can be used for any input signal. One example of such an algorithm is an artificial neural network (ANN), which is vaguely inspired by the way the human brain transmits and processes information. ANN is composed of a collection of units (i.e. artificial neurons) connected to other

units in certain architectures (e.g. feed-forward network, recurrent network). Although ANNs were successfully used by monkeys to control a robotic arm or computer cursor [7], [42], [73], [151], they were not widely used as BMI decoder; most BMI studies still linear decoders such as KF and WF. This is possibly because linear decoders offer simplicity with comparable decoding accuracy to that of ANNs. The development of faster computing hardware and optimisation algorithms combined with the abundance of data has driven renewed interest in ANNs and popularised the field of deep learning. Deep learning is a subset of artificial intelligence (AI) which are based on ANNs. It has achieved state-of-the-art performance in a wide variety of applications such as image and speech recognition, natural language processing, and medical diagnostics [58], [59]. Glaser *et al.* proposed deep learning decoders and showed that their performance better than conventional methods [60]. Some proposed deep learning decoder were based on standard recurrent neural network (SRNN) [60], [152], [153] and long short-term memory (LSTM) [60], [62]. However, there are still relatively few studies that utilise deep learning for BMI applications. This opens a new opportunity for the development of deep learning based BMI decoders. The summary of the decoding algorithms used in BMI studies is given in Table 2.1. For reviews of decoding algorithms excluding the deep learning methods, refer to [96], [154], [155].

Table 2.1 Summary of decoding algorithms used in BMI studies

Signal Input	Subject	Decoding Algorithm
Single-unit activity (SUA)	Monkey	PVA [9], [10], [67], [68], [138], WF [7], [8], [14], [42], [43], WCF [50], KF [44]–[47], UKF[54]–[57], PF [145], [146], [156], PPF [101], [147], [149], [150], ANN [7], [42], [73], [151], LSTM [60], [62]
	Human	LR [16]–[18], WF [11], [15], KF [12], [13], [48]
Multiunit activity (MUA)	Monkey	LR [27], [38], KF [38], [131], [157], [158], PF [26], SRNN [152], [153],
	Human	KF [75], [76], [90], [103], [159]–[161]
Local field potential (LFP)	Monkey	WF [36], WCF [35], [51]–[53], KF [22], [32], [33], [37], [41], [92], [93]
	Human	WF [162], KF [92], LWMA [163]

PVA: population vector algorithm, LR: linear regression, WF: Wiener filter, WCF: Wiener cascade filter, KF: Kalman filter, UKF: unscented Kalman filter, PF: particle filter, PPF: point-process filter, SRNN: standard recurrent neural network, LSTM: long short-term memory, LWMA: linear weighted moving average

Part I

Improving Decoding Accuracy of MUA-based BMIs

3 | Firing Rate Estimation Using Bayesian Adaptive Kernel Smoother (BAKS)

This chapter addresses the challenges in estimating firing rate from single-trial spike trains as commonly performed during feature extraction step in spike-based BMIs. We present a new method based on kernel smoothing technique, referred to as Bayesian adaptive kernel smoother (BAKS), which considers the bandwidth parameter as a random variable with a prior distribution that is adaptively updated under an empirical Bayesian framework. We evaluate the performance of BAKS on both synthetic and real spike trains and demonstrate improved performance compared to other methods.

This chapter has been adapted from the following published article:
N. Ahmadi, T. G. Constandinou, and C.-S. Bouganis, "Estimation of neuronal firing rate using Bayesian adaptive kernel smoother (BAKS)", *PLoS One*, vol. 13, no. 11, e0206794, 2018. [1]

3.1 Introduction

In neural systems, signalling and interneuronal communication can be observed through the characteristic of action potentials (or 'spikes'). A sequence of spikes, known as a spike train, may encode information based on different schemes. Currently, there are two main hypotheses of neural coding schemes: *temporal* coding and *rate* coding. The temporal coding represents the information by the precise timing or occurrence of spikes. On the other hand, the rate coding represents the information by the rate or frequency at which a neuron "fires" spikes, also known as "firing rate", and has been the most commonly used scheme to characterise the neuronal or network responses to external stimuli or behavioural tasks [164], [165]. The firing rate is typically estimated in an offline analysis by averaging spiking responses across

multiple repeated experiments known as trials. In practice, however, spiking responses may differ considerably even though the trial setting remains approximately the same. This is partly due to the inherently stochastic nature of neurons and the difference of cognitive states during the trials [91], [166]. Averaging out many variably similar trials can obscure the temporal dynamics, which may contain useful information, within each single trial. Furthermore, there are many research problems that require firing rate to be estimated on a single trial basis. For example, quantifying trial-to-trial variability of neuronal responses [167], [168], decoding task parameters in brain-machine interface (BMI) applications [91], [167], and measuring neuronal responses in cognitive-related tasks such as decision making, motor planning, learning and memory [169]–[171]. Therefore, it is essential to be able to accurately estimate firing rate based on single trials.

Estimating the firing rate as a continuous-time function from a single trial is a challenging task since the underlying process provides only a sparse representation of the spiking data. A widely used method known as peri-stimulus time histogram (PSTH) results in a coarse estimate [172], [173]. To produce a smooth estimate of firing rate, several methods have been proposed such as optimised kernel smoother (OKS) [174], variable kernel smoother (VKS) [174], local polynomial fit (Locfit) [175], and Bayesian adaptive regression splines (BARS) [176]. OKS and VKS employ kernel density estimation technique in which the choice of the kernel bandwidth profoundly impacts the accuracy of estimation. Both methods automatically compute the bandwidth based on the mean integrated squared error (MISE) minimisation principle. However, in computing the bandwidth, these methods assume that spikes are generated by a Poisson process. Even though superimposed spike trains across many trials approximate a Poisson process, in the case of a single trial, a spike train has been shown to depart from this assumption [177], [178]. Single-trial spike train exhibits history-dependent properties such as refractory period and bursting, which cannot be modelled by Poisson process [177], [178]. The deviation from Poisson assumption can lead both OKS and VKS to exhibit poor performance under single-trial cases. Locfit employs a generalised nonparametric regression technique where the firing rate is approximated by a polynomial. The estimation accuracy of Locfit depends mostly on a smoothing parameter (bandwidth). How this bandwidth is selected along with

the Poisson assumption of the spike train are the main caveats of this method. Like Locfit, BARS also employs a generalised regression technique, except that it estimates the firing rate using splines (several polynomials connected at some points or knots). The challenge of using a spline-based method is determining the number and location of the knots since these will significantly impact the estimation. To determine the optimal knot configurations (number and location), BARS utilises a reversible-jump Markov chain Monte Carlo (MCMC) engine with Bayesian information criterion (BIC). The flexibility and powerfulness of BARS come at a price of relatively high computational complexity [176], [179]. In addition, similarly to above-mentioned methods, BARS also assumes that spikes are generated from a Poisson process. It is intended to be used for firing rate estimation after pooling spikes across multiple trials [179].

Within this context, it is highly desirable to have a method for estimating the neuronal firing rate from a single trial spike train, which features an adaptive capability and low computational complexity. An adaptive capability is crucial as spiking activity may change rapidly within a single trial. The proposed method should be data-driven and able to accurately estimate the underlying spike dynamics. Low computational complexity is needed to perform the computation within a short time and is necessary for real-time BMI applications. Also, it is crucial to have a spike train model that mimics the spiking behaviour encountered in the real neural recording. Using a particular assumption on the underlying rate function (i.e. the "ground truth" is known), single spike trains can be stochastically generated which can then be used to evaluate the performance of the firing rate estimation method.

In this chapter, we propose a new method for the estimation of firing rate that addresses the issues listed above. This method employs a kernel smoothing technique due to its simplicity. The critical parameter, bandwidth, is considered as a random variable with a prior distribution and is adaptively determined under an empirical Bayesian framework. With the appropriate selection of kernel and prior distribution functions, an analytical expression of posterior distribution can be attained, which reduces the computational complexity. We refer to this method as Bayesian adaptive kernel smoother (BAKS). We evaluate BAKS with synthetic data generated from two biologically plausible models: inhomogeneous Gamma (IG) and inhomogeneous inverse Gaussian (IIG). BAKS is then tested with real neural data recorded from the motor

and visual cortex of non-human primates (NHPs).

3.2 Methods

In this section, we first introduce a kernel smoothing technique for estimating firing rate. We then describe our proposed method, BAKS, a new variant of kernel-based firing rate estimation method that incorporates an adaptive bandwidth. Lastly, we explain two models that we used to generate synthetic spike train data for evaluating the performance of the proposed method. The BAKS code and all the datasets that we synthesised (in Matlab) have been made publicly available through `https://github.com/nurahmadi/BAKS`.

3.2.1 Kernel-based firing rate estimation

Let $t_1, t_2, \cdots, t_n$ be a sequence of spike times (i.e. a spike train) which can be expressed mathematically as

$$\rho(t) = \sum_{i=1}^{n} \delta(t - t_i) \tag{3.1}$$

where $\delta(t)$ is Dirac function and n is the total number of spikes. The underlying rate function also known as firing rate, $\lambda(t)$, can be estimated by using kernel smoothing, a method which convolves the spike train with a kernel function $K(t)$ as follows,

$$\hat{\lambda}(t) = \int_{-\infty}^{\infty} K(\tau)\rho(t - \tau)d\tau \tag{3.2}$$

Equation (3.2) can also be represented as the sum over kernel functions centred at spike times t_i,

$$\hat{\lambda}(t) = \sum_{i=1}^{n} K(t - t_i) \tag{3.3}$$

The effectiveness (in a minimum integrated squared error (MISE) sense) of kernel smoothing technique depends on the choice of a kernel function and the selection of a smoothing parameter (i.e. bandwidth). A kernel function is required to be a non-negative, normalised to a unit area, having a zero first moment and a finite variance [167], [174]. Examples of kernel functions that have been widely used are Gaussian and Epanechnikov [180], [181]. In many occasions,

the kernel bandwidth is set fixed over the whole observation interval. A significant amount of literature has been reported in the field of statistics on selecting the proper value for this fixed bandwidth [182]–[186]. Even though a near-optimal fixed bandwidth selection (e.g. [174], [187], [188]) may yield a better estimate compared to an arbitrary choice, it may still suffer from simultaneously under- and over-smoothing depending on the underlying spike dynamics. A rapid change in spiking activity is sometimes encountered in neural responses and is of interest to neuroscientists. Thus, it is highly desirable to find an optimal adaptive bandwidth selection method that can adaptively grasp the slow and rapid changes of firing rate.

3.2.2 Bayesian adaptive kernel smoother (BAKS)

BAKS employs a kernel smoothing technique and incorporates an adaptive bandwidth at the estimation points, meaning the bandwidth at which firing rate is being estimated can adapt to the dynamics of the underlying process. In so doing, BAKS considers the bandwidth as a random variable with prior distribution and adaptively updates the posterior bandwidth given spiking data using an empirical Bayesian framework.

Selection of kernel function

There are several choices of kernel functions that can be utilised for firing rate estimation. Gaussian and Epanechnikov are among the popular kernel functions in statistical research [180], [181]. In terms of minimising asymptotic mean integrated squared error (AMISE), Gaussian kernel is slightly less efficient than Epanechnikov kernel [180]. However, Gaussian offers more interesting properties, e.g. the availability of several types of conjugate prior distributions. This conjugate property enables an analytical expression of posterior distribution which simplifies the computation and avoids using a numerical approximation technique. Due to this mathematical convenience, in our proposed method, we select a Gaussian kernel with adaptive bandwidth which can be expressed as,

$$K(t) = \frac{1}{\sqrt{2\pi}h(t)} \exp\left\{-\frac{t^2}{2h(t)^2}\right\} \tag{3.4}$$

where $h(t)$ is the adaptive bandwidth. The bandwidth should be small (large) at the region of high (low) spike density.

Selection of prior distribution

We select a prior distribution of bandwidth that incorporates prior belief about spiking data (i.e. informative prior) and also leads to an analytical expression of the posterior distribution. The informative prior is especially useful when the number of spiking data is relatively small as this prior can give more weight than the likelihood function. At the same time, the analytical expression is crucial to simplify the computation and avoid a numerical approximation technique (e.g. Markov chain Monte Carlo). In Equation (3.4), the Gaussian kernel uses parameter bandwidth (i.e. standard deviation in statistical literature) that describes how spread the observed data are around the mean. We can also represent the parameter in terms of precision (inverse of square bandwidth) that describes how concentrated the observed data are around the mean. In computing firing rate estimation, the means of Gaussian kernel are set to the spike times. These spike times can be represented as the sum of the interspike intervals (ISIs) that can be conveniently modelled by Gamma distribution [177], [189]–[192]. Since the sum of independent Gamma random variables follows Gamma distribution [193], the spike times can also be represented as Gamma distribution. Hence, we propose a Gamma prior distribution on the precision parameter $\sigma(t)$, where $\sigma(t) = 1/h(t)^2$. As Gamma distribution is a conjugate prior for Gaussian distribution with precision parameter [194], the choice of Gamma prior distribution results in an analytical expression of the posterior distribution. This Gamma prior distribution is given by

$$\pi(\sigma(t)) = \frac{\sigma(t)^{\alpha-1}}{\Gamma(\alpha)\beta^{\alpha}} \exp\left\{-\frac{\sigma(t)}{\beta}\right\}, \quad \sigma > 0 \tag{3.5}$$

where $\alpha > 0$ is the shape parameter, $\beta > 0$ is the scale parameter, and $\Gamma(\alpha)$ is Gamma function. By the change-of-variable formula and transformation technique, we can express the

prior distribution $\pi(\sigma(t))$ as a function of $h(t)$:

$$\begin{aligned}\pi(h(t)) &= \pi(\sigma(t)) \times \left|\frac{d\sigma(t)}{dh(t)}\right| \\ &= \frac{2h(t)^{-2\alpha-1}}{\Gamma(\alpha)\beta^{\alpha}} \exp\left\{-\frac{1}{\beta h(t)^2}\right\}\end{aligned} \tag{3.6}$$

BAKS modeling and inference

Under the Bayesian framework, the statistical inference depends on both likelihood function and prior distribution. The likelihood function is the probability density function of the observed spike train $\rho(t)$ viewed as a function of the unknown bandwidth parameter $h(t)$, which can be approximated by

$$\begin{aligned}\hat{f}(\rho(t)|h(t)) &= \frac{1}{n}\sum_{i=1}^{n} K_{h(t)}(t-t_i) \\ &= \frac{1}{n}\sum_{i=1}^{n} \frac{1}{\sqrt{2\pi}h(t)} \exp\left\{-\frac{(t-t_i)^2}{2h(t)^2}\right\}\end{aligned} \tag{3.7}$$

where $K_{h(t)}(t-t_i)$ is Gaussian kernel as defined in Equation (3.4) with adaptive bandwidth and centred at spike event time t_i.

Using Bayes' theorem, the posterior distribution of kernel bandwidth, $\pi(h(t)|\rho(t))$, can be computed by

$$\pi(h(t)|\rho(t)) = \frac{\hat{f}(\rho(t)|h(t))\pi(h(t))}{\int \hat{f}(\rho(t)|h(t))\pi(h(t))dh(t)} \tag{3.8}$$

Computing the denominator of Equation (3.8), also called marginal density, can be problematic if there is no analytical solution of its integral expression since an approximation method will be required instead. The choice of prior distribution in Equation (3.6) coupled with Gaussian kernel in Equation (3.7) leads to an analytical expression for the denominator in Equation (3.8) as follows,

$$\pi(h(t)|\rho(t)) = \frac{\sum_{i=1}^{n} h(t)^{-2\alpha-2} \exp\left\{-\frac{1}{h(t)^2}\left[\frac{(t-t_i)^2}{2}+\frac{1}{\beta}\right]\right\}}{\frac{1}{2}\Gamma(\alpha+\frac{1}{2})\sum_{i=1}^{n}\left[\frac{(t-t_i)^2}{2}+\frac{1}{\beta}\right]^{(-\alpha-\frac{1}{2})}} \tag{3.9}$$

The adaptive bandwidth can then be estimated under squared error loss function by using the

posterior mean formulated as

$$\hat{h}(t) = \int h(t)\pi(h(t)|\rho(t))dh(t) \tag{3.10}$$

The closed-form expression of Equation (3.10) is given by

$$\hat{h}(t) = \frac{\Gamma(\alpha)\sum_{i=1}^{n}\left[\frac{(t-t_i)^2}{2}+\frac{1}{\beta}\right]^{-\alpha}}{\Gamma(\alpha+\frac{1}{2})\sum_{i=1}^{n}\left[\frac{(t-t_i)^2}{2}+\frac{1}{\beta}\right]^{-\alpha-\frac{1}{2}}} \tag{3.11}$$

Equations (3.9)-(3.11) define what is called a mixture of t-distributions in the statistical literature. By substituting the adaptive bandwidth in Equation (3.11) into Equation (3.3), we can compute firing rate estimation using the following formula:

$$\hat{\lambda}(t) = \sum_{i=1}^{n}\frac{1}{\sqrt{2\pi}\hat{h}(t)}\exp\left\{-\frac{(t-t_i)^2}{2\hat{h}(t)^2}\right\} \tag{3.12}$$

BAKS is graphically illustrated in Figure 3.1. The derivation of the closed-form expressions of bandwidth posterior distribution and the adaptive bandwidth estimate are given in Appendices A.1 and A.2, respectively.

Parameter setting of prior distribution

The effectiveness of firing rate estimation using BAKS depends on the adaptive bandwidth estimate, which is influenced by the setting of parameter shape (α) and scale (β) of the prior distribution. According to prior distribution of bandwidth in Equation (3.6), we can calculate its mean and variance as follows,

$$\begin{aligned}\mathbb{E}[h(t)] &= \int h(t)\pi(h(t))dh(t) \\ &= \frac{\Gamma(\alpha-\frac{1}{2})}{\Gamma(\alpha)\beta^{\frac{1}{2}}}\end{aligned} \tag{3.13}$$

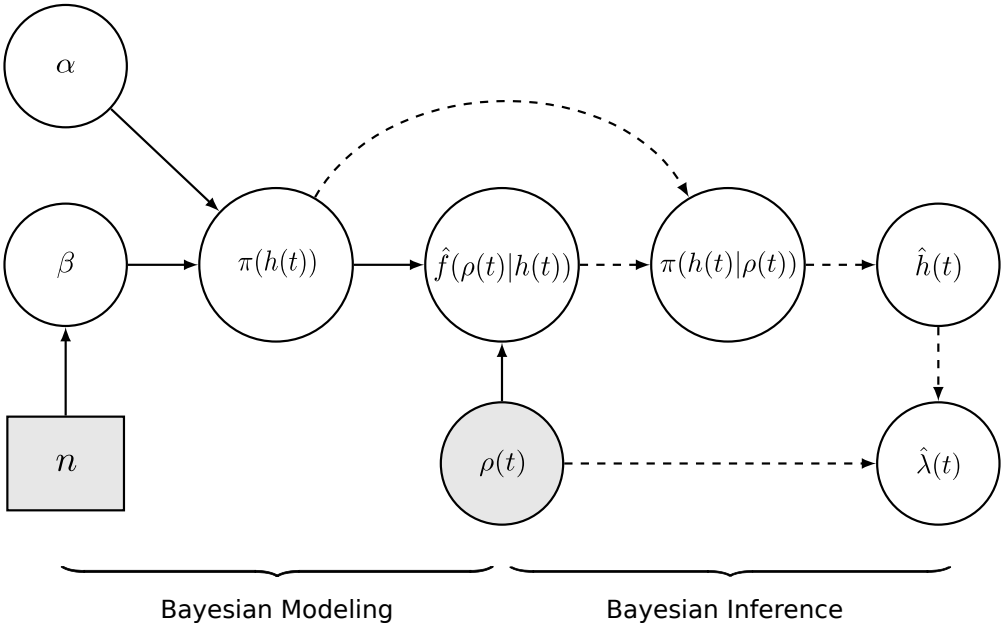


Figure 3.1 Graphical representation of Bayesian adaptive kernel smoother (BAKS). The various blocks describe the different components as follows: Circular blocks denote continuous variables, and rectangle blocks denote discrete variables. Shaded blocks represent observed variables, whereas white blocks represent hidden variables. Solid and dashed arrow indicate Bayesian modeling and inference phase, respectively.

$$\begin{aligned}\mathbb{V}[h(t)] &= \int (h(t)-\mu)^2 \pi(h(t)) dh(t), \quad \mu = \mathbb{E}[h(t)] \\ &= \frac{\Gamma(\alpha)\Gamma(\alpha-1) - \Gamma(\alpha-\frac{1}{2})^2}{\Gamma(\alpha)^2 \beta}\end{aligned} \tag{3.14}$$

where $\mathbb{E}[h(t)]$ and $\mathbb{V}[h(t)]$ are the mean and variance of $\pi(h(t))$, respectively. It can be observed from both Equations (3.13) and (3.14) that the mean and variance are inversely proportional to the value of parameter β. As the number of spike events (n) increases, the bandwidth decreases. Hence, to obtain consistency of the estimation, the value of β is set to be a function proportional to the number of spike events (n). Here, we propose $\beta = n^{4/5}$ in accordance with MISE convergence rate of Gaussian kernel [195].

To yield an accurate estimate of firing rate, we tune parameter α by minimising mean integrated squared error (MISE). The error is defined as the difference between the underlying rate function and the estimated firing rate. Since $h(t) > 0$, the numerator of Equations (3.13) and (3.14) must be greater than zero, which in turn requires $\alpha > 1$. The value of parameter α is tuned such that it minimises MISE function given fixed value of $\beta = n^{4/5}$. This tuning

process is mathematically expressed as follows,

$$\alpha_t = \arg\min_{\alpha}\{\text{MISE}(\alpha)\} \tag{3.15}$$

MISE is a common metric used in evaluating goodness-of-fit of density estimation and has been used in previous studies of firing rate estimation [167], [174], [196]. It is computed between the estimated firing rate ($\hat{\lambda}(t)$) and the true underlying rate function ($\lambda(t)$) with the following formula,

$$\begin{aligned} \text{MISE} &= \int \mathbb{E}[\hat{\lambda}(t) - \lambda(t)]^2 dt \\ &\approx \Delta t \sum \mathbb{E}[\hat{\lambda}(t) - \lambda(t)]^2 \end{aligned} \tag{3.16}$$

where $\mathbb{E}$ denotes the expectation with respect to stochastic process of spike generation model and the integration/summation is performed over observation interval.

The tuning of parameter α was performed using synthetic spike train data generated by biologically plausible models —inhomogeneous Gamma (IG) model and inhomogeneous inverse Gaussian (IIG) model— which are described in the subsequent section. Parameter α value that gives the smallest MISE during this tuning process was then fixed and used for firing rate estimation during performance evaluation (testing) phase of our proposed method. The testing phase was performed using distinct synthetic datasets (i.e. different than that of during tuning phase) and real neural datasets.

3.2.3 Spike train generation model

It is essential to have a model for generating spike trains in order to tune the parameter of BAKS and to evaluate its performance. A spike train can be modelled by using a point process, that is, a stochastic process that describes localised events or points in real (e.g. time, space) domain. The most commonly used class of point processes to model a spike train is inhomogeneous Poisson (IP) process [91], [167], [174], [177], [187]. In this class, spike counts (increments) within any intervals vary across time (i.e. nonstationary). The spike counts depend on the integral value of the firing intensity function over the interval in question but do not depend on the past spike times (i.e. remain independent). As a consequence, IP process still possesses

memoryless property; a spike occurring at a particular time does not depend on the past spiking activities [197]. It has been shown that the IP process can be used to approximate the behaviour of neural spikes when the spikes are superimposed across trials [174], [178]. However, a Poisson process cannot model the history-dependent properties (e.g. refractory period and bursting) of neural spike trains in single-trial cases. Hence, another class of point process is required to model single-trial spike trains.

The renewal process is an alternative class of point process that can describe the history-dependent property of spike times by assuming that a spike fired at any point in time depending only on the last spike, not the spikes before it. In the renewal process, spike times are no longer independent, but it is their inter-spike intervals (ISIs) that remain independent. Spike trains can be conveniently represented by their ISIs drawn from a particular distribution. Two classes of distribution which have often been used to model non-Poisson spike train are Gamma and inverse Gaussian distributions [168], [177], [189]–[192], [198]. The former is then called inhomogeneous Gamma (IG) model while the latter is called inhomogeneous inverse Gaussian (IIG) model.

Inhomogeneous Gamma (IG) model

The inhomogeneous Gamma (IG) model generalises a Poisson model by allowing more flexible ISI distribution that is controlled by a shape parameter (γ) [177], [189]–[192]. The gamma probability density of the ISI is defined as

$$f_\tau(\tau) = \frac{1}{\Gamma(\gamma)\theta^\gamma}\tau^{\gamma-1}\exp\left\{-\frac{\tau}{\theta}\right\} \tag{3.17}$$

where $\tau > 0$ denotes the ISI, $\gamma > 0$ represents the shape parameter, θ is the scale parameter, and $\Gamma(\gamma)$ is the Gamma function. As shown in Equation (3.17), when $\gamma = 1$, the ISI follows an exponential distribution as in a Poisson process. If $\gamma < 1$, the probability density value declines faster than exponential, which means the ISI tends to become smaller. This can be used to describe neuron's rapid firing (bursting) phenomena. If $\gamma > 1$, the probability density value will increase from zero up to a certain peak value and then decrease again to zero. This indicates the refractory period property where a neuron is less likely to fire again immediately

after it fires a spike.

To generate a spike train from such a model, we employ time-rescaling theorem which transforms the original spike times into new rescaled spike times where the ISIs are independently and identically drawn from fixed distribution (e.g. Gamma) [191]. In the IG model, this transformation is defined as the integration of underlying rate function, $\lambda(t)$, on the interval $(0, t]$ and multiplied by the shape parameter (γ). This transformation is formulated as

$$\Lambda(t) = \gamma \int_0^t \lambda(u)du \tag{3.18}$$

Since $\lambda(t)$ is a non-negative function, $\Lambda(t)$ is a monotonically increasing and one-to-one function and usually called integrated intensity function. Thus, we can obtain the original spike times from *i.i.d.* ISI samples (τ) by performing the inverse of time-rescaling transform as follows,

$$\tau_i \equiv \Lambda_{t_{i-1}}(t_i) = \Lambda(t_i) - \Lambda(t_{i-1}) = \gamma \int_{t_{i-1}}^{t_i} \lambda(u)du \tag{3.19}$$

$$t_i = \Lambda_{t_{i-1}}^{-1}(\tau_i) \tag{3.20}$$

where $0 < t_1 < t_2 <, \cdots, < t_n \leq T$ represent the spike times within interval $(0, T]$ and $\Lambda_{t_{i-1}}^{-1}(t)$ is the inverse of time-rescaling transform.

Inhomogeneous inverse Gaussian (IIG) model

It has been suggested that the inhomogeneous inverse Gaussian (IIG) model is more biologically plausible to describe the characteristic of neural spike train than IG model [177], [199]. The IIG model uses inverse Gaussian distribution for the ISI and has been applied in multiple studies [147], [200]. The inverse Gaussian ISI distribution used in this model is given by

$$f_\tau(\tau) = \Big[\frac{\gamma}{2\pi\tau^3}\Big]^{1/2} \exp\left\{-\frac{\gamma(\tau-\mu)^2}{2\mu^2\tau}\right\} \tag{3.21}$$

where $\tau > 0$ denotes the ISI, γ is the shape parameter, and μ is the location parameter. In the IIG model, the value of ISI density is zero at the origin, then increases to certain peak value, and decreases again to zero. How quick the density value rises or falls is controlled by the shape parameter (γ), which demonstrates the flexibility of the IIG model in describing different neural spike characteristics such as rapid bursting and refractory period. The smaller (larger) value of γ, neuron tends to fire more bursty (regularly).

To generate a spike train using the IIG model, we employ a similar procedure as that of the IG model, with the exception that the time-rescaling transform here is expressed as

$$\Lambda(t) = \int_0^t \lambda(u)du \tag{3.22}$$

where the interval between subsequent spike times (ISIs) in the rescaled domain are drawn from *i.i.d.* inverse Gaussian distribution. The spike times in the original domain can be computed by using the inverse of time-rescaling transform as shown in Equation (3.20).

3.3 Results

3.3.1 Synthetic datasets

In our synthetic data, the spike trains were generated from IG and IIG models with different underlying rate functions that represent non-stationary processes usually encountered in empirical datasets. These rate functions include (1) a continuous process with changing (heterogeneous) frequency, (2) a continuous process with oscillatory (homogeneous) frequency, and (3) a discontinuous process with sudden rate changes. In this chapter, the first, second, and third rate functions are consecutively referred to as 'chirp', 'sine', and 'sawtooth' rate functions, and are mathematically expressed as,

$$\lambda_c(t) = \eta + A\sin(2\pi f t^2 + \phi) \tag{3.23}$$

$$\lambda_s(t) = \eta + A\sin(2\pi f t + \phi) \tag{3.24}$$

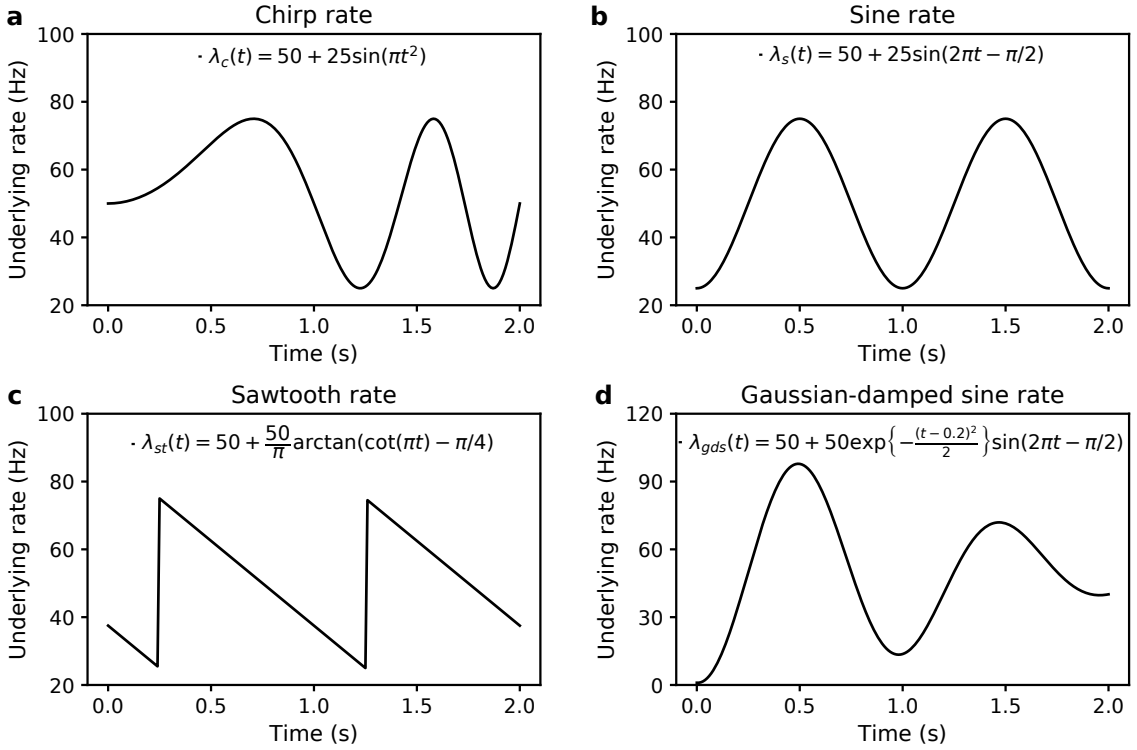


Figure 3.2 Illustration of the underlying rate functions used in this chapter. (a) Chirp rate function with $\eta = 50$, $A = 25$, $f_c = 0.5$, and $\phi = 0$. (b) Sine rate function with $\eta = 50$, $A = 25$, $f_s = 1$, and $\phi = -\pi/2$. (c) Sawtooth rate function with $\eta = 50$, $A = 25$, $f_{st} = 1$, and $\phi = -\pi/4$. (d) Gaussian-damped sinusoidal rate function with $\eta = 50$, $A = 1$, $t_0 = 0.2$, $\sigma = 1$, $f_{gds} = 0.5$, and $\phi = -\pi/2$.

$$\lambda_{st}(t) = \eta + \frac{2A}{\pi}\arctan(\cot(\pi f t + \phi)) \tag{3.25}$$

where η indicates the base or average number of spikes per second and A denotes the intensity (or amplitude) which controls the dynamic range of the rate function. Parameter f is frequency whereas ϕ is phase. The chirp, sine and sawtooth rate functions are represented by $\lambda_c(t), \lambda_s(t)$ and $\lambda_{st}(t)$, respectively.

The chirp rate function used in this chapter is similar to that of Rao and Teh [201], whereas the sine and sawtooth processes are similar to that of Shimazaki and Shinomoto [174] but with different intensity, frequency, and spike train model. Examples of chirp, sine, and sawtooth rate functions with certain parameter settings are illustrated in Figures 3.2a, 3.2b, 3.2c, respectively.

During parameter tuning of BAKS (tuning phase), we set η to 50 spikes/s and A to 25 spikes/s (corresponds to dynamic intensity range between 25 and 75 spikes/s) as in [190].

These intensity values are referred to as medium intensity. This setting is reasonable as, in practice, cortical neurons do not often fire more than 100 spikes/s [202]. The frequency of chirp (f_c), sine (f_s) and sawtooth (f_{st}) rate functions were set to 0.5, 1 and 1, respectively. These frequency values are referred to as medium frequency. We generated randomly 100 single spike trains (2 s duration) from two models (IG and IIG) and three rate functions (chirp, sine, and sawtooth). In the IG model, the shape parameter (γ) that determines the regularity of firing rate was set to 4, while the scale parameter (θ) was set to 1. This setting is comparable to real neural spiking data and also used in [189], [198]. Analogously, the shape (γ) and location (μ) parameters in the IIG model were respectively set to 4 and 1. The parameter value (α) of prior distribution in BAKS was tuned by minimising the average MISE from two models and three rate functions mentioned above. The same procedure was performed to tune the smoothing parameter value of Locfit.

For evaluating the performance of BAKS (testing phase), we generated five datasets, each containing 100 repetitions from both IG and IIG models with the same and different setting from that of parameter tuning phase. Dataset testing 1 was generated using the same setting as in the tuning phase. Dataset testing 1 was used to evaluate BAKS when the underlying rate functions are the same, but their stochastic spiking data realisations are different. Dataset testing 2 was used to evaluate BAKS when the frequency and intensity of the underlying rate functions are different. The frequency of chirp, sine and sawtooth rate functions were respectively set to low (0.25, 0.5 and 0.5) and high (0.75, 1.5 and 1.5) while keeping the intensity same as that of tuning phase. Then, the dynamic intensity range of these three rate functions were set to low (2–20) and high (25–175) while keeping the frequency same as that of tuning phase. In dataset testing 3, the model setting was equal to during the tuning phase except that the parameter shape (γ) was varied between 1 and 10 with an increment of 1. Dataset testing 4 was used to assess the performance of BAKS under multiple trials ($tr = \{5, 10, 20, 30\}$). These multi trial spike trains were obtained by superimposing spike trains from a number of trials (5, 10, 20 or 30) into single spike trains with the same setting as of tuning phase. In dataset testing 5, spike trains were generated using Gaussian-damped sinusoidal rate function with varying intensity and frequency. The use of this rate function was inspired by Mazurek's work [202].

Table 3.1 Model settings for the synthetic dataset generation

Phase	Rate function	Intensity	Frequency	Parameter shape (γ)	Number of trial (tr)
Tuning	{Chirp, sine, sawtooth}	Medium	Medium	4	1
Testing 1	{Chirp, sine, sawtooth}	Medium	Medium	4	1
Testing 2	{Chirp, sine, sawtooth}	{Low, high}	{Low, high}	4	1
Testing 3	{Chirp, sine, sawtooth}	Medium	Medium	$\{1, 2, \cdots, 10\}$	1
Testing 4	{Chirp, sine, sawtooth}	Medium	Medium	4	$\{5, 10, 20, 30\}$
Testing 5	{Gaussian-damped sine}	{Low, medium, high}	{Low, medium, high}	4	1

Tuning: medium intensity ($\eta = 50, A = 25$), medium frequency ($f_c = 0.5, f_s = 1, f_{st} = 1$)
Testing 1: medium intensity ($\eta = 50, A = 25$), medium frequency ($f_c = 0.5, f_s = 1, f_{st} = 1$)
Testing 2: low intensity ($\eta = 11, A = 9$), high intensity ($\eta = 100, A = 75$), low frequency ($f_c = 0.25, f_s = 0.5, f_s t = 0.5$), high frequency ($f_c = 0.75, f_s = 1.5, f_{st} = 1.5$)
Testing 3: medium intensity ($\eta = 50, A = 25$), medium frequency ($f_c = 0.5, f_s = 1, f_{st} = 1$)
Testing 4: medium intensity ($\eta = 50, A = 25$), medium frequency ($f_c = 0.5, f_s = 1, f_{st} = 1$)
Testing 5: low intensity ($\eta = 10, A = 1$), medium intensity ($\eta = 50, A = 1$), high intensity ($\eta = 100, A = 1$), low frequency ($f_{gds} = 0.5$), medium frequency ($f_{gds} = 1.5$), high frequency ($f_{gds} = 2$)

Mathematically, the Gaussian damped sinusoidal rate function is given by

$$\lambda_{gds}(t) = \eta + \eta A \exp\left\{-\frac{(t - t_0)^2}{2\sigma^2}\right\} \sin(2\pi f t + \phi) \tag{3.26}$$

During the generation of dataset testing 5, the frequency (f_{gds}) was set to low (0.5), medium (1.5) and high (2) while keeping the dynamic intensity range to medium (0 to 100) which corresponds to $\eta = 50$ and $A = 1$; the dynamic intensity range was varied between low (0 to 20), medium (0 to 100) and high (0 to 200) while setting $f_{gds} = 1$. An example of Gaussian-damped sinusoidal rate function is illustrated in Figure 3.2d. The summary of model settings for generating the synthetic datasets used in tuning and testing phase is shown in Table 3.1.

3.3.2 Tuning the shape parameter of prior distribution

We investigated the choice of a parameter value that yielded the most accurate firing rate estimates measured by MISE metric. The tuned parameter (α_t) obtained from the tuning phase was then used as the parameter value for our proposed method during the testing phase. Spike trains used during the tuning phase were stochastically generated from two models (IG and IIG) and three different underlying rate functions (chirp, sine, sawtooth) as described in the previous section. We performed firing rate estimation over tuning dataset using parameter α varying from 1 to 10 with an increment of 0.5. This estimation was carried out while keeping

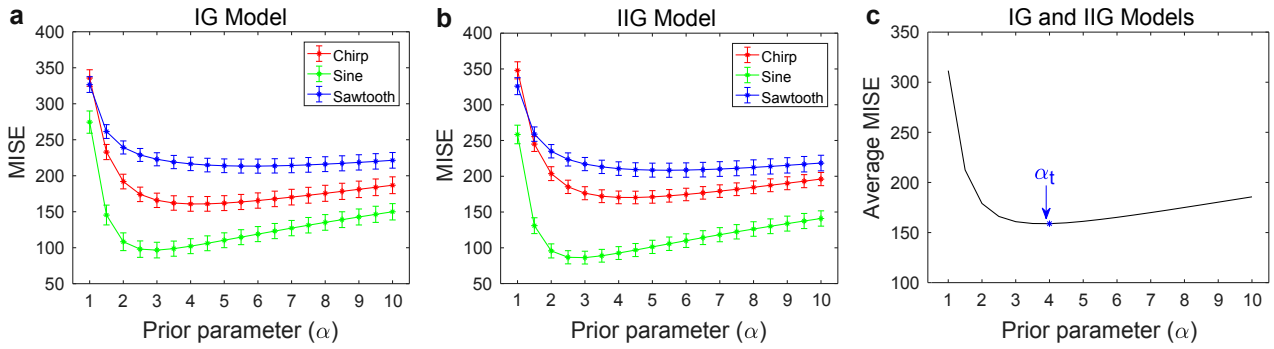


Figure 3.3 MISE comparison under different values of prior parameter (α). (a) MISE values (star marks) and their 95% confidence intervals (vertical bars) for IG model. (b) MISE values (star marks) and their 95% confidence intervals (vertical bars) for IIG model. (c) Average MISE values across 2 models and 3 rate functions. We set $\alpha_t = 4$ as it yields the smallest average MISE.

parameter β fixed to $n^{4/5}$, where n represents the total number of spikes within the observed duration.

The performance of estimation was quantified in terms of MISE, mean of integrated squared error between the underlying and estimated firing rates over the observed duration (2 s). From 100 repetitions and 19 variation of parameter values ($\alpha = \{1, 1.5, 2, \cdots, 10\}$), we computed the MISE values along with their 95% confidence intervals from these 6 scenarios (2 models and 3 rate functions). The MISE values and their confidence intervals for IG and IIG models are shown in Figures 3.3a and 3.3b. In the IG model, α values that resulted in the smallest MISE for chirp, sine and sawtooth rate functions are 4, 3 and 6, respectively. In the IIG model, α values associated with the smallest MISE for chirp, sine and sawtooth rate functions are 4.5, 3 and 5.5, respectively. We tuned parameter α by computing the average MISE values across six scenarios. The value of α that corresponds to the smallest average MISE is referred to as the tuned parameter (α_t). According to the results, as shown in Figure 3.3c, we set $\alpha_t = 4$ and used this value during the testing phase.

3.3.3 Comparison with established methods

We evaluated and compared the performance of the proposed method (BAKS) with established methods which include optimised kernel smoother (OKS) [174], variable kernel smoother (VKS) [174], local polynomial fit (Locfit) [175] and Bayesian adaptive regression splines (BARS) [176].

The performances were quantified using MISE, as expressed in Equation (3.16). We did not include the histogram (PSTH) method in the comparison as this cannot produce a smooth estimate under the single-trial case. The performance of PSTH was shown to be significantly lower than OKS, VKS, Locfit, and BARS [174].

The two kernel smoothing methods used for performance comparison, OKS and VKS[1], were developed by Shimazaki and Shinomoto [174]. In OKS, the bandwidth is fixed for the whole duration and automatically selected based on global MISE minimisation principle. On the contrary, VKS employs a variable bandwidth that is automatically determined by minimising local MISE function. Thus, both OKS and VKS do not require manual user intervention in selecting optimal bandwidth parameter.

Locfit is part of Chronux analysis software[2] that estimates the firing rate by maximising local log-likelihood where the log-density function is approximated by a local polynomial. It has some parameters such as degree of the polynomial, weight function, and bandwidth. However, it is the bandwidth that is considered as the most crucial parameter affecting the accuracy of estimation [175]. We used the nearest neighbour (NN) bandwidth so that the local neighbourhood always contains sufficient data (spikes). This can reduce data sparsity problem that may arise in real neural data. The NN bandwidth parameter was determined by a tuning process similar to that of BAKS. In contrast, the parameters of the degree of a polynomial and weight function were set to the default values, which are two and tricube ('tcub'), respectively. We performed MISE comparison from three underlying rate processes (chirp, sine, and sawtooth) with NN bandwidth varying from 0.2 to 0.8 with an increment of 0.1. We set the NN bandwidth parameter to 0.4 as this value yielded the smallest average MISE. It means that Locfit uses 40% of the total data in each estimation point.

BARS[3] estimates the firing rate by using a cubic spline basis function with free parameters on the number and location of knots [176]. The optimal knot configuration is determined by a fully Bayesian approach with a reversible-jump Markov chain Monte Carlo (MCMC) engine and locality heuristic procedure. BARS takes spike counts within bin intervals (histogram) centred on estimation times of interest. We used a Poisson prior distribution on the number of

[1] `http://www.neuralengine.org/res/kernel.html`
[2] `http://chronux.org/`
[3] `http://lib.stat.cmu.edu/~kass/bars/bars.html`

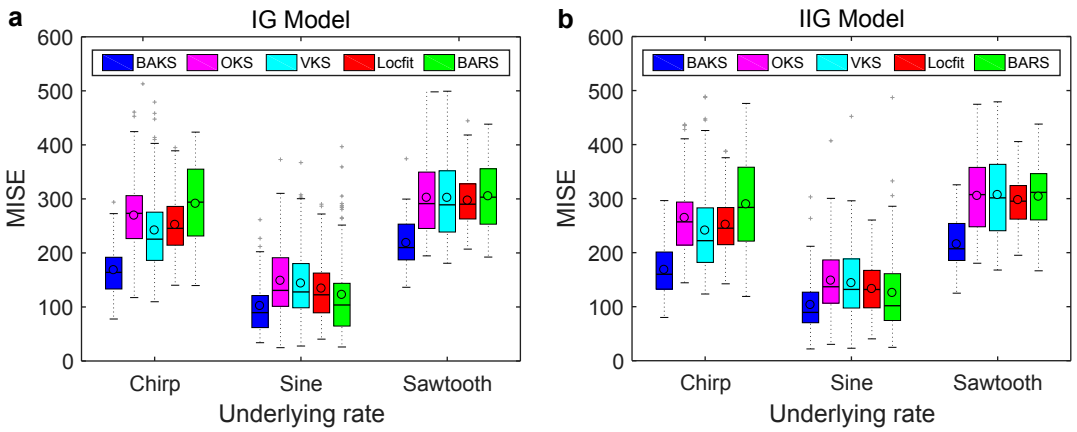


Figure 3.4 MISE comparison of BAKS with other methods. (a) MISE comparison for three rate functions (chirp, sine, and sawtooth) from IG model. (b) MISE comparison for three rate functions (chirp, sine, and sawtooth) from IIG model. In each boxplot, black lines show the medians; black circles indicate the means; coloured solid boxes represent interquartile ranges; whiskers extend 1.5× from upper and lower quartiles; grey crosses represent outliers.

knots and set the sample iterations to 5000 and burn-in samples to 500. We used spike counts within 10 ms bin intervals as the input and mean of fitted function as the output estimate.

We performed 100 repetitions of single-trial firing rate estimation using dataset testing 1. MISE comparison of all the methods for three underlying rate functions and two models is plotted as a boxplot (Figure 3.4). BAKS yielded the smallest medians, means, and interquartile ranges of MISE compared to other competing methods in all six scenarios. As shown in Figure 3.4a for IG model and Figure 3.4b for IIG model, BAKS (blue boxplot) yields significantly lower MISE than the other methods. This demonstrates the effectiveness of BAKS, which features an adaptive bandwidth in estimating the underlying rate from single-trial spike train. Examples of the underlying rate functions and the estimated firing rates from all methods are given in Figure 3.5. BAKS (blue line) is shown to yield qualitatively good firing rate estimates across six scenarios.

3.3.4 Comparison across different values of intensity and frequency

We studied the effect of different intensity and frequency values of the underlying rate functions on the performance of BAKS. In real neural data, the number of spikes (intensity) and the temporal fluctuation of spikes (frequency) may change slowly and rapidly. Therefore, we varied the values of intensity and frequency parameters, as described in Table 3.1 (Testing 2). Using

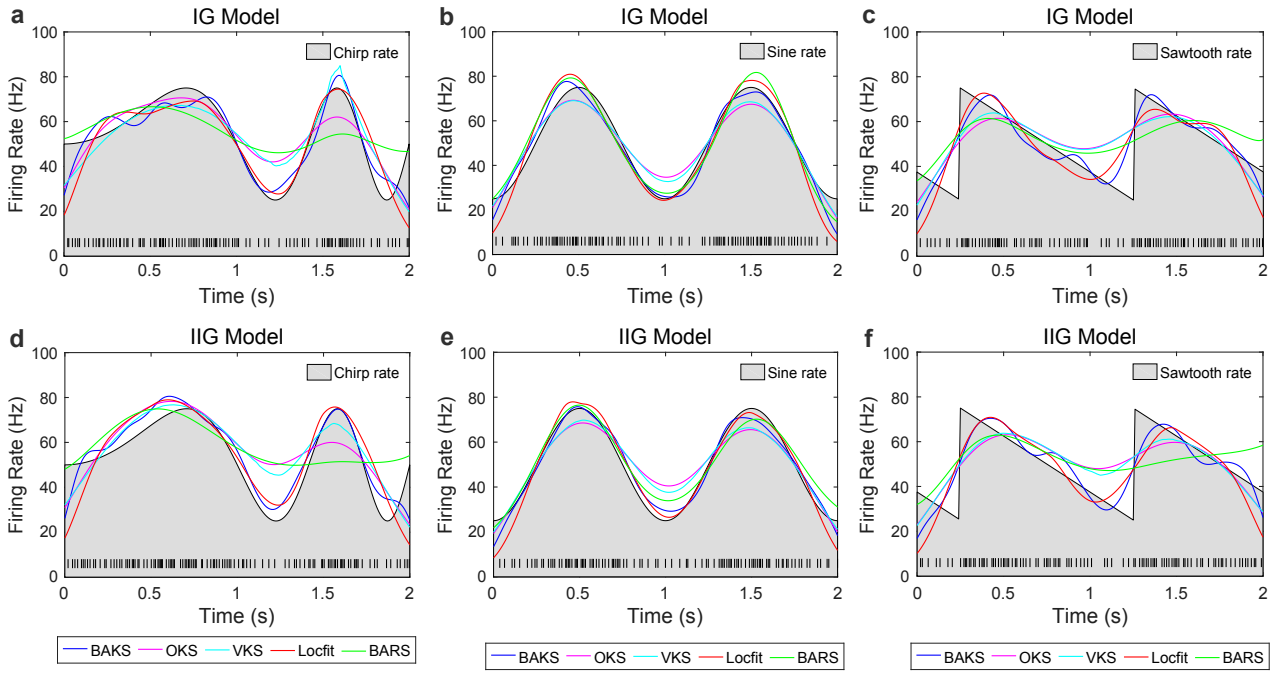


Figure 3.5 Comparison of firing rate estimates across all methods. (a)-(c) Firing rate estimates from IG model with chirp, sine, and sawtooth rate functions, respectively. (d)-(f) Firing rate estimates from IIG model with chirp, sine, and sawtooth rate functions, respectively. Black lines with grey-shaded regions indicate the underlying rate functions. Black raster in the bottom of each plot represents the spike train generated from the associated model and underlying rate function.

dataset testing 2, we performed 100 repetitions of single-trial firing rate estimation from all methods. The statistical summaries of MISE comparison across all methods for the cases of IG and IIG models are shown in Figures 3.6 and B.1, respectively. From a total of 24 cases (2 models, 3 rate functions and 4 variations of intensity and frequency), BAKS outperformed other competing methods in 16 cases (66.67%) as shown in Figures 3.6 and B.1. On average, BAKS performed better than other methods. Significant performance improvements by BAKS were achieved in the cases of chirp and sawtooth rate functions with high frequency. This suggests that BAKS is more effective at estimating a continuous or discontinuous rate function with rapidly changing spike dynamic. On the other hand, BAKS exhibits poor performance in the case of sine rate function with low frequency. This suggests that BAKS is not suitable for estimating a continuous rate function with slowly changing spike dynamic.

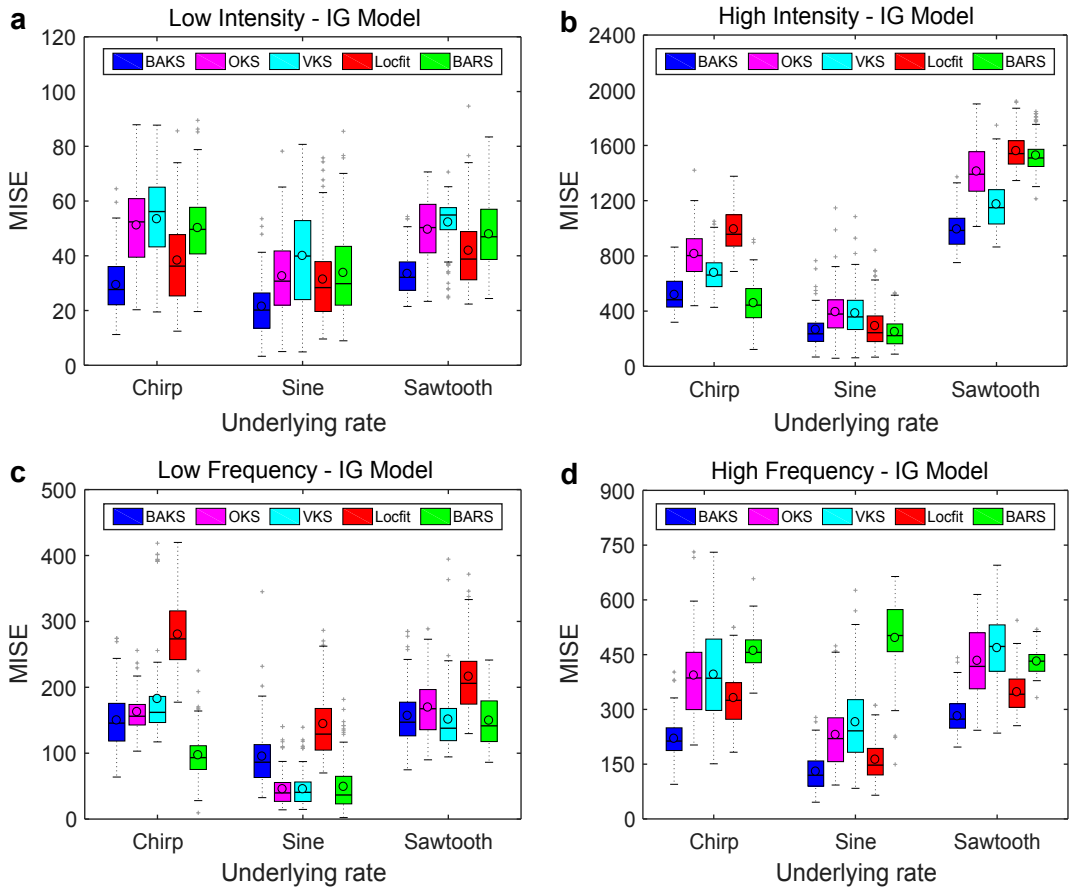


Figure 3.6 MISE comparison across different values of intensity and frequency for IG model. (a) MISE comparison for the case of low intensity. (b) MISE comparison for the case of high intensity. (c) MISE comparison for the case of low frequency. (d) MISE comparison for the case of high frequency. In each boxplot, black lines show the medians; black circles indicate the means; coloured solid boxes represent interquartile ranges; whiskers extend 1.5× from upper and lower quartiles; grey crosses represent outliers.

3.3.5 Comparison across different values of shape parameter

We examined the flexibility of BAKS when the assumption of ISI shape deviates from that of used during the tuning phase. Using dataset testing 3 ($\gamma = \{1, 2, \cdots 10\}$), we performed 100 repetitions of single-trial firing rate estimation. From a total of 60 cases (2 models, 3 rate functions and 10 variations of γ values), BAKS showed superior performance over the other methods in 55 cases (91.67%) as shown in Figure 3.7. In the remaining cases, BAKS resulted in comparable performance to OKS and VKS, which performed good under inhomogeneous Poisson model (corresponding to $\gamma = 1$). We also investigated the performance of BAKS when $\gamma < 1$, which corresponds to bursting activity. We performed firing rate estimations using

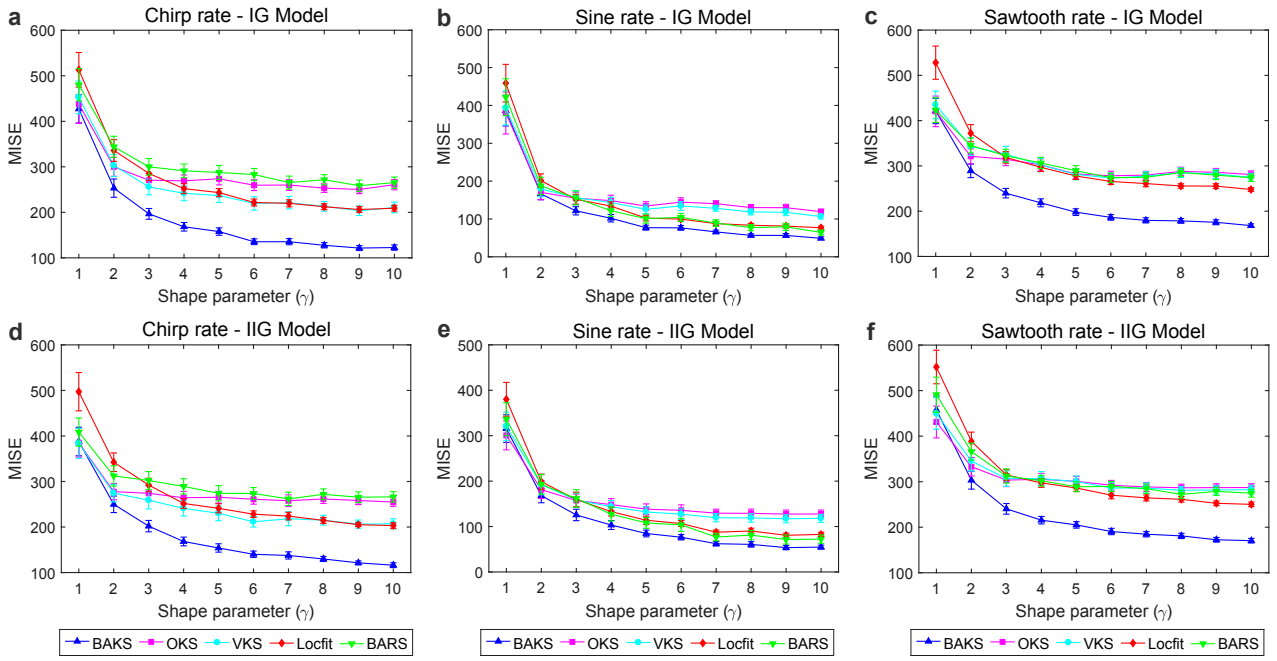


Figure 3.7 MISE comparison across various values of shape parameter (γ). (a)-(c) MISE comparison for chirp, sine, and sawtooth rate functions, respectively, from IG model. (d)-(f) MISE comparison for chirp, sine, and sawtooth rate functions, respectively, from IIG model. Vertical bars represent the 95% confidence intervals.

synthetic data generated with $\gamma = \{0.25, 0.5, 0.75\}$. The results showed the superiority of BAKS to other methods in all 18 scenarios (2 models, 3 rate functions, 3 variations of γ values) as depicted in Figure B.2. These results demonstrate the flexibility of BAKS in estimating the underlying rates across various ISI characteristics of spike train models.

3.3.6 Comparison across different numbers of trials

In offline analyses, the firing rate is typically estimated using spike trains from many similar trials aggregated into a single, compact, spike train. To study the effect of the increasing number of trials, we assessed the performance of BAKS across different numbers of trials ($tr = \{5, 10, 20, 30\}$) using dataset testing 4. We performed firing rate estimation using all methods, each for 100 times. MISE comparison for these multi-trial cases is depicted in Figure 3.8. The increasing number of trials improved the performance of all methods as indicated by the decreasing MISE values. However, with the increasing number of trials, the rate of improvement declined as the MISE values reached their convergences. BAKS on average performed better than other methods except against BARS. Compared to Locfit, BAKS always

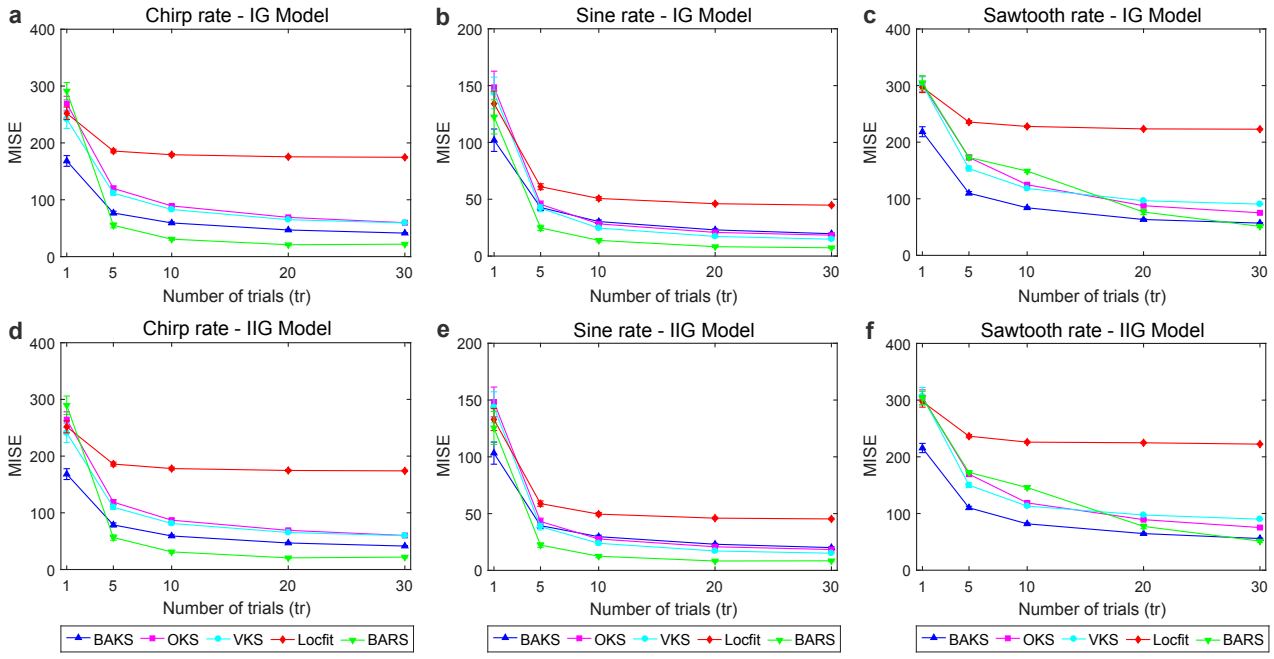


Figure 3.8 MISE comparison across different numbers of trials. (a)-(c) MISE comparison for chirp, sine and sawtooth rate functions, respectively, for IG model. (d)-(f) MISE comparison for chirp, sine and sawtooth rate functions, respectively, for IG model. Vertical bars (not clearly seen for $tr \geq 5$) represent the 95% confidence intervals.

showed significantly better performance in all multi-trial cases. In comparison to OKS and VKS, BAKS shows worse performance only in the cases of sine rate functions for $tr \geq 10$ (Figures 3.8b and 3.8e). However, compared to BARS, BAKS yielded poor performance in most multi-trial cases except for sawtooth rate cases ($tr < 30$) as shown in Figures 3.8c and 3.8f. These overall results suggest that BAKS performs well on multi-trial cases with a moderate number of spikes. Nevertheless, when it comes to multi-trial cases with a large number of spikes, BARS performs better than the others.

3.3.7 Comparison across different underlying rate functions

In practice, the true underlying rate function that generates the spiking data is unknown. There are infinite spaces of rate function that underlie the spiking generation. During the tuning phase, we used three underlying rate functions (chirp, sine and sawtooth) to tune the parameter of BAKS. Next, we studied the impact of different underlying rate function along with its intensity and frequency variations on the performance of BAKS. We performed 100 repetitions of firing rate estimation using dataset testing 5, which corresponds to single-trial spike trains generated

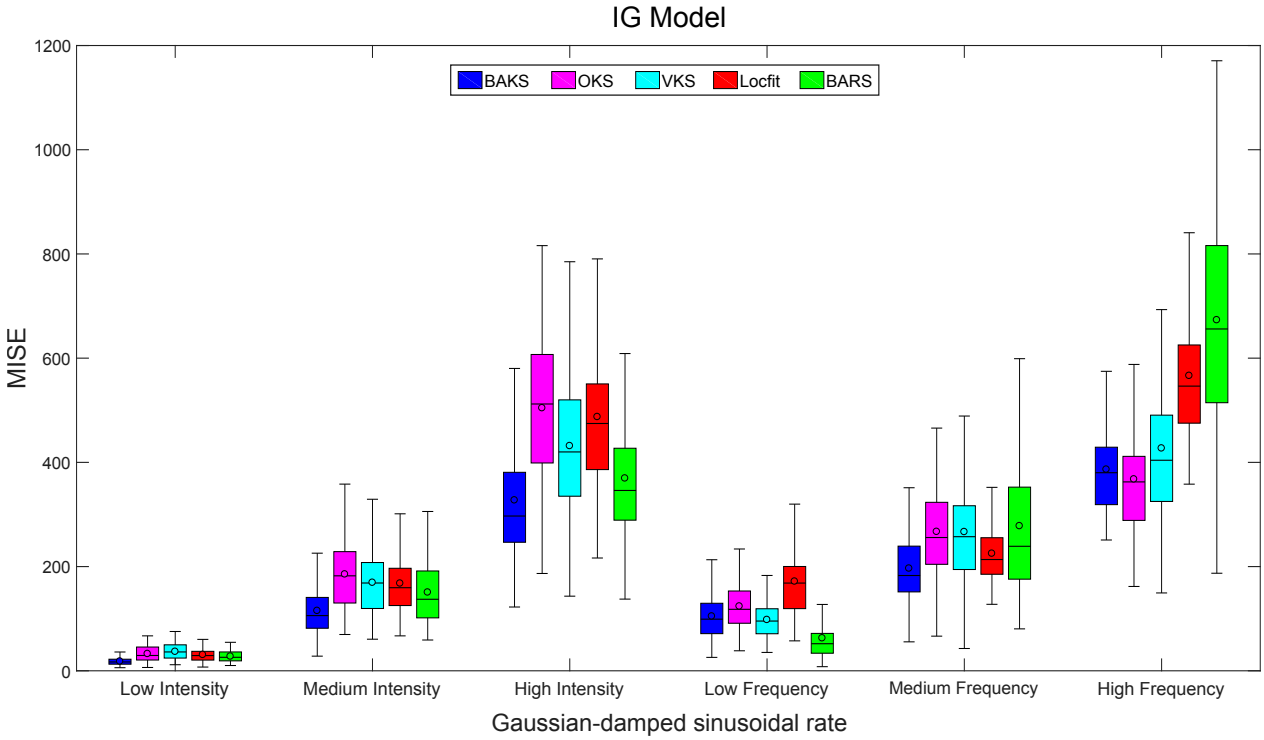


Figure 3.9 MISE comparison across different underlying rate functions for IG model. In each boxplot, black lines show the medians; black circles indicate the means; coloured solid boxes represent interquartile ranges; whiskers extend 1.5× from upper and lower quartiles; grey crosses represent outliers.

from Gaussian-damped sinusoidal rate function as in Equation (3.26). This rate function was chosen as another representation of continuous rate functions with changing spike dynamic. The performance comparisons across all methods using this dataset for IG and IIG models are plotted in Figures 3.9 and B.3, respectively. From a total of 12 cases, BAKS outperformed all the other methods in 8 cases (66.67%).

We also evaluated the performance of BAKS using synthetic data generated from square rate function that represents discontinuous rate functions with changing spike dynamic. The choice of square rate function was inspired by other studies [174], [203]. We performed firing rate estimations using synthetic data generated from the IG model with square rate function. We varied the values of frequency and intensity of the square rate function. From a total of 5 cases, BAKS consistently showed superior performance than other methods as depicted in Figure B.5. These overall results demonstrate the reliability of BAKS even when the underlying rate functions depart from that of used during the tuning phase. Examples of estimated firing rates from all methods for IG and IIG models with Gaussian-damped sinusoidal rate function

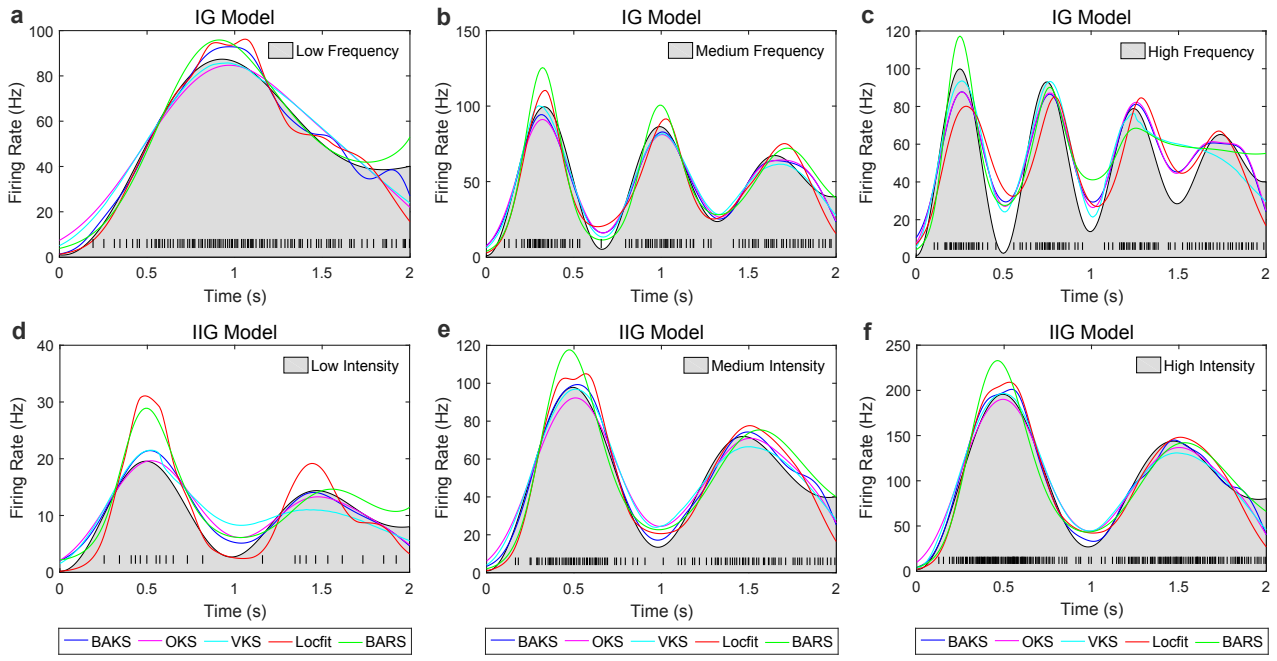


Figure 3.10 Firing rate estimate comparison for IG model with Gaussian-damped sinusoidal rate function. (a)-(c) Firing rate estimates for the cases of low, medium and high frequency, respectively. (d)-(f) Firing rate estimates for the cases of low, medium and high intensity, respectively. Black lines with grey-shaded regions indicate the underlying rate functions. Black raster in the bottom of each plot represents a spike train generated from the associated underlying rate function.

are shown in Figures 3.10 and B.4, respectively. Examples of estimated firing rates for IG model with square rate function are depicted in Figure B.6.

3.3.8 Comparison of computational complexity

Computational complexity affects how fast the computation of each method is completed. A computationally fast method allows a significant reduction in the data analysis process when working with a large number of iterations. Furthermore, the fast computation of firing rate is necessary for online spike-based BMI experiments to generate real-time feedback to the subject.

Kernel smoothing technique has the advantage of being relatively simple and computationally fast. This is especially the case when the bandwidth is fixed throughout the observation interval. OKS uses binned spike counts within certain bin intervals (centred at estimation times) and kernel function with fixed bandwidth to estimate firing rate. The firing rate computation is performed by convolving the binned spike counts with the kernel function. OKS incorporates a Fast Fourier Transform (FFT) for computing the convolution to further reduce the computation

time. The bandwidth is selected by minimising mean integrated squared error (MISE) function over the whole duration [174]. An extension to OKS is VKS, which incorporates a variable bandwidth. This variable bandwidth is computed by minimising local MISE, which requires a large number of iterations with varying local interval. This iterative process makes VKS significantly more complex than OKS. Unlike OKS and VKS, Locfit uses a polynomial to fit log-rate function by maximising a local likelihood function. Locfit has relatively low complexity because it uses a fixed bandwidth selected manually; it does not employ an automatic selection of bandwidth. In this chapter, the bandwidth (in terms of nearest neighbour) of Locfit was set to 0.4, meaning that the computation involves 40% of the total data within the whole duration.

Even though BAKS incorporates an automatic selection of adaptive bandwidth, it still offers relatively low computational complexity. This advantage arises from the simple kernel smoothing technique with a proper choice of prior distribution and kernel function, which leads to closed-form expression of posterior bandwidth. This in turn simplifies the computation process of determining the adaptive bandwidth. This type of convenient closed-form expression cannot be obtained in the case of BARS; thus, a numerical approximation technique is required. BARS uses an iterative procedure involving computationally expensive Markov chain Monte Carlo (MCMC) technique and Bayes information criterion (BIC) to find the optimal smoothing parameters (i.e. number and location of knots). This process takes relatively long computation to yield "converged" results.

The computational complexity among these methods can be indicated by the time required for completing the firing rate estimation in a computer simulation (i.e. computational runtime). Since the code implementation of each method impacts this runtime comparison, this should be viewed as an estimation of the real computational complexity of each method. All the program codes were written and run in Matlab R2016b on 64-bit PC with 8 Intel cores i7-4790 @3.6 GHz and RAM 16 GB. This comparison uses 100 repetitions of single-trial synthetic spike train data (2 s duration) generated by the IG model with chirp rate function.

Locfit and OKS were the two fastest methods, completing the computation within the order of few milliseconds (Table 3.2). VKS took around two orders of magnitude longer time than both Locfit and OKS, whereas BARS took the longest time (in the order of seconds). In

Table 3.2 Average runtime comparison of BAKS with other methods (in second)

Single trial case	BAKS	OKS	VKS	Locfit	BARS
Low intensity	0.0016	0.0017	0.2340	0.0012	4.8438
Medium intensity	0.0062	0.0018	0.3306	0.0012	4.6563
High intentisy	0.0126	0.0017	0.3306	0.0014	4.5527

BARS parameter setting, we set burn-in iterations to 500 and sample iterations to 5000. The BARS runtime can be reduced by setting the burn-in and sample iterations to smaller values. However, this may result in decreasing accuracy as the trade-off. Therefore, these parameters should be carefully set to find a good trade-off between accuracy and runtime. Wallstrom et al. suggested that the default values for burn-in and sample iterations are 500 and 2000, respectively [204].

The runtime performance of BAKS was significantly better than that of both VKS and BARS, but worse than that of Locfit and OKS. Table 3.2 shows that BAKS runtime is influenced by the intensity of the underlying rate function (i.e. number of spikes), whereas other methods' runtime is relatively consistent. This is because, as opposed to BAKS, other methods incorporate a binning procedure for the spiking data before their core computation. This makes the number of input data fed to the core computation always uniform regardless of the number of spikes within the observation interval. Our current BAKS code is a straightforward implementation of BAKS formula (see Section 3.2.2). In this chapter, we have not considered the efficient implementation of BAKS. It is important to note, as neurons have a property of refractory period, the number of spikes within observation interval is limited. This guarantees that under single-trial cases, even with the current implementation code, the runtime performance of BAKS will only decrease up to certain bound.

3.3.9 Application to real neural data

We applied BAKS for estimating the neuronal firing rate from real data obtained from two public neural databases: database for reaching experiments and models (DREAM)[4] and neural signal archive (NSA)[5]. In the DREAM database, we used Flint_2012 dataset which was

[4] `http://crcns.org/`

[5] `http://www.neuralsignal.org/`

recorded the from the primary motor cortex area (M1) of a monkey while performing centre-out reaching task. Single unit spikes were obtained by using thresholding and offline sorting technique. More detailed information on the recording tools and experimental setup can be found in [35]. In the NSA database, we used nsa2004.1 dataset recorded from the visual cortex (MT/V5) area of a monkey while receiving random-dot stimuli [205]. Details on the electrophysiological recording setup is described elsewhere [206].

In the case of real data (unlike synthetic data), the true underlying rate function (i.e. ground truth) is unknown. Therefore, the ground truth underlying rate in the real data was estimated by averaging and smoothing the spike counts across many similar trials. This procedure is similar to the work of Cunningham *et al.* [189]. We assumed that neurons respond similarly upon given the same tasks/stimuli, e.g. moving hand to the same target direction or observing the same visual stimuli. These similar trials were selected such that each trial contains greater or equal to 50 spikes/s within observation interval (1 s for the Flint_2012 dataset, 2 s for the nsa2004.1 dataset). This limited number of spikes was taken on the assumption that neurons likely fire more spikes when performing tasks or receiving stimuli. We considered only neurons that satisfy this criterion in more than 30 trials to obtain sufficiently small error as observed in multi-trial synthetic data (Figure 3.8). To this end, we obtained 47 (4) sub-datasets with a total of 1791 (134) trials for the Flint_2012 (nsa2004.1) dataset. In the Flint_2012 dataset, we aligned the spiking responses over same-direction reaching tasks to the time when the monkey started the actual hand movement (indicated by the cursor movement). To make the observation interval the same from inherently different trial duration for each trial, we used on average of 200 ms before and 800 ms after the movement (total duration of 1 s). In the nsa2004.1 dataset, the duration was fixed to 2 s and the spiking responses were aligned to the time when random moving dot stimuli was first presented to the monkey.

To estimate the ground truth underlying rate, we employed trial averaging and smoothing processes. The trial averaging process was done by superimposing the spike trains from individual neurons and averaging the spike counts within predefined bin interval (10 ms) across all similar trials. This averaging process still resulted in a coarse or jagged estimate. To get a smooth estimate of the underlying rate, a smoothing process was performed using BARS.

BARS was selected because, based on the synthetic data, it demonstrated the most superior performance in multi-trial cases with a large number of trials (see Figure 3.8). In these multi-trial cases where there exists spike train variability across trials, we increased the sample iteration to 30,000 and burn-in samples to 5,000 to ensure the convergence of results. One of the advantages of BARS is that it provides the output estimate along with its credible interval. When computing MISE, we took into account this uncertainty and used 95% credible interval. Before squaring and integrating across observation interval, we normalised the error between the ground truth underlying rate obtained from multi-trials (act as a reference) and single-trial estimated firing rate from the method of interest by dividing it with upper or lower credible interval. The upper (lower) interval was used when the estimated firing rate is larger (smaller) than the reference. By doing so, we imposed more (less) weight when the credible interval is smaller (larger) to adjust the uncertainty brought by the BARS estimation. We call this normalised MISE as a weighted MISE (WMISE) and formulate it as follows,

$$\text{WMISE} \approx \Delta t \sum \mathbb{E}\left[\frac{\hat{\lambda}(t) - \lambda(t)}{C(t)}\right]^2 \tag{3.27}$$

where $C(t)$ is set to the upper credible interval when $\hat{\lambda}(t) \geq \lambda(t)$ and the lower credible interval when $\hat{\lambda}(t) < \lambda(t)$. These upper and lower credible intervals calculated by BARS are not uniform.

We compared the performance across methods under WMISE function using three different metrics. First, we investigated the WMISE performance across the total number of trials. Based on 1791 single-trial firing rate estimation from 47 sub-datasets in the Flint_2012 dataset, BAKS produced the smallest WMISE mean (10.34) and median (6.85) as shown in Figure 3.11a. BAKS also yielded the best performance (mean = 19.77 and median = 13.56) in the case of nsa2004.1 dataset with a total of 134 trials from 4 sub-datasets (Figure 3.11d). Second, we measured the number of times (in %) one method outperformed all the other methods. In both Flint_2012 and nsa2004.1 datasets, BAKS outperformed the other methods more frequently (41.99% and 42.54% respectively) as observed in Figures 3.11b and 3.11e. In these cases, OKS came as the second best method with 30.65% and 17.16%. Third, we assessed the

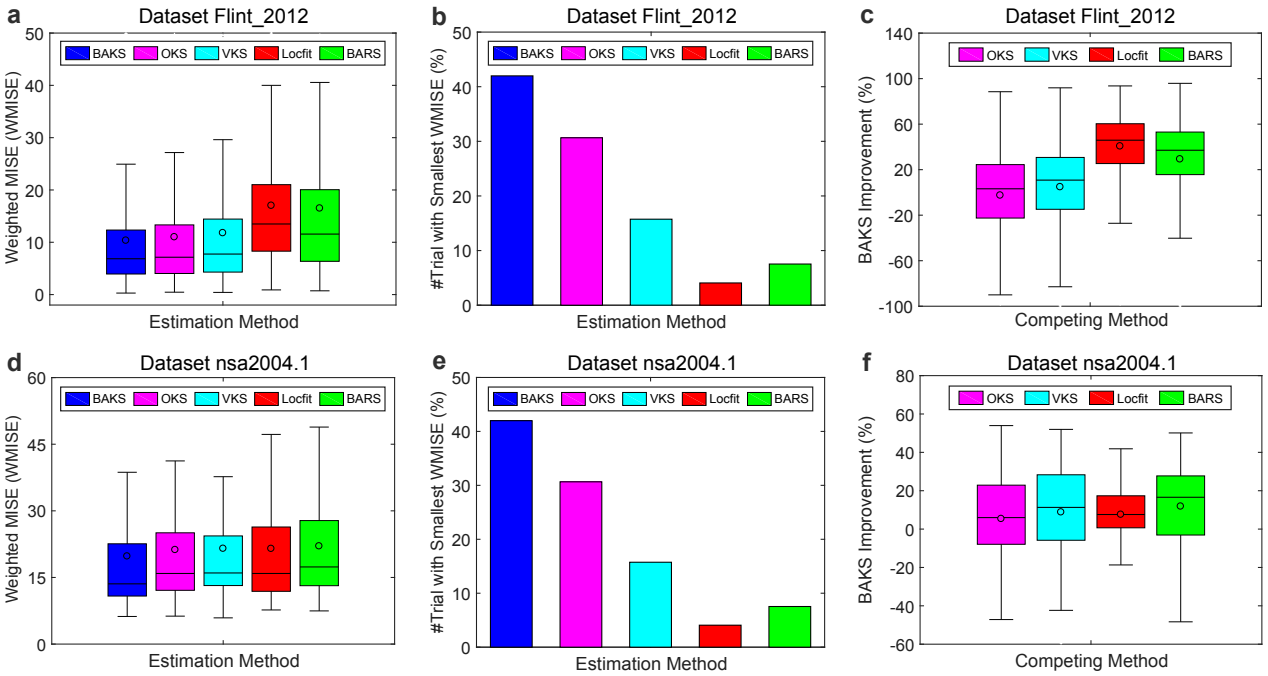


Figure 3.11 WMISE comparison across all firing rate estimation methods on real neural data. (a) Average WMISE comparison across all trials in Flint_2012 dataset. (b) Number of times (in %) BAKS outperforms other methods in Flint_2012 datasets. (c) Single-trial performance improvement made by BAKS over competing methods in Flint_2012 datasets. (d)-(f) Similar to that of (a)-(c) but with nsa2004.1 dataset. In each boxplot, black lines show the medians; black circles indicate the means; coloured solid boxes represent interquartile ranges; whiskers extend 1.5× from upper and lower quartiles.

improvement (WMISE decrease in %) per trial made by BAKS against other methods. Figures 3.11c and 3.11f describe the statistical summary of BAKS performance compared to other methods for the Flint_2012 and nsa2004.1 datasets, respectively. A positive (negative) value means that the BAKS method outperforms (is outperformed by) the others. As illustrated in Figure 3.11c, BAKS performed better than other methods except against OKS (see magenta boxplot; mean = -2.58% and median = 3.20%). This seems to contradict the results when using the first and second metrics, in which BAKS outperformed all other methods. To investigate this contradictory, we analysed the performance of BAKS against OKS across 1791 trials. It turns out that although BAKS outperforms OKS more frequently, in a few trials, the magnitude of improvement made by OKS against BAKS is significantly large. In the nsa2004.1 dataset, BAKS consistently outperformed all other methods (Figure 3.11f). These results are in good agreement with the two other metrics and the results obtained from the synthetic datasets. As a summary, BAKS on average performs well compared to other methods in both real and

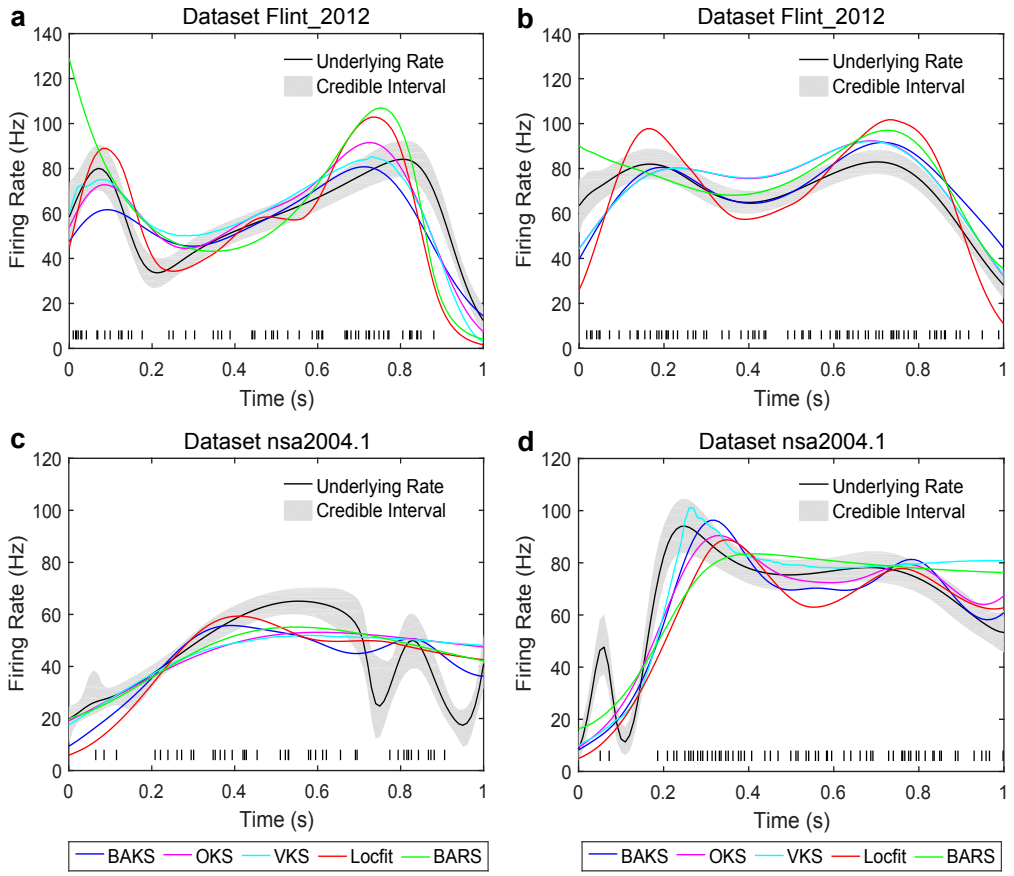


Figure 3.12 Firing rate estimates comparison for real neural data. (a) Firing rate estimates from 'N184' of Flint_2012 dataset (b) Firing rate estimates from 'E164' of Flint_2012 dataset (c) Firing rate estimates from 'j032_25.6' of nsa2004.1 dataset. (d) Firing rate estimates from 'j032_51.2' of nsa2004.1 dataset. Black lines and grey-shaded regions indicate the 'true' underlying rates and their 95% credible intervals, respectively. Black raster in the bottom of each plot represents a single spike train generated from the associated underlying rate.

synthetic datasets. Some examples of single-trial firing rate estimations from all methods for both datasets are displayed in Figure 3.12. Figures 3.12a and 3.12b show the firing rate estimates from 'N184' and 'E164' cases in the Flint_2012 dataset. Figures 3.12c and 3.12d show the firing rate estimates from 'j032_25.6' and 'j032_51.2' cases in the nsa2004.1 dataset.

3.4 Discussion

In this chapter, we propose a new method, referred to as BAKS, for estimating single-trial neuronal firing rates. BAKS employs a kernel smoothing technique with adaptive bandwidth. This differs from other kernel-based firing rate estimation methods in that the bandwidth parameter is adaptively determined by an empirical Bayesian approach. BAKS has been developed with

the motivation to estimate firing rate from single spike train generated from underlying rate function that dynamically changes over the observed duration (i.e. non-stationary).

We tuned the parameter of BAKS using synthetic spike train data stochastically sampled from 3 rate functions (as the representation of non-stationary underlying processes), termed as chirp, sine, and sawtooth rate functions. We then evaluated the performance of BAKS using five synthetic datasets, representing various setting and combination of underlying rate functions along with their intensity and frequency variations, ISI shape (parameter γ), and the number of trials. Extensive simulations demonstrated that BAKS performed better than two other kernel-based methods (OKS and VKS) and two generalised nonparametric regression methods (Locfit and BARS). On average, BAKS outperformed the other methods in single-trial estimation across various settings. The adaptive bandwidth featured in the BAKS can adjust the different spike densities within the observation interval. The results suggest that BAKS is suitable for the single-trial analysis of neural data. The flexibility of BAKS was evaluated spike train data generated from the same models with different values of shape parameter ($\gamma = \{0.25, 0.5, 0.75, 1, 2, 3, \cdots 10\}$). These various values represent diverse neuronal activities which include bursting, irregular, and regular spiking. BAKS was also tested on synthetic datasets generated from two other underlying rate functions, namely Gaussian-damped sinusoidal rate and square rate functions. Consistent results were obtained despite using different characteristics of spike trains. BAKS does not assume a specific distribution on the spike train; rather, it uses appropriately chosen prior distribution on the bandwidth parameter. The prior distribution of the bandwidth is derived from Gamma prior distribution on the precision parameter (inverse of square bandwidth). The precision parameter describes how concentrated observed data are around the means of Gaussian kernel, which are set to the spike times. Since these spike times (i.e. the sum of ISI) are conveniently modelled with Gamma distribution [177], [189]–[192], the precision parameter is also assumed to be Gamma distribution. This choice has been shown to yield good performance. On the other hand, all other competing methods (OKS, VKS, Locfit, and BARS) use a Poisson assumption [174]–[176], which is less likely for the case of single-trial spike train; neurons have certain properties (e.g. refractory and bursting) that cannot be described by the Poisson model [177], [178]. Numerous works

have shown the inadequacy of the Poisson model and proposed other more biophysically plausible models (e.g. IG and IIG) [168], [177], [189]–[192], [198]. The deviation from Poisson assumption under single-trial cases could lead OKS, VKS, Locfit and BARS to poor performance [174], [179]. Some topics of interest in neuroscience use multi-trial spike train to obtain the firing rate. To investigate the implication of increasing number of spikes within the same duration, we compared the performance of BAKS across multi-trial cases. The results showed that all methods revealed a similar trend; the increasing number of trials up to a certain value (threshold) improves the performance. However, as the number of trial increases, the rate of improvement of each method and the threshold value differ from each other. BARS produced the most significant performance improvement with the increasing number of trials, which is driven by two factors. First, BARS is built on the assumption that spike counts within the bin intervals follow a Poisson distribution. This assumption is suitable for multi-trial cases since superimposed spike trains across trials approximate a Poison process [174], [178]. Second, BARS employs a fully Bayesian approach and locality heuristic procedure to update the prior distribution of parameter (knots configuration) [176]. With the increasing number of trials (i.e. number of spikes), BARS can effectively find the optimal knots configuration regardless of the prior at the cost of high computational complexity. Nevertheless, in comparison to OKS, VKS, and Locfit methods, BAKS yielded better performance. The overall results suggest that BAKS is good at estimating firing rate from a low to a moderate number of spikes (represented by single or few trials) from non-stationary underlying rate functions. BAKS performed significantly better in the cases of sawtooth and square rate functions, indicating its suitability for estimating firing rate from discontinuous underlying rate functions.

Next, BAKS was also tested on real neural data recorded from the motor and visual cortex of NHPs. The motor neural data is associated with centre-out reaching tasks, whereas the visual neural data is associated with moving random-dot visual stimuli. Measuring the performance in real neural data is challenging due to unknown underlying rate. Hence, the underlying rate was estimated using multi-trial cases on the assumption that neurons respond similarly upon given the same tasks/stimuli. This procedure is similar to Cunningham's work [189]. However, our procedure differs from [189] in that we use BARS instead of PSTH for

the smoothing process. In practice, neuronal response may considerably differ across similar trials. To minimise large variation in the spike trains, subsets from two datasets (Flint_2012 and nsa2004.1) were selected with constraints explained in the previous section. BARS was chosen to obtain a smooth estimate of the ground truth underlying rate because it yielded the smallest MISE in multi-trial cases of synthetic data. To account for the estimation uncertainty produced by BARS, weighted MISE (WMISE) was used for comparison. From a total of 6 cases (3 metrics and 2 real datasets), BAKS outperformed all other methods in five cases. The overall results show that, on average, BAKS performs well compared to all other competing methods. This is in good agreement with the results obtained from single-trial synthetic data, which further demonstrates the effectiveness of the proposed method in estimating single-trial neuronal firing rates.

The selection of prior parameter (α) value of BAKS is crucial since it will impact the performance of the firing rate estimation. The value of α was tuned by minimising MISE on synthetic dataset during the training phase. This dataset was generated from three rate functions (chirp, sine, sawtooth) with medium intensity and frequency. During the testing phase, we used datasets with various settings to evaluate the robustness of BAKS with this tuned parameter ($\alpha_t = 4$). According to Equation (3.13), the mean of prior bandwidth is inversely proportional to the value of α. In the cases of low-frequency rate functions with the same intensity as of during the tuning phase, the spike density tends to decrease, which in turn requires larger bandwidth. To adapt to this, the value of α that minimises MISE tends to become smaller. Since we use a fixed α_t, this results in poor performance of BAKS. On the other hand, in the cases of high frequency, although there was performance degradation, BAKS still showed better performance compared to all other methods. This could be attributed to relatively small performance degradation resulting from the use of α_t as opposed to the larger value of α (for adapting to higher spike density). As can be seen in Figure (3.3), the performance difference from the use of $\alpha_t = 4$ to the right (larger value of α) is smaller than to the left (smaller value of α). In other cases related to different values of intensity (the frequency is kept the same), BAKS performed consistently better than other methods. The change in intensity (thus the associated bandwidth) could be partially compensated by the change in

Table 3.3 Comparison summary of BAKS against other methods

	BAKS	OKS	VKS	Locfit	BARS
Adaptability to underlying dynamics	✓	–	✓	–	✓
Bayesian/probabilistic approach	✓	–	–	–	✓
Automatic selection of smoothing parameter	✓	✓	✓	–	✓
Single trial (low to moderate number of spikes)	◇◇◇	◇◇	◇◇	◇◇	◇◇
Multi trials (large number of spikes)	◇◇	◇◇	◇◇	◇	◇◇◇
Computational complexity (runtime)	◇◇◇	◇◇◇	◇◇	◇◇◇	◇

More diamonds mark (◇) indicates better (desirable) property.

scale parameter (β) which was set to a function proportional to the number of spikes. This indicates that the performance of BAKS is less (more) sensitive to α_t value when there is a low (high) variation of frequency (i.e. temporal fluctuation) of the underlying rate function.

In practice, the value of α parameter of BAKS can be tuned using synthetic datasets generated from spike train model whose parameters derived from a statistical summary of real datasets of interest. By analysing repeated trial spike trains on real neural datasets with the same tasks/stimuli, we can estimate important underlying parameters such as temporal fluctuation, mean and dynamic range of intensity, ISI characteristic, and shape of the rate function. This information can then be fed to the spike train model to generate synthetic datasets for the tuning purpose. Using the same principle of minimising MISE, we can tune α and use the tuned value for firing rate estimation during the testing phase.

BAKS offers simplicity similar to the standard kernel-based method and is effective in grasping sudden and slow changes of firing rate in different regions within the observation interval. Unlike BARS which is computationally demanding, BAKS is relatively fast owing to an analytical expression of bandwidth posterior density. This analytical expression leads to the adaptive bandwidth determined in an exact way (not the numerical approximation), reducing the computational complexity. With good performance and relatively low complexity, BAKS is suitable for research that requires single-trial firing rate estimation, such as neural encoding mechanism in cognitive-related tasks and neural decoding in brain-machine interface (BMI) applications. As a summary, the comparison of BAKS with other methods is given in Table 3.3.

3.5 Conclusion

We have presented a simple yet accurate method, referred to as BAKS, for estimating single-trial neuronal firing rate based on a kernel smoothing technique with adaptive bandwidth. The key idea of BAKS is to consider the bandwidth parameter as a random variable under an empirical Bayesian framework. By using Bayes' theorem with the proper choice of kernel and prior distribution functions, the bandwidth can be adaptively determined in an exact and quick way. We have demonstrated, through extensive evaluations on both synthetic and real neural data, the good performance of BAKS compared to other established methods. This suggests that BAKS has the potential to improve single-trial analysis in neuroscience studies and decoding performance of spike-based brain-machine interfaces (BMIs).

4 | Enhanced MUA-based BMIs with BAKS and Deep Learning

This chapter approaches the decoding accuracy issue of MUA-based BMIs from both feature extraction and decoding algorithm perspectives. Many studies employ a simple binning method for the former and linear decoder for the latter. Here, we propose BAKS (introduced in Chapter 3) as a feature extraction method and long short-term memory (LSTM) as a decoding algorithm.

This chapter is an extended version of a published conference paper. It was submitted to *IEEE Access* and deposited to *techRxiv* preprint server.
N. Ahmadi, T. G. Constandinou, and C.-S. Bouganis, "Spike rate estimation using Bayesian adaptive kernel smoother (BAKS) and its application to brain-machine interfaces", in *40th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, 2018, pp. 2547–2550. [2]
N. Ahmadi, T. G. Constandinou, and C.-S. Bouganis, "Improved Spike-based Brain-Machine Interface Using Bayesian Adaptive Kernel Smoother and Deep Learning", in *techRxiv*, 2020. [3]

4.1 Introduction

MUA has been proposed by many researchers as an alternative BMI input signal to SUA [19], [22], [26], [27]. Most MUA-based BMIs utilise a binning method for extracting features (firing rates) and linear decoders (KF and WF) for decoding movement parameters. Binning estimates firing rates by counting the number of spikes within a predefined bin/window width. Despite being simple and fast, binning results in a coarse/noisy estimate of firing rates. As movements are usually smooth or continuous over time, a method that can yield a smooth estimate of firing rates could potentially improve decoding performance of MUA-based BMIs. Additionally, the

use of linear decoders with their inherent assumptions lead to suboptimal decoding performance since neural signals often exhibit non-linear, non-stationary, and non-Gaussian characteristics [207]. Therefore, it is highly desirable to develop a decoding algorithm that is robust against the above neural signal characteristics. One promising method is long short-term memory (LSTM), which does not require any assumption on the data. LSTM is a variant of recurrent neural networks (RNNs) that has achieved state-of-the-art performance in a wide range of tasks, especially those related to time-series data [208]. However, there has been no study that utilises LSTM for MUA-based BMIs.

This chapter aims to improve the decoding performance of MUA-based BMIs by simultaneously proposing BAKS as the firing rate estimation method and LSTM as the decoding algorithm. We evaluate BAKS and LSTM for decoding (offline) hand kinematics from neural data chronically recorded from the motor cortex (M1) area of a monkey. The proposed methods are compared against two other firing rate estimation methods (binning and Gaussian kernel smoother) and two other decoding algorithms (KF and WF). Furthermore, we compare the decoding performance between MUA and SUA using the proposed methods.

4.2 Methods

Block diagram of spike-based BMI system used in this chapter is illustrated in Figure 4.1. Before being fed to each decoder (KF, WF, and LSTM), the firing rates obtained from each method were standardised (i.e. z-transformed) to zero mean and unit variance, whereas the kinematic data were normalised to zero mean. The LSTM decoder was implemented using Keras/TensorFlow deep learning framework [209]. All the experiments and analyses were done in Python.

4.2.1 Neural recordings

Neural recordings were obtained from a public dataset deposited by Sabes lab [210]. The neural data were recorded from primary motor cortex (M1) area of an adult male Rhesus macaque monkey (*Macaca mulatta*), indicated by I. The recordings were made by using a 96-channel Utah microelectrode array (platinum contact, 400 kΩ nominal impedance, 400 μm interelectrode

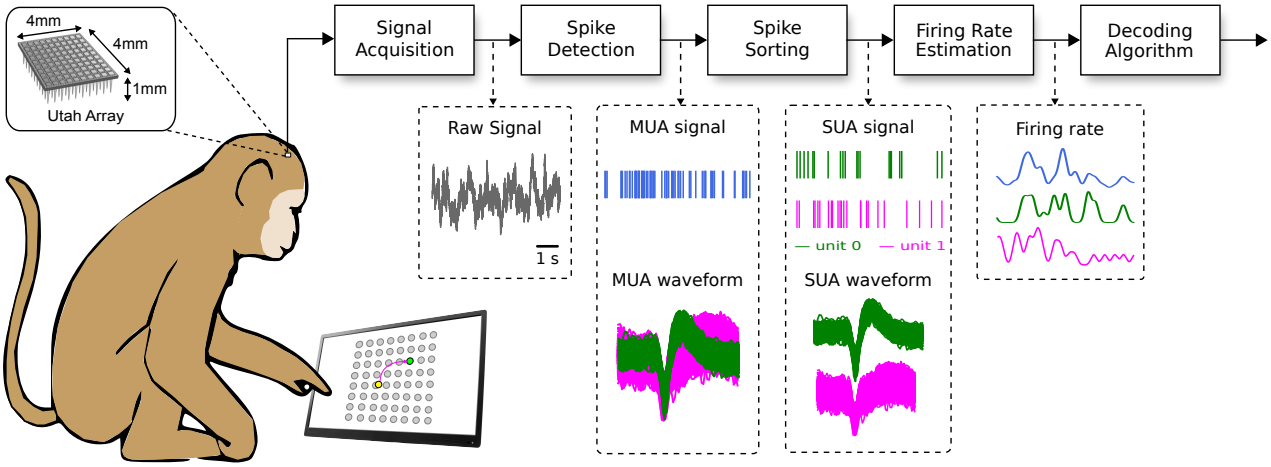


Figure 4.1 Block diagram of spike-based BMI

spacing, 1 mm electrode length) referenced to a silver wire placed under the dura (several cm away from the electrodes). The recordings were preamplified and filtered by using a 4th-order low-pass filter at 7.5 kHz with a roll-off of 24 dB per octave. The recordings were then digitised with 16-bit resolution at 24.4 kHz sampling rate. These digitised recordings are hereinafter referred to as raw neural signals. Details of the experimental setup including the subject, surgical procedure, and recording setup have been described elsewhere [211]. The neural data used in this chapter consist of 26 recording sessions spanning 7.3 months between the first (I20160627_01) and last (I20170131_02) sessions. The first and last sessions correspond to 110 and 328 days since the electrode implantation, respectively. The recording duration varies from 6 to 13.6 minutes with an average of 8.88 ± 1.96 minutes (mean ± SD).

4.2.2 Behavioural task

The behavioural task has been previously described in detail elsewhere [211]. Briefly, the monkey was trained to perform a point-to-point task, that is, to reach randomly drawn circular targets which were uniformly distributed around an 8×8 square grid, as illustrated in Figure 4.1. The target was acquired when the monkey reached the target using his fingertip and held it for 450 ms. Upon every target acquisition, a new random target was presented immediately without an inter-trial interval. The fingertip position of the reaching hand and the target position (in x-y Cartesian coordinates with mm unit) were both sampled at 250 Hz. The position data were then low pass filtered with a non-causal, 4th-order Butterworth filter at 10 Hz to reject sensor

noise. Velocity and acceleration data were computed by using first and second derivative of the position data.

4.2.3 Spike signal processing

Single-unit activity (SUA)

The raw neural signals were band-pass filtered with a causal, 4th-order Butterworth IIR filter between 500 to 5000 Hz. Spikes were then detected whenever the absolute value of the band-passed signals crossed a threshold value (typically set to between 3.5 and 4.0 times the standard deviation). SUA was obtained by sorting (i.e. classifying) the detected spikes into distinct putative single units via principal component analysis and template matching. This means each channel could contain more than one SUA. More detailed information on the spike detection and sorting processes can be found in [211]. We Only included SUA with spike rates above 0.5 Hz, which resulted in a varying number of SUA ranging from 91 to 157 units with an average of 125.73 ± 13.95 units (mean $\pm$ SD) across 26 recording sessions.

Multi-unit activity (MUA)

In this chapter, MUA is defined as all the spikes detected through threshold crossing technique. To obtain MUA, we performed spike detection (as described in the above SUA processing) but without spike sorting. Thus, for each channel, we obtained a maximum of one MUA. Similar to that of SUA, only MUA with spike rates exceeding 0.5 Hz were included. This yielded a number of MUA that varied from 77 to 95 units with an average of 87.08 ± 4.44 (mean $\pm$ SD) across 26 recording sessions.

4.2.4 Firing rate estimation

Firing rate estimation from each method was performed in an overlapping fashion to obtain a feature sample every 4 ms (to match the timescale of the kinematic data). Each firing rate estimation method is briefly described as follows:

- *Binning method* estimates firing rate by computing the number of spikes within a 256 ms rectangular window.

- *Gaussian kernel smoother* estimates firing rate by convolving spike train with a Gaussian kernel with 256 ms window width and 110 ms bandwidth (standard deviation).

- *Bayesian adaptive kernel smoother (BAKS)* estimates firing rate from spike train with 256 ms Gaussian window, parameter shape (α) = 4, and parameter scale (β) = $n^{4/5}$; n represents the number of spikes. The formula of BAKS and its parameter setting has been described in Section 3.2.2 of Chapter 3.

4.2.5 Decoding algorithm

Kalman filter (KF)

KF, primarily developed by Rudolf E. Kalman in 1960s, is one of the most well-known estimation algorithms used to estimate states of interest from a sequence of indirect and noisy measurements. It has been employed in numerous BMI research to decode limb kinematics or kinetics from neural signals [44], [46], [55], [75], [99], [143]. KF combines process and measurement models with the assumption that both models are linear and Gaussian. The process model defines the evolution of the state (assumed to be Markovian) from previous timestep $t-1$ to current timestep t as:

$$\mathbf{y}_t = \mathbf{F}\mathbf{y}_{t-1} + \mathbf{q}_t, \quad \mathbf{q}_t \sim \mathcal{N}(0, \mathbf{Q}) \tag{4.1}$$

where $\mathbf{F}$ denotes the state transition matrix that linearly maps the previous state $\mathbf{y}_{t-1}$ to the current state $\mathbf{y}_t$, $\mathbf{q}_t$ is the process noise that is assumed to be drawn from Gaussian distribution with zero mean and covariance $\mathbf{Q}$. The state $\mathbf{y}_t$ at timestep t is formulated as:

$$\mathbf{y}_t = [s_x, s_y, v_x, v_y, a_x, a_y]_t^T \tag{4.2}$$

where $s_x, s_y, v_x, v_y, a_x, a_y$ denote the hand position, velocity, and acceleration in x and y directions, respectively, and d refers to the dimension of the state. The measurement model describes the relationship between the state and the measurement at current timestep t which

is mathematically expressed as:

$$\mathbf{x}_t = \mathbf{H}\mathbf{y}_t + \mathbf{r}_t, \quad \mathbf{r}_t \sim \mathcal{N}(0, \mathbf{R}) \tag{4.3}$$

where $\mathbf{H}$ denotes the measurement matrix that linearly transform the state $\mathbf{y}_t$ into the measurement $\mathbf{x}_t$, $\mathbf{r}_t$ is the measurement noise that is assumed to be drawn from Gaussian distribution with zero mean and covariance $\mathbf{R}$.

KF consists of two steps, *prediction* and *update*, which are performed in a recursive manner. In the prediction step, *a priori* state estimate ($\hat{\mathbf{y}}'_t$) and *a priori* error covariance ($\mathbf{P}'_t$) at the current timestep are predicted from previous state estimate ($\hat{\mathbf{y}}'_{t-1}$) and error covariance ($\mathbf{P}_{t-1}$) using following formula:

$$\begin{aligned} \hat{\mathbf{y}}'_t &= \mathbf{F}\hat{\mathbf{y}}_{t-1} \\ \mathbf{P}'_t &= \mathbf{F}\mathbf{P}_{t-1}\mathbf{F}^T + \mathbf{Q} \end{aligned} \tag{4.4}$$

In the update step, *a posteriori* state estimate ($\hat{\mathbf{y}}_t$) and *a posteriori* error covariance ($\mathbf{P}_t$) at the current timestep are obtained from the predicted state estimate and error covariance combined with the information from the current measurement using equations below:

$$\begin{aligned} \hat{\mathbf{y}}_t &= \hat{\mathbf{y}}'_t + \mathbf{K}_t(\mathbf{x}_t - \mathbf{H}\hat{\mathbf{y}}'_t) \\ \mathbf{P}_t &= (\mathbf{I} - \mathbf{K}_t\mathbf{H})\mathbf{P}'_t \end{aligned} \tag{4.5}$$

where $\mathbf{K}_t$ is the Kalman gain that represents how much weight given to the measurements to refine the current state estimate.

Parameters $\mathbf{F}$, $\mathbf{H}$, $\mathbf{Q}$, and $\mathbf{R}$ were assumed to be invariant over time and were estimated using least square regression with ℓ_2 or Tikhonov regularisation (i.e. ridge regression) as follows:

$$\begin{aligned} \mathbf{F} &= \mathbf{Y}_2\mathbf{Y}_1^T(\mathbf{Y}_1\mathbf{Y}_1^T + \alpha\mathbf{I})^{-1} \\ \mathbf{H} &= \mathbf{X}\mathbf{Y}^T(\mathbf{Y}\mathbf{Y}^T + \alpha\mathbf{I})^{-1} \\ \mathbf{Q} &= (\mathbf{Y}_2 - \mathbf{F}\mathbf{Y}_1)(\mathbf{Y}_2 - \mathbf{F}\mathbf{Y}_1)^T/(N-1) \\ \mathbf{R} &= (\mathbf{X} - \mathbf{H}\mathbf{Y})(\mathbf{X} - \mathbf{H}\mathbf{Y})^T/N \end{aligned} \tag{4.6}$$

where α denotes the ridge regression parameter and N is the number of samples or data points

in the training data. The value of α was set to 0 (default value).

Wiener filter (WF)

WF, introduced by Norbert Wiener in 1940s, is a linear filter used to estimate an unknown signal of interest (e.g. kinematic data) from noisy observations (e.g. neural data) under the assumption of known stationary signal and additive noise. WF produces an optimal estimate of the desired signal in the sense of minimum mean squared error. WFs were used as a decoding algorithm in several prior animal and human BMI studies non-human primates [7], [8], [11], [14], [15], [42], [43]. An estimate of hand kinematics, $\hat{y}(t)$, through Wiener filtering is given by

$$\hat{y}(t) = \sum_{i=1}^{C} \sum_{\tau=0}^{L-1} w_i(\tau) x_i(t-\tau) \tag{4.7}$$

where $w_i(\tau)$ denotes a filter kernel and x_i represents features (i.e. firing rates) estimated from each method. C is the number of channels, whereas L is the number of lags/taps. The objective is to find the filter kernel weights $w(\tau)$ such that the squared error between actual and decoded hand kinematics is minimised. Equation (4.7) can be written in vector and matrix forms as

$$\mathbf{y} = \mathbf{X}\mathbf{W} \tag{4.8}$$

The direct solution for Equation (4.8) based on minimised squared error sense is given by

$$\mathbf{W} = (\mathbf{X}^T\mathbf{X} + \alpha\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y} \tag{4.9}$$

In this chapter, L was empirically set to 15 and the filter weights were estimated by using linear least squares with ℓ_2 regularisation. The regularisation parameter (α) was set to 0 (default value).

Long short-term memory (LSTM)

LSTM, proposed by Hochreiter and Schmidhuber in 1997 [212], is one of the most popular deep learning architectures and has achieved state-of-the-art performance in a wide range of machine

learning problems, especially those dealing with time-series data [208]. It addresses the problem of vanishing or exploding gradient commonly found in traditional recurrent neural networks (RNNs). LSTMs can effectively learn long-term temporal dependencies via a memory cell that maintains its state over time and gating mechanism that controls the flow of information into and out of the memory cell. In this chapter, we employed a commonly used variant of LSTM architectures which consists of three gates (forget, input, output) and a single memory cell. The states of LSTM components at timestep t are formulated as:

$$\begin{aligned}
\mathbf{f}_t &= \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f) \\
\mathbf{i}_t &= \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i) \\
\tilde{\mathbf{c}}_t &= \tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1} + \mathbf{b}_c) \\
\mathbf{o}_t &= \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o) \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)
\end{aligned} \tag{4.10}$$

where $\mathbf{x}, \mathbf{h}, \mathbf{f}, \mathbf{i}, \mathbf{o}, \mathbf{c}$ consecutively represent the input, output, forget gate, input gate, output gate, and memory cell. The operators σ and tanh denote the logistic sigmoid and hyperbolic tangent activation functions, respectively. The operator $\odot$ denotes the element-wise multiplication. Parameters that are learnt during training include input and recurrent weights ($\mathbf{W}, \mathbf{U}$) and bias vectors ($\mathbf{b}$).

We used 1-layer LSTM with number of timesteps (i.e. sequence length) empirically set to 2. The last timestep from the LSTM output was connected to a fully connected layer to obtain the final output. Other hyperparameters including number of units, number of epochs, batch size, dropout rate and learning rate were determined through hyperparameter optimisation from predefined ranges (see Table 4.1) using Bayesian optimisation package called Hyperopt [213]. The hyperparameter optimisation was conducted separately for each spike signal input and firing rate estimation method, and was run for 300 iterations. The resulting optimised hyperparameters are given in Table 4.1. To save the computational time of experiments, the hyperparameter optimisation was performed only once using the first recording session; for the subsequent sessions, we used the same hyperparameter configuration. The LSTM was trained

Table 4.1 Hyperparameter configuration of SUA- and MUA-driven LSTM decoders across firing rate estimation methods

Hyperparameter	Value range	SUA			MUA		
		Binning	Gaussian	BAKS	Binning	Gaussian	BAKS
Number of units	$\{50, 75, \cdots, 200\}$	200	200	200	150	175	200
Number of epochs	$\{2, 3, \cdots, 8\}$	6	7	8	6	5	6
Batch size	$\{64, 96, 128\}$	64	96	64	64	32	32
Dropout rate	$\{0, 0.1, \cdots, 0.5\}$	0	0.2	0.2	0	0.5	0.1
Learning rate	$\{5, 10, \cdots, 50\} \times 10^{-4}$	0.002	0.004	0.005	0.0035	0.0035	0.003

using RMSprop optimiser and root mean squared error (RMSE) loss function.

4.2.6 Performance evaluation and metrics

The dataset was firstly divided into 10 non-overlapping contiguous blocks of equal size which were categorised further into three sets: training (8 concatenated blocks), validation (1 block) and testing (1 block). The training set was used to train (fit) the decoder; the validation set was used to optimise hyperparameters of the decoder; the testing set was used to evaluate the optimised decoder. Decoding performance was evaluated using two commonly used metrics: (1) root mean square error (RMSE) and (2) Pearson's correlation coefficient (CC) [32], [33], [41]. RMSE represents the average magnitude of the decoding error; CC which is a translation and scale invariant metric assesses the linear correlation (shape similarity) between the true and decoded hand kinematics. RMSE and CC are formulated as follows:

$$\text{RMSE} = \sqrt{\sum_{i=1}^{N} (\hat{y}_i - y_i)^2 / N} \tag{4.11}$$

$$\text{CC} = \frac{\sum_{t=1}^{N} (y_t - \bar{y})(\hat{y}_t - \bar{\hat{y}}_t)}{\sqrt{\sum_{t=1}^{N} (y_t - \bar{y})^2} \sqrt{\sum_{t=1}^{N} (\hat{y}_t - \bar{\hat{y}}_t)^2}} \tag{4.12}$$

where y_t and $\hat{y}_t$ denote the true and decoded hand kinematics at timestep t, respectively, and N represents the total number of samples.

For each session, the mean and confidence interval of the decoding performance were evaluated on 10 different blocks within the testing set. To test statistical significance between a pair of different firing rate estimation methods or different decoding algorithms, a two-tailed paired t-test was used if the difference between the pairs follows normal distribution; otherwise

a two-tailed paired Wilcoxon signed-rank test was used. The significance level (α) was set to 0.05.

Boxplot was used to visualise the results. The horizontal line and circle mark inside each box represent the median and mean, respectively. The coloured solid box represents interquartile range (from 25th to 75th percentiles). The whisker extends 1.5 times the interquartile range.

4.3 Results

4.3.1 Performance comparison across firing rate estimation methods

First, we evaluated the decoding performance of each firing rate estimation method using MUA-driven KF decoder. Figures 4.2a and 4.2b present long-term decoding performance comparison across firing rate estimations methods, measured in RMSE and CC, respectively. We found that BAKS outperformed both binning and Gaussian kernel smoother (or Gaussian for short) methods in all metrics across all recording sessions (100%). The average decoding performance of each method was as follows (sorted from highest to lowest): BAKS (RMSE = 60.08 ± 1.44, CC = 0.73 ± 0.01), binning (RMSE = 62.70 ± 1.40, CC = 0.71 ± 0.01), and Gaussian (RMSE = 65.21 ± 1.60, CC = 0.70 ± 0.01). The RMSE and CC values are written in terms of mean $\pm$ standard error of the mean (SEM). Compared to binning method, BAKS yielded an average performance improvement of 4.2% (RMSE) and 3.3% (CC); however, Gaussian exhibited an average performance degradation of 3.9% (RMSE) and 1.0% (CC), as can be seen in Figures 4.2d and 4.2f. Statistical tests showed that the performance of BAKS differed significantly from that of other methods (in both RMSE and CC metrics) as shown in Figures 4.2c and 4.2e. Snippet examples of actual and decoded velocities (in x- and y-directions) from binning and BAKS are illustrated in Figures 4.2g-j.

To determine whether the above findings were also observed when using different decoding algorithms, we performed decoding comparison across firing rate estimation methods using WF and LSTM decoders. We found that BAKS consistently outperformed other methods as shown in Figure 4.3 for WF decoder and Figure 4.4 for LSTM decoder. There was a statistical

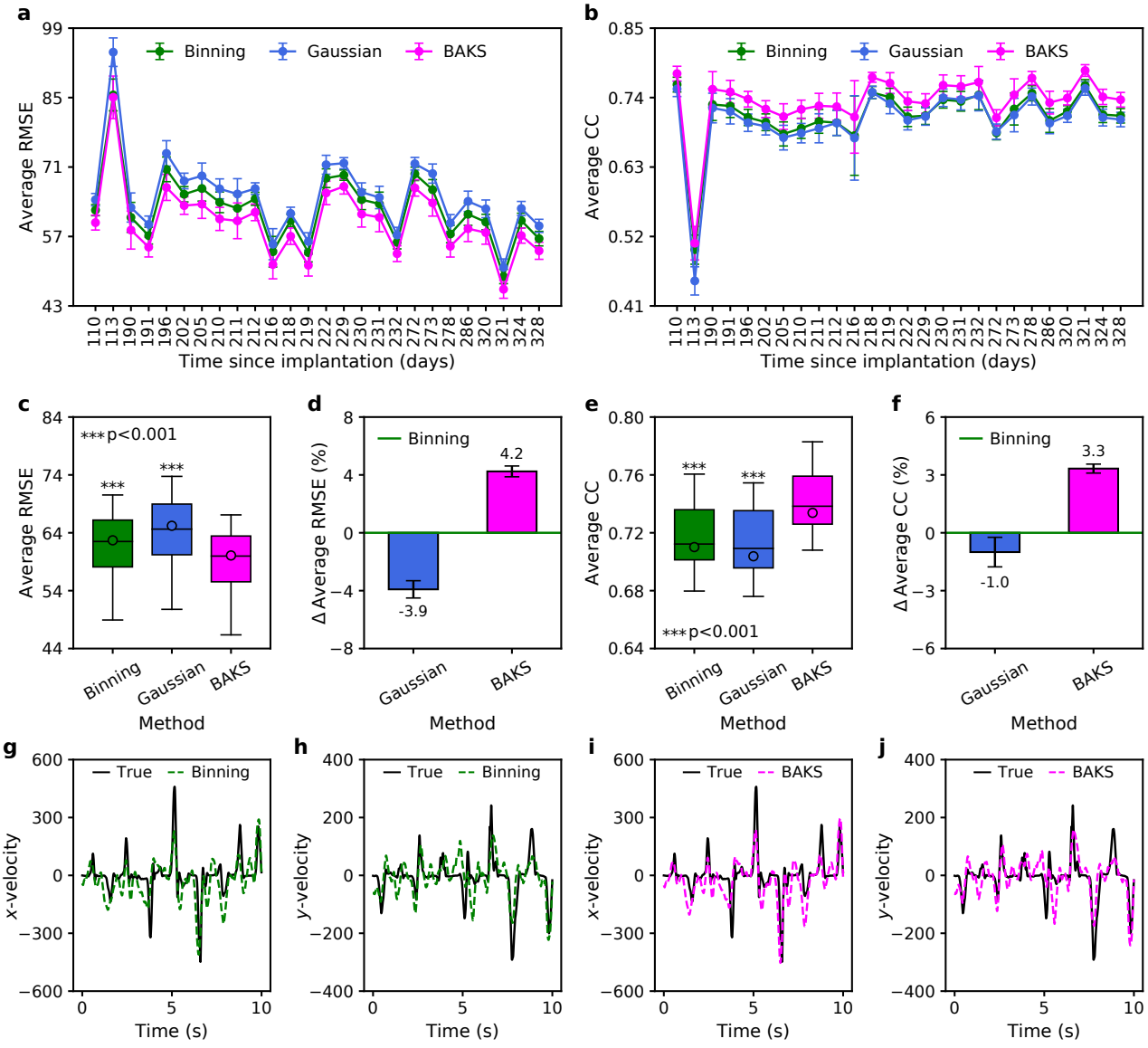


Figure 4.2 Comparison of decoding performance of MUA-driven KF decoder across different firing rate estimation methods. (a),(b) Performance comparison over long-term recording sessions measured in RMSE and CC, respectively. (c),(e) Boxplot comparison across sessions measured in RMSE and CC, respectively. Asterisks indicate firing rate estimation methods whose performances differed significantly from that of BAKS (*** $p<0.001$). (d),(f) Performance improvement/degradation (in percent RMSE and CC, respectively) relative to binning method. Positive (negative) value indicates performance improvement (degradation). Black error bars denote 95% confidence intervals. (g)-(j) Snippet examples of true and decoded velocities in x- and y- coordinates from different firing rate estimation methods (data from recording session I20170131_02).

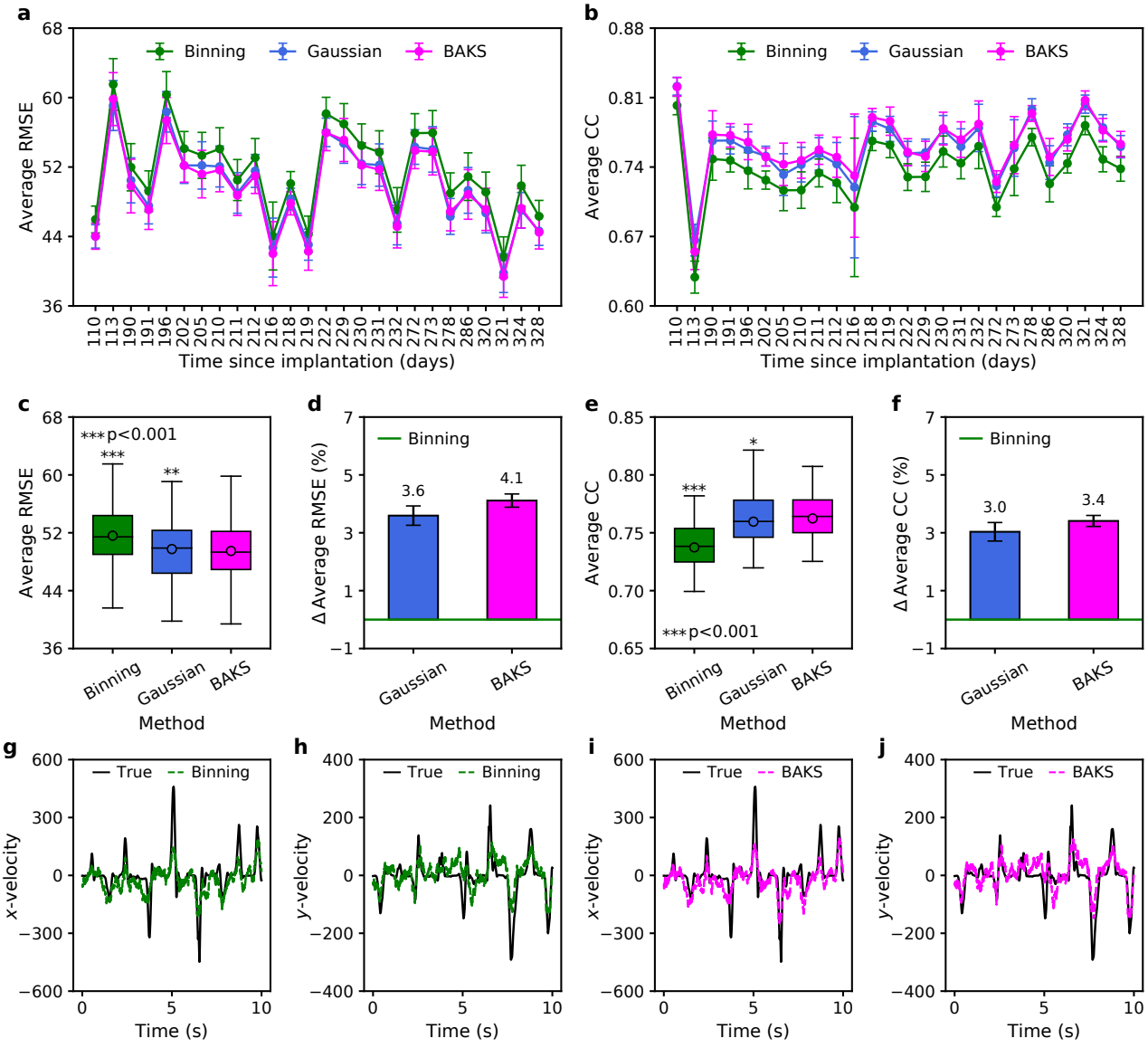


Figure 4.3 Comparison of decoding performance of MUA-driven WF decoder across different firing rate estimation methods. (a),(b) Performance comparison over long-term recording sessions measured in RMSE and CC, respectively. (c),(e) Boxplot comparison across sessions measured in RMSE and CC, respectively. Asterisks indicate firing rate estimation methods whose performances differed significantly from that of BAKS (* $p<0.05$, ** $p<0.01$, *** $p<0.001$). (d),(f) Performance improvement/degradation (in percent RMSE and CC, respectively) relative to binning method. Positive (negative) value indicates performance improvement (degradation). Black error bars denote 95% confidence intervals. (g)-(j) Snippet examples of true and decoded velocities in x- and y- coordinates from different firing rate estimation methods (data from recording session I20170131_02).

significant difference in decoding performance between BAKS and other methods as illustrated in Figures 4.3c,e (WF decoder) and Figures 4.4c,e (LSTM decoder). In the case of WF decoder, according the decoding performance, we found the following descending order: BAKS (RMSE = 49.49 ± 0.98, CC = 0.76 ± 0.01) > Gaussian (RMSE = 49.75 ± 0.96, CC = 0.76 ± 0.01) > binning (RMSE = 51.60 ± 0.99, CC = 0.74 ± 0.01). Relative to binning method, BAKS achieved 4.11% (3.41%) average performance improvement in RMSE (CC), whereas Gaussian method gained 3.59% (3.03%) average performance improvement in RMSE (CC), as shown in Figure 4.3d (4.3f). In the case of LSTM decoder, we obtained the following descending order: BAKS (RMSE = 41.56 ± 0.84, CC = 0.84 ± 0.01) > binning (RMSE = 41.87 ± 0.85, CC = 0.83 ± 0.01) > Gaussian (RMSE = 42.75 ± 0.84, CC = 0.83 ± 0.00). The average performance improvement achieved by BAKS relative to binning method in the case of LSTM was considerably smaller (RMSE = 0.76%, CC = 0.35%) than in the cases of KF and WF (see Figures 4.4d,f). Examples of actual and decoded velocities (in x- and y-directions) in the cases of WF and LSTM are plotted in Figures 4.3g-j and Figures 4.4g-j, respectively.

4.3.2 Impact of window width on decoding performance

Next, we assessed the impact of firing rate estimation methods under different window widths on decoding performance. For each firing rate estimation method, we varied the value of window width from 16 ms to 400 ms with an increment of 16 ms. We used KF and LSTM decoders for performance comparison. Figures 4.5a,b and 4.5c,d illustrate the impact of varying window widths on decoding performance using KF and LSTM decoders, respectively. Results showed that increasing the window width up to a certain value would improve the decoding performance; above this value, however, the decoding performance reached a plateau or tended to decrease. Binning and BAKS were found to reach the plateau level faster than Gaussian method in both KF and LSTM decoders. All firing rate estimation methods reached the plateau level faster in the case of LSTM decoder than in the case of KF decoder.

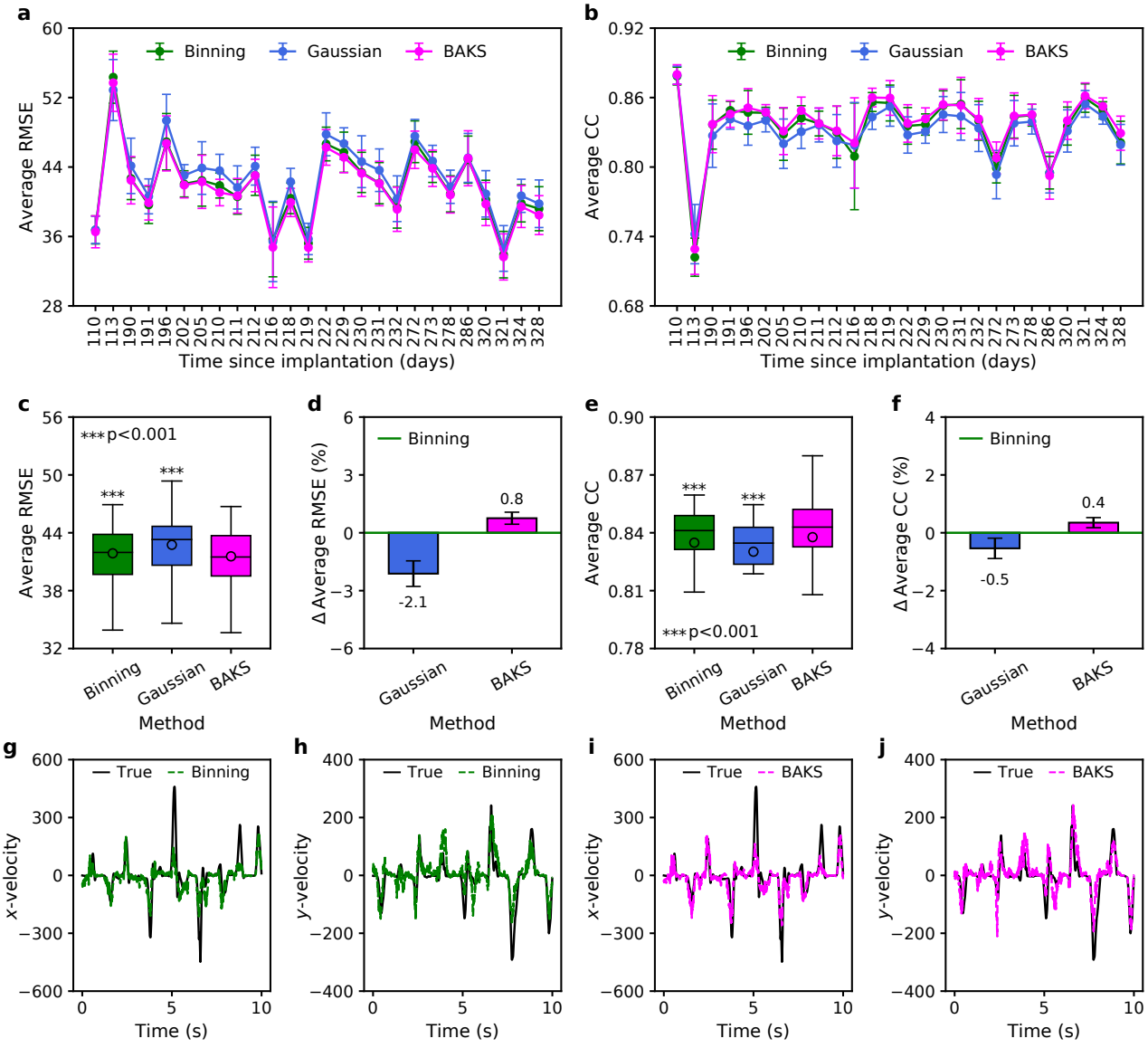


Figure 4.4 Comparison of decoding performance of MUA-driven LSTM decoder across different firing rate estimation methods. (a),(b) Performance comparison over long-term recording sessions measured in RMSE and CC, respectively. (c),(e) Boxplot comparison across sessions measured in RMSE and CC, respectively. Asterisks indicate firing rate estimation methods whose performances differed significantly from that of BAKS (*** $p<0.001$). (d),(f) Performance improvement/degradation (in percent RMSE and CC, respectively) relative to binning method. Positive (negative) value indicates performance improvement (degradation). Black error bars denote 95% confidence intervals. (g)-(j) Snippet examples of true and decoded velocities in x- and y- coordinates from different firing rate estimation methods (data from recording session I20170131_02).

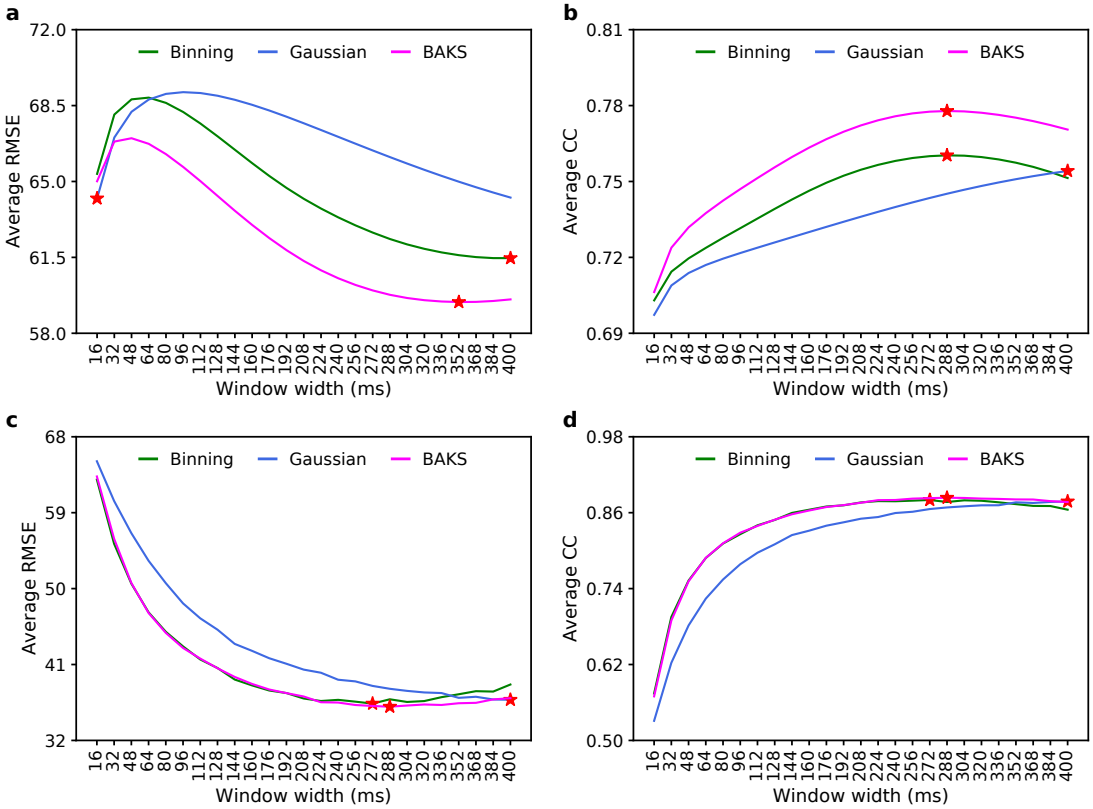


Figure 4.5 Decoding performance comparison across different firing rate estimation methods with varying window widths. (a),(b) Performance comparison using MUA-driven KF decoder measured in RMSE and CC, respectively. (c),(d) Performance comparison using MUA-driven LSTM decoder measured in RMSE and CC, respectively. Red stars indicate window widths that yielded the best decoding performances on validation set from session I20160627_01.

4.3.3 Performance comparison across decoding algorithms

Using BAKS as the firing rate estimation method, we then evaluated the decoding performance of different decoders (KF, WF, and LSTM). Figures 4.6a and 4.6b present the decoding performance comparison over 26 sessions in terms of RMSE and CC, respectively. We found that LSTM outperformed other decoders in all sessions. By sorting descendingly according to the decoding performance, we obtained the following order: LSTM (RMSE $= 41.56 \pm 0.84$, CC $= 0.84 \pm 0.01$), WF (RMSE $= 49.49 \pm 0.98$, CC $= 0.76 \pm 0.01$), and KF (RMSE $= 60.08 \pm 1.44$, CC $= 0.73 \pm 0.01$) as observed from Figures 4.6c and 4.6e. Relative to KF, LSTM yielded an average performance improvement of 30.64% (RMSE) and 14.56% (CC), whereas WF yielded an average performance improvement of 17.41% (RMSE) and 4.20% (CC) as shown in Figures 4.6d and 4.6f. Statistical tests showed that there were statistically significant differences in decoding performance between LSTM and other decoders ($p<0.001$).

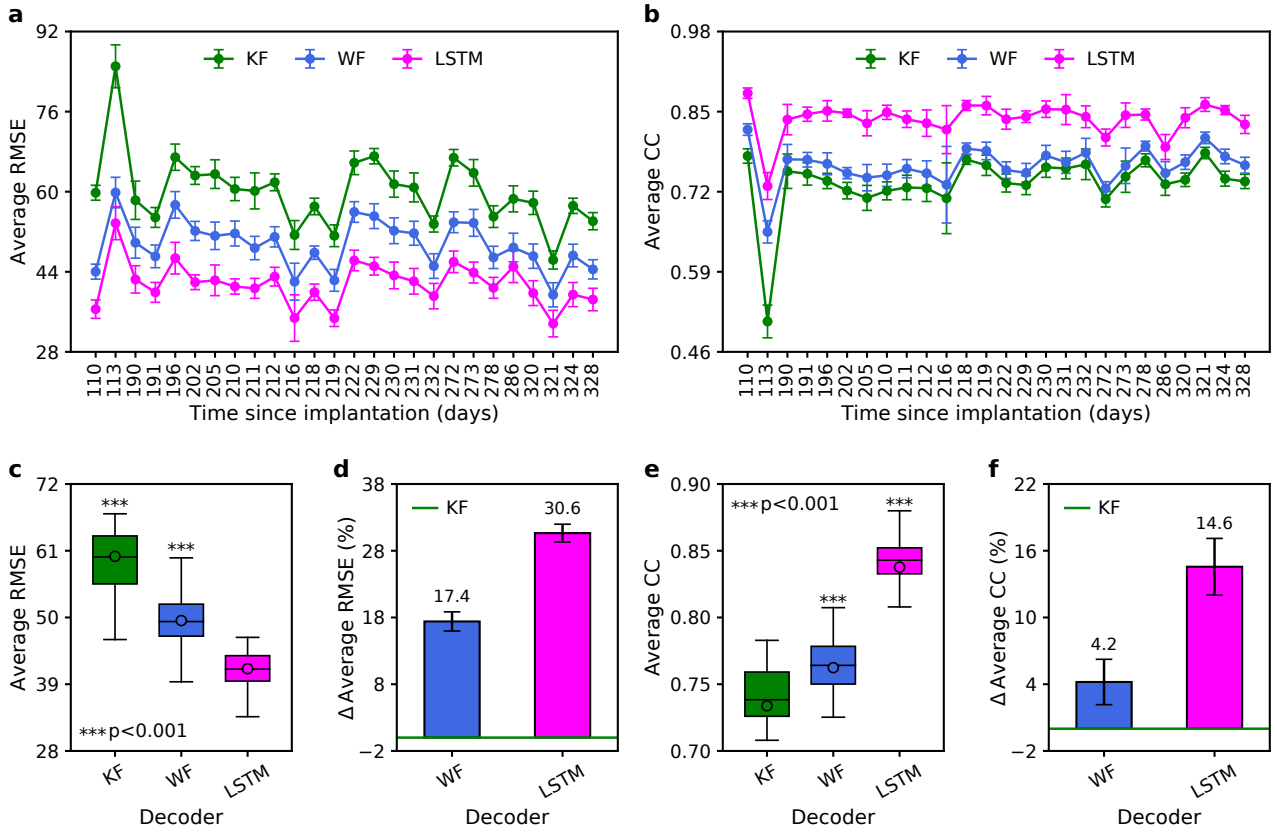


Figure 4.6 Comparison of decoding performance of MUA-driven decoders with firing rates computed by BAKS. (a),(b) Performance comparison over long-term recording sessions measured in RMSE and CC, respectively. (c),(e) Boxplot comparison across sessions measured in RMSE and CC, respectively. Asterisks indicate decoders that yielded statistically significant different performances from that of binning method (*** $p<0.001$). (d),(f) Performance improvement/degradation (in percent RMSE and CC, respectively) relative to binning method. Positive (negative) value indicates performance improvement (degradation). Black error bars denote 95% confidence intervals.

4.3.4 Decoding performance comparison between SUA and MUA

Lastly, we compared the decoding performance of MUA against SUA using BAKS coupled with different decoders. Figures 4.7a-b, Figures 4.7c-d, and Figures 4.7e-f plot the decoding performance comparison between MUA and SUA across 26 sessions for the case of KF, WF, and LSTM decoders, respectively. Extensive results revealed that MUA yielded significantly better decoding performance than SUA in all sessions regardless of the decoders. Relative to SUA, average performance improvement achieved by MUA from each decoding algorithm was as follows: KF (RMSE = 7.53%, CC = 6.34%), WF (RMSE = 5.94%, CC = 5.21%), and LSTM (RMSE = 7.26%, CC = 3.94%).

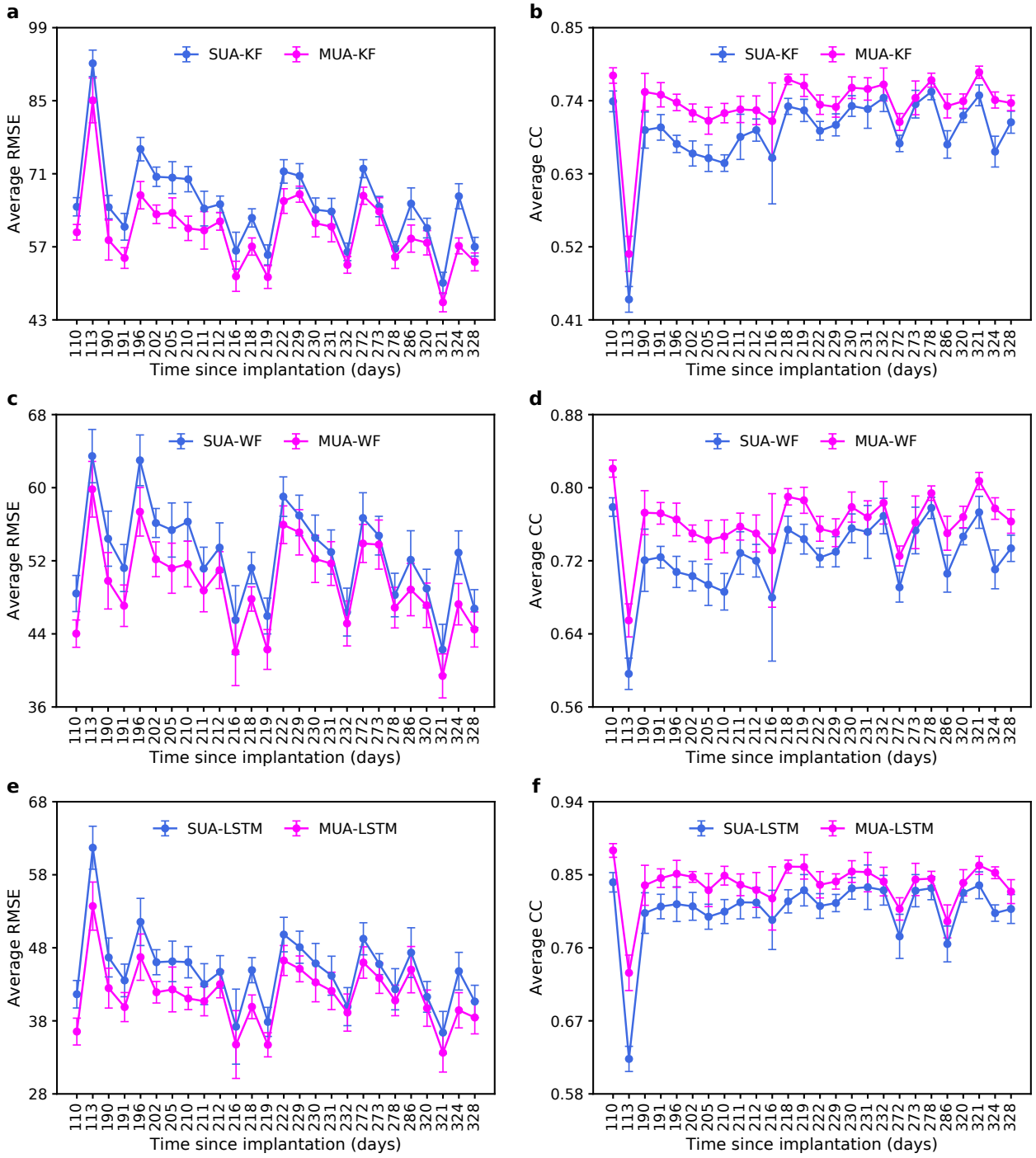


Figure 4.7 Comparison of decoding performance between SUA- and MUA-driven decoders with firing rates computed by BAKS. (a),(b) Long-term decoding comparison between SUA- and MUA-driven KF decoders measured in RMSE and CC, respectively. (c),(d) Long-term decoding comparison between SUA- and MUA-driven WF decoders measured in RMSE and CC, respectively. (e),(f) Long-term decoding comparison between SUA- and MUA-driven LSTM decoders measured in RMSE and CC, respectively.

4.4 Discussion

In this chapter, we attempt to improve the decoding performance of MUA-based BMI by proposing BAKS as a feature extraction method coupled with LSTM as a decoding algorithm. Firstly, we compared the decoding performance of BAKS against two commonly used feature extractions methods (binning and Gaussian) under three different decoding algorithms (KF, WF, and LSTM). Comparison results demonstrated that BAKS consistently outperformed other methods across different decoding algorithms. When varying the window width values, we also observed consistently better performance of BAKS compared to other methods. This could be attributed to BAKS' capability to obtain a smoother and more accurate estimate of firing rate. BAKS incorporates a data-driven and adaptive bandwidth parameter that allows for more accurate estimation of firing rate when there is a rapidly changing spike dynamic. On the other hand, both binning and Gaussian methods employ a fixed, predefined bandwidth parameter; thus, they cannot accurately estimate the firing rate from a spike train with rapidly changing spike dynamic. The good performance of BAKS comes at the expense of increased computational complexity compared to binning and Gaussian methods.

We found that the average performance improvement of BAKS relative to other methods was more significant in the case of linear decoders (KF and WF) than deep learning decoder (LSTM). This might be because LSTM can compensate for the differences in estimated firing rates through gradient-based optimisation algorithm during the training. In other words, LSTM is less sensitive to the smoothness and accuracy of estimated firing rates than both KF and WF. Using BAKS as the feature extraction algorithm, we then compared the performance of LSTM against KF and WF. Results showed that LSTM significantly outperformed both KF and WF, which demonstrates the effectiveness of LSTM in capturing the complex, non-linear relationship between neural signals and hand kinematic data.

Lastly, using BAKS coupled with LSTM, we compared the decoding performance of MUA to that of SUA. Empirical results revealed that MUA achieved significantly higher decoding performance than SUA. When using BAKS coupled with KF or WF, we also observed the same finding. These results contradict several prior studies where SUA was shown to yield

better decoding performance than MUA [22], [26], [27], [39]. It is difficult to find the exact reason to this contradiction due to the differences in recording setup, behavioural task, signal processing, decoding algorithm, etc. across studies. One possible explanation is that in our study, to obtain SUA, we only used well-isolated (sorted) spikes and discarded unsorted spikes (also known 'noise' or 'hash' units). Hash units contained all spikes that did not match any of the operator's defined templates used for spike sorting. Todorova *et al.* have recently shown that hash units contained some information about movement and discarding this information could degrade the decoding performance [38]. On the contrary, when computing MUA, we used all the detected spikes, including the hash units, which potentially contributed to improved decoding performance. In summary, overall results demonstrated that our proposed methods, BAKS and LSTM, achieved better decoding decoding performance than other methods.

4.5 Conclusion

We have proposed BAKS and LSTM and evaluated their impact on decoding performance of MUA-based BMI. We have shown that BAKS and LSTM significantly outperformed combinations of other firing rate estimation methods and decoding algorithms. This suggests the feasibility and the potential use of BAKS and LSTM for improving the decoding performance of MUA-based BMIs.

Part II

Improving Decoding Accuracy of LFP-based BMIs

5 | Decoding Hand Kinematics from LFPs Using Deep Learning

This chapter proposes decoding algorithms based on two different deep learning methods, namely recurrent neural network (RNN) and convolutional neural network (CNN). Particularly, we utilise an RNN with LSTM architecture (described in Chapter 4) driven by features that are extracted manually. Next, we exploit a special type of CNNs called temporal convolutional network (TCN) to learn the features automatically and decode hand kinematics from LFPs in a single unified approach. By doing so, we can eliminate the time-consuming process of manual feature extraction and learn features that are optimised for the decoding tasks.

This chapter is based on the following conference papers:
N. Ahmadi, M. L. Cavuto, P. Feng, *et al.*, "Towards a distributed, chronically-implantable neural interface", in *9th Int. IEEE/EMBS Conf. Neural Eng. (NER)*, 2019, pp. 719–724. [4]
N. Ahmadi, T. G. Constandinou, and C.-S. Bouganis, "Decoding hand kinematics from local field potentials using long short-term memory (LSTM) network", in *9th Int. IEEE/EMBS Conf. Neural Eng. (NER)*, 2019, pp. 415–419. [5]
N. Ahmadi, T. G. Constandinou, and C.-S. Bouganis, "End-to-end hand kinematic decoding from LFPs using temporal convolutional network", in *IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, 2019, pp. 1–4. [6]

5.1 Introduction

In recent years, there has been an increasing interest in using local field potential (LFP) as an alternative or complementary input signal for BMIs [25], [35], [37]. This arises from desirable features of LFP (stability and low bandwidth) and community is thus interested in attempting to retrieve informative content. Many LFP-based BMIs employ KF decoder with its restricting

assumptions. A more flexible and effective decoding method is required to improve the decoding performance of LFP-based BMIs. In addition, most LFP-based BMIs still make use of hand-crafted features. Several commonly used LFP features are local motor potential (LMP) and average spectral power on different frequency bands [22], [25], [35], [37]. LFP frequency bands are usually selected based on intuition, previous literature or visual inspection of the data [94] which can be suboptimal since it is not optimised for the decoding task. Furthermore, selecting the optimal set of features from all available features can be time-consuming. It is therefore desirable to have a unified method capable of performing automatic feature extraction and decoding that are optimised for the task at hand.

One such method is deep learning, which has demonstrated state-of-the-art performance in a wide range of real-world applications such as image and speech recognition, language translation, and medical diagnostics. Long short-term memory (LSTM) and Temporal convolutional network (TCN) are two variants of deep learning methods and have shown high performance in sequence modeling tasks [214]. Although there is an increasing adoption of deep learning in the field of BMIs (e.g. [60], [62]), no studies have been found that utilise both LSTM and TCN for hand kinematic decoding from LFPs.

This chapter aims to improve the decoding performance of LFP-based BMIs by utilising these two variants of deep learning methods. First, we propose LSTM as decoding algorithm driven by hand-crafted LFP features. Second, we propose an end-to-end system approach based on TCN that can automatically extract features and decode hand kinematics directly from raw LFP signals. We then compare decoding performance of the proposed methods against LFP-, SUA-, and MUA-driven KF decoders. Furthermore, we also compared the performance of the proposed methods to SUA- and MUA-driven LSTM decoders. Lastly, we analyse the TCN-extracted features and report the estimated inference time and power consumption of the TCN.

5.2 Methods

Figure 5.1 presents a block diagram of LFP-based BMI system using LSTM decoder. Another decoder, TCN, is schematically illustrated in Figure 5.2. Both LSTM and TCN decoders were

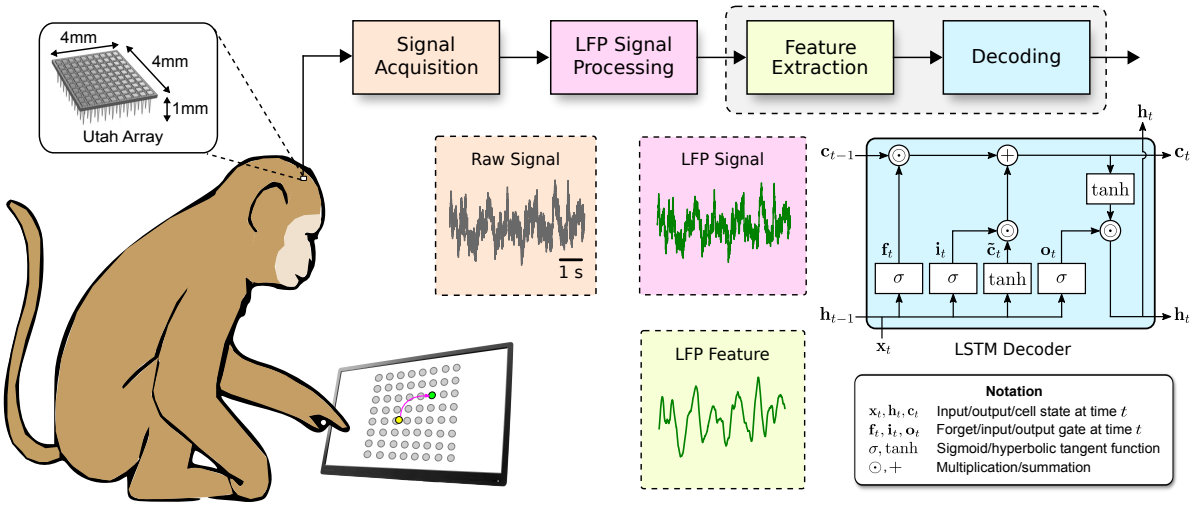


Figure 5.1 Block diagram of LFP-based BMI using LSTM decoder

implemented using Keras/TensorFlow deep learning framework [209]. All the experiments and analyses were conducted in Python and run on an Ubuntu 16.04 OS based machine with Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz and NVIDIA GeForce GTX 1080 GPU.

5.2.1 Neural recording and behavioural task

We used two recording sessions (session I: I20170124_01 and session II: I20170127_03) from the same neural dataset used in Chapter 4 (see Section 4.2.1 and 4.2.2). Briefly, the neural data were recorded using 96-channel Utah array while the monkey was performing self-paced reaching tasks without inter-trial intervals within 8-by-8 square grid. The neural data were sampled at 24.4 kHz and filtered with a 4^{th} oder low-pass filter at 7.5 kHz. Recording duration for session I and session II were 9 and 12 minutes, respectively. Fingertip position in x-y coordinates was sampled at 250 Hz. Velocity and acceleration were computed from position by using discrete derivative.

5.2.2 Neural signal processing

LFP signal processing

LFPs were obtained by low pass filtering the raw neural signals using a 4^{th} order Butterworth filter at 300 Hz and downsampling them to 1 kHz. The filtering was performed on the forward and backward directions to avoid any phase distortion. To remove common noise that is often

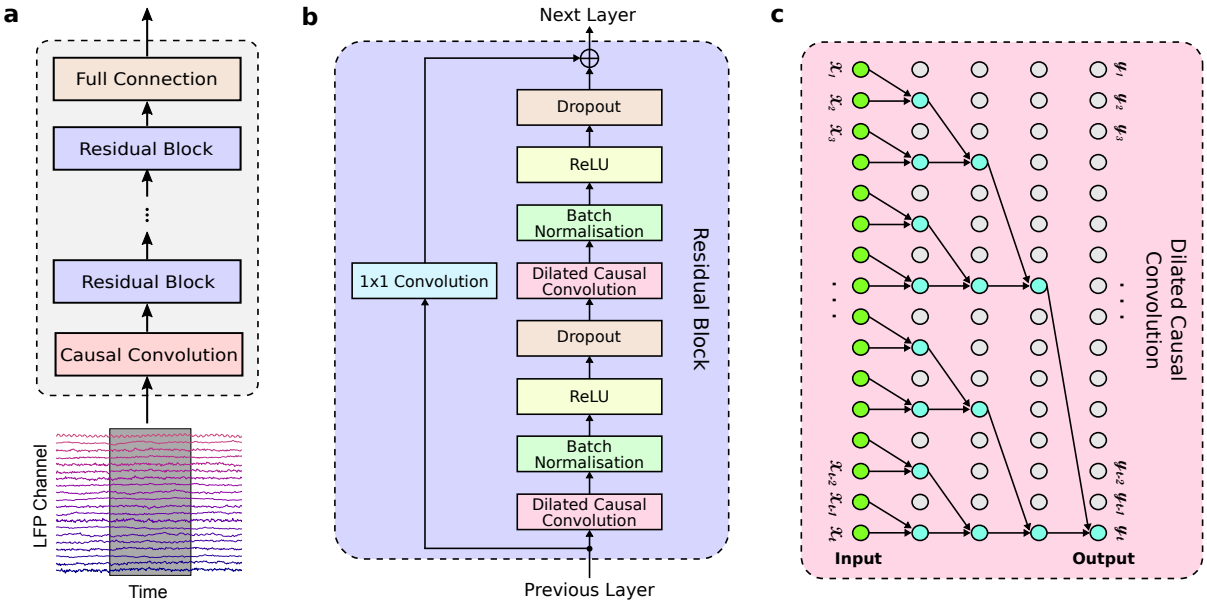


Figure 5.2 Schematic overview of TCN decoder. (a) A simplified block diagram of TCN. (b) The residual block of TCN. (c) An illustration of dilated causal convolution with dilation factors $d = 1, 2, 4, 8$ and filter size $k = 2$.

contaminating LFPs, we performed common average reference (CAR) by computing the average of LFPs across all channels and subtracting it from LFP in each channel. All LFP channels were included for experiments and analyses.

Spike signal processing

We used the same spike signal processing described in Section 4.2.3 to obtain SUA and MUA. Briefly, spikes were extracted from each channel by band pass filtering (from 500 to 5000 Hz) the raw neural signal and then detecting the filtered signal amplitudes that exceed a predetermined threshold value. All the detected spikes within single channel are referred to as MUA. The detected spikes were then sorted into distinct putative single units to obtain SUA.

5.2.3 Feature extraction

LFP feature extraction

For each LFP channel, six different features were extracted (one feature in time domain and five features in frequency domain). The time domain feature was an average amplitude called local motor potential (LMP) whereas the frequency domain features were average spectral

power in five different frequency bands: delta (1–4 Hz), theta (3–10 Hz), alpha (12–23 Hz), beta (27–38 Hz) and gamma (50–300 Hz). The selection of these frequency bands followed a previous study by Stavisky *et al.* [37]. The LMP was obtained by using a moving average filter with 256 ms rectangular window. To extract the spectral power features, we first band-pass filtered the LFP signals using 3rd-order Butterworth filter at each frequency band and then applied short-time Fourier transform (STFT) with a 256 ms Hanning window. All the feature extractions were performed in an overlapping fashion (252 ms overlap) to yield a sample every 4 ms (matching the sampling rate of the kinematic data). Before being fed to LSTM decoder, the LFP features were standardised (i.e. z-transformed) to zero mean and unit variance, whereas the kinematic data were normalised to zero mean.

Spike feature extraction

Spike features (firing rates) were extracted using BAKS with a 256 ms Gaussian window and the same parameters as described in Section 4.2.4. BAKS was used because it yielded better decoding performance than binning and Gaussian kernel smoother methods as shown in Chapter 4. Units with firing rates below 0.5 Hz were excluded thus leaving only 130 (124) SUA and 91 (94) MUA for session I (II). The spike features were also standardised before being fed to LSTM decoder.

5.2.4 Decoding algorithm

Long short-term Memory (LSTM)

LSTM is a type of recurrent neural networks (RNNs) that can learn long-term temporal dependencies without suffering from a vanishing gradient problem. In Chapter 4, we have shown that LSTM significantly outperformed both KF and WF in hand kinematic decoding from spike signals. In this chapter, we used the same LSTM architecture but different hyperparameter configuration. The LSTM decoder was trained using RMSprop optimiser and root mean squared error (RMSE) loss function.

Temporal convolutional network (TCN)

Temporal convolutional network (TCN) is a variant of convolutional neural network (CNN) architectures that makes use of 1D dilated causal convolutional layers. The major difference from regular 1D convolution is that the dilated causal convolution holds three characteristics: (1) the output has the same length as the input; (2) the output only depends on the present and past inputs (i.e. an output at time t is convolved only with the inputs at time t and earlier); (3) the introduction of a dilation factor (i.e. fixed step) between every adjacent filter taps which can increase the receptive field exponentially [215]. Given input sequences $x(t)$, the output sequences $y(t)$ obtained from a dilated causal convolution is formulated as follows:

$$y[t] = \sum_{i=0}^{k-1} W_i \, x[t - di] \tag{5.1}$$

where W denotes the filter weights, k is the filter size and d is the dilation factor. This type of convolution has been previously used in the popular and powerful WaveNet architecture for realistic speech synthesis [216].

The TCN architecture used in this chapter was inspired by the work of Bai et. al. [214] with one modification, that is, batch normalisation [217] was used instead of weight normalisation. A simplified and more detailed block diagram of TCN is given in Figure 5.2. In each residual block, the dilation rate was increased by a factor of 2 ($d = 2^{i-1},\ i = 1, 2, \cdots, r$), where r is the total number of residual blocks. The TCN combines feature extraction and decoding processes into single unit and is optimised based on single objective function. It was trained using Adam optimiser and root mean squared error (RMSE) loss function.

Hyperparameter optimisation

Similar to that of in Chapter 4, the data were divided into 10 non-overlapping contiguous blocks of equal size and categorised into training set (8 concatenated blocks), validation set (1 block) and testing set (1 block). The hyperparameters' values of LSTM and TCN decoders were optimised by using Hyperopt library [213] on the validation set of session I and run for 300 iterations. The search space for the hyperparameter optimisation and the resulting optimised

Table 5.1 Hyperparameter search space for TCN and LSTM decoders

Hyperparameter	Value Range	
	TCN	LSTM
Number of residual blocks	$\{6, 7, 8\}$	–
Number of filters	$\{16, 24, \cdots, 64\}$	–
Filter size	$\{3, 4, 5\}$	–
Number of layers	–	$\{1, 2, 3\}$
Number of units	–	$\{50, 75, \cdots, 200\}$
Number of timesteps	$\{128, 192, 256\}$	$\{2, 3, 4\}$
Number of epochs	$\{2, 3, \cdots, 8\}$	
Batch size	$\{64, 96, 128\}$	
Dropout rate	$\{0, 0.1, \cdots, 0.5\}$	
Learning rate	$\{5, 10, \cdots, 50\} \times 10^{-4}$	

Table 5.2 Hyperparameter configuration of TCN and LSTM decoders

Hyperparameter	Optimised Value						
	TCN	LSTM					
	Raw LFP	LMP	1-4 Hz	3-10 Hz	12-23 Hz	27-38 Hz	50-300 Hz
Number of residual blocks	8	–	–	–	–	–	–
Number of filters	48	–	–	–	–	–	–
Filter size	4	–	–	–	–	–	–
Number of layers	–	1	1	1	1	1	1
Number of units	–	125	200	150	175	100	75
Number of timesteps	256	2	2	2	2	2	2
Number of epochs	6	8	7	2	5	3	4
Batch size	64	64	64	96	64	96	64
Dropout rate	0.3	0	0	0.2	0.1	0.3	0.1
Learning rate	0.0015	0.003	0.003	0.0005	0.0035	0.0005	0.0005

values for both decoders are given in Table 5.1 and 5.2, respectively.

5.2.5 Performance evaluation and metrics

The decoding performance was evaluated using root mean square error (RMSE) and Pearson's correlation coefficient (CC) as in the previous chapter. The mean and confidence interval of the decoding performance were computed using 10 different blocks within the testing set. Statistical significance test was conducted using a two-tailed paired t-test when the difference between the paired data follows normal distribution, otherwise using a two-tailed paired Wilcoxon signed-rank test. The significance level (α) was set to 0.05.

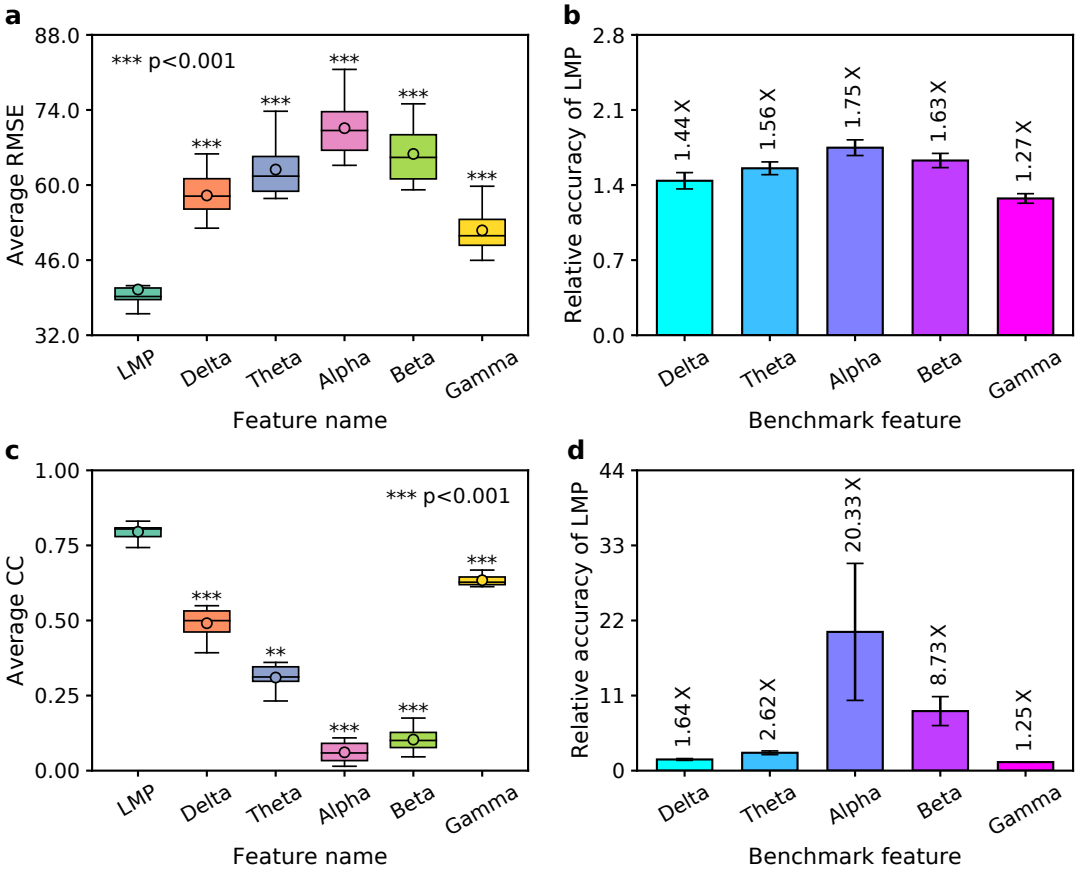


Figure 5.3 Decoding performance comparison across different LFP features using LSTM (data from recording session I). (a),(c) Boxplot comparison across LFP features measured in RMSE and CC, respectively. Asterisks indicate features that had statistically significant different performances from that of LMP (** $p<0.01$, *** $p<0.001$). (b),(d) Relative performance (in terms of RMSE and CC, respectively) of LMP relative to other LFP features. Larger value indicates better relative performance. Black error bars denote 95% confidence intervals.

5.3 Results

5.3.1 Performance comparison across different LFP features

We were interested in investigating which LFP feature carries the highest movement-related information. In doing so, we quantified the informativeness of each LFP feature by measuring its decoding performance using independently optimised LSTM decoder. Empirical results revealed that LMP yielded the highest decoding performance (smallest RMSE and largest CC) as depicted by boxplots in Figures 5.3a and 5.3c. The circle mark and horizontal line within each boxplot denote the mean and median, respectively. The coloured solid box represents the interquartile range between 25th and 75th percentiles. The whisker extends to 1.5 times the

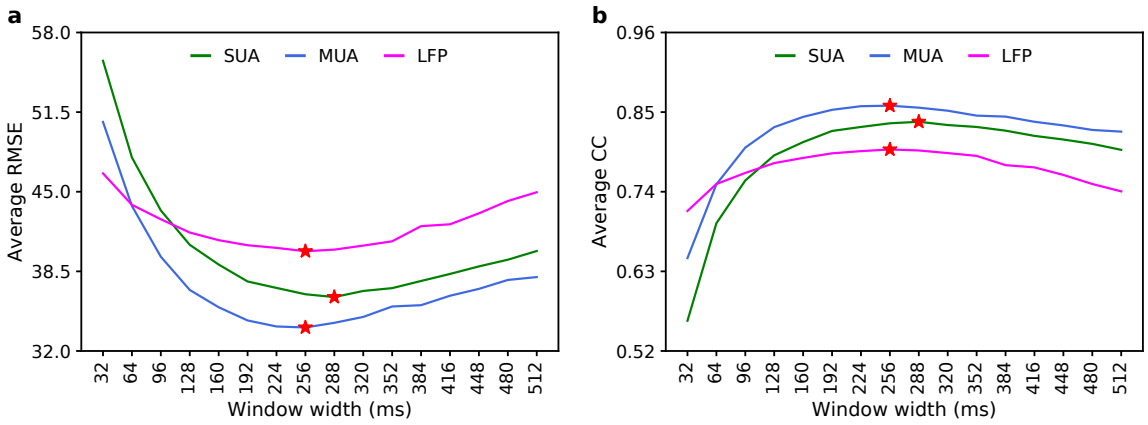


Figure 5.4 Decoding performance comparison of SUA-, MUA-, and LFP-driven LSTM decoders across different window widths. Red stars indicate window widths that yielded the best decoding performances on validation set from session I.

interquartile range. By sorting descendingly the average decoding performance, we obtained the following order: LMP (RMSE $= 40.52 \pm 1.27$, CC $= 0.80 \pm 0.01$) $>$ gamma (RMSE $= 51.54 \pm 1.42$, CC $= 0.63 \pm 0.01$) $>$ delta (RMSE $= 58.06 \pm 1.42$, CC $= 0.49 \pm 0.02$) $>$ theta (RMSE $= 62.89 \pm 1.72$, CC $= 0.31 \pm 0.01$) $>$ beta (RMSE $= 65.80 \pm 1.75$, CC $= 0.10 \pm 0.01$) $>$ alpha (RMSE $= 70.60 \pm 1.79$, CC $= 0.06 \pm 0.01$). Both RMSE and CC values are written in terms of mean $\pm$ standard error of the mean (SEM). Statistical tests showed that the performance of LMP differed significantly from that of other LFP features. LMP achieved, on average, 1.27 to 1.75 times better RMSE and 1.25 to 20.33 times better CC relative to other LFP features (see Figures 5.3b and 5.3d). Due to its highest decoding performance, LMP was selected for subsequent experiments.

5.3.2 Impact of window width on decoding performance

Next, we examined the impact of different window widths when computing LMP on decoding performance of LFP-driven LSTM decoder. We varied the window width value from 32 ms to 512 ms with an increment of 32 ms. We found that the increasing window width up to a particular point would increase the decoding performance (Figure 5.4); above this point, however, the performance would reach a plateau and tend to decline. The highest decoding performance was achieved when using a window width of 256 ms. We also investigated whether the same trend was observed in the case of SUA- and MUA-driven LSTM decoders. As shown

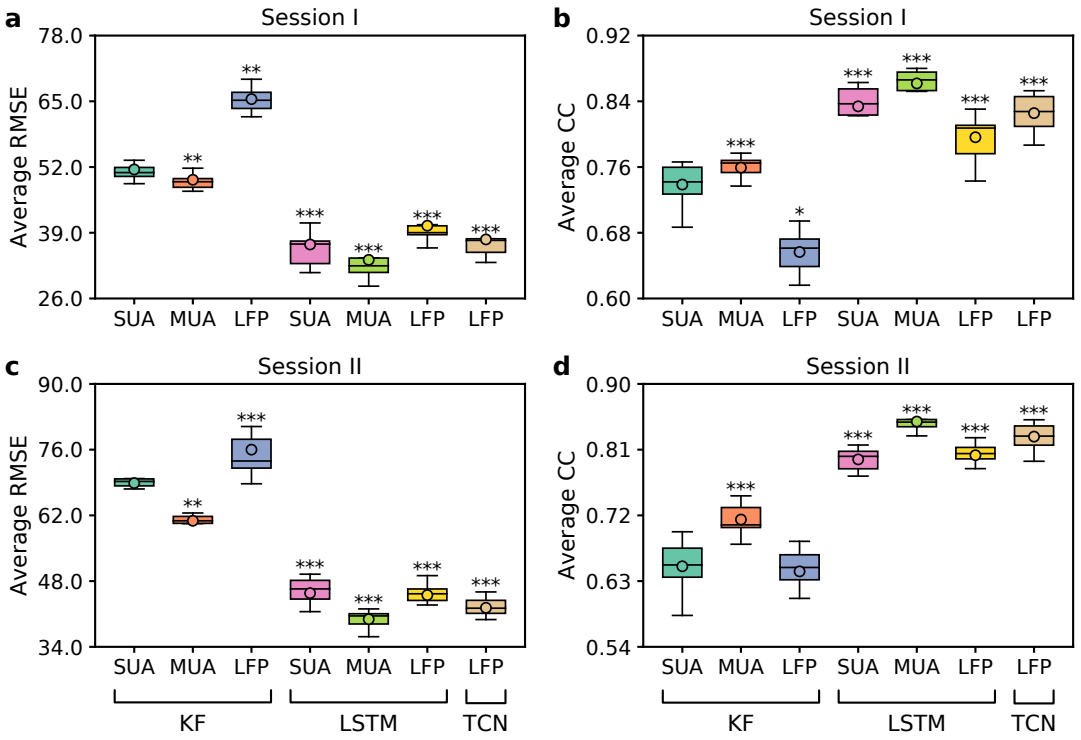


Figure 5.5 Decoding performance comparison across different decoders driven by SUA, MUA, and LFP. (a),(b) Boxplot comparison from session I measured in RMSE and CC, respectively. (c),(d) Boxplot comparison from session II measured in RMSE and CC, respectively. Asterisks indicate decoders whose decoding performance differed significantly from that of SUA-KF (* $p<0.05$, ** $p<0.01$, *** $p<0.001$).

in Figure 5.4, we found the same trend where increasing the window width to a particular value (around 256–288 ms) led to improved decoding performance; above this value, the decoding performance would decline.

5.3.3 Performance comparison across input signals and decoders

Further, we compared the decoding performance of our proposed LFP-driven LSTM and TCN decoders (referred to as LFP-LSTM and LFP-TCN, respectively) against LFP-, SUA-, and MUA-driven KF decoder (referred to as LFP-KF, SUA-KF, and MUA-KF, respectively). As shown in Figure 5.5, LFP-LSTM and LFP-TCN yielded statistically significantly better decoding performance than LFP-KF, SUA-KF, and MUA-KF across both metrics and both sessions. LFP-LSTM yielded RMSE = 40.40 ± 1.26 (45.03 ± 1.02) and CC = 0.80 ± 0.01 (0.80 ± 0.01) for session I (II). Relative to SUA-KF, LFP-LSTM achieved a decoding improvement of RMSE = 21.72% and CC = 7.84% (RMSE = 34.65% and CC = 23.57%) for session I (II), as shown

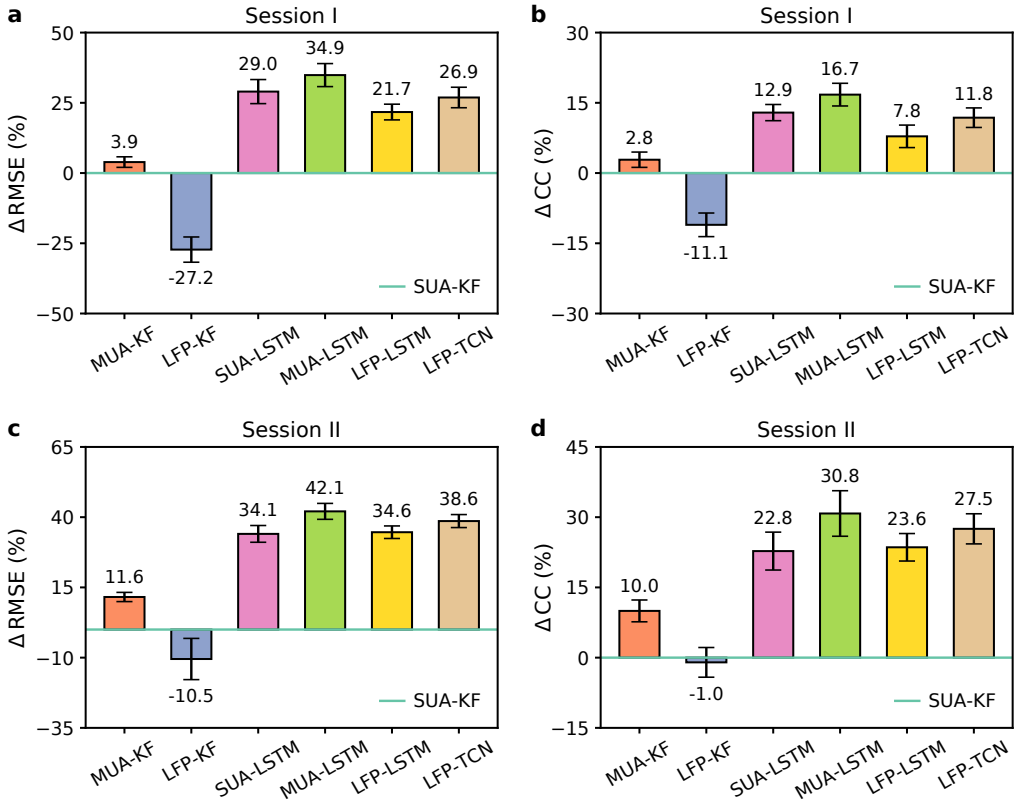


Figure 5.6 Decoding performance comparison across different decoders relative to SUA-KF. (a),(b) Comparison from session I measured in ΔRMSE (%) and ΔCC (%), respectively. (c),(d) Comparison from session II measured in ΔRMSE (%) and ΔCC (%), respectively. A positive (negative) value indicates performance improvement (degradation). Black error bars represent 95% confidence intervals.

in Figure 5.6. LFP-TCN yielded RMSE = 37.68 ± 1.16 (42.31 ± 1.11) and CC = 0.83 ± 0.01 (0.83 ± 0.01) for session I (II). This corresponded to a decoding performance improvement of RMSE = 26.89% and CC = 11.82% (RMSE = 38.62% and CC = 27.51%) for session I (II) with respect to SUA-KF.

In addition, we also measured the decoding performance of SUA- and MUA-driven LSTM decoders (referred to as SUA-LSTM and MUA-LSTM). SUA-LSTM yielded RMSE = 36.68 ± 1.52 (45.44 ± 1.20) and CC = 0.83 ± 0.01 (0.80 ± 0.01) for session I (II). MUA-LSTM yielded RMSE = 33.64 ± 1.40 (39.89 ± 0.98) and CC = 0.86 ± 0.01 (0.85 ± 0.00) for session I (II). The comparison results revealed that MUA-LSTM outperformed all other decoders. Compared to SUA-KF, MUA-LSTM achieved a decoding improvement of RMSE = 34.87% and CC = 16.75% (RMSE = 42.07% and CC = 30.79%) for session I (II), as shown in Figure 5.6. By sorting descendingly all the decoders according to their average decoding performance, we

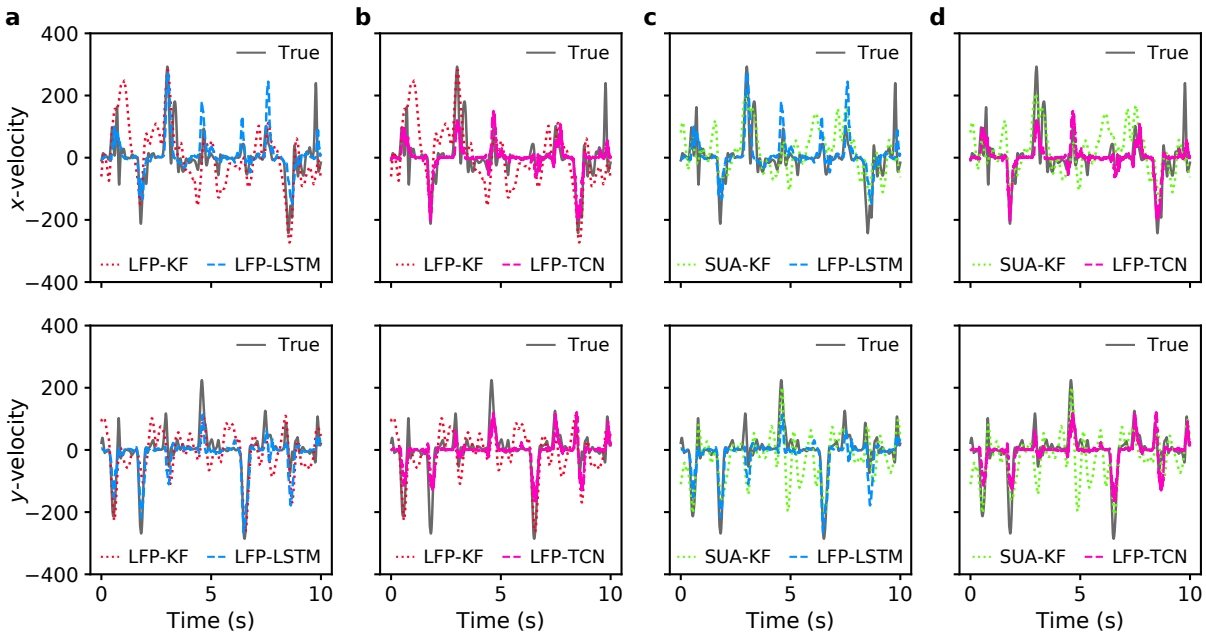


Figure 5.7 Examples of actual and decoded velocities from different decoders in x- and y-directions (data from session I).

obtained the following order: MUA-LSTM > LFP-TCN > SUA-LSTM > LFP-LSTM > MUA-KF > SUA-KF > LFP-KF. Examples of actual and decoded x- and y- velocities from SUA-KF, LFP-KF, LFP-LSTM and LFP-TCN are given in Figure 5.7.

5.3.4 Cross-correlation between learnt and hand-crafted LFP features

As TCN could learn informative features automatically from the raw LFPs, we next sought to determine whether TCN-learnt features exhibit cross-correlation with hand-crafted features in the same order as observed in Section 5.3.1. TCN-learnt features were obtained by taking the weight values of the last layer (before fully-connected layer) of TCN architecture (see Figure 5.2). We then computed Pearson's correlation coefficient between TCN-learn features and each of six hand-crafted LFP features. Results showed the cross-correlation scores between TCN-learnt and hand-crafted features indeed exhibited the same order with LMP having the highest score and alpha having the lowest score (Figure 5.8). According to the cross-correlation scores, LFP features could be sorted in descending order as follows: LMP > gamma > delta > theta > beta > alpha.



Figure 5.8 Cross-correlation between TCN-learnt and hand-crafted LFP features. Data were obtained from session I.

5.3.5 Inference time and power consumption

Lastly, we computed the inference time and power consumption of TCN decoder. The inference time was measured using the Python built-in function *perf_ counter()* while the power consumption was obtained via the NVIDIA System Management Interface (SMI). Results show that, for the inference (i.e. testing) time, TCN decoder took an average of 0.29 ms/output and consumed an average of 109.47 Watt.

5.4 Discussion

In this chapter, we aim to improve the decoding performance of LFP-based BMI by proposing two variants of deep learning methods, LSTM and TCN, as a decoding algorithm. First, we selected the most informative feature within LFPs by evaluating the decoding performance of each feature independently and selecting the feature with the highest decoding performance. We found that LMP yielded the highest decoding performance compared to other LFP features. This finding is in good agreement with Stavisky *et al.*'s study [37]. Next, using LMP as the feature, we compared the decoding performance of LFP-driven LSTM decoder to those of SUA-, MUA-, and LFP-driven KF decoders. We found that LFP-driven LSTM decoder achieved better decoding performance than KF decoders, demonstrating LSTM's capability in capturing nonlinear and complex relationship between LFPs and hand kinematics. However,

when comparing the decoding performance of SUA, MUA, and LFP under LSTM decoders, we found that the performance of LFP was the lowest among others. This could indicate that the movement-related information contained within LFPs was lower than within SUA and MUA. These results are in line with several prior published studies [33], [35], [39]–[41] except [31]. Mehring *et al.* [31] used 16 LFP channels, while we and other studies [33], [35], [39]–[41] used a significantly higher number of LFP channels. LFPs are known to exhibit relatively high correlation (redundancy) among channels. Thus, individual LFP channel generally contains higher movement-related information than individual SUA and MUA, as reflected by its decoding performance. However, as the number of LFP channels increases, the decoding performance of LFP reaches a plateau faster than those of SUA and MUA. When the number of channels/units is sufficiently high, SUA and MUA usually outperform LFPs [33]–[35], [41].

To improve further the decoding performance of LFPs while alleviating the need for the tedious manual feature extraction process, we proposed a TCN decoder that combines feature extraction and decoding algorithm into a single, unified, end-to-end approach. Comparison results showed that LFP-driven TCN decoder could outperform SUA- and LFP-driven LSTM decoder. This demonstrates the effectiveness of TCN decoder to automatically learn the optimal representation (i.e. features) of LFPs and decode hand kinematics. Upon examining the cross-correlation between TCN-learnt LFP features and hand-crafted LFP features, we found that LMP yielded the highest cross-correlation score. This indicates that TCN could indeed learn the most informative features within LFPs. The performance improvement achieved by TCN comes at the expense of increased computational complexity and power consumption.

TCN decoder was implemented and run on a desktop-based GPU which requires high power consumption. One possible direction for future work is to develop and optimise TCN decoder to be run on a low-power embedded platform. Compared to MUA-driven LSTM decoder, TCN decoder yielded a lower decoding performance. Therefore, developing LFP-based BMIs that can outperform MUA-based BMIs remains a challenging task and opens new avenues for future work.

5.5 Conclusion

This chapter has presented two variants of deep learning methods, LSTM and TCN, as a decoding algorithm. We have shown that LSTM significantly outperforms KF. Further, we have shown that TCN can improve decoding performance while alleviating the need for the laborious process of manual feature extraction. Overall results suggest the potential use of deep learning, particularly LSTM and TCN, in providing high decoding performance for LFP-based BMIs.

6 | Impact of Referencing Scheme on Decoding Performance of LFP-based BMI

This chapter considers the problem of common noise that often contaminates LFPs when recorded with respect to a unipolar reference. Several referencing schemes have been proposed to remove common noise, such as bipolar reference, current source density (CSD), and common average reference (CAR). We investigate the impact of these referencing schemes on decoding performance of LFP-based BMI. We also examine the impact of interelectrode distance and interchannel correlation on decoding performance. The performance evaluation is undertaken across long-term recording sessions using different features and deep learning decoders.

A version of this chapter was submitted to *Journal of Neural Engineering* and deposited to *bioRxiv* preprint server with the following title:
N. Ahmadi, T. G. Constandinou, and C.-S. Bouganis, "Impact of referencing scheme on decoding performance of LFP-based brain-machine interface", *bioRxiv*, 2020. [7]

6.1 Introduction

A great deal of literature has been published on the use of LFPs for BMIs in non-human primates [22], [25], [32], [33], [35], [37], [52] and humans [90], [103], [163], [218]. However, most LFP-based BMIs record neural signals with respect to a single common reference (hereinafter referred to as unipolar reference) which are susceptible to common noise due to different sources including non-silent reference and volume conduction [219], [220]. Since there is no reference site in the body free from electrical activity [221] and neural signals are measured as the potential differences between the active electrodes and reference, electrical activity in the reference will affect the recordings in all active electrodes. In addition, electrical activity from distant sources

can also contaminate the recordings through the volume conduction mechanism. This common noise may degrade LFP's signal-to-noise ratio (SNR) and lead to suboptimal BMI decoding performance. Several referencing schemes have been proposed to eliminate the common noise such as bipolar reference, common source density (CSD), and common average reference (CAR). Previous studies have assessed the impact of these referencing schemes on the analyses of power, phase, coherence, and Granger causality in LFPs [219], [220], [222]–[224]. So far, however, there is no study that examines the impact of these referencing schemes on decoding performance of LFP-based BMIs.

This chapter aims to address the above knowledge gap by systematically evaluating and comparing the impact of different referencing schemes on decoding performance of LFP-based BMIs. We used long-term LFP data that were chronically recorded from the motor cortex area of a macaque monkey (*Macaca mulatta*) while performing self-paced reaching tasks. We employed two deep learning methods, namely long short-term memory (LSTM) and gated recurrent unit (GRU) networks for decoding hand kinematics from LFPs. Firstly, We examined the informativeness of different LFP features across referencing schemes. Secondly, we assessed the impact of interelectrode distance in bipolar reference on decoding performance. We then compared the decoding performance across referencing schemes. Lastly, we investigated the impact of referencing schemes on the interchannel correlation of LFPs and its relation to decoding performance.

6.2 Methods

The block diagram of LFP-based BMI system used in this chapter is illustrated in Figure 6.1. All the experiments and analyses were done in Python. Keras/TensorFlow deep learning framework [209] was used to implement the decoding algorithms.

6.2.1 Neural recordings and Behavioural task

We used the same neural dataset as in Chapter 4 (see Section 4.2.1). This dataset contains long-term raw data and corresponding behavioural task (a total of 26 sessions with over 7.3-month span), allowing us to analyse the impact of referencing scheme over a long period of time.

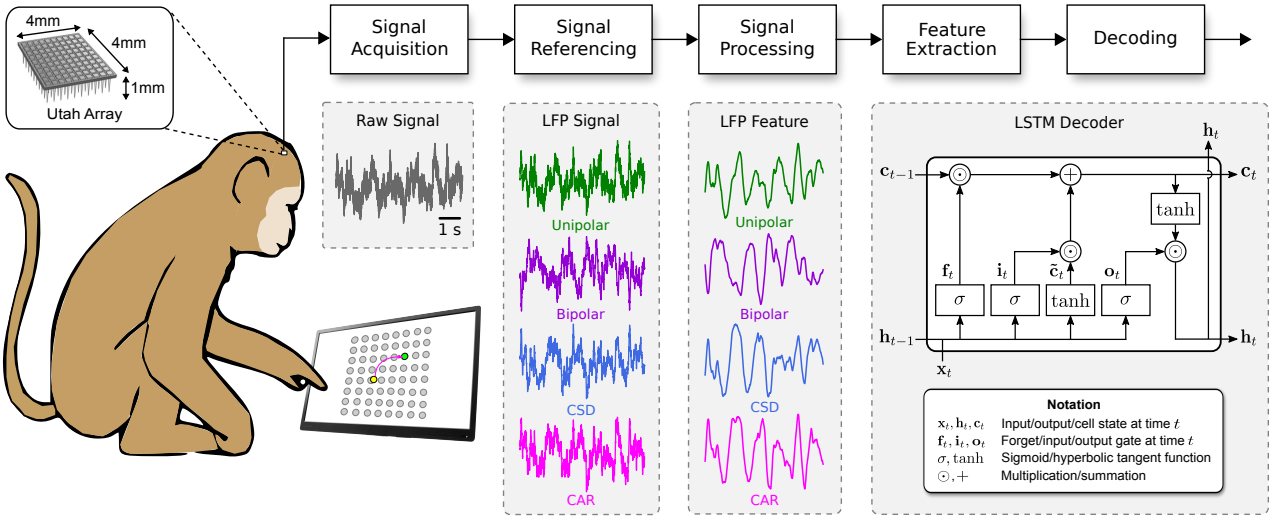


Figure 6.1 Block diagram of LFP-based BMI system with different referencing schemes

The neural data were recorded from the primary motor cortex (M1) area of a monkey with a 96-channel Utah array (platinum coating, 1 mm electrode length, 400 μm spacing, 400 kΩ impedance). Due to the fixed length of Utah array, the analysis in this chapter is confined to signals recorded at a specific depth. Therefore, referencing schemes are compared on 2D spatial properties parallel to the cortical surface. The behavioural task has been described in Section 4.2.2.

6.2.2 Signal referencing

We used four referencing schemes namely unipolar reference, bipolar reference, current source density (CSD), and common average reference (CAR), each of which is described as follows.

Unipolar reference

Unipolar reference in the form of a silver wire placed under the dura (several cm away from the electrodes) was originally used during neural signal acquisition. The recorded neural signal at electrode i, represented by V_i^{uni}, is measured as:

$$V_i^{uni} = V_i - V_r, \quad 0 \leq i < 96 \tag{6.1}$$

where V_i and V_r denote the electric potential at the electrode i and reference, respectively. In total there was 96 unipolar signals (i.e. channels). The characteristics of the recording setup, e.g. very low added noise by instrumentation, high-resolution data conversion, and broadband data, allow us to emulate other referencing scheme in software. All filtering operation was performed digitally in forward and backward directions to avoid any phase distortion.

Bipolar reference

Bipolar referencing, initially used in EEG studies [225], was obtained by computing the difference between a pair of unipolar signals as follows:

$$V_k^{bi} = V_i^{uni} - V_j^{uni}, \quad r_{ij} = 400, 800, \cdots, 2400\,\mu m \tag{6.2}$$

where r_{ij} denotes a predefined distance between electrode i and electrode j ranging from 400 μm to 2400 μm with an increment of 400 μm, which yielded a total of 172, 152, 132, 112, 172, and 72 bipolar signals, respectively.

Current source density (CSD)

CSD, also known as Laplacian referencing [226], was computed by subtracting the average of four nearest neighbouring electrodes with equal distance of 400 μm as formulated below:

$$V_i^{CSD} = V_i^{uni} - \frac{1}{4}\sum_{j}^{4} V_j^{uni} \tag{6.3}$$

Only electrodes having 4 equidistant neighbours were included for further experiments and analyses, yielding a total of 64 CSD signals (i.e. electrodes at the edge were excluded).

Common average reference (CAR)

Originally used for ECG [227] in 1930s and firstly adopted for EEG in 1950s [228], CAR has become one of the most popular referencing schemes. CAR was obtained by subtracting the

average of all the unipolar signals as described by the following equation:

$$V_i^{CAR} = V_i^{uni} - \frac{1}{N}\sum_{i}^{N} V_i^{uni} \tag{6.4}$$

CAR referencing resulted in the same number of signals as the original unipolar reference (96 channels).

6.2.3 Signal processing and feature extraction

To obtain LFP signals, the raw neural signals were low-pass filtered with a 4th-order Butterworth filter at 300 Hz and then downsampled to 1 kHz. The filtering was performed on the forward and backward directions to avoid any phase shift. Same as in Chapter 5, we extracted six different features: local motor potential (LMP) and average power spectra from five narrow-bands, which include delta (1–4 Hz), theta (3–10 Hz), alpha (12–23 Hz), beta (27–38 Hz) and gamma (50–300 Hz). All the feature extractions were performed in an overlapping fashion to yield a sample every 4 ms (matching the sampling rate of the kinematic data). All features were then standardised before being used for training/evaluation of the decoders. All the LFP channels were included for experiments and analyses.

6.2.4 Decoding algorithm

Long short-term memory (LSTM)

We used the same LSTM architecture as previously described in Section 4.2.5 but with different hyperparameter configuration.

Gated recurrent unit (GRU)

GRU is another variant of recurrent neural networks (RNNs) proposed by Cho *et al.* in 2014 [229] that uses a gating mechanism similar to LSTM. It has less parameters and less structural complexity than LSTM since it combines the forget gate and input gate into a single update gate and combines the memory cell and hidden state. Each component of GRU is defined by

the following equations:

$$\begin{aligned}
\mathbf{z}_t &= \sigma(\mathbf{W}_z\mathbf{x}_t + \mathbf{U}_z\mathbf{h}_{t-1} + \mathbf{b}_z) \\
\mathbf{r}_t &= \sigma(\mathbf{W}_r\mathbf{x}_t + \mathbf{U}_r\mathbf{h}_{t-1} + \mathbf{b}_r) \\
\tilde{\mathbf{h}}_t &= \tanh\left(\mathbf{W}_h\mathbf{x}_t + \mathbf{U}_h(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h\right) \\
\mathbf{h}_t &= (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t
\end{aligned} \tag{6.5}$$

where $\mathbf{z}, \mathbf{r}, \mathbf{h}$ represent the update gate, reset gate, and hidden state, respectively. Chung *et al.* demonstrated empirically that the performance of GRU was comparable to that of LSTM across three sequence modelling tasks [230].

Decoder training and optimisation

As in the previous chapter, the data were divided into 10 non-overlapping contiguous blocks of equal size which were categorised further into three sets: training (8 concatenated blocks), validation (1 block) and testing (1 block). Each decoder was trained using RMSprop optimiser and root mean squared error (RMSE) loss function. The decoder hyperparameters were optimised from predefined ranges shown in Table 6.1 using Hyperopt library [213]. The resulting optimised hyperparameters for each LFP feature and referencing scheme are given in Table 6.1 and Table 6.2, respectively. The hyperparameter optimisation was performed only on the first recording session; the same hyperparameter configuration was used for the subsequent sessions.

Table 6.1 Hyperparameter configuration of LSTM decoders across LFP features.

Hyperparameter	Value Range	LFP Feature					
		LMP	Delta	Theta	Alpha	Beta	Gamma
Number of units	$\{50, 75, \cdots, 200\}$	175	175	100	75	175	200
Number of epochs	$\{2, 3, \cdots, 8\}$	3	2	2	2	2	2
Batch size	$\{64, 96, 128\}$	64	64	64	64	64	64
Dropout rate	$\{0, 0.1, \cdots, 0.5\}$	0.4	0.2	0.2	0.3	0.2	0.3
Learning rate	$\{5, 10, \cdots, 50\} \times 10^{-4}$	0.0005	0.001	0.0005	0.0005	0.0005	0.0005

Table 6.2 Hyperparameter configuration of LSTM decoders across referencing schemes

Hyperparameter	Value Range	Referencing Scheme			
		Unipolar	Bipolar	CSD	CAR
Number of units	$\{50, 75, \cdots, 200\}$	175	150	50	175
Number of epochs	$\{2, 3, \cdots, 8\}$	3	3	3	5
Batch size	$\{64, 96, 128\}$	64	96	96	64
Dropout rate	$\{0, 0.1, \cdots, 0.5\}$	0.4	0.5	0.4	0.4
Learning rate	$\{5, 10, \cdots, 50\} \times 10^{-4}$	0.0005	0.0045	0.001	0.0045

6.2.5 Performance evaluation and metrics

we evaluated decoding performance of each model using root mean square error (RMSE) and Pearson's correlation coefficients (CC). We calculated mean and confidence interval of the decoding performance for each session (n=10 blocks) and across all sessions (n=26 sessions). Statistical significant tests between a pair of different LFP features or different referencing schemes were conducted using a paired two-tailed t-test whenever the difference between the pairs follows normal distribution, otherwise using Wilcoxon signed-rank test. The significance level (α) was 0.05.

6.3 Results

6.3.1 LFP feature comparison and selection

Firstly, we assessed the informativeness of LFP features by measuring their decoding performance under unipolar reference using independently optimised LSTM decoders. Experimental results revealed that LMP yielded the highest decoding performance (smallest RMSE and largest CC), followed by gamma, delta, theta, alpha, and beta, respectively (see Figure 6.2). The circle mark and horizontal line within each boxplot denote the mean and median, respectively. The coloured solid box represents the interquartile range between 25th and 75th percentiles. The whisker extends to 1.5 times the interquartile range. As shown in Figures 6.2a and 6.2c, LMP and gamma were the two most informative features, yielding RMSE $= 45.65 \pm 0.90$ and CC $= 0.81 \pm 0.01$ for LMP and RMSE $= 60.62 \pm 1.50$ and CC $= 0.61 \pm 0.01$ for gamma. In this chapter, RMSE and CC performance are written in term of mean $\pm$ standard error of the

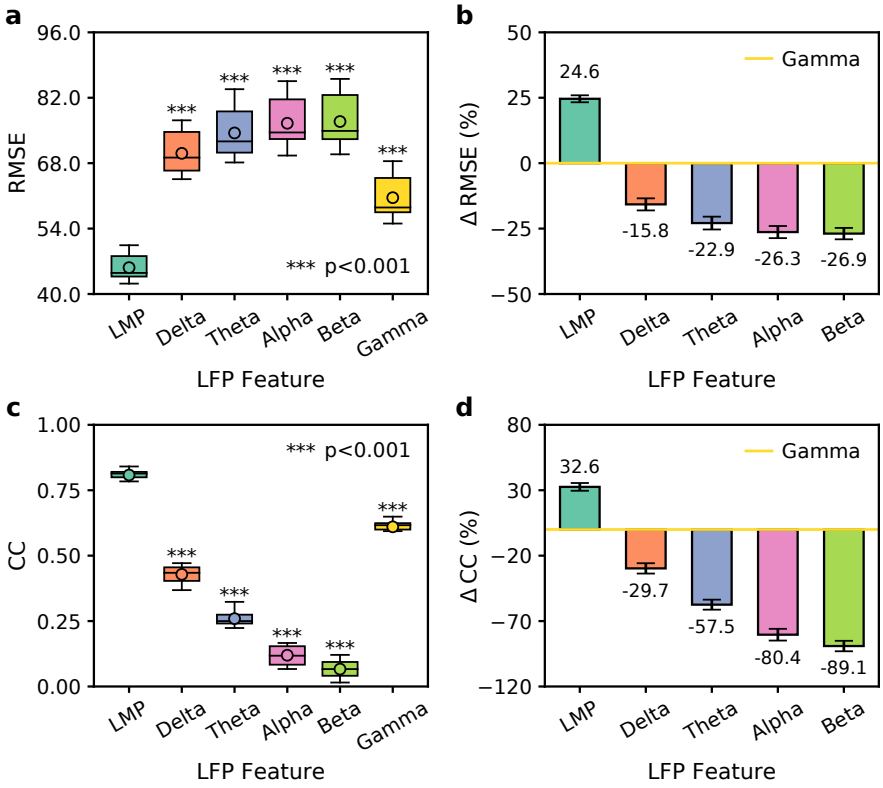


Figure 6.2 Decoding performance comparison across different LFP features in unipolar reference (data from recording session I20160627_01). (a),(c) Boxplot comparison across features measured in RMSE and CC, respectively. Asterisks indicate features that had statistically significant different performances from that of LMP (*** p<0.001). (b),(d) Performance improvement/degradation (in percent RMSE and CC, respectively) relative to gamma. Positive (negative) value indicates performance improvement (degradation). Black error bars denote 95% confidence intervals.

mean (SEM) unless otherwise noted. Relative to gamma, the performance of LMP in terms of RMSE and CC improved on average by 24.6% and 32.6%, respectively, as can be seen from Figures 6.2b and 6.2d. The performance of other LFP features (delta, theta, alpha, and beta) degraded on average by 15.8%–26.9% (RMSE) and 29.7%–89.1% depending on the feature.

We then examined whether the same trend was also observed from different referencing schemes. Results from bipolar reference showed the same order of feature informativeness as of unipolar reference (in descending order): LMP > gamma > delta > theta > alpha > beta. There were some variations in the order of feature informativeness in the cases of CSD and CAR, which were LMP > gamma > theta > delta > beta > alpha and LMP > gamma > delta > theta > beta> alpha, respectively. These extensive results revealed that LMP and gamma consistently emerged as the first and second most informative features. Therefore, for

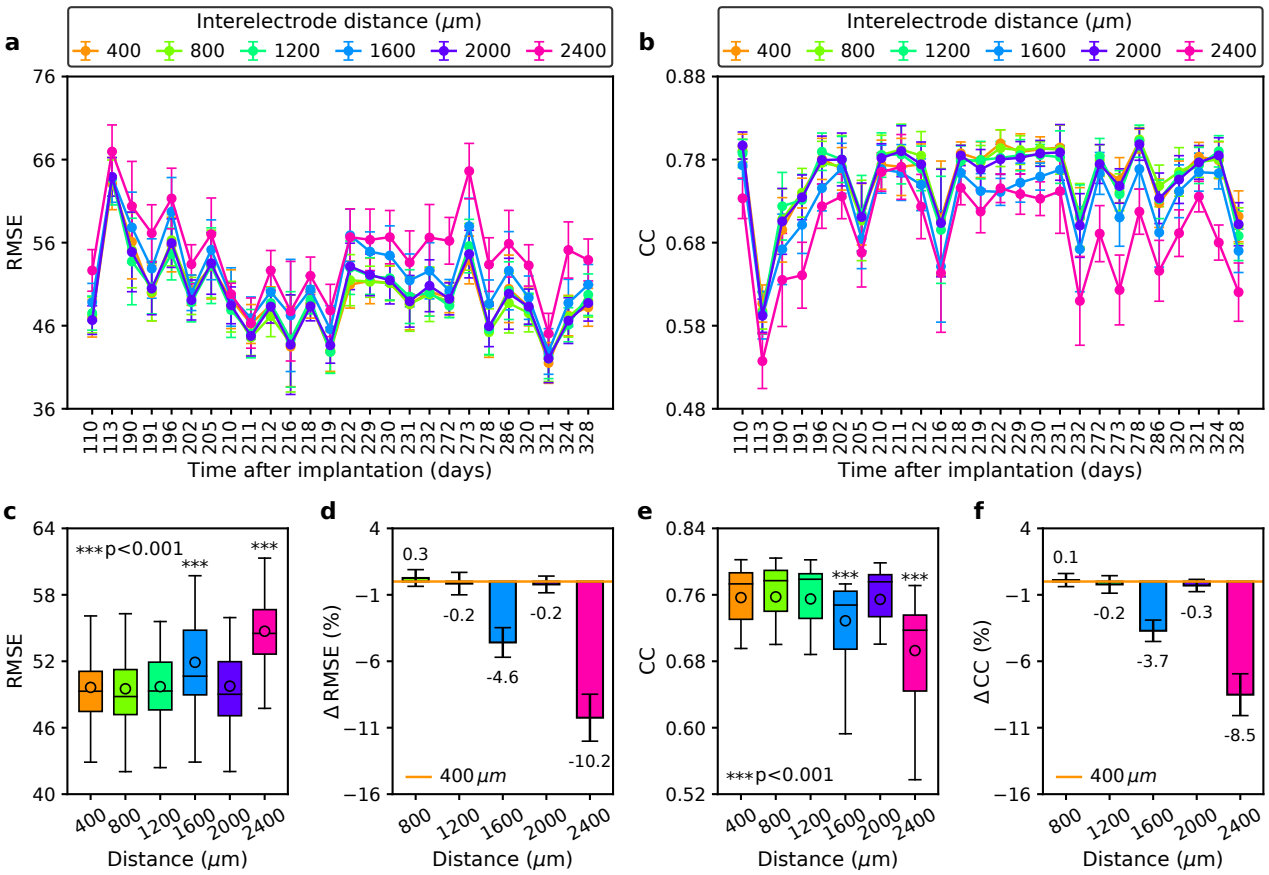


Figure 6.3 Decoding performance comparison across different interelectrode distances in bipolar reference with LMP feature. (a),(b) Performance comparison over long-term recording sessions measured in RMSE and CC, respectively. (c),(e) Boxplot comparison across distances and sessions measured in RMSE and CC, respectively. Asterisks indicate distances whose performances were statistically significant difference from that of 400 μm (*** p<0.001). (d),(f) Performance improvement/degradation (in percent RMSE and CC, respectively) relative to that of 400 μm. Positive (negative) value indicates performance improvement (degradation). Black error bars denote 95% confidence intervals.

subsequent experiments, we selected to use only these two features.

6.3.2 Impact of interelectrode distance in bipolar reference

Next, we investigated the impact of interelectrode distance in bipolar reference by comparing the decoding performance from varying distances ($400, 800, \cdots, 2400\,\mu$m) over long-term recording sessions. Figures 6.3 and 6.4 display the decoding performance comparison across different interelectrode distances over 26 sessions from LMP and gamma features, respectively. In the case of LMP, the decoding performance from the distances of 400, 800, 1200, and 2000 μm (with exceptions from the distances of 1600 and 2400 μm) were similar as can be observed in

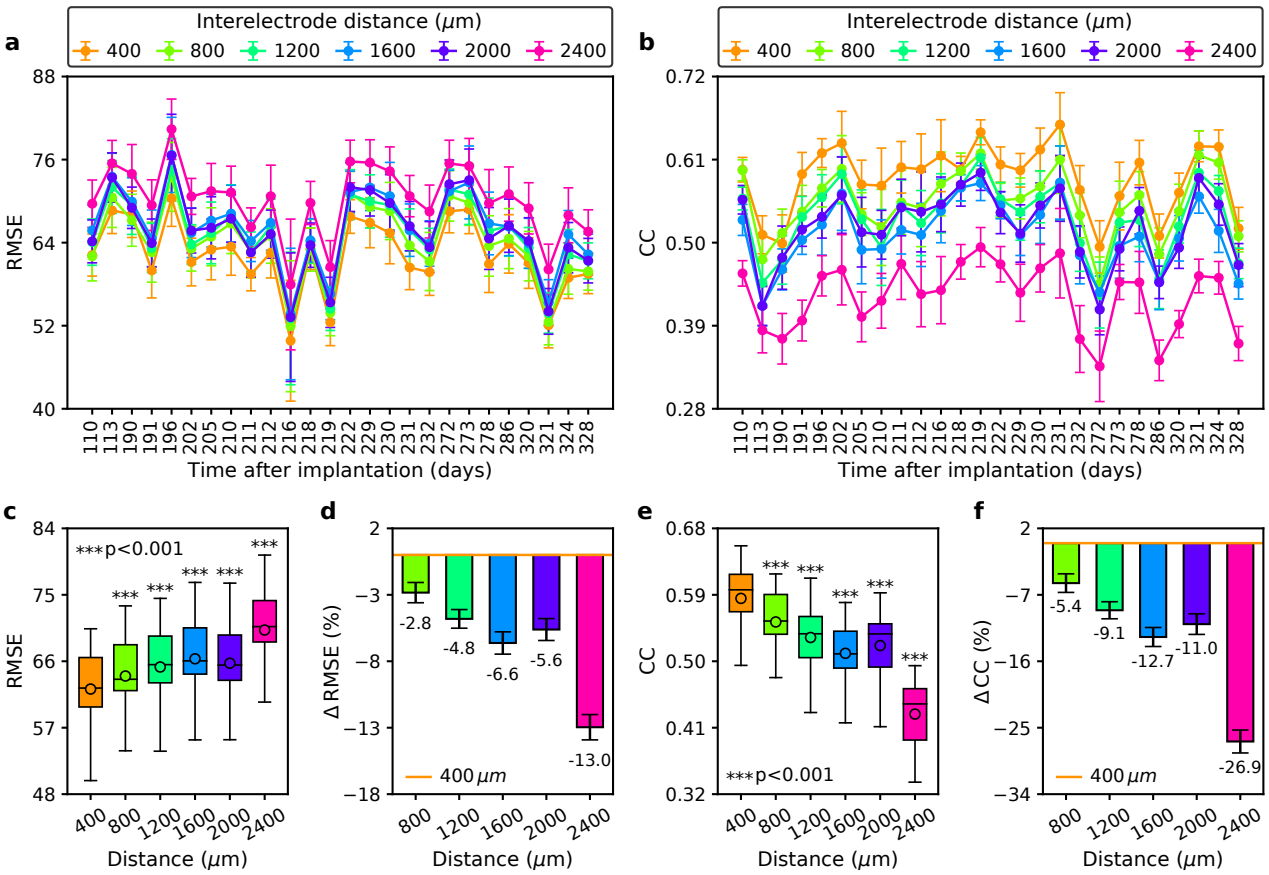


Figure 6.4 Decoding performance comparison across different interelectrode distance in bipolar reference with gamma feature. (a),(b) Performance comparison over long-term recording sessions measured in RMSE and CC, respectively. (c),(e) Boxplot comparison across distances and sessions measured in RMSE and CC, respectively. Asterisks indicate distances whose performances were statistically significant difference from that of 400 μm (*** p<0.001). (d),(f) Performance improvement/degradation (in percent RMSE and CC, respectively) relative to that of 400 μm. Positive (negative) value indicates performance improvement (degradation). Black error bars denote 95% confidence intervals.

Figures 6.3a and 6.3b. The average RMSE and CC for the distance of 400 μm were 49.64 ± 0.88 and 0.76 ± 0.01, respectively (Figures 6.3c and 6.3e). Increasing the distances from 400 μm to 2400 μm did not improve the decoding performance, but, on the contrary, it tended to degrade the decoding performance as clearly visible in Figures 6.3d and 6.3f. Specifically, the decoding performance from the distances of 1600 μm and 2400 μm were statistically significantly lower than that of 400 μm (*** p<0.001). With respect to 400 μm, the performance of 1600 μm degraded by 4.6% (RMSE) and 3.7% (CC), whereas the performance of 2400 μm degraded by 10.2% (RMSE) and 8.5% (CC).

In the case of gamma, the average decoding performance of 400 μm was 62.24 ± 1.03

(RMSE) and 0.59 ± 0.01 (CC) and was consistently better than that of larger distances across sessions as illustrated in Figures 6.4a and 6.4b. Statistical significant tests showed that the decoding performance of larger distances exhibited significant difference than that of 400 μm (*** $p<0.001$) as evident from Figure 6.4c for RMSE and Figure 6.4e for CC. Although in essence similar to the case of LMP, the impact of increasing interelectrode distances was more clearly observed in the case of gamma. As shown in Figures 6.4d and 6.4f, increasing the distances resulted in performance degradation of 2.8%–13.0% (RMSE) and 5.4%–26.9% (CC) relative to that of 400 μm. We thus selected an interelectrode distance of 400 μm for subsequent experiments.

6.3.3 Impact of referencing scheme on decoding performance

We then evaluated the impact of different referencing schemes on decoding performance of LSTM decoder driven by LMP feature. The decoding performance comparison across referencing schemes from LMP-driven LSTM decoder over 26 recording sessions measured in RMSE and CC are illustrated in Figures 6.5a and 6.5b, respectively. The decoding performance of each referencing scheme averaged across sessions was as follows: unipolar (RMSE = 50.97 ± 0.96, CC = 0.76 ± 0.01), bipolar (RMSE = 49.64 ± 0.88, CC = 0.76 ± 0.01), CSD (RMSE = 61.72 ± 1.11, CC = 0.61 ± 0.01), and CAR (RMSE = 48.08 ± 0.97, CC = 0.77 ± 0.01). Sorting the referencing schemes descendingly according to their decoding performance resulted in the following order: CAR > bipolar > unipolar > CSD (see Figures 6.5c and 6.5e). Statistical tests showed that, compared to that of unipolar, the performance of CSD and CAR had significant differences in both RMSE and CC metrics (*** $p<0.001$). CSD resulted in average performance degradation of 21.2% (RMSE) and 19.1% (CC), while CAR yielded an average performance improvement of 5.6% (RMSE) and 1.9% (CC) as can be seen in Figures 6.5d and 6.5f. The performance of bipolar, however, only showed a significant difference in RMSE metric (** $p<0.01$). Snippet examples of decoded x- and y-velocities along with their ground truth values from four referencing schemes are illustrated in Figures 6.5g-j.

To check whether a similar trend was observed when using gamma feature, we compared the decoding performance of gamma-driven LSTM decoder across referencing schemes. The

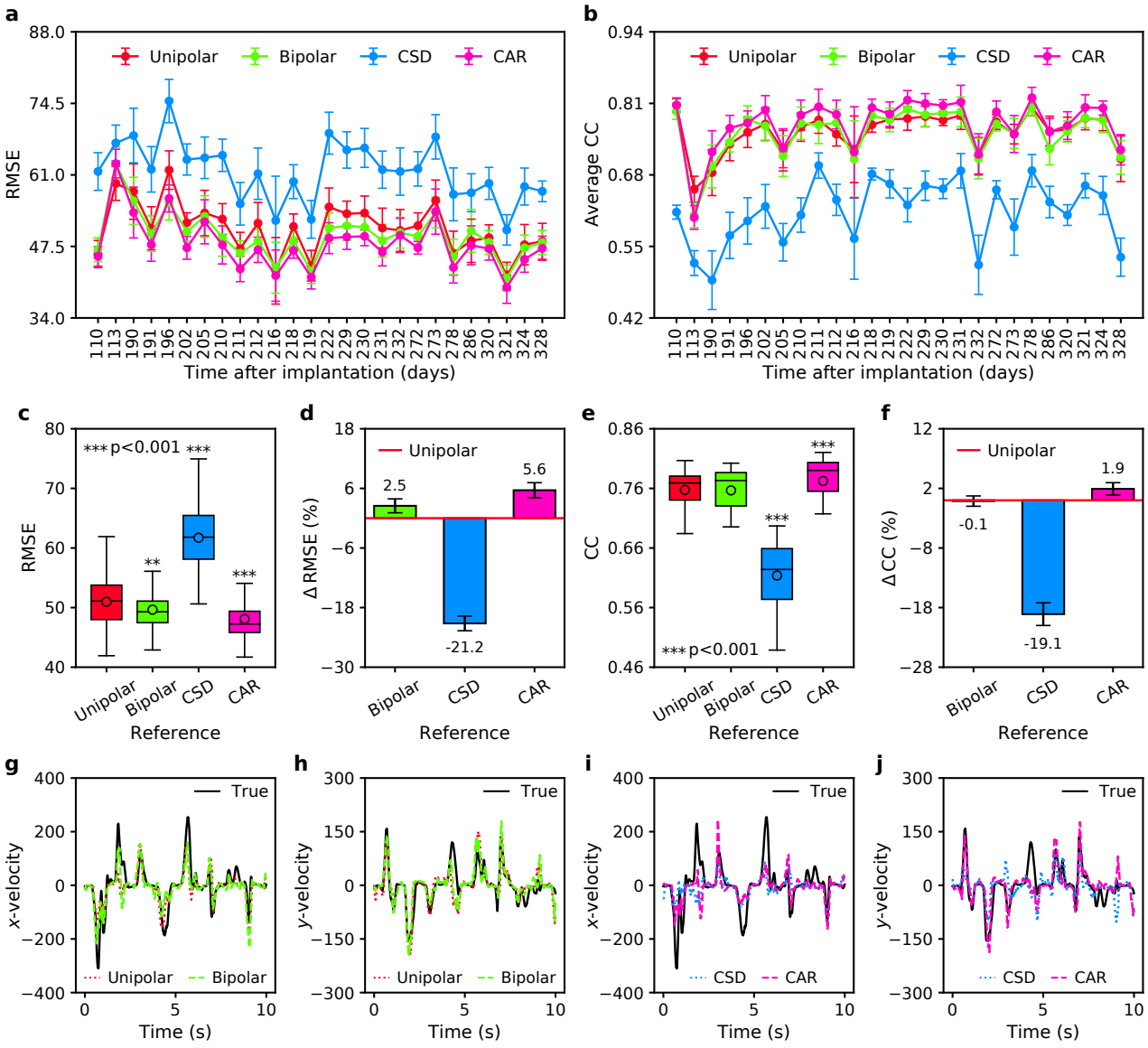


Figure 6.5 Decoding performance comparison across different referencing schemes using LMP-driven LSTM decoder. (a),(b) Performance comparison over long-term recording sessions measured in RMSE and CC, respectively. (c),(e) Boxplot comparison across sessions measured in RMSE and CC, respectively. Asterisks indicate referencing schemes that yielded statistically significant different performances from that of unipolar reference (** $p<0.01$, *** $p<0.001$). (d),(f) Performance improvement/degradation (in percent RMSE and CC, respectively) relative to unipolar reference. Positive (negative) value indicates performance improvement (degradation). Black error bars denote 95% confidence intervals. (g)-(j) Snippet examples of true and decoded velocities in x- and y- coordinates from different referencing schemes (data from recording session I20160627_01).

result of RMSE and CC performance comparison over 26 recording sessions are depicted in Figures 6.6a and 6.6b, respectively. In comparison to the case of LMP, here we found contrasting results when using bipolar and CAR, but consistent result when using CSD. The order of decoding performance across referencing schemes (from high to low) was bipolar (RMSE = 62.24±1.03, CC = 0.59±0.01), CAR (RMSE = 64.03±1.12, CC = 0.55±0.01), unipolar (RMSE = 63.88 ± 1.10, CC = 0.55 ± 0.01), and CSD (RMSE = 68.11 ± 1.11, CC = 0.46 ± 0.01). With respect to unipolar, bipolar and CSD yielded significant differences in the decoding performance (*** $p<0.001$), whereas CAR did not differ significantly as can be seen in Figures 6.6c and 6.6e. Relative to unipolar, bipolar improved the average decoding performance by 2.5% (RMSE) and 6.7% (CC), while CSD degraded the average decoding performance by 6.7% (RMSE) and 16.9% (CC) as plotted in Figures 6.6d and 6.6f. It is worth noting that the decoding performance from gamma was consistently lower than that from LMP regardless of the referencing scheme. We next sought to assess whether the impact of referencing scheme on decoding performance remain the same when using different decoding algorithm. We thus employed a new decoder based on gated recurrent unit (GRU) network. We compared the decoding performance across referencing schemes using LMP-driven GRU decoder. Figures 6.7a and 6.7b illustrate the decoding performance comparison over 26 sessions in terms of RMSE and CC, respectively. We observed the same pattern of decoding performance across referencing schemes as in the case of LMP-driven LSTM decoder. Sorted from high to low decoding performance, we obtained the following order: CAR (RMSE = 48.62±0.96, CC = 0.77±0.01), bipolar (RMSE = 50.00±0.89, CC = 0.75 ± 0.01), unipolar (RMSE = 52.19 ± 0.86, CC = 0.73 ± 0.01), and CSD (RMSE = 61.86 ± 1.06, CC = 0.58 ± 0.01) as depicted in Figures 6.7c and 6.7e. Relative to unipolar, bipolar yielded performance improvement of 4.2% (RMSE) and 3.5% (CC), whereas CAR yielded performance improvement of 6.9% (RMSE) and 5.6% (CC) as shown in Figures 6.7d and 6.7f. On the other hand, the performance of CSD degraded by 18.6% (RMSE) and 20.3% (CC). Although the same pattern was observed between LMP-driven GRU and LMP-driven LSTM decoders, the magnitude of the decoding performance (RMSE and CC) obtained from LSTM were consistently better than that of GRU across all referencing schemes.

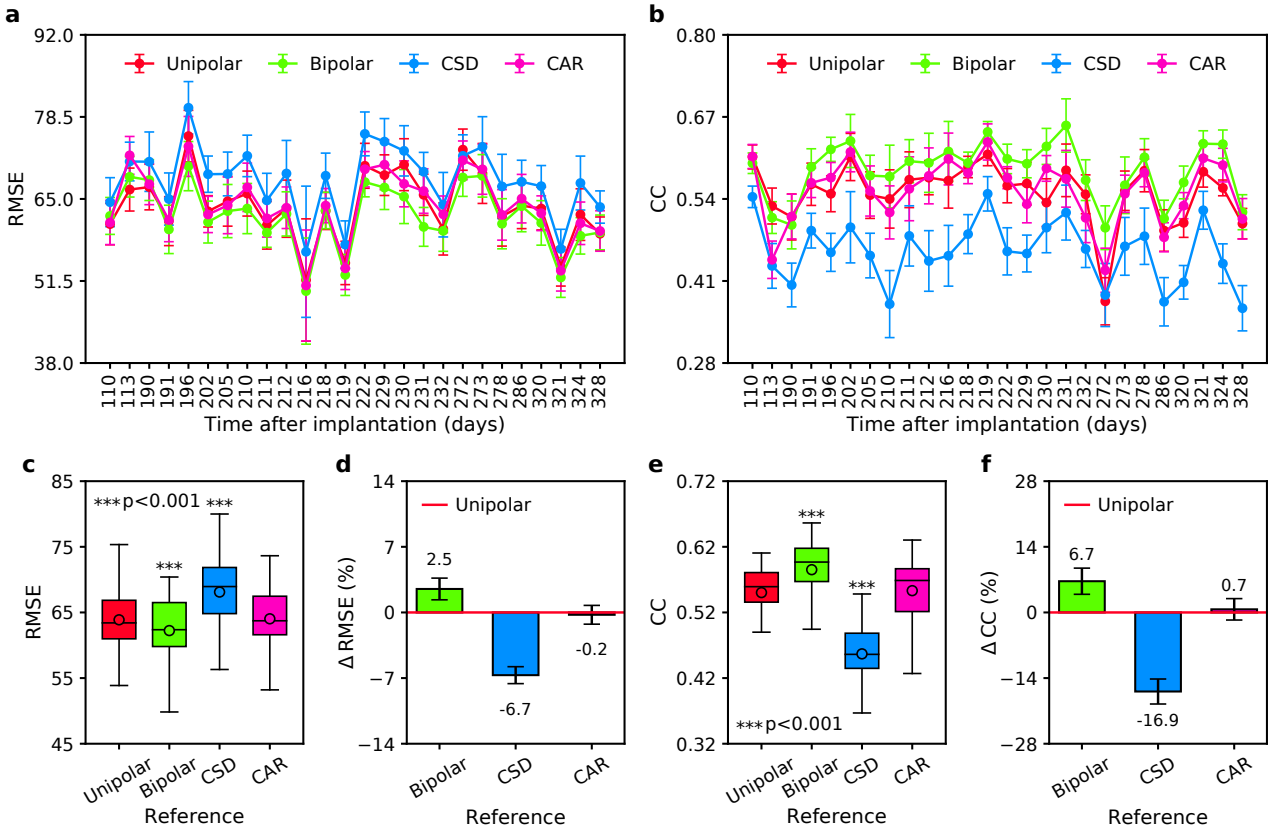


Figure 6.6 Decoding performance comparison across different referencing schemes using gamma-driven LSTM decoder. (a),(b) Performance comparison over long-term recording sessions measured in RMSE and CC, respectively. (c),(e) Boxplot comparison across sessions measured in RMSE and CC, respectively. Asterisks indicate referencing schemes that yielded statistically significant different performances from that of unipolar reference (*** $p<0.001$). (d),(f) Performance improvement/degradation (in percent RMSE and CC, respectively) relative to unipolar reference. Positive (negative) value indicates performance improvement (degradation). Black error bars denote 95% confidence intervals.

6.3.4 Impact of referencing scheme on interchannel correlation

Lastly, we examined the impact of referencing scheme on the interchannel correlation of LFPs by computing Pearson's correlation coefficient among all unique possible pair combinations of LFP channels. Comparison of interchannel correlation across referencing schemes over 26 recording sessions is plotted in Figure 6.8a. Our results showed that, in comparison to unipolar, other referencing schemes yielded significantly lower average interchannel correlation (*** $p<0.001$). By sorting the average interchannel correlation descendingly, we obtained the following order: unipolar (0.56 ± 0.10) > CAR (0.20 ± 0.07) > CSD (0.12 ± 0.04) > bipolar (0.09 ± 0.02). Figure 6.8b illustrates heatmaps of interchannel correlation matrix from each scheme.

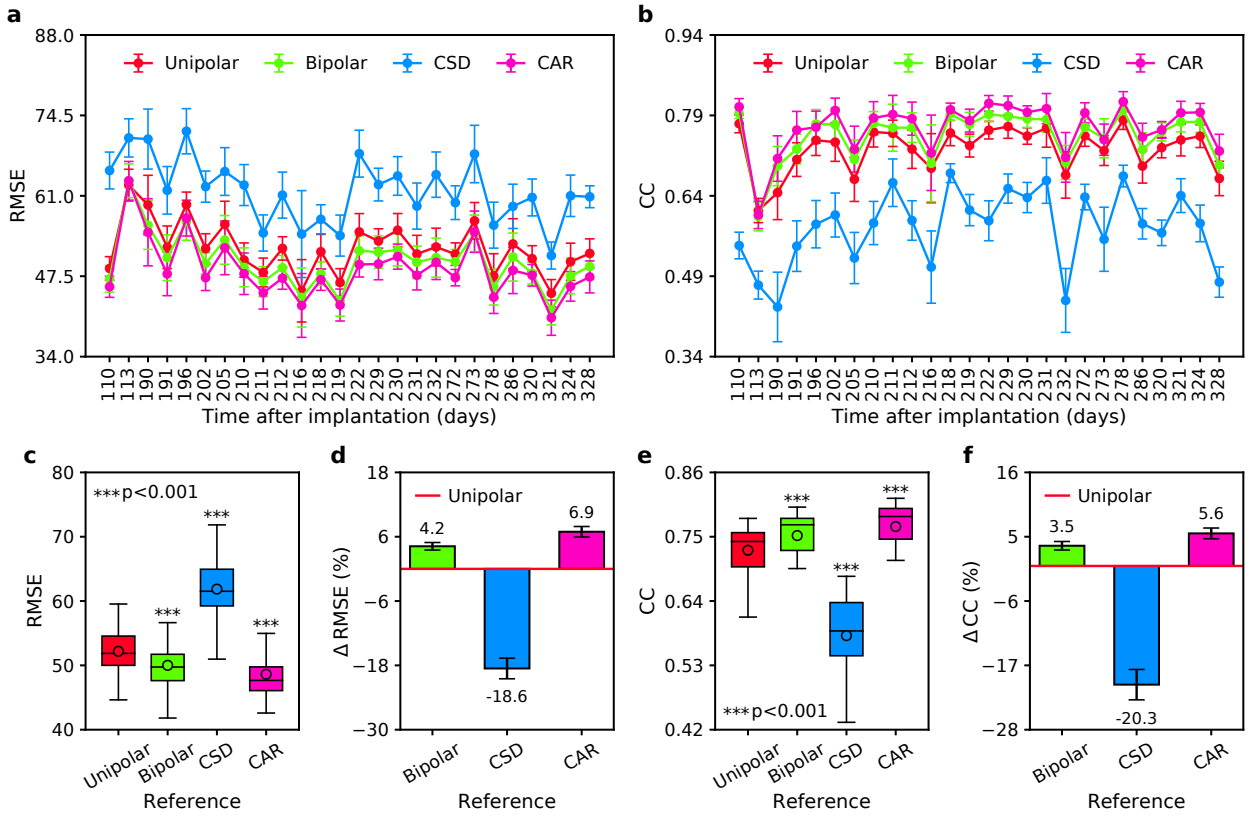


Figure 6.7 Decoding performance comparison across different referencing schemes using LMP-driven GRU decoder. (a),(b) Performance comparison over long-term recording sessions measured in RMSE and CC, respectively. (c),(e) Boxplot comparison across sessions measured in RMSE and CC, respectively. Asterisks indicate referencing schemes that yielded statistically significant different performances from that of unipolar reference (*** $p<0.001$). (d),(f) Performance improvement/degradation (in percent RMSE and CC, respectively) relative to unipolar reference. Positive (negative) value indicates performance improvement (degradation). Black error bars denote 95% confidence intervals.

6.4 Discussion

In this chapter, we present an experimental investigation of the impact of four referencing schemes, namely unipolar, bipolar, current source density (CSD), and common average reference (CAR), on the decoding performance of LFP-based BMI. Firstly, we seek to determine which feature within LFP carries the highest movement-related information as characterised by its decoding performance. Several prior studies have shown that power in the high-frequency band (i.e. high-gamma) of LFPs hold the highest information about movement compared to other frequency bands [32], [41], [92]. The high-gamma power is believed to reflect synchronised spiking activity arising from a local population of neurons near the recording electrodes [231],

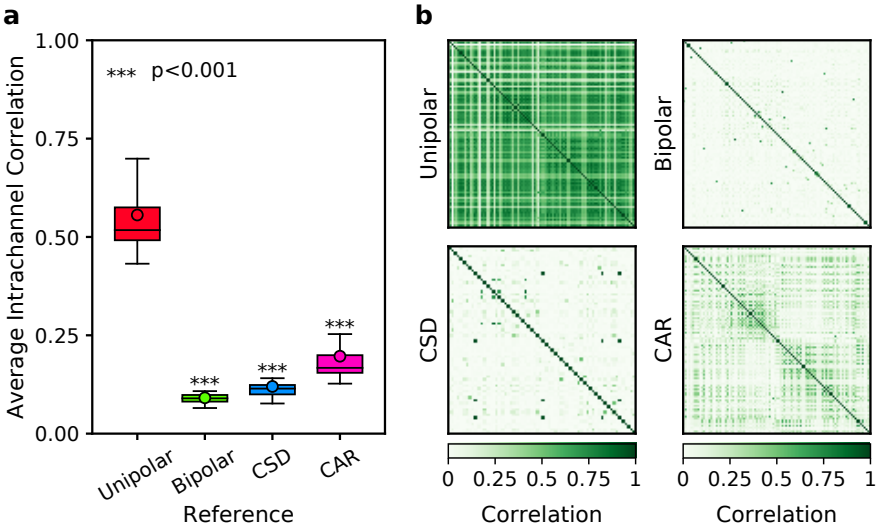


Figure 6.8 Comparison of interchannel correlation of LFPs across different referencing schemes. (a) Boxplot comparison of average interchannel correlation over 26 recording sessions. Asterisks indicate referencing schemes that yielded statistically significant different performances from that of unipolar reference (*** $p<0.001$). (b) Comparison of interchannel correlation matrix across referencing schemes (data from recording session I20160627_01).

[232]. In these studies, however, local motor potential (LMP), which is a time-domain LFP feature, was not included in the experiments and analyses.

Here, we extend the above studies by comparing six LFP features consisting of LMP and power within five frequency bands: delta, theta, alpha, beta, and gamma. Our empirical results revealed that LMP significantly outperformed all the other LFP features. It was observed that LMP consistently achieved higher decoding accuracy across different referencing schemes. The high informativeness of LMP could be related to the view that LMP represents evoked LFP changes in response to behavioural tasks [232], [233]. Heretofore, it remains unclear what the physiological origin of LMP is. Different hypotheses have suggested that LMP is associated with firing rate modulation of neuronal population in the vicinity of the recording electrode [233] or activity of a distant neuronal population connected to the recording site [234], [235].

We also found that gamma always yielded the second-highest decoding performance after LMP. These findings are in good agreement with Stavisky *et al.*'s study [37], but contradict Flint *et al.*'s study [35] which shows that gamma was more informative than LMP. This contradiction could be due to the difference in the way we divide the LFP frequency bands and measure the informativeness of LFP features. Following Stavisky *et al.*'s study [37], we define gamma frequency band as 50–300 Hz, while Flint *et al.* split gamma frequency band into 70–200 Hz

(low-gamma) and 200–300 Hz (high-gamma) [35]. In this chapter and [37], the informativeness of LFP feature is measured independently according to its decoding performance, whereas in [35] it is calculated based on relative contribution (i.e. filter weight) from top 150 mixed LFP features. Regardless of this contradiction, these three studies support the notion that LMP and gamma are both highly predictive for decoding kinematic data.

Next, we examined the impact of different interelectrode distances $(400, 800, \cdots, 2400\,\mu m)$ on the decoding performance in bipolar reference. Results from both LMP and gamma features showed that increasing the interelectrode distance tended to degrade the decoding performance. One possible reason for this trend could be related to the spatial spread of LFP which has been reported within around 200–400 μm [236]–[238], although other studies suggest a larger spatial spread up to a few millimetres [239], [240]. Bipolar referencing is computed by subtracting one electrode from the other in an attempt to remove common noise component, yielding a more localised neural activity. Common noise arising from a distant source through volume conduction can be effectively removed if the interelectrode distance is sufficiently close [219]. If this is too close (high correlation between a pair of electrodes), bipolar referencing may pose a risk of subtracting out signal from approximately the same source. However, if this is too far apart, bipolar referencing could subtract out signal from a different source with a very little common noise component. An interelectrode distance of 400 μm may reflect sufficiently different signal sources between the electrode pairs but exhibit considerably high common noise. This common noise could be effectively eliminated through bipolar referencing, which in turn lead to better decoding performance.

We then compared the impact of referencing schemes on the decoding performance using LMP-driven LSTM decoder. Results showed that CAR consistently outperformed all the other referencing schemes over long-term recording sessions. There were statistically significant differences in the decoding performance between CAR and other referencing schemes. The high decoding performance of CAR may be attributed to the fact that we use a relatively high number and density of electrode array (96 channels). It has been previously suggested that CAR would perform reasonably well if the number and density of the electrodes are high [221]. When using a different decoder, gated recurrent unit (GRU) network [229], [230], we also observed the

same trend, which indicates the robust performance of CAR. Another finding was that bipolar referencing yielded the second-highest decoding performance after CAR. Apart from the ability to remove common noise, bipolar referencing could also add more informative features (due to many combinations of electrode pairs) to the decoder. A rather surprising finding was that, although CSD could remove common noise, its decoding performance was lower than the other referencing schemes. This is likely ascribed to the fact that electrodes at the edges (not having four neighbouring electrodes) were excluded, effectively throwing away some information. We further examined whether the above results were dependent on the choice of the feature. When using gamma-driven LSTM decoder, it turned out that bipolar yielded the highest decoding performance followed by CAR, unipolar, and CSD, respectively. Upon investigation on this phenomena, we found that the gamma features obtained from unipolar and CAR were highly correlated (average CC = 0.78), whereas LMP features were less correlated (average CC = 0.50). This suggests that CAR is not effective when using gamma as the feature. It is important to note that the decoding performance of CAR coupled with LMP was found to be significantly better than bipolar coupled with gamma.

Lastly, we examined the impact of referencing schemes on the interchannel correlation. We observed that bipolar, CSD, and CAR decreased the interchannel correlation substantially with respect to unipolar. However, the decrease in interchannel correlation thought to reflect some degree of common noise does not necessarily translate into better decoding performance. The decoding performance is not only affected by the use of referencing schemes, but also by other factors such as feature type and feature size among others. For example, as previously described, although CSD had substantially lower interchannel correlation, it also had smaller number of features, which led to lower decoding performance relative to unipolar.

The impact of referencing schemes on the analysis of LFP signals (e.g. power, phase, coherence, and Granger causality) have been comparatively studied before. This chapter extends these prior studies by comparing the impact of referencing schemes on the decoding performance of LFP-based BMI. To the best of our knowledge, this chapter is the first study that comprehensively investigates the impact of referencing schemes for BMI application using LFPs. Similar studies using EEG signals have reported contradictory findings, that is, CSD

(i.e. Laplacian) yielded the best performance [241]–[243]. However, it is difficult to make a direct comparison between these studies and ours due to notable differences in several factors such as signal sources (scalp vs intracortical), subjects (human vs monkey), behavioural tasks (movement imagery vs movement execution), signal processing and decoding algorithms among others.

In summary, our overall results suggest the use of CAR coupled with LMP for LFP-based BMIs. Both CAR and LMP are simple, computationally lightweight, and amenable for on-chip implementation and real-time application [37], [244]. On top of that, LFPs have been demonstrated to possess long-term signal stability [22], [35], [52], which is critical for widespread clinical use of BMI.

6.5 Conclusion

We have presented a systematic and comprehensive investigation of the impact of the referencing schemes on the decoding performance of LFP-based BMI. Our experimental results showed that LMP was the most informative LFP feature regardless of the referencing schemes, indicating its richness of movement-related information. Using LMP-driven deep learning algorithms, we further showed that CAR significantly and consistently yielded better decoding performance relative to other referencing schemes. These findings provide empirical justification for the use of CAR coupled with LMP for enhancing the decoding performance of LFP-based BMI. The simplicity and amenability of both CAR and LMP for on-chip implementation along with the long-term stability of LFP signal could pave the way towards low power, fully implantable, and clinically viable BMI.

Part III

Developing Robust and Accurate BMIs

7 | Robust and Accurate Decoding of Hand Kinematics from Entire Spiking Activity Using Deep Learning

This chapter aims to improve both robustness and accuracy of BMIs by exploring a new, alternative input signal and decoding algorithm. We propose entire spiking activity (ESA) as the input signal and quasi-recurrent neural network (QRNN) as the decoder. The proposed ESA-driven QRNN decoder is compared against SUA-, MUA-, and LFP-driven decoders developed in the previous chapters. We also examine the stability characteristics of each input signal over time. Moreover, we assess the impact of different decoder's training schemes and the amount of training data on decoding performance.

A version of this chapter was submitted to *Journal of Neural Engineering* and deposited to *bioRxiv* preprint server with the following title:
N. Ahmadi, T. G. Constandinou, and C.-S. Bouganis, "Robust and accurate decoding of hand kinematics from entire spiking activity using deep learning", *bioRxiv*, 2020. [8]

7.1 Introduction

The robustness and accuracy of BMI systems are greatly influenced by the type of input signal and decoding algorithm. Despite the appealing advantages of MUA and LFP, a considerable number of published studies have reported that the decoding accuracy of MUA [22], [26], [27], [38], [39] and LFP [33], [35], [39]–[41] are lower than that of SUA. Therefore, it is highly desirable to have an input signal that is not only stable but also yields high decoding accuracy.

Many BMI studies employ linear decoders, e.g. Wiener filter (WF) [14], [15], [125]

and Kalman filter (KF) [11], [32], [33], [37], [44], [75], which could yield suboptimal decoding accuracy as neural signals are known to exhibit nonlinear and nonstationary properties [49]. Although there exists a nonlinear extension of WF and KF, called Wiener cascade filter (WCF) [35], [52], [53] and unscented Kalman filter (UKF) [54], [57], WCF and UKF assume that the noise is (additive) stationary and Gaussian, respectively. If these a priori assumptions are violated, the decoding performance of both decoders will not be optimal. The rise of artificial intelligence, particularly deep learning, with its state-of-the-art performance across a wide range of tasks (e.g. image and speech recognition, language translation, and medical diagnostics) [58], [59], has raised questions of whether such method can significantly improve the decoding performance of BMIs. Recent studies have demonstrated that different variants of deep learning decoders, such as standard recurrent neural network (SRNN), long short-term memory (LSTM), and gated recurrent unit (GRU), outperform classical decoders [60]–[62], [153], Despite this, the amount of BMI research utilising deep learning is still quite limited. Thus there is much room for further improvement.

The issues of alternative input signal and decoder for improving the robustness and accuracy of BMIs have been so far addressed separately. In this chapter, we aim to address these both issues simultaneously by exploring a new input signal and deep learning decoder. Specifically, we propose entire spiking activity (ESA) as the input signal and quasi-recurrent neural network (QRNN) as the decoder. We evaluate the performance of ESA-driven QRNN decoder for offline prediction of hand kinematics from neural data chronically recorded from primary motor cortex area of a nonhuman primate. We benchmark our proposed approach against combinations of three input signals (SUA, MUA, and LFP) and seven decoders (SRNN, LSTM, GRU, WF, WCF, KF, and UKF). We also examine the characteristics (stability and quality) of each input neural signal over time. Lastly, we assess the impact of different training schemes and the amount of training data on decoding performance.

7.2 Methods

The block diagram of LFP-based BMI system used in this chapter is illustrated in Figure 7.1. All the experiments and analyses were done in Python. TensorFlow deep learning framework

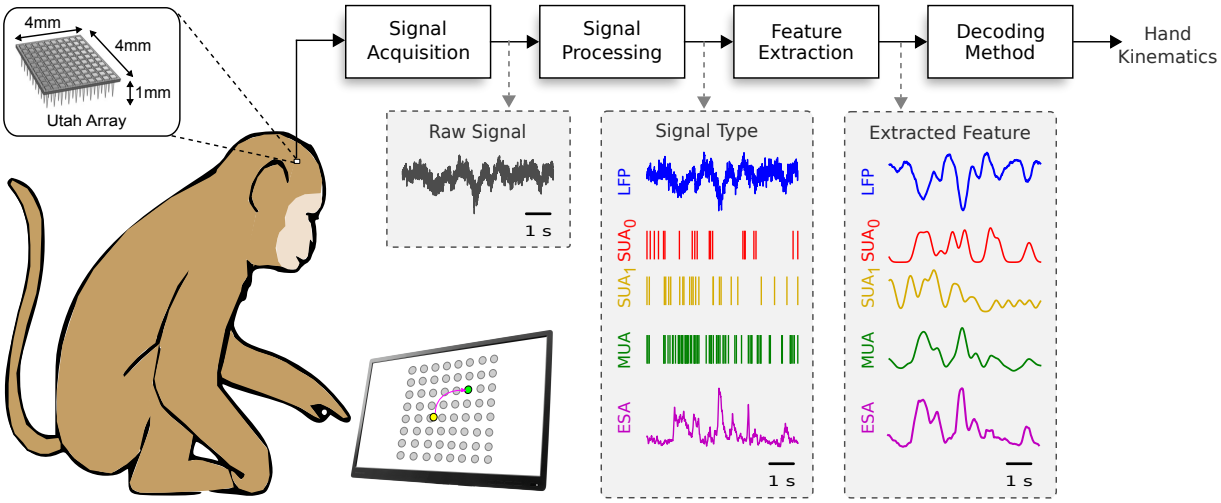


Figure 7.1 Schematic diagram of the brain-machine interface (BMI) system. Raw neural signals were recorded using a 96-channel Utah microelectrode array from primary motor cortex (M1) area of a monkey while performing continuous reaching tasks. Four different types of neural signals were obtained from the raw neural signals, namely local field potential (LFP), single-unit activity (SUA), multiunit activity (MUA), and entire spiking activity (ESA). Features extracted from these neural signals were used as inputs for different decoding methods.

[209] was used to implement the decoding algorithms.

7.2.1 Neural recordings and behavioural task

We used the same neural dataset as in Chapter 4. The neural recording setup and behavioural task are described in Section 4.2.1 and 4.2.2, respectively.

7.2.2 Signal processing and feature extraction

From the raw neural signals, we computed four different types of neural signals, namely local field potential (LFP), single-unit activity (SUA), multiunit activity (MUA), and entire spiking activity (ESA). Examples of these four neural signals along with their associated features are shown in Figure 7.1. The signal processing and feature extraction algorithms were all implemented in Python and are described briefly as follows.

Local field potential (LFP)

The raw neural signals were first digitally re-referenced using common average reference (CAR) to eliminate common noise. CAR referencing has been shown in Chapter 6 to yield better

decoding performance. LFPs were obtained by low-pass filtering the re-referenced neural signals with a 4th-order Butterworth filter at 300 Hz followed by downsampling them to 1 kHz. To avoid any phase shift, the filtering processes in this chapter were by default performed on the forward and backward directions (unless otherwise stated). Local motor potential (LMP) was extracted by using a moving average filter with 256 ms rectangular window in an overlapping fashion (252 ms overlap) to yield a sample every 4 ms, thereby matching the timescale of the kinematic data. LMP was selected because it yielded better decoding performance than other LFP features (Section 5.3.1 and 6.3.1). LFP features from all the 96 channels were included in our experiments and analyses.

Single-unit activity (SUA)

SUA processing has been described in Section 4.2.3. Briefly, spikes were obtained by band-pass filtering the raw neural signals (between 500 to 5000 Hz) and detecting any amplitudes crossing a threshold value. SUA was then obtained through spike sorting procedure. Features (firing rates) were estimated from SUA using BAKS with 256 ms Gaussian window sliding every 4 ms. Only SUA with firing rates > 0.5 Hz were included, resulting in 91 to 157 SUA (average: 125.73 ± 13.95) across 26 recording sessions.

Multi-unit activity (MUA)

MUA processing has been described in Section 4.2.3. Briefly, MUA is defined as all the spikes detected through threshold crossing technique (without spike sorting). MUA Features were extracted using BAKS with 256 ms Gaussian window sliding every 4 ms. Only MUA with firing rates exceeding 0.5 Hz were included, yielding 77 to 95 MUA (average: 87.08 ± 4.44) across 26 recording sessions.

Entire spiking activity (ESA)

The term ESA was firstly used by Drebitz *et al.* in 2018 [245] but its underlying principle has been known and used for several decades [246], [247]. ESA was previously utilised to measure evoked responses (e.g. receptive field) in the visual cortex of monkeys upon receiving various

types of visual stimuli [237], [248]–[251]. While both SUA and MUA are represented by a sequence of binary signals, ESA is represented by a continuous signal reflecting an instantaneous measure of the number and size of spikes from a population of neurons surrounding the electrode tip [246], [252]. ESA was obtained by digitally re-referencing the raw neural signals with common average reference (CAR) and followed by high-pass filtering with 1st-order Butterworth filter at 300 Hz. The filtered signals were then full-wave rectified (i.e. taking the absolute value), low-pass filtered with 1st-order Butterworth filter at 12 Hz and downsampled to 1 kHz. We extracted a time-domain average amplitude feature (similar to LMP) using a moving average filter with a 256 ms rectangular window overlapping by 252 ms (yielding a feature sample every 4 ms).

7.2.3 Decoding methods

We benchmarked eight decoding algorithms which can be categorised into classical decoders and deep learning decoders; each category consists of four decoding algorithms. The classical decoders include Wiener filter (WF), Wiener cascade filter (WCF), Kalman filter (KF), and unscented Kalman filter (UKF), while the deep learning decoders include standard recurrent neural network (SRNN), long short-term memory (LSTM), gated recurrent unit (GRU), and quasi-recurrent neural network (QRNN). TensorFlow (Keras) deep learning framework was used to implement the deep learning decoders. Each of these eight decoders is briefly described as follows.

Wiener filter (WF)

We used the same WF described in Section 4.2.5. The regularisation parameter (α) was set to 0 (by default) and increased to a large value if there was a significant drop in decoding performance.

Wiener cascade filter (WCF)

WCF is an extension of WF to deal with nonlinear dynamic systems. It is composed of two stages: a linear dynamic system followed by a static nonlinearity [35], [52], [53]. The first stage

was implemented by a WF, whereas the second stage was implemented by fitting a 3rd order polynomial between the WF output and the kinematic data. The number of taps and the regularisation parameter (α) were set equal to that of the WF.

Kalman filter (KF)

We used the same KF described in Section 4.2.5. The regularisation parameter (α) was by default set to 0 except in a few sessions (increased to a large value) when a significant drop in decoding performance was encountered.

Unscented Kalman filter (UKF)

UKF is an extension of KF to deal with nonlinear systems. It utilises an unscented transform, that is, a method to calculate the statistics of a random variable passing through a nonlinear transformation. It builds on the intuition that approximating a probability distribution is easier than approximating an arbitrary nonlinear function [144]. This can be accomplished by using a minimal set of carefully and deterministically chosen sample points known as sigma points. We followed the UKF implementation described in Li *et al.*'s study [54]. The state transition and measurement matrices were estimated using the same method as of the KF. The scaling parameter (κ) and number of taps were both set to 1.

Standard recurrent neural network (SRNN)

SRNN is a class of neural networks where the outputs from the previous timestep are used as inputs in the current timestep. SRNN has a "memory" in the form of a "hidden state" which allows it to remember sequential information that has been calculated so far. SRNN offers several advantages, including ability to process variable length input, taking into account historical information, and shared weights across timesteps. The output at timestep t in SRNN can be formulated as:

$$\mathbf{h}_t = \tanh(\mathbf{W}\mathbf{x}_t + \mathbf{U}\mathbf{h}_{t-1} + \mathbf{b}_h) \tag{7.1}$$

where $\mathbf{x}, \mathbf{h}$, and tanh denote the input, hidden state, and hyperbolic tangent activation function, respectively. $\mathbf{W}, \mathbf{U}$ and $\mathbf{b}$ represent input weight, recurrent weight, and bias vector. The

graphical illustration of SRNN is shown in Figure B.7a.

Long short-term memory (LSTM)

LSTM is graphically illustrated in Figure B.7b. The LSTM architecture used in this study has been previously described in Section 4.2.5.

Gated recurrent unit (GRU)

GRU is graphically illustrated in Figure B.7c. The GRU architecture used in this study has been previously described in Section 6.2.4.

Quasi-recurrent neural network (QRNN)

QRNN was recently introduced by Bradbury *et al.* in 2016 [253] that combines the parallelism power of convolutional neural networks (CNNs) with the capability of RNNs in learning temporal dependencies of sequential data. It is composed of two components: convolutional component that performs convolutions in parallel across timesteps and pooling component that handles temporal dependencies in parallel across feature dimensions. The full set of equations in QRNN is defined as follows:

$$\begin{aligned}
\mathbf{z}_t &= \tanh(\mathbf{W}_z * \mathbf{X} + \mathbf{b}_z) \\
\mathbf{f}_t &= \sigma(\mathbf{W}_f * \mathbf{X} + \mathbf{b}_f) \\
\mathbf{o}_t &= \sigma(\mathbf{W}_o * \mathbf{X} + \mathbf{b}_o) \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + (1 - \mathbf{f}_t) \odot \mathbf{z}_t \\
\mathbf{h}_t &= \mathbf{o}_t \odot \mathbf{c}_t
\end{aligned} \tag{7.2}$$

where $\mathbf{z}, \mathbf{f}, \mathbf{o}, \mathbf{c}, \mathbf{h}$ consecutively represent the candidate vectors, forget gate, output gate, memory cell, and hidden state. Operator $*$ denotes a masked convolution which is a type of convolutions that depends only on the past and present inputs (i.e. cannot access inputs from future timesteps). QRNN is graphically illustrated in Figure B.7d.

7.2.4 Hyperparameter optimisation

For each deep learning decoder, the final output was obtained by connecting the last timestep of output from the RNN layer with a fully connected layer. The number of layers and timesteps were empirically set to 1 and 2, respectively. Other hyperparameters such as the number of units, number of epochs, batch size, dropout rate and learning rate were determined through hyperparameter optimisation.

We split each recording session into 10 non-overlapping contiguous segments of equal size which were then categorised into three different sets: training set (8 concatenated segments), validation set (1 segment) and testing set (1 segment). The training set was used to train the deep learning (DL) decoders; the validation set was used to optimise the hyperparameters of the DL decoders; the testing set was used to evaluate the performance of the optimised DL decoders. All features being fed to the DL decoders were standardised (i.e. z-transformed) to have zero mean and unit variance. The DL decoders were trained using a root mean squared error (RMSE) loss function and an RMSprop optimiser. A Bayesian optimisation library called Hyperopt [213] was used for the hyperparameter optimisation which was conducted separately for each type of neural signal and run for 300 iterations. To save the computational time of experiments, the hyperparameter optimisation was performed only once using the first recording session. The hyperparameters for each DL decoder were optimised from predefined hyperparameter search spaces listed in Table B.1. The value ranges for number of units were specified so that the minimum and maximum possible number of parameters (weights) are comparable across different decoders. The hyperparameter optimisation result for the case of ESA-driven QRNN decoder is visually illustrated in Figure B.8. The resulting optimised hyperparameters for each decoder driven by ESA and for QRNN decoder driven by LFP, SUA, and MUA are provided in Table 7.1. The optimised hyperparameters for each decoder driven by LFP, SUA, and MUA are given in Table B.2.

7.2.5 Decoder training schemes

According to the training scheme being used, we divided the decoders into three categories: (1) fixed decoder, (2) retrained decoder, and (3) resumed decoder. Each of these categories is

briefly described as follows:

- *Fixed decoder.* The training was performed at the first session and the trained weights were kept fixed for the rest of the sessions. If in the subsequent sessions there were channels that did not record neural activity, these channels were assigned with zeros to keep the input structure (i.e. the number of channels) remain the same.

- *Retrained decoder.* The training was performed at every session without taking into account the information from the previous session. All the decoders were by default trained by using this scheme (unless otherwise stated).

- *Resumed decoder.* The training was performed at every session while keeping the information from the previous session. This is done by storing the model (i.e. decoder) from the previous session and resuming the training from where we left off using the current session data.

7.2.6 Performance evaluation and metrics

The decoding performance was evaluated using root mean square error (RMSE) and Pearson's correlation coefficient (CC). The stability of each neural signal over time was assessed by calculating the slope of its regression line. The slope was calculated using Huber regression, one type of robust linear regression methods, that is less sensitive to outliers without ignoring their effect. We tested whether the neural signal was statistically significantly stable using a one-tailed one-sample t-test. For each session, the decoding performance was evaluated on 10 different segments within the testing set to obtain the mean and confidence interval. To test whether there was statistical difference between a pair combination of neural signal and decoder, we used a Wilcoxon signed-rank test. The significance level (α) was set to 0.05.

Boxplots were used to visualise the decoding performance comparison among neural signals/decoders across 26 sessions. The horizontal line and circle mark inside each box represent the median and mean, respectively. The coloured solid box represents the interquartil range (between 25th and 75th percentiles). The whisker extends 1.5 times the interquartile range. All the analyses were conducted in Python.

Table 7.1 Hyperparameter configurations of deep learning decoders driven by ESA and QRNN decoders driven by other neural signals.

Hyperparameter	ESA Signal				QRNN Decoder		
	SRNN	GRU	LSTM	QRNN	LFP	SUA	MUA
Number of units	450	175	200	400	350	400	400
Number of epochs	3	8	5	2	2	4	4
Batch size	96	96	32	64	64	96	64
Dropout rate	0.3	0.2	0.5	0.5	0.1	0.4	0.4
Learning rate	0.0005	0.004	0.004	0.0015	0.0005	0.003	0.005

7.3 Results

Figure 7.1 presents the schematic diagram of processing steps used in this study, which employs four neural signals (LFP, SUA, MUA, and ESA) and eight decoders that are categorised into classical decoders and deep learning decoders. The classical decoders include WF, WCF, KF, and UKF, whereas the deep learning decoders include SRNN, LSTM, GRU, and QRNN. The result of hyperparameter optimisation for deep learning decoders driven by ESA and QRNN decoder driven by other neural signals is shown in Table 7.1. Hyperparameter configurations for other combinations of deep learning decoders and neural signals are listed in Table B.2.

7.3.1 Characteristic comparison across neural signals

Firstly, we quantified the characteristics of stability and quality of SUA and MUA over time with three metrics: (1) number of units, (2) average peak-to-peak voltage amplitude (Vpp; in μm) across units, and (3) mean spike rate (Hz) across units. Since the number of channels for LFP and ESA remained the same (96 channels) throughout the recording sessions, to quantify the characteristics of LFP and ESA, we only used their mean amplitude (μm) across channels. Figure 7.2 shows the characteristic comparison across different neural signals. We found statistically significant decreasing trends for SUA in all three metrics: the number of units declined by 11.19% (* $p < 0.05$), the Vpp declined by 18.26% (*** $p < 0.001$), and the mean spike rate declined by 0.11% (* $p < 0.05$) per day, as shown in Figures 7.2a,b,c, respectively. As for MUA (see Figures 7.2a,b,d), the number of units and Vpp increased by 1.48% and 186.57% per day, respectively, whereas the mean spike rate statistically significantly decreased by 0.56%

per day (*** $p < 0.001$). In the case of LFP, the mean amplitude was found to increase by 1.27% per day (Figure 7.2e). We did not observe a statistically significant decreasing trend in the mean amplitude of ESA although the rate was negative (-0.88% per day) as can be seen in Figure 7.2f. We then examined the redundancy characteristics of neural signals by measuring their inter-

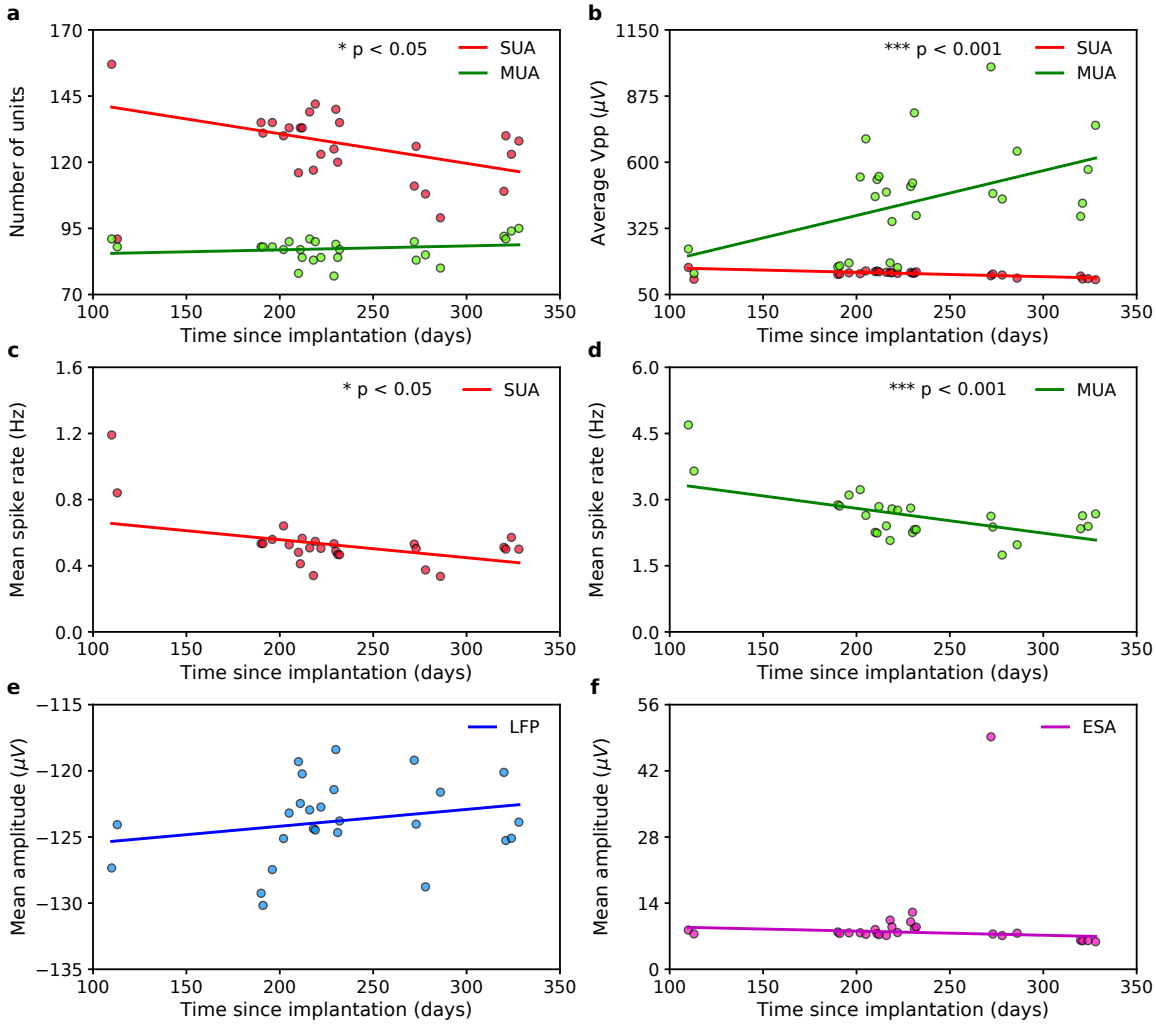


Figure 7.2 Long-term stability of different types of neural signals. (a) Number of SUAs and MUAs over time. (b) Average peak-to-peak voltage amplitudes (Vpp; in μm) of SUA and MUA over time. (c),(d) Mean spike rates (Hz) of SUA and MUA over time, respectively. (e),(f), Mean amplitudes (μm) of LFP and ESA over time, respectively. Huber based robust linear regression was used to test whether or not the trends significantly linearly decreased. Asterisks represent the trends of neural signals that significantly linearly decreases (one-tailed one-sample t-test; * $p < 0.05$, *** $p < 0.001$).

channel/unit correlation. For each neural signal, we calculated the correlation between all possible pairs of units/channels and averaged them across units/channels. The average inter-channel/unit correlation for all neural signals is displayed in Figure 7.3. SUA was found to have the lowest average correlation (0.10 $\pm$ 0.01; mean $\pm$ SD), whereas LFP had the highest

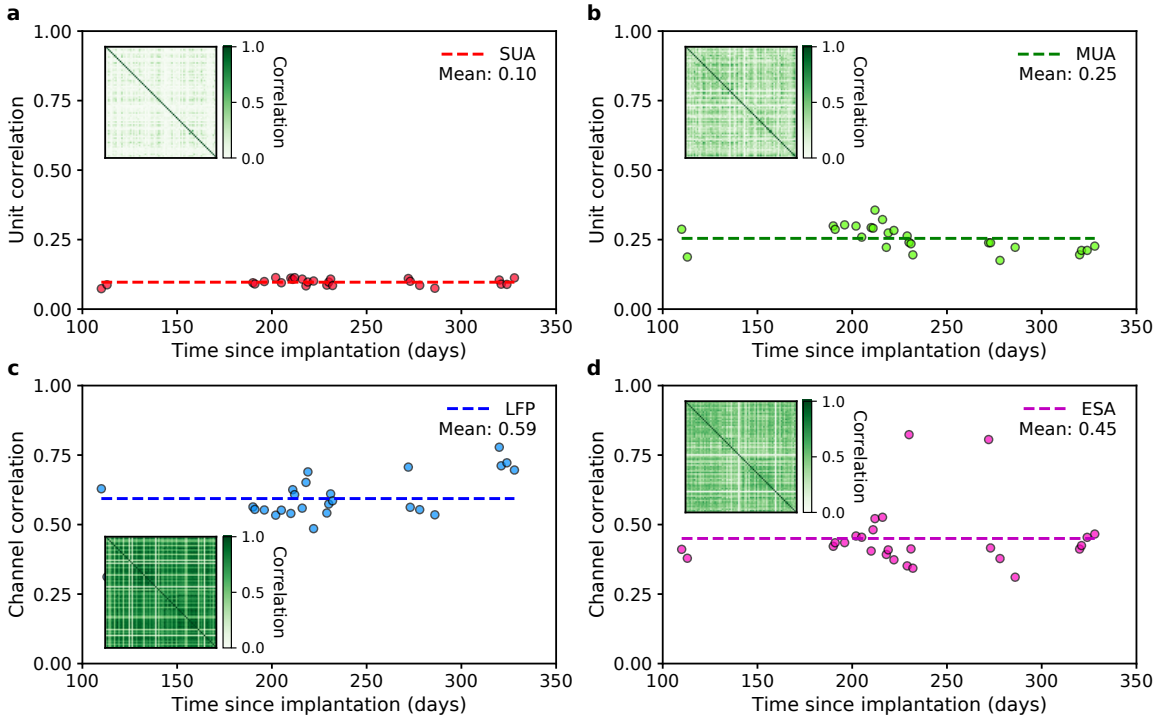


Figure 7.3 Inter-channel/unit correlation of neural signals over time. (a),(b) Average inter-unit correlation across all possible pairs of units within SUA and MUA, respectively. (c),(d) Average inter-channel correlation across all possible pairs of channels within LFP and ESA, respectively. The dashed lines represent the mean of correlation across all the sessions (n = 26). The insets depict the heatmap of inter-channel/unit correlation from different neural signals taken from the first session (I20160627_01).

average correlation (0.59 ± 0.09) as can be seen in Figure 7.3a and Figure 7.3d, respectively. On the other hand, MUA and ESA had an average correlation of 0.25 ± 0.05 and 0.45 ± 0.12, respectively, as shown in Figure 7.3b and Figure 7.3c, respectively. By sorting according to the average correlation scores, we could obtain the following descending order: LFP > ESA > MUA > SUA. It is important to note that the average inter-channel correlation of LFP and ESA above were obtained from recordings referenced to a single common reference (referred to as unipolar reference). To eliminate correlated noise commonly found in unipolar-based recordings, we applied a common average reference (CAR) referencing to the raw neural data. The CAR referencing significantly reduced the average inter-channel correlation for LFP (0.25 ± 0.10; 57.33%) but only marginally for ESA (0.45 ± 0.10; 0.2%).

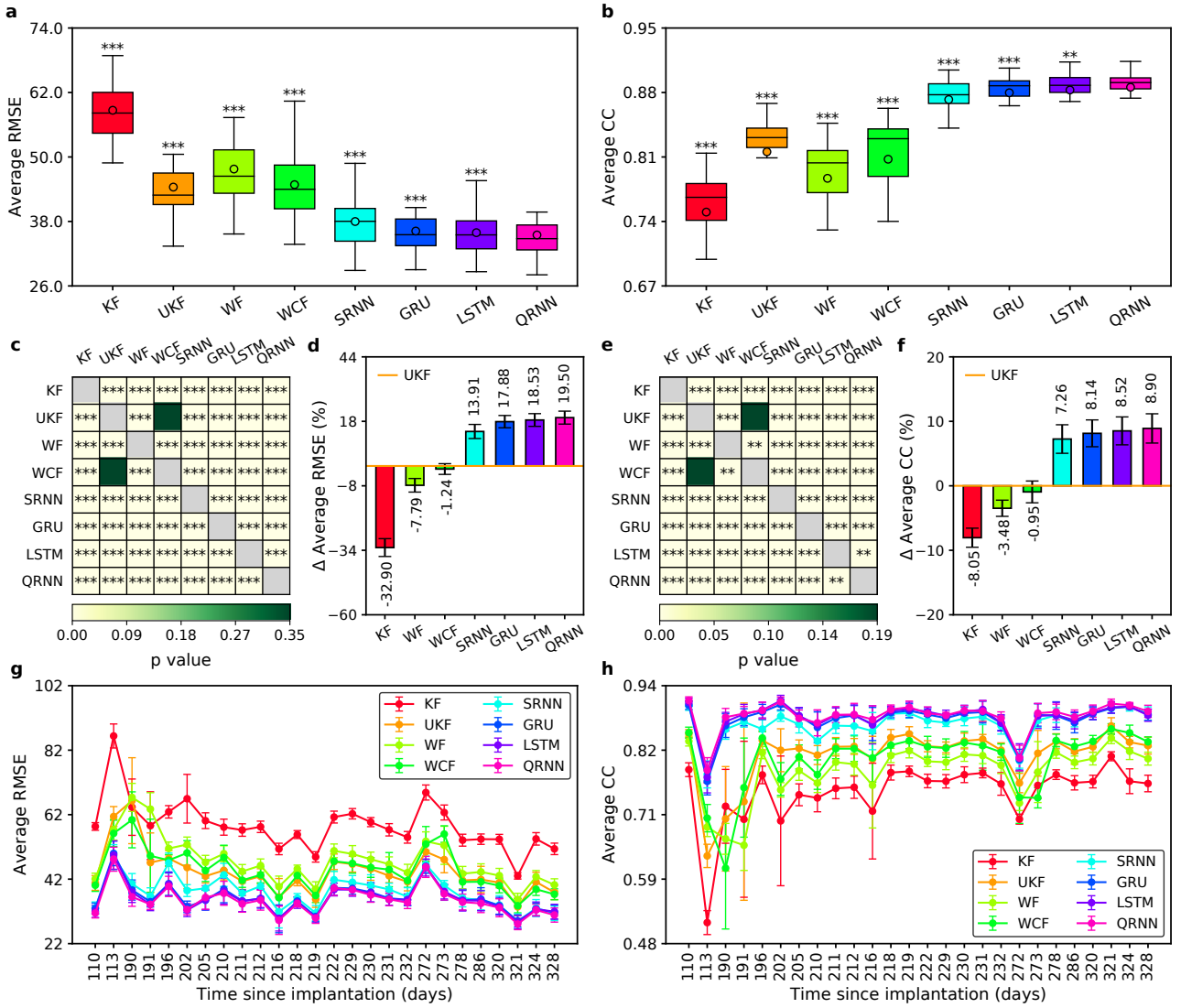


Figure 7.4 Long-term decoding performance comparison across different decoders driven by ESA signal. (a),(b) Boxplot comparison of long-term decoding performance across decoders as measured by average RMSE and CC, respectively. The horizontal lines inside the boxes denote the medians whereas the circles denote the means. Asterisks represent decoders whose decoding performance differed significantly from that of QRNN (two-tailed paired Wilcoxon signed-rank test; ** $p < 0.01$, *** $p < 0.001$). (c),(e) Heatmap of statistical significant matrix of decoding performance between all possible pairs of decoders as measured by average RMSE and CC, respectively. Each cell is coloured based on the p-value except cells in the main diagonal (grey; excluded from statistical test calculation). Asterisks indicate whether there were statistical significant differences between the pairs (two-tailed paired Wilcoxon signed-rank test; ** $p < 0.01$, *** $p < 0.001$). (d),(f) Average performance improvement/degradation (in %) of decoders with respect to that of UKF as measured by RMSE and CC metrics, respectively. Positive (negative) value indicates performance improvement (degradation). Black numeral texts represent the means whereas black error bars represent the 95% confidence intervals ($n = 26$). (g),(h) Decoding performance comparison over time across decoders in terms of average RMSE and CC, respectively. Coloured circles represent the means whereas coloured error bars represent the 95% confidence intervals ($n = 10$) for each session. The x-axes represent the number of days since electrode implantation. The x-axes' ticks correspond to different sessions having irregular gaps between the consecutive sessions.

7.3.2 Decoding performance comparison across decoders

Next, we compared offline decoding of hand velocity from ESA signal using different decoders over time. The long-term decoding performance comparison results across ESA-driven decoders are depicted in Figure 7.4. In the case of classical decoders, UKF yielded the highest average decoding performance (RMSE = 44.39 ± 1.38, CC = 0.82 ± 0.01; mean ± SEM), whereas KF yielded the lowest average decoding performance (RMSE = 58.69 ± 1.56, CC = 0.75 ± 0.01). In the case of deep learning decoders, QRNN produced the highest average decoding performance with RMSE of 35.48 ± 0.89 and CC of 0.89 ± 0.01, while SRNN produced the lowest average decoding performance with RMSE of 37.99 ± 0.99 and CC of 0.87 ± 0.01 (see Figures 7.4a,b). Overall, the order of decoders from the highest to the lowest decoding performance was as follows: QRNN > LSTM > GRU > SRNN > UKF > WCF > WF > KF. As evident from Figures 7.4c,e, all the deep learning decoders significantly outperformed all the classical decoders (two-tailed paired Wilcoxon signed-rank test; *** $p < 0.001$). By taking UKF, the best performing classical decoder, as the baseline, the deep learning decoders improved the average decoding performance by 13.91%–19.50% (RMSE) and 7.26%–8.90% (CC) as illustrated in Figures 7.4d,f. In particular, our proposed QRNN decoder yielded decoding performance improvement on average by 19.50% ± 2.61% (RMSE) and 8.90% ± 2.27% (CC). Compared to the other classical decoders, QRNN achieved even higher decoding performance improvement. We found that QRNN yielded statistically significantly higher decoding performance than the other deep learning decoders (** $p < 0.01$, *** $p < 0.001$). We further sought to determine whether the superior decoding performance of the deep learning decoders over the classical decoders was consistently observed on each session. Our extensive results showed that all the deep learning decoders always outperformed their classical counterparts in each of 26 sessions for both RMSE and CC metrics as depicted in Figure 7.4g and Figure 7.4h, respectively. In addition, QRNN outperformed other deep learning decoders in 26/26 sessions (100% against SRNN), 25/26 sessions (96.15% against GRU), and 20/26 sessions (76.92% against LSTM).

One may be interested in investigating whether above findings were also found in the cases of different neural signals. We therefore compared the decoding performance across decoders driven by SUA, MUA, and LFP. The decoding performance comparison results for the

cases of SUA, MUA, and LFP are shown in Figure B.9, Figure B.10, and Figure B.11, respectively. We found consistent findings where all the deep learning decoders yielded significantly higher decoding performance than all the classical decoders (* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$). Moreover, we found that on average QRNN outperformed the other deep learning decoders across 3 neural signals, 26 sessions, and 2 metrics.

7.3.3 Decoding performance comparison across neural signals

Further, we assessed the decoding performance of QRNN decoders across different neural signals. Figure 7.5 presents the decoding performance comparison of QRNN decoders driven by LFP, SUA, MUA, and ESA. The results revealed that ESA yielded the highest average decoding performance with RMSE of 35.48 ± 0.89 and CC of 0.89 ± 0.01 (mean ± SEM). After ESA, the next highest average decoding performances (in descending order) were achieved by MUA (RMSE = 41.71 ± 0.86, CC = 0.84 ± 0.01), SUA (RMSE = 44.42 ± 1.01, CC = 0.81 ± 0.01), and LFP (RMSE = 47.45 ± 0.98, CC = 0.79 ± 0.01) as shown in Figures 7.5a,b. Statistical tests showed that there were statistically significant differences between all possible pair combinations of neural signals (two-tailed paired Wilcoxon signed-rank test; *** $p < 0.001$) as illustrated in Figures 7.5c,e. By taking SUA as the baseline, ESA improved the decoding performance by 20.12% ± 1.78 (RMSE) and 9.33% ± 1.52 (CC) as can be observed from Figures 7.5d,f. The decoding performance improvements gained by ESA with respect to the other neural signals were as follows: 25.15% ± 2.63% (RMSE) and 12.99% ± 2.00% (CC) against LFP, and 15.02% ± 1.68% (RMSE) and 5.98% ± 0.80% (CC) against MUA. The superior decoding performance of ESA compared to the other neural signals was consistently observed across all 26 sessions (see Figures 7.5g,h). To investigate whether the superior performance of ESA was dependence on the presence of spikes recorded on the same electrodes, we removed the spike components from ESA signal. We followed the procedure described in Stark *et al.*'s study [254], that is, by replacing all the detected spikes with high-frequency noise. This is done by subtracting smoothed spikes (computed through a five-sample moving average) from the associated raw neural signals. These spikeless ESA signals are referred to as despiked ESA (ESA_D). We then performed the same offline decoding of hand kinematics using ESA_D-driven QRNN

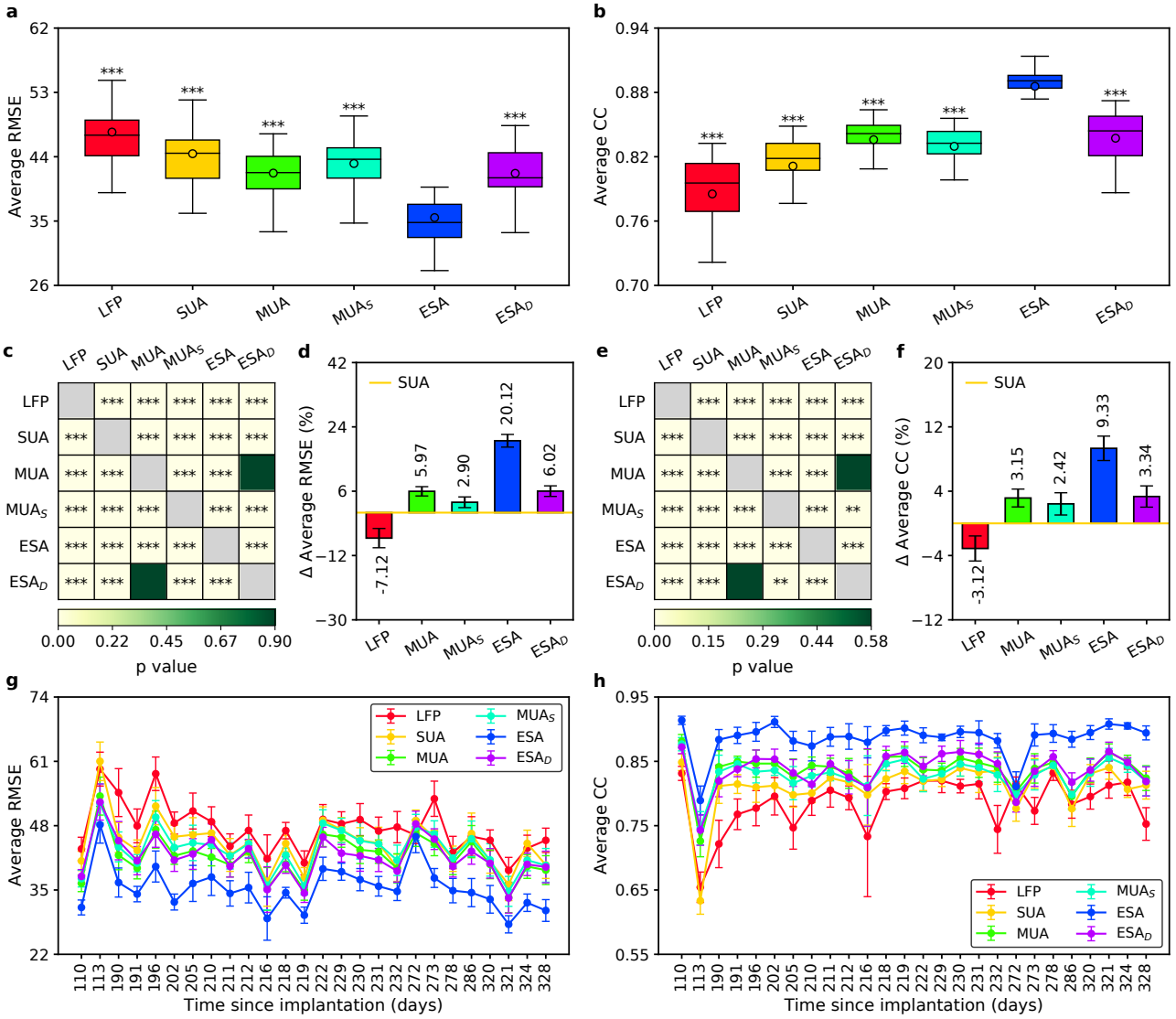


Figure 7.5 Long-term decoding performance comparison across QRNN decoders driven by different neural signals. (a),(b) Boxplot comparison of long-term decoding performance across different neural signals as measured by average RMSE and CC, respectively. The horizontal lines inside the boxes denote the medians whereas the circles denote the means. Asterisks represent neural signals whose decoding performance differed significantly from that of ESA (two-tailed paired Wilcoxon signed-rank test; *** $p < 0.001$). (c),(e) Heatmap of statistical significant matrix of decoding performance between all possible pairs of neural signals as measured by average RMSE and CC, respectively. Each cell is coloured based on the p-value except cells in the main diagonal (grey; excluded from statistical test calculation). Asterisks indicate whether there were statistical significant differences between the pairs (two-tailed paired Wilcoxon signed-rank test; ** $p < 0.01$, *** $p < 0.001$). (d),(f) Average performance improvement/degradation (in %) of neural signals with respect to that of SUA as measured by RMSE and CC metrics, respectively. Positive (negative) value indicates performance improvement (degradation). Black numeral texts represent the means whereas black error bars represent the 95% confidence intervals ($n = 26$). (g),(h) Decoding performance comparison over time across neural signals in terms of average RMSE and CC, respectively. Coloured circles represent the means whereas coloured error bars represent the 95% confidence intervals ($n = 10$) for each session. The x-axes represent the number of days since electrode implantation. The x-axes' ticks correspond to different sessions having irregular gaps between the consecutive sessions.

decoder. The results revealed that although there was significant degradation (by 17.85% ± 2.20% in RMSE and 5.46% ± 0.62% in CC) in the average decoding performance relative to ESA, the average decoding performance of ESA_D was still significantly better than that of LFP and SUA (*** $p < 0.001$) and comparable to that of MUA, as shown in Figures 7.5a,b and Figures 7.5c,e). Specifically, the decoding performance improvements achieved by ESA_D were 12.02% ± 2.12% (RMSE) and 6.77% ± 1.57% (CC) against LFP, 6.02% ± 1.48% (RMSE) and 3.34% ± 1.32% (CC) against SUA, and 0.02% ± 1.38% (RMSE) and 0.18% ± 0.61% (CC) against MUA.

We also examined whether MUA inherently contained the same amount of information (i.e. spiking activity) as ESA and differed only from the signal processing point of view. We thus approximated the feature of ESA by convolving a Gaussian kernel with spike train from MUA. The smoothing parameter of the Gaussian kernel was selected by minimising RMSE between the actual ESA and approximated ESA (referred to as smoothed MUA or MUA_S). We subsequently performed the same procedure of offline decoding using MUA_S-driven QRNN decoder. The results showed that the decoding performance of MUA_S was significantly lower (RMSE = 43.05 ± 0.85, CC = 0.83 ± 0.00) than that of MUA and ESA, indicating that MUA and ESA have a different amount of information content.

We extended the analysis of decoding performance comparison across neural signals using the other deep learning decoders. The comparison results for SRNN, GRU, and LSTM are presented in Figure B.12, Figure B.13, and Figure B.14, respectively. From these results, we found the same trend as above, that is, ESA produced statistically significantly higher decoding performance than LFP, SUA, and MUA (*** $p < 0.001$). The decoding performance of ESA corresponded to an improvement of 9.85%–30.24% (RMSE) and 4.76%–20.42% (CC) with respect to the other neural signals.

In order to examine the impact of different numbers of features (units or channels) on decoding performance, we selected n units or channels randomly from each neural signal. The n values used for SUA were $\{1, 10, 20, 30, \cdots, 150\}$, while the n values used for LFP, MUA, ESA were $\{1, 10, 20, 30, \cdots, 90\}$. We then compared the average decoding performance from n units/channels of each neural signal using QRNN decoder. We performed the same procedure

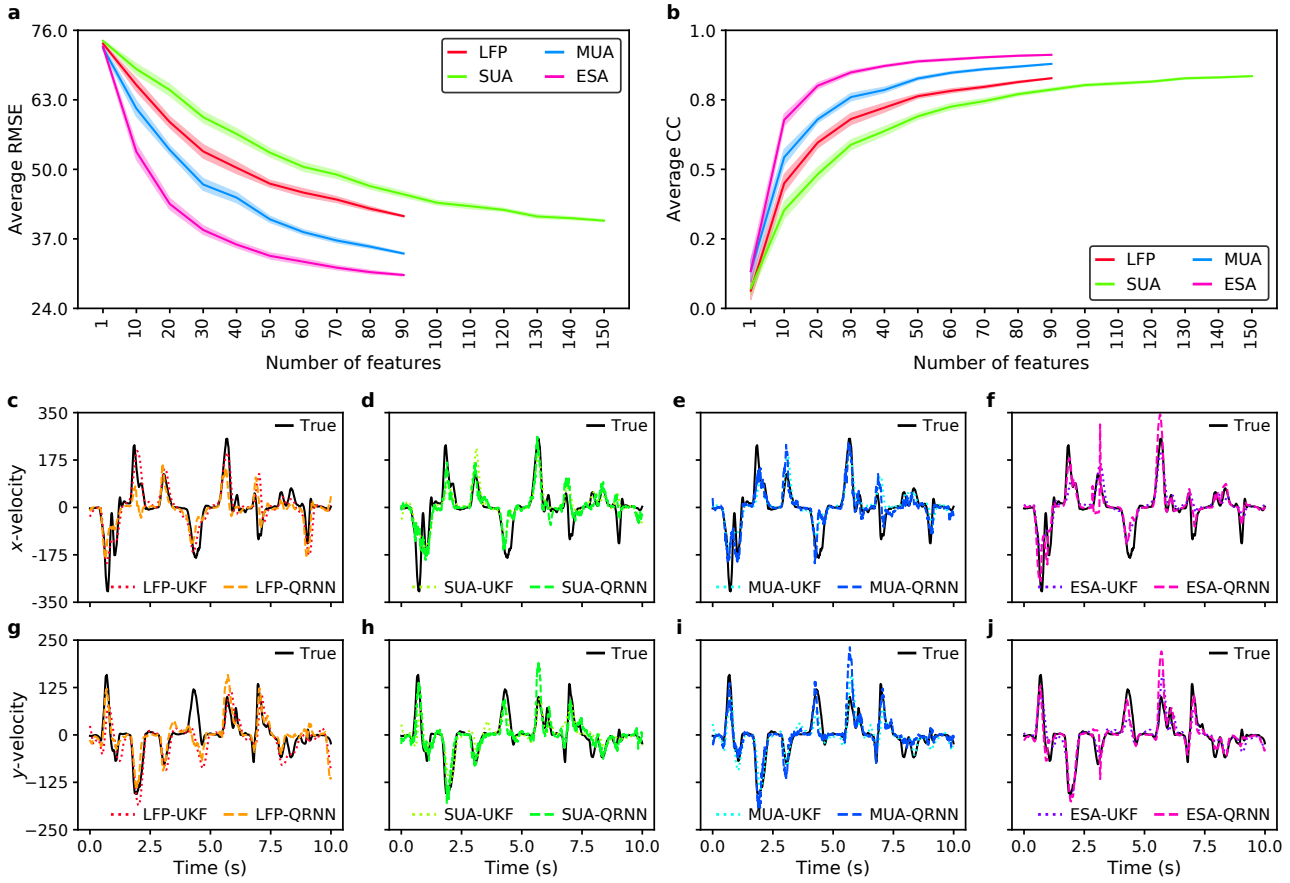


Figure 7.6 Decoding performance comparison across different types of neural signals. (a),(b) Impact of different number of features (units or channels) on the decoding comparison of QRNN decoders driven by different neural signals as measured by average RMSE and CC, respectively. The shaded areas represent the 95% confidence intervals ($n = 30$). (c)-(f) Snippet examples of true and decoded x-velocities from LFP-, SUA-, MUA-, and ESA-driven decoders, respectively. (g)-(j) Snippet examples of true and decoded y-velocities from LFP-, SUA-, MUA-, and ESA-driven decoders, respectively. Data are taken from the first session (I20160627_01).

for 30 iterations to obtain the mean and confidence interval. The comparison results from the first session (I20160627_01) across different neural signals are presented in Figures 7.6a,b. We found that when $n = 1$, MUA yielded the highest decoding performance, followed by ESA, LFP, and SUA. However, when $n > 1$, ESA always produced the highest decoding performance compared to all the other neural signals. when $n < 130$, LFP showed better decoding performance than SUA but when $n \geq 130$, LFP was outperformed by SUA. As observed in Figures 7.6a,b, the performance of all neural signals improved with the increasing n and reached a plateau after a certain value of n. ESA was found to reach a plateau quickest, whereas SUA reached a plateau slowest than the others. Similar findings were also observed when using data from the last session (I20170131_02) as displayed in Figure B.15. Snippet examples (10 s) of the true and

decoded velocities (x-coordinate) from UKF and QRNN decoders driven by LFP, SUA, MUA, and ESA are graphically visualised in Figures 7.6c-f. Likewise, snippet examples of the true and decoded velocities in y-coordinate from the same decoders are plotted in Figures 7.6g-j.

7.3.4 Impact of different training schemes and the amount of training data

To investigate how frequently a decoder should be updated or whether a static decoder could sustain high decoding performance over time, we trained QRNN decoder using three training schemes: (1) trained the decoder only at the first session and kept it fixed for the rest sessions (referred to as fixed QRNN), (2) retrained the decoder at every session without taking into account the data from the previous sessions (referred to as retrained QRNN), and (3) retrained the decoder at every session by resuming (i.e. continuing) the training from the previous session (referred to as resumed QRNN). The training was performed on 80% of data from each session, while the testing was performed on 10% of data sequentially located after the training data to mimic the real application scenario. The results showed that, on average, the retrained QRNN yielded slightly higher decoding performance than the resumed QRNN and significantly higher decoding performance than the fixed QRNN as illustrated in Figures 7.7a,b. Although the fixed QRNN exhibited degraded performance compared to the retrained QRNN and resumed QRNN, its decoding performance was still moderately good except at 3 out of 26 sessions.

The high accuracy of deep learning is often attributed to a large amount of data it uses during training. To test this preconception, we varied the amount of training data from 1 to 10 minutes with an increment of 1 minute. We analysed and compared the decoding performance of deep learning decoders with that of the classical decoders. Figure 7.7c and Figure 7.7d plot the impact of increasing the amount of training data on the decoding performance across eight different decoders measured by RMSE and CC, respectively. We found that, on average, the decoding performance of all decoders improved the increasing amount of training data. However, all the decoders on average had already reached a plateau when the training data equals to 5 minutes. Another finding was that both QRNN and LSTM outperformed UKF (the best performing classical decoder) even when the training data was small ($\leq$ 3-minute length).

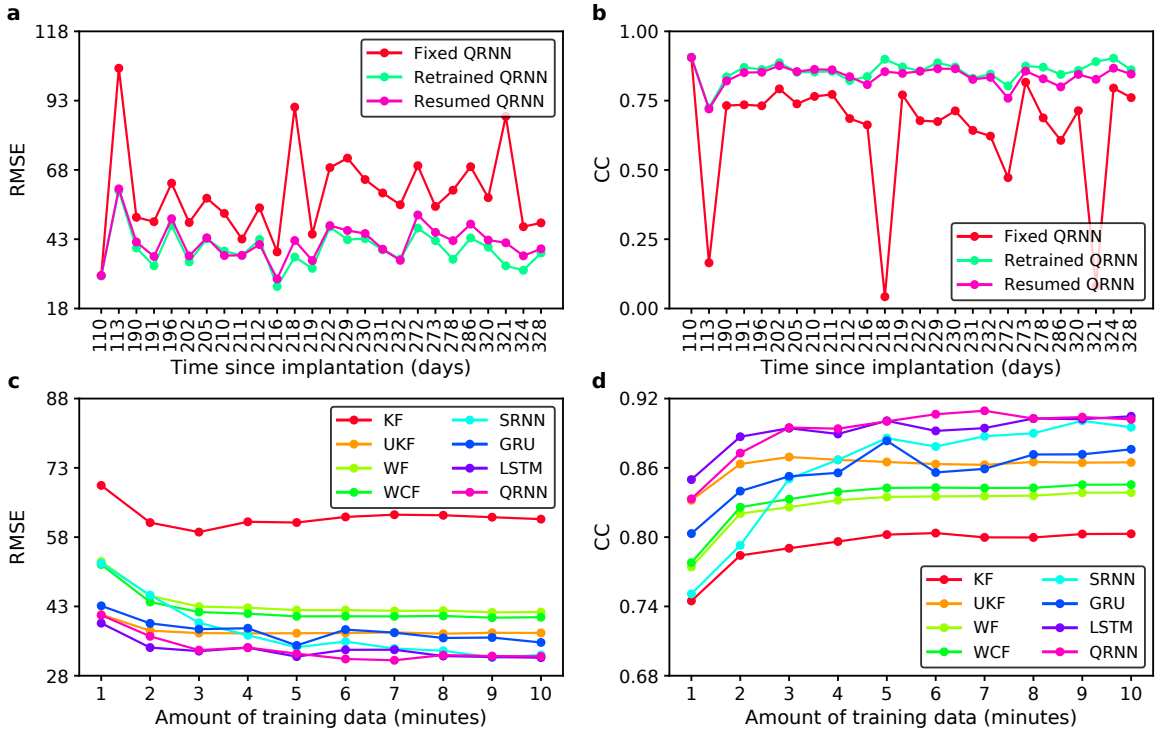


Figure 7.7 Decoding performance comparison across decoders under different training schemes and amount of training data. (a),(b) Impact of the training schemes on long-term decoding performance of QRNN decoder across sessions as measured by RMSE and CC, respectively. (c),(d) Impact of the amount of training data on decoding performance of different decoders as measured by RMSE and CC, respectively. Data are taken from the first session (I20160627_01).

On the other hand, SRNN and GRU were outperformed by UKF when using a small amount of training data. Regardless of the amount of training data, KF was always found to yield the lowest decoding performance.

7.4 Discussion

In this chapter, we address the issues of robustness and accuracy of BMIs simultaneously by exploring a new, alternative input neural signal and decoding algorithm (i.e. decoder). We propose entire spiking activity (ESA) as an input neural signal and quasi-recurrent neural network (QRNN) as a decoder and comprehensively compare its decoding performance (i.e. accuracy) over a long period of time (i.e. robustness). Firstly, we characterised the long-term stability and quality of our proposed entire spiking activity (ESA) signal and compared its characteristics with commonly used neural signals which include SUA, MUA, and LFP. As expected, SUA showed statistically significant decrease in all three characteristics over time:

(1) number of recorded units, (2) average peak-to-peak amplitude (Vpp; in μm), and (3) mean spike rate (Hz). MUA was found to exhibit a significant decrease only in the mean spike rate, indicating its better stability than SUA. On the other hand, LFP did not show a decreasing trend in the mean amplitude, while ESA showed a slow (not statistically significant) decreasing trend, suggesting that both LFP and ESA are more stable over time than SUA and MUA. One approach to mitigate the signal instability issue faced by SUA-based BMIs is by utilising MUA as an alternative input neural signal [19], [26], [27], [39]. Another approach is to use LFP, which has been believed and demonstrated to be stable over a long period of time even when the spikes have already deteriorated or lost [22], [35], [52], [218], [255]. Our results confirm the previous studies by showing that both LFP and MUA exhibit better signal stability than SUA. We further extend the previous findings by demonstrating that ESA also possesses the characteristic of long-term signal stability, which could potentially be used as an alternative input neural signal for BMIs.

To examine the redundancy characteristics of neural signals, we compared the inter-channel/unit correlation between all possible pair combinations of channels/units within each neural signal. LFP was found to have the highest average correlation with a significantly large gap from the other neural signals, while SUA was found to have the lowest average correlation. This is in good agreement with prior studies [34], [35], [39], [256]. Overall comparison results on long-term data yielded the following descending order: LFP $>$ ESA $>$ MUA $>$ SUA. The results may also indicate the degree of independence between channels/units within each neural signal. The high redundancy of LFP might be attributed to the contamination of correlated noise arising from different sources, e.g. non-silent reference and volume conduction [219], [220] as commonly observed in LFP recordings with unipolar reference. To remove the correlated noise, we re-referenced the raw neural signals using common average reference (CAR) and re-computed the LFP. The resulting CAR-based LFP had substantially lower inter-channel correlation. In addition, the different level of redundancy could be related to the spatial integration or coverage of each neural signal. LFP is thought to mainly reflect summed synaptic activity from a local neuronal population ranging from a radius of a few hundred micrometres to a few millimetres around the recording electrodes [30], [238]–[240]. As the Utah array has an inter-electrode

distance of 400 μm, the recorded LFP on each electrode could therefore contain (to some degree) overlapping neural activity. On the other hand, SUA represents spike (action potential) activity from individual neurons which can be well discriminated within a radius less than $\sim$50 μm [257] or up to $\sim$130 μm [258] from the recording electrodes. The small spatial coverage of SUA may partly explain its low inter-unit correlation and high degree of independence. MUA, which represents the aggregate spikes from an ensemble of neurons near the electrodes, has a spatial coverage larger than SUA but less than LFP. Several studies reported that the spatial coverage of MUA is within a radius of $\sim$140–300 μm [117]–[120]. As for ESA, it is considered to reflect an instantaneous measure of the number and size of spikes from a population of neurons around the electrodes [246], [252]. The spatial coverage of ESA is likely larger than that of MUA (but smaller than that of LFP) as it includes the contribution from small amplitude spikes (either from small nearby neurons or big farther neurons) which may not be detected through threshold crossing technique. [249], [251].

Next, using ESA as the input, we benchmarked offline decoding performance of our proposed QRNN decoder against three deep learning decoders (SRNN, GRU, and LSTM) and four classical decoders (WF, WCF, KF, and UKF). Our analysis resulted in several findings as follows. First, QRNN yielded significantly higher decoding accuracy compared to all the other decoders and could maintain its high accuracy over a long period of time. Second, all the deep learning decoders significantly outperformed all the classical decoders. Third, the superiority of the deep learning decoders, particularly QRNN, over the classical counterparts were consistently observed across different neural signals (SUA, MUA, and LFP) and sessions. Forth, within the category of classical decoders, UKF emerged as the highest performing decoder. These findings reinforce previous studies highlighting the high performance of deep learning decoders (SRNN, GRU, LSTM) in comparison to classical decoders [60]–[62], [153]. This chapter further expands the previous studies by demonstrating that QRNN yielded better decoding accuracy and robustness over time than the other deep learning decoders. There is concern that the high performance of deep learning decoders is attributable to a large enough amount of training data. To test this preconception, we varied the amount of training data from 1 to 10 minutes and compared the decoding performance across deep learning and classical decoders.

Comparison results demonstrated that all the decoders exhibited performance decline with the decreasing amount of training data. However, we found that QRNN and LSTM still performed considerably well compared to the others even when the training data were small. This finding is in line with a recent report by Glaser *et al* [60]. One potential implication of this finding is that the data collection for decoder training could be shortened which in turn speeds up the decoder calibration and deployment processes while minimising the subject's learning fatigue. Due to the variability and nonstationarity of neural signals, a fixed (i.e. static) decoder may degrade over the course of hours, days, or months. To address this issue, the decoder needs to be recalibrated regularly at the beginning of each session or when there is a significant drop in the decoding performance. Here, we investigate the impact of three training schemes on the decoding performance: retraining only at the first session (fixed decoder); retraining from scratch at every session (retrained decoder); and resuming the training from the previous session (resumed decoder). Extensive results using different deep learning decoders showed that retrained decoders performed slightly better than resumed decoders and substantially better than fixed decoders. Taking into account the information from previous sessions during training as in resumed decoders on average did not improve the decoding performance compared to retraining from scratch at every session (retrained decoders). This could indicate the disparity between the previous and current sessions arising probably from changes in neural encoding pattern or recording degradation [22], [259]. Even though fixed decoders exhibited a significant performance decline (as anticipated), their decoding performance was still moderately good in most (except a few) sessions. This suggests that under circumstances where decoder recalibration cannot be carried out frequently on per-session basis, fixed decoder can be leveraged. The recalibration process can then be performed only when the performance of a fixed decoder substantially deteriorates.

Further, we quantified movement-related information content within each neural signal by comparing their decoding performance using QRNN decoder. Results revealed that ESA significantly outperformed the other neural signals and could sustain its high decoding performance across all the sessions. The order of neural signals according to their decoding performance (from high to low) was ESA $>$ MUA $>$ SUA $>$ LFP. The same trend was con-

sistently observed when using different deep learning decoders, indicating that ESA holds rich information about the movement parameters. To gain more insights into whether the superiority of ESA is dependence on the presence of spikes recorded from the same electrodes, we removed the spike components from the raw neural signals and recomputed ESA (termed as depsiked ESA or ESA_D) using a method described by Stark *et al* [254]. We found that ESA_D still produced better decoding performance than SUA, MUA, and LFP, suggesting that ESA contains additional information other than spikes detected through threshold crossing technique (as in SUA and MUA). Spike detection through threshold crossing technique relies on manual or automatic selection of threshold value. This could be problematic in chronic recordings with low SNR or high variability over time. If the threshold value is too small, it may detect some background noise as spikes, increasing the false positive rate. Conversely, too high threshold value could leave true spikes undetected. In addition, threshold crossing technique may provide a biased estimate of spikes in favour of large neurons (with large amplitude), thus leaving spikes from small neurons undetected [249], [251]. Contrary to SUA and MUA, ESA uses a simple, automated, and threshold-less method that allows a more reliable and less biased estimate of spikes and takes into account the contribution from all neurons (including the small ones) in the vicinity of the electrodes [249], [251]. We examined this issue further by approximating ESA from MUA (referred to as smoothed MUA or MUA_S) through a Gaussian kernel-based smoothing with an optimised width parameter. Results showed that MUA_S could not approach the level of decoding performance of ESA. This may support the idea that there is additional information in ESA which is not available in MUA.

Another finding in this chapter is that LFP yielded the lowest decoding performance despite its better long-term signal stability, indicating its less movement-related information content than the other neural signals. This finding is consistent with a considerable number of prior studies [33], [35], [37], [41], [92]. However, in the long run, as the number of detected spikes decreases, LFP could outperform both SUA and MUA as demonstrated in previous reports by Flint *et al.* [52], [53] and Wang *et al* [22]. In addition, our analysis on unit/channel-dropping curve showed that LFP yielded higher decoding performance than SUA when the number of channels/units were relatively small. When using all the available channels/units,

SUA outperformed LFP, which compares well with previous works [33]–[35], [41]. This could act as an additional indication that LFP is more redundant (less independence) than SUA. On the other hand, ESA always yielded the highest decoding performance compared to the other neural signals across different decoders, sessions, and numbers of channels. With the increasing number of channels/units, the performance of ESA reached a plateau quicker than the others, suggesting that ESA exhibits some degree of redundancy. The implication of this is that we could obtain accurate and robust decoding performance using ESA even without using all the available channels, which can reduce computational complexity, memory footprint, and power consumption.

ESA has been previously used to measure evoked responses in the visual cortex upon receiving various visual stimuli [248]–[251], whereas QRNN has been mainly used for natural language processing (NLP) applications [253], [260]. There is no previous study that uses both ESA and QRNN for intracortical BMI application. To the best of our knowledge, this chapter is the first study to systematically and comprehensively compare long-term signal and decoding stability across different neural signals and decoders simultaneously. Our results agree well with prior studies that used a neural signal which in principle is similar to ESA [254], [261]. However, there are differences in the signal processing point of view to obtain such signal. After high-pass filtering, this study uses absolute operation (i.e. full-wave rectification) followed by a low-pass filtering at 12 Hz, whereas those prior studies use root mean square (RMS) operation followed by a low-pass filtering at 100 Hz. Absolute operation is simpler and easier than RMS operation especially when implemented on hardware (i.e. chip). Lower frequency cut-off enables signals to be downsampled into a lower sampling rate, which in turn reduces the computational complexity.

The limitation of this study is that we use neural data recorded approximately from 3.6 to 10.9 months after implantation, corresponding to a total of 26 sessions. Although there is a gradual decline over time, the number of detected spikes at the last session is still sufficiently high. It remains to be tested whether ESA-based QRNN decoder could maintain its high decoding performance for many years when spikes have severely diminished or even lost. Our analysis using despiked ESA suggests that ESA could still provide good performance even when

the spikes are lost. A recent study using a signal similar to ESA demonstrated that such a signal could sustain good performance over three years worth of recordings [261].

7.5 Conclusion

We have proposed ESA as an alternative input signal and QRNN as a decoding algorithm and presents a systematic comparison between various combinations of input signals and decoding algorithms. Our overall results suggest that ESA signal coupled with deep learning decoder, particularly QRNN, could provide robust and accurate decoding over time. In addition, the simple, automated, threshold-less processing along with a relatively low sampling rate of ESA may enable the implementation of a fully implantable wireless BMI system. This, in turn, could pave the way to a clinically viable BMI.

8 | Inferring Entire Spiking Activity from LFPs with Deep Learning

This chapter attempts to shed light on the relationship between ESA and LFP by investigating whether we can infer ESA solely from simultaneously recorded LFPs. We employ an LSTM based deep learning as an inference method and evaluate it on neural recordings from monkeys' motor cortex during point-to-point and instructed delay reach-to-grasp tasks. We analyse which LFP features are most informative about ESA. Next, we compare the inference accuracy of ESA with that of SUA and MUA. Finally, we examine the contribution of each LFP channel and the effect of the number of LFP channels on the inference accuracy.

A version of this chapter was submitted to *Scientific Reports* and deposited to *bioRxiv* preprint server with the following title:
N. Ahmadi, T. G. Constandinou, and C.-S. Bouganis, "Inferring entire spiking activity from local field potentials with deep learning", *bioRxiv*, 2020. [9]

8.1 Introduction

Understanding the relationship between the LFP and spikes is important for brain-machine interface (BMI) research, for example, to measure movement-related information within different neural signals [33], [256] and extract indirect spiking information features from LFPs in biofeedback-based BMIs [36]. In addition, it plays a critical role in addressing many issues in neuroscience research, such as neuronal functional organisation and connectivity, neuronal communication between different brain areas, cell assembly formation, neural coding and information processing [235], [262]–[264]. As LFPs are thought to represent mainly the input to local neuronal networks, while spiking activity represents the output from local neuronal networks [265], it is conceivable to relate the two signals based on system identification-based

inference approach. Previous studies have shown that SUA [36], [232], [266] and MUA [256], [267] can be inferred solely from LFPs with moderately good accuracy. So far, however, there has been no study that investigates the relationship between the LFPs and ESA.

In light of this, this chapter aims to address the above-mentioned issue by using extracellular recordings from the motor cortex area of two macaque monkeys while performing two different tasks: point-to-point task and instructed delay reach-to-grasp task. We assess quantitatively how well ESA can be inferred from LFPs with deep learning, specifically long short-term memory (LSTM) network. We also analyse which features within LFPs that are highly predictive of ESA. Furthermore, we compare the inference accuracy of ESA with that of SUA and MUA. Finally, we examine the LFP channel importance/contribution to and the effect of the number of LFP channels on the inference accuracy.

8.2 Methods

Figure 8.1 illustrates a schematic overview of signal processing and deep learning (DL) based inference steps. The DL algorithm was implemented using Pytorch deep learning framework [268]. All the experiments and analyses were conducted in Python.

8.2.1 Neural recording and behavioural task

Electrophysiological recordings used in this chapter were obtained from two public neural datasets [210], [269] herein referred to as dataset I and dataset II. These datasets were recorded from motor cortex area of two adult male Rhesus macaque monkeys (*Macaca mulatta*) while performing predefined tasks with 96-channel silicon-based intracortical microelectrode (Utah) array. The data acquisition and behavioural task associated with each dataset are briefly described below.

Dataset I

This dataset is described in our previous chapter (Section 4.2.1 and 4.2.2). It consists of 26 recording sessions spanning 7.3 months between the first (I20160627_01) and last (I20170131_02) sessions. The recording duration varies from 6 to 13.6 minutes (average: 8.88 $\pm$ 1.96 minutes).

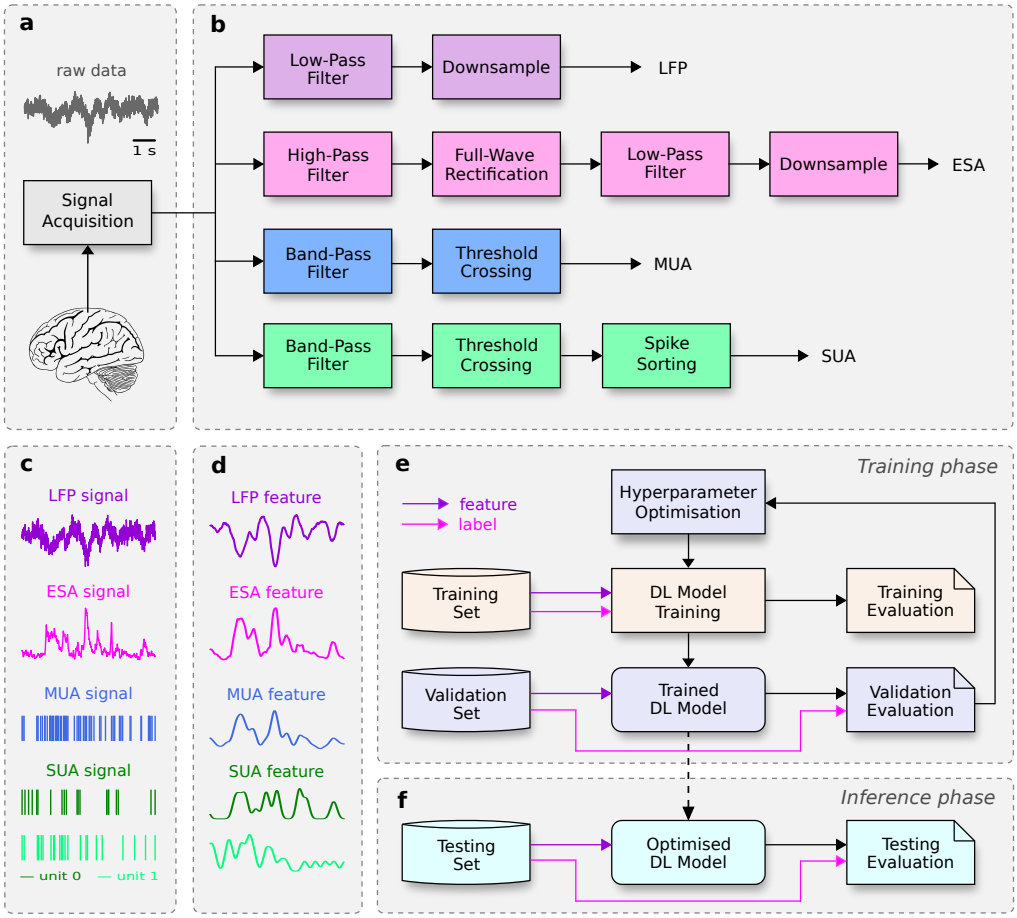


Figure 8.1 Schematic illustration of signal processing and deep learning (DL) based inference steps. (a) Raw neural signal acquisition from motor cortex area of monkey with a 96-channel intracortical Utah array. (b) Signal processing steps for different types of neural signals (LFP, ESA, MUA, and SUA). (c) LFP, ESA, MUA, and SUA signals obtained from the processing steps. (d) Extracted features from LFP, ESA, MUA, and SUA signals. (e) DL model training and optimisation phase. (f) DL model inference phase.

The neural data were recorded from a monkey while performing a point-to-point task, that is, to reach randomly drawn circular targets which were uniformly distributed around an 8-by-8 square grid. A sequence of new random targets was presented immediately and continuously after target acquisition without an inter-trial interval. The recordings were pre-amplified and filtered with a 4th-order 7.5 kHz low-pass filter and then sampled at 24.4 kHz (16-bit resolution).

Dataset II

This dataset was recorded from a monkey (named as monkey N) while performing an instructed delayed reach-to-grasp task. The monkey had to grasp an object with one of grip types (side grip or precision grip) and displace it with either a high or low pulling force. In each trial, the

monkey had to perform one of four possible trial types (from combination of two grip types and two force types) randomly drawn from an equiprobable distribution. Before initiating the movement, the monkey had to wait for 1000 ms (preparatory delay). The trial was successful if the monkey could reach, grasp, pull and hold the object for 500 ms within the position window. The recordings were amplified and filtered with a 1st-order 0.3 Hz high-pass filter and a 3rd-order 7.5 kHz Butterworth low-pass filter. The recordings were then digitised with 16-bit resolution at 30 kHz sampling rate, which hereinafter called as raw neural signals. This dataset contains only one recording session (identified as I140703-001) with a duration of 16.43 minutes. A comprehensive description on the experimental setup and behavioural task along with the corresponding metadata is provided elsewhere [270].

8.2.2 Signal processing and feature extraction

Signal processing steps from the raw neural signals to obtain LFP, ESA, MUA, and SUA along with their associated features are illustrated in Figure 8.1a–d. We briefly describe the signal processing and feature extraction steps as follows.

Local field potential (LFP)

LFP was obtained by low-pass filtering the raw neural signal with a 4th-order Butterworth filter at 100 Hz and then downsampling it to 1 kHz. 100 Hz cut-off frequency was selected to eliminate possible contamination from multiunit spiking activity within higher LFP band ($100 - 300$ Hz). The filtering was performed on the forward and backward directions to avoid any phase shift. We extracted six different LFP features consisting of local motor potential (LMP) and average power spectra from narrow-bands: delta ($0.5 - 4$ Hz), theta ($4 - 8$ Hz), alpha ($8 - 12$ Hz), beta ($12 - 30$ Hz) and gamma ($30 - 100$ Hz). The LFP feature extraction processes have been described in Section 5.2.3.

Entire spiking activity (ESA)

As described in Section 7.2.2, ESA was obtained by first digitally re-referencing the raw neural signals with common average reference (CAR) and high-pass filtering with 1st-order Butter-

worth filter at 300 Hz. The filtered signals were then full-wave rectified, low-pass filtered with 1st-order Butterworth filter at 12 Hz and downsampled to 1 kHz. All the filtering processes were performed in both forward and backward directions. We extracted one ESA feature using time-domain moving average filter with a rectangular window of 256 ms width and 252 ms overlap (similar to that of LMP feature).

Multiunit activity (MUA)

Spike waveforms were extracted by first band-pass filtering the raw neural signals with Butterworth filter (4th-order, from 500 Hz to 5kHz for dataset I; 2nd-order, from 250 Hz to 5 kHz for dataset II) and then storing snippets of the filtered signals (48 or 64 samples for dataset I; 38 samples for dataset II) that crossed a threshold value. These extracted spike waveforms would be used later for classifying spikes into distinct putative single units (i.e. spike sorting). In each channel, MUA was defined as all the detected spikes and represented by their spike times. Details on the spike detection for each dataset has been described in other studies [211], [270]. We extracted MUA features (firing rates) by using BAKS with 256 ms Gaussian window sliding every 4 ms (see Section 4.2.3). Only MUAs with spike rates exceeding 0.5 Hz were included for our experiments and analyses.

Single-unit activity (SUA)

SUA was obtained by aligning the extracted spike waveforms, reducing the dimensionality to a few principal components, and then sorting them into single units via certain algorithms (operator defined templates for dataset I; K-Means and Valley Seeking for dataset II). Details on how spike sorting was performed for each dataset can be found in other studies [211], [270]. We extracted SUA features using the same method as for MUA. In this chapter, we only used one SUA per channel. If a channel had more than one SUAs, one representative of SUAs with highest spike rate was selected. In addition, only SUAs with spike rates above 0.5 Hz were included.

Table 8.1 Hyperparameter setting of LSTM models for inferring different types of spike signals. For inferring ESA signal, hyperparameter values were optimised separately on each of six LFP features. For inferring MUA and SUA signals, hyperparameter values were optimised on the LMP feature of LFP.

Hyperparameter	Value range	Optimised value for different signal types							
		ESA						MUA	SUA
		LMP	Delta	Theta	Alpha	Beta	Gamma		
# of units	$\{5, 10, \cdots, 25\}$	25	15	5	5	10	25	25	25
# of epochs	$\{2, 3, \cdots, 7\}$	2	3	5	4	4	2	2	3
Batch size	$\{32, 64, 96\}$	32	64	96	64	96	96	64	64
Dropout rate	$\{0, 0.1, \cdots, 0.4\}$	0.2	0.4	0.4	0.4	0.2	0.4	0.2	0.3
Learning rate	$\{5, 10, \cdots, 50\} \times 10^{-4}$	0.002	0.001	0.0005	0.0005	0.0005	0.0005	0.001	0.001

8.2.3 Long short-term memory (LSTM)

We used the same LSTM architecture as previously described in Section 4.2.5 but with different hyperparameter configuration. We divided each dataset into 10 non-overlapping contiguous blocks of equal size which were then categorised into three sets: training set (8 concatenated blocks), validation set (1 block) and testing set (1 block). The training set was used for training the model, the validation set was used for optimising the hyperparameters, whereas the testing set was used for the final performance evaluation. The schematic of the LSTM model training, optimisation and evaluation is illustrated in Figure 8.1e,f. We standardised (i.e. z-transformed) the LFP and spike features to have zero mean and unit standard deviation. The model was trained using RMSprop optimiser and mean squared error (MSE) loss function. Hyperopt library [213] was used to optimise the LSTM hyperparameter values from predefined hyperparameter search spaces shown in Table 8.1. We optimised the hyperparameters for inferring ESA, MUA, and SUA signals separately using the same procedure. In the case of ESA, the hyperparameter optimisation was performed on each of six different LFP features. In the cases of MUA and SUA, the hyperparameter optimisation was only performed on the LMP feature of LFP. The optimised hyperparameter values for each case are provided in Table 8.1.

8.2.4 LFP channel importance/contribution

The importance/contribution of each LFP channel on inference performance was measured using a method introduced by Breiman in 2001 [271] known as permutation importance. Specif-

ically, we used a model-agnostic version of the feature importance proposed by Fisher *et al.* which they call model reliance [272]. It measures the importance of a channel (i.e. variable) by computing the difference of model's inference accuracy after temporally permuting the data in that channel. This permutation breaks the temporal relationship between the data in that channel and output variable being inferred, which increases the model's accuracy difference. The larger the increase in the model's accuracy difference after channel permutation, the more important is that channel for the model's inference.

The procedure to calculate the channel importance is as follows: *first*, compute the accuracy of the original trained model (without permutation) based on the testing set and record it as the baseline accuracy; *second*, for each channel, permute the data in that channel, compute and record the accuracy based on the permuted data; *lastly*, compute the channel importance as the difference between the baseline accuracy and the accuracy based on the permuted data. We implemented this procedure using a Python library called MLxtend [273].

8.2.5 Effect of number of LFP channels on inference accuracy

To examine the effect of number of LFP channels on the inference accuracy, we selected randomly n distinct LFP channels where $n = \{1, 5, 10, 15, \cdots, 90, 95\}$ from a total of 96 channels. We then used these n LFP channels to train LSTM model evaluate its performance in inferring SUA, MUA, and ESA. This procedure was repeated for 30 iterations to obtain the average performance along with its confidence interval. The inference accuracy of SUA, MUA, and ESA from the same channel number were compared and analysed.

8.2.6 Performance evaluation and metrics

The performance evaluation of LFP to spiking activity (ESA/MUA/SUA) inference was conducted using the optimised LSTM model on the testing set with Pearson's correlation coefficients (CC) metric. CC measures the linear correlation between the actual and inferred spiking activity and has been used in several prior studies [36], [256], [274]–[276]. We also evaluated the inference performance using root mean square error (RMSE). It is a measure of the average magnitude of the inference error and has also been previously used in other studies [256], [275].

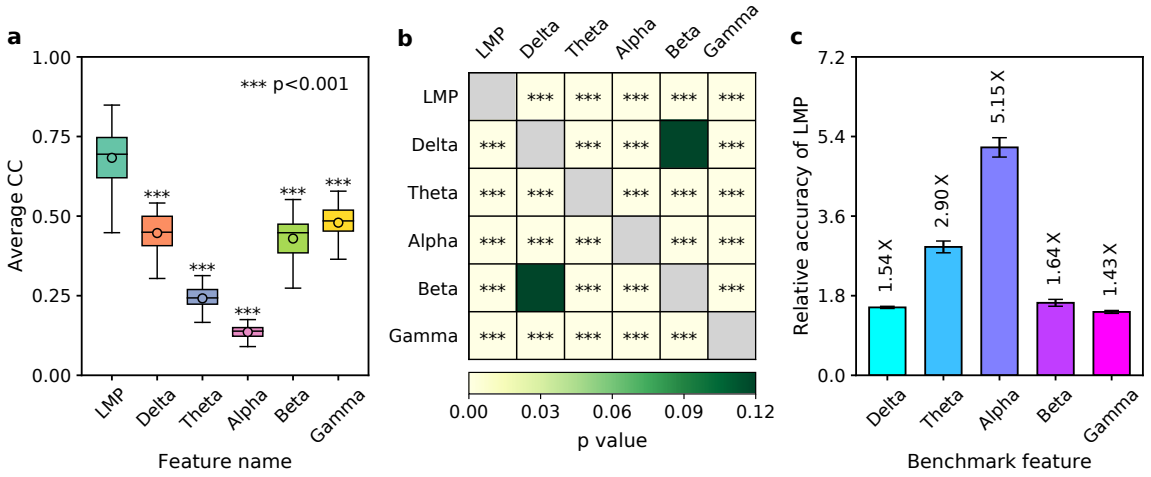


Figure 8.2 Performance comparison of ESA inference across different LFP features (data from recording session i20160627_01). (a) Boxplot comparison of average CC across LFP features. Asterisks indicate which features were statistically significant difference from that of LMP (*** $p<0.001$). (b) Heatmap of statistical significance matrix among LFP features. Grey boxes in the main diagonal did not have p-value (excluded from statistical test calculation). (c) Relative inference accuracy of LMP with respect to other LFP features being benchmarked against. Larger value indicates better relative performance. Black error bars denote 95% confidence intervals.

For each session, the mean and confidence interval of the inference performance were evaluated on 10 different blocks within the testing set. To test for significant effects between a pair of different LFP features or different spiking activity, we used a paired two-tailed t-test whenever the difference between the pairs follows normal distribution, otherwise we used Wilcoxon signed-rank test. The significance level (α) was set to 0.05.

8.3 Results

8.3.1 LFP feature informativeness for ESA inference

We evaluated the informativeness of different LFP features for the inference of ESA using extracellular recordings from a subject while performing a point-to-point task (see Dataset I in Methods). We trained and optimised an LSTM model for each LFP feature separately and then quantified how well this model and LFP feature predict ESA. Figure 8.2a shows boxplots comparing the inference performance of six LFP features measured in average Pearson's correlation coefficient (CC). From 910 cases (10 blocks $\times$ 91 ESA channels), LMP was found to

yield the highest accuracy with CC = 0.68 ± 0.008 (mean ± SEM), while power spectra in the alpha band (shortly referred to as alpha) yielded the lowest accuracy with CC = 0.14 ± 0.002. The order of LFP feature from highest to lowest accuracy was LMP > gamma > delta > beta > theta > alpha. The statistical tests performed between a pair of LFP features from all possible combinations revealed that there were statistically significant differences (*** $p < 0.001$) between LFP features (as shown in Figure 8.2b). Relative to other features, LMP yielded on average 1.43 to 5.15 times better CC as illustrated in Figure 8.2c. Due to the highest predictive power, LMP was thus selected as the feature for subsequent processing and analysis.

8.3.2 Comparison of inference performance: SUA vs MUA vs ESA

Using LMP feature from 96 LFP channels in the first recording session of dataset I (i20160627_01), we then performed inference of three different spiking activities: SUA, MUA, and ESA. In this recording session, from a total of 96 channels, we only inferred 83 and 91 channels for SUA and MUA, respectively; other channels were excluded due to very few number of detected spikes (≤ 0.5 Hz). For the case of MUA, we inferred all the 96 channels but for the performance comparison we only took into account the inference from the same channels as MUA. The inference performance (measured in CC) of each channel of SUA, MUA, and ESA are plotted into heatmap grid spatially corresponding to Utah array configuration as shown in Figure 8.3a-c, respectively. Numbers within the red boxes in Figure 8.3a-b indicates the excluded SUA and MUA channels. From 83 SUA channels, we obtained average CC of 0.43 ± 0.017 (mean ± SEM) with minimum CC of 0.13 and maximum CC of 0.79. From 91 MUA channels, we obtained higher inference accuracy (CC = 0.62 ± 0.014 with $CC_{min} = 0.07$ and $CC_{max} = 0.86$). The highest inference accuracy was obtained from ESA inference with an average CC of c across 91 channels with $CC_{max} = 0.45$ and $CC_{max} = 0.85$. The order of spiking activities from highest to lowest inference accuracy was ESA > MUA > SUA.

To investigate whether this trend was consistently observed across different recording sessions, we performed the inference using long-term neural recordings with a total of 26 sessions spanning more than 7 months between the first and last sessions. Across the entire recording sessions, the number of SUA channels being included for analyses varied from 71 to 89 with

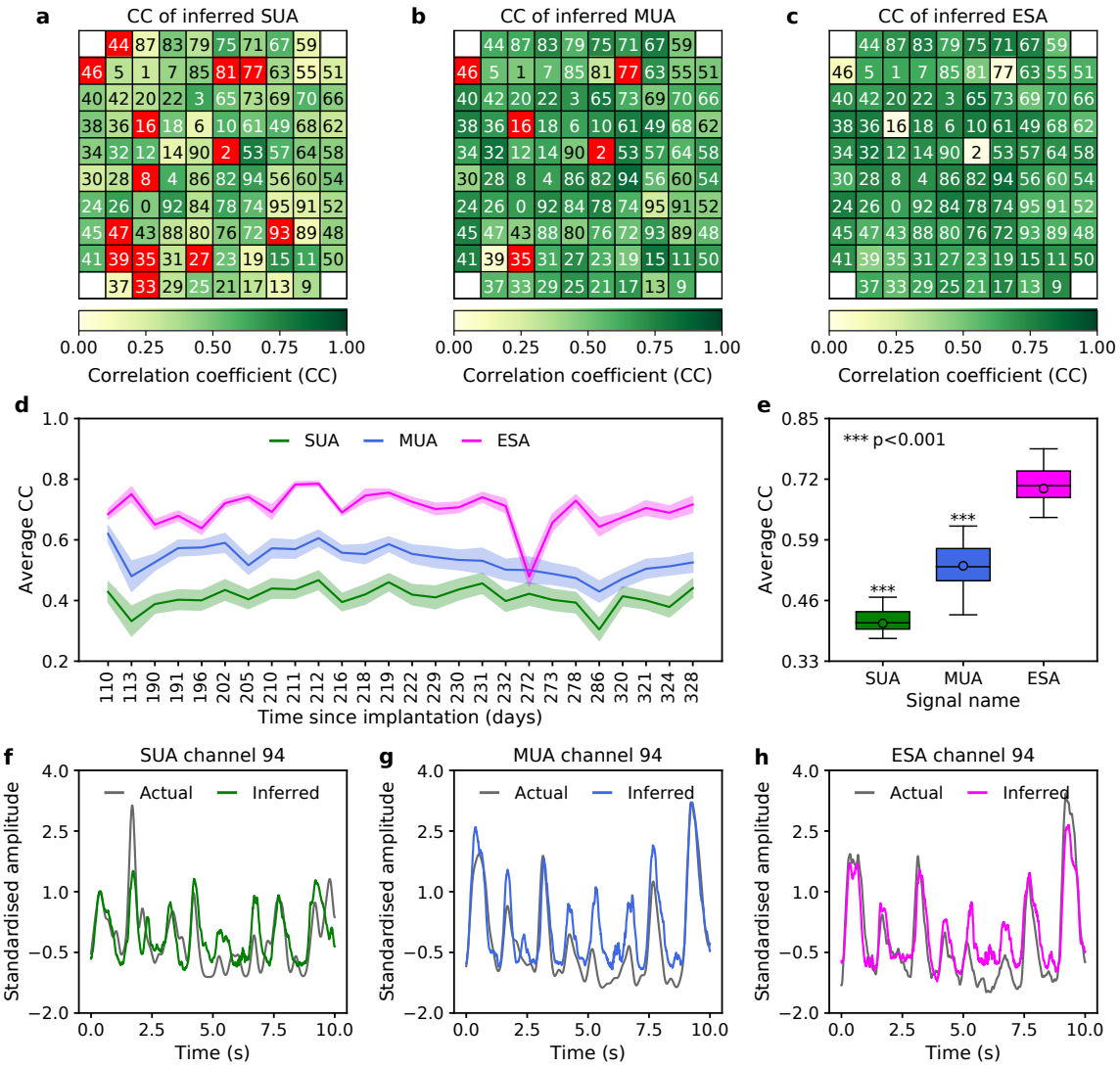


Figure 8.3 Comparison of inference accuracy (CC) among different types of spiking activities (SUA, MUA, and ESA). (a-c) Heatmap of inference accuracy (CC) of 96-channel SUA, MUA, and ESA, respectively mapped onto a 10×10 grid spatially corresponding to Utah electrode array configuration. Black/white numbers inside the grids denote each of the 96-channel numbers. Grey boxes on the grid corners represent unused (unconnected) electrodes. Red boxes indicate channels that were excluded from the inference due to too few number of recorded spikes (≤ 0.5 Hz). Data were taken from recording session i20160627_01. (d) Comparison of inference accuracy (average CC) among SUA, MUA, and ESA over long periods of time lasting up to 328 days after implantation. (e) Boxplot comparison (average CC across all the recording sessions) among SUA, MUA, and ESA. Asterisks indicate the statistical significance of other spiking activities being compared to ESA (*** $p < 0.001$). (f-h) A snippet example of actual and inferred SUA, MUA, and ESA, respectively (taken from channel 94 of recording session i20160627_01).

an average of 80.62 ± 5.36 (mean $\pm$ SD); for both MUA and ESA, the number of included channels were from 77 to 95 channels with an average of 87.08 ± 4.44 (mean $\pm$ SD). As can be seen in Figure 8.3d, the inference accuracy of ESA was consistently better than that of SUA and MUA. Compared to SUA, the inference accuracy of ESA was higher in 26 of 26 sessions (100%), whereas compared to MUA, the inference accuracy of ESA was higher in 25 of 26 sessions (96.15%). The average accuracy (mean $\pm$ SEM) of SUA, MUA, and ESA were 0.41 ± 0.007, 0.53 ± 0.009, and 0.70 ± 0.012, respectively. Figure 8.3e displays boxplot of ESA inference that was statistically significant different than that of SUA and MUA (*** $p < 0.001$). The relative inference accuracy of ESA was 1.71 and 1.32 times higher with respect to SUA and MUA, respectively. These results indicate that LFPs contain substantial information about ESA. Examples of actual and inferred SUA, MUA, and ESA are illustrated in Figure 8.3f-h.

CC is a translation and scale invariant metric that only assesses the shape similarity between actual and inferred signals. In the presence of a large but constant inference error, the CC value can be high (implying falsely good accuracy). Therefore, as an additional metric for the inference performance, we used root mean square error (RMSE) that represents an average magnitude of inference error between actual and inferred signals. We found the same trend of the LFP feature informativeness as of CC. LMP yielded the highest accuracy (i.e. smallest RMSE) with RMSE $= 0.73 \pm 0.007$ (mean $\pm$ SEM), whereas alpha yielded the lowest accuracy (i.e. largest RMSE) with RMSE $= 0.99 \pm 0.002$. The LFP feature informativeness in descending order revealed the same pattern as of CC: LMP > gamma > delta > beta > theta > alpha (see Figure 8.4a). There were statistically significant differences (*** $p < 0.001$) among combinations of LFP feature pairs as illustrated in a heatmap of p-values shown in Figure 8.4b). In comparison to other features, LMP yielded on average 1.20 to 1.37 times better accuracy (Figure 8.4c).

Identical to that of CC metric, ESA inference accuracy (average RMSE 0.74 ± 0.008) was better than that of SUA (0.89 ± 0.008) and MUA (0.77 ± 0.010). Figure 8.4d-f display heatmaps of inference accuracy (RMSE) from each channel of SUA, MUA, and ESA, respectively. Greener grid indicates better accuracy (smaller RMSE) whereas yellower grid indicates worse accuracy (larger RMSE). The order of spiking activities from highest to lowest inference accuracy was

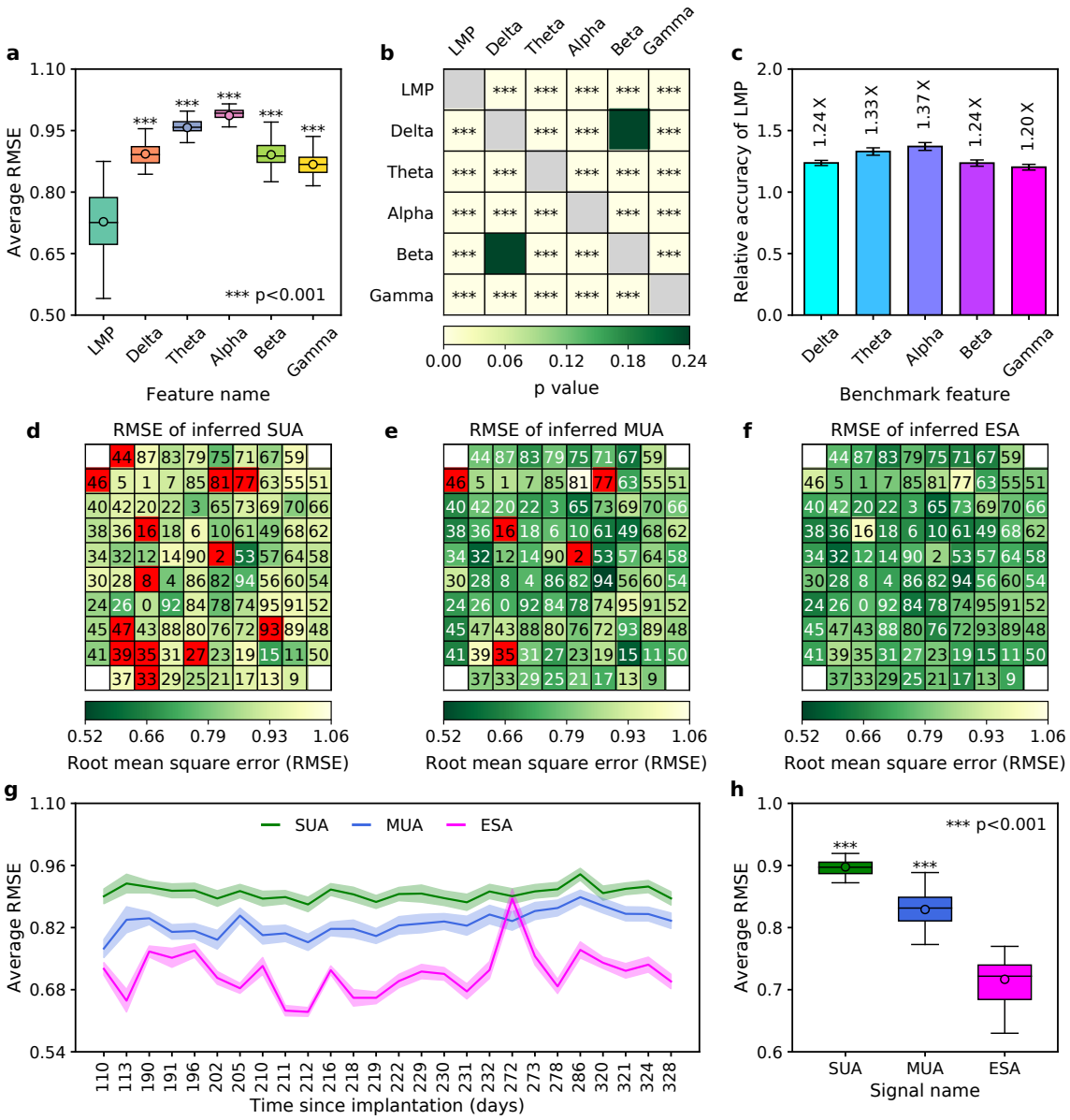


Figure 8.4 Comparison of inference accuracy measured with RMSE metric. (a) Boxplot comparison of ESA inference across LFP features. Asterisks indicate which features were statistically significant difference from that of LMP (*** $p<0.001$). (b) Heatmap of statistical significance matrix among LFP features. Grey boxes in the main diagonal did not have p-value (excluded from statistical test calculation). (c) Relative inference accuracy of LMP with respect to other LFP features being benchmarked against. Larger value indicates better relative performance. Black error bars denote 95% confidence intervals. (d-f) Heatmap of inference accuracy (RMSE) of 96-channel SUA, MUA, and ESA, respectively mapped onto a 10×10 grid spatially corresponding to Utah electrode array configuration. Black/white numbers inside the grids denote each of the 96-channel numbers. White boxes on the grid corners represent unused (unconnected) electrodes. Red boxes indicate channels that were excluded from the inference due to too few number of recorded spikes (≤ 0.5 Hz). Data were taken from recording session i20160627_01. (g) Comparison of inference accuracy (average RMSE) among SUA, MUA, and ESA over long periods of time lasting up to 328 days after implantation. (h) Boxplot comparison (average RMSE across all the recording sessions) among SUA, MUA, and ESA. Asterisks indicate the statistical significance of other spiking activities being compared to ESA (*** $p<0.001$).

ESA > MUA > SUA. As evident from Figure 8.4g, the same trend was also observed across long-term recording sessions where the inference accuracy of ESA consistently outperformed its counterparts. The average accuracy (mean ± SEM) of SUA, MUA, and ESA were 0.90 ± 0.003, 0.83 ± 0.005, and 0.72 ± 0.010, respectively. Figure 8.4h displays boxplot of ESA inference showing its statistically significant difference from that of SUA and MUA (*** $p < 0.001$). The inference accuracy of ESA was 1.26 and 1.16 times better relative to SUA and MUA, respectively.

8.3.3 LFP channel importance/contribution

Next, we assessed the importance or contribution of LFP channels for the inference of spiking activity at particular channels. The LFP channel importance was measured and ranked using permutation feature importance as described in Methods section. Figure 8.5a-c show an example of LFP channel importance scores mapped into 10-by-10 heatmap grids for the inference of channel 94 of SUA, MUA, and ESA, respectively. The channel of spiking activity being inferred (channel 94) is displayed in bold yellow text. The LFP channel with highest importance score for the inference of SUA, MUA, and ESA was channel 37, 24, and 94, respectively. From the analysis across different inferred channel numbers, we can draw two insights. First, the LFP from the same channel as the inferred spiking activity on average had high importance score (for example, see Figure 8.5a-b) than other channels but did not necessarily have the highest importance score (Figure 8.5a). This indicates that LFP from the same channel contains rich information about spiking activity. Second, for the same channel number, the LFP importance score for ESA inference was higher than that for SUA and MUA. This may indicate that ESA contains additional information that is not available in SUA and MUA.

8.3.4 Effect of number of LFP channels on inference accuracy

Figure 8.5d-e illustrates the effect of using different number of LFP channels on the inference accuracy. We found that increasing the number of LFP channels to a certain number dramatically improved the inference accuracy. However, after that point, the inference accuracy only improved marginally or even decreased (as evident in the case of SUA from Figure 8.5d-e).

Analysis across different inferred spiking activity channels revealed that the inference accuracy of SUA reached plateau in significantly less number of LFP channels than that of MUA and ESA. Compared to SUA and MUA, ESA needed higher number of LFP channels to reach the maximum inference accuracy (shown by the red star markers in Figure 8.5d-e). This may indicate that ESA has larger spatial coverage and more information than SUA and MUA.

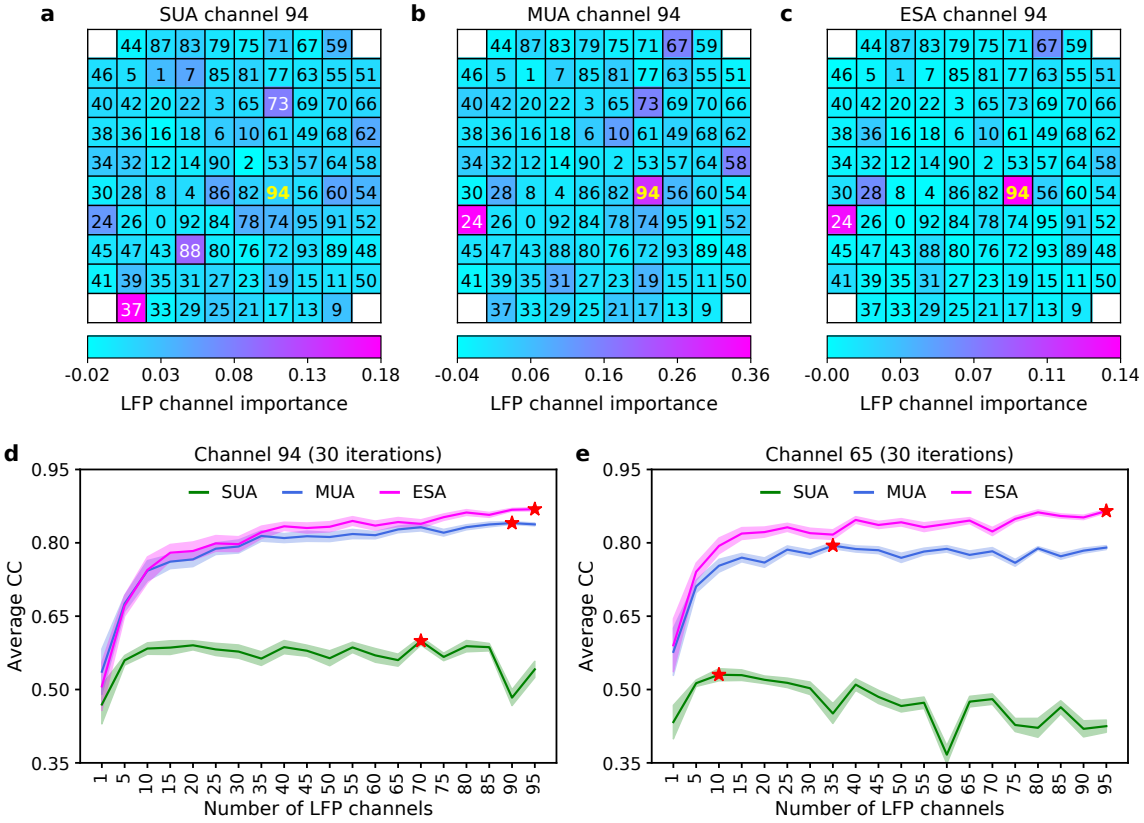


Figure 8.5 The importance of each LFP channel and the effect of the number of LFP channels to inference accuracy. Data were taken from recording session i20160627_01. (a-c) Heatmap of LFP channel importance score for the inference of SUA, MUA, and ESA in channel 94, respectively mapped onto a 10×10 grid spatially corresponding to Utah electrode array configuration. Black/white numbers inside the grids denote each of the 96 LFP channel numbers. Bold yellow number (i.e. 94) indicates the channel of spiking activities being inferred. White boxes on the grid corners represent unused (unconnected) electrodes. The larger the channel importance score, the more important is the channel for the inference. (d-e) The effect of increasing number of LFP channels for the inference accuracy (average CC) of spiking activities in channel 94 and 64, respectively. The shaded areas represent 95% confidence intervals ($N = 30$). The red star markers indicate the number of LFP channels that corresponds to highest average CC score for each spiking activity.

8.3.5 Performance valuation on different subject and task

Lastly, we next sought to determine whether the trend in above results was associated to any particular subject or task. We performed the same inference procedure using a different subject performing different task (see Dataset II in Methods). Analysis on LFP feature informativeness revealed the same trend where LMP yielded the highest inference accuracy, whereas alpha yielded the lowest inference accuracy. However, there was minor difference in the order of LFP feature informativeness in which delta feature showed better predictive performance than gamma. The order of LFP feature informativeness (from highest to lowest) was as follows: LMP $>$ delta $>$ gamma $>$ beta $>$ theta $>$ alpha (Figure 8.6a). The statistical tests among pairs of LFP features also showed that there were statistically significant difference among the features (Figure 8.6b). The relative performance of LMP varied from 1.28 to 10.53 times better than that of other features (Figure 8.6c). Using LMP as the feature, we plotted the inference accuracy of each channel of SUA, MUA, and ESA into 10-by-10 heatmap grids (see Figure 8.6d-f). The inference accuracy of ESA (average CC 0.71 ± 0.013) was also found to be statistically significant higher than SUA (0.55 ± 0.020) and MUA (0.52 ± 0.021) as illustrated in Figure 8.6g. Relative to SUA and MUA, ESA yielded on average 1.28 and 1.35 times better inference accuracy. Similar to the results obtained with dataset I, the inference accuracy of SUA reached plateau in less number of LFP channels than MUA and ESA (Figure 8.6h). In good agreement with the results from dataset I, the LFP from the same channel had high but not necessarily highest importance score for the inference of spiking activity (Figure 8.6i-k). Moreover, for the same channel, the importance score of ESA was on average also higher than that of SUA and MUA.

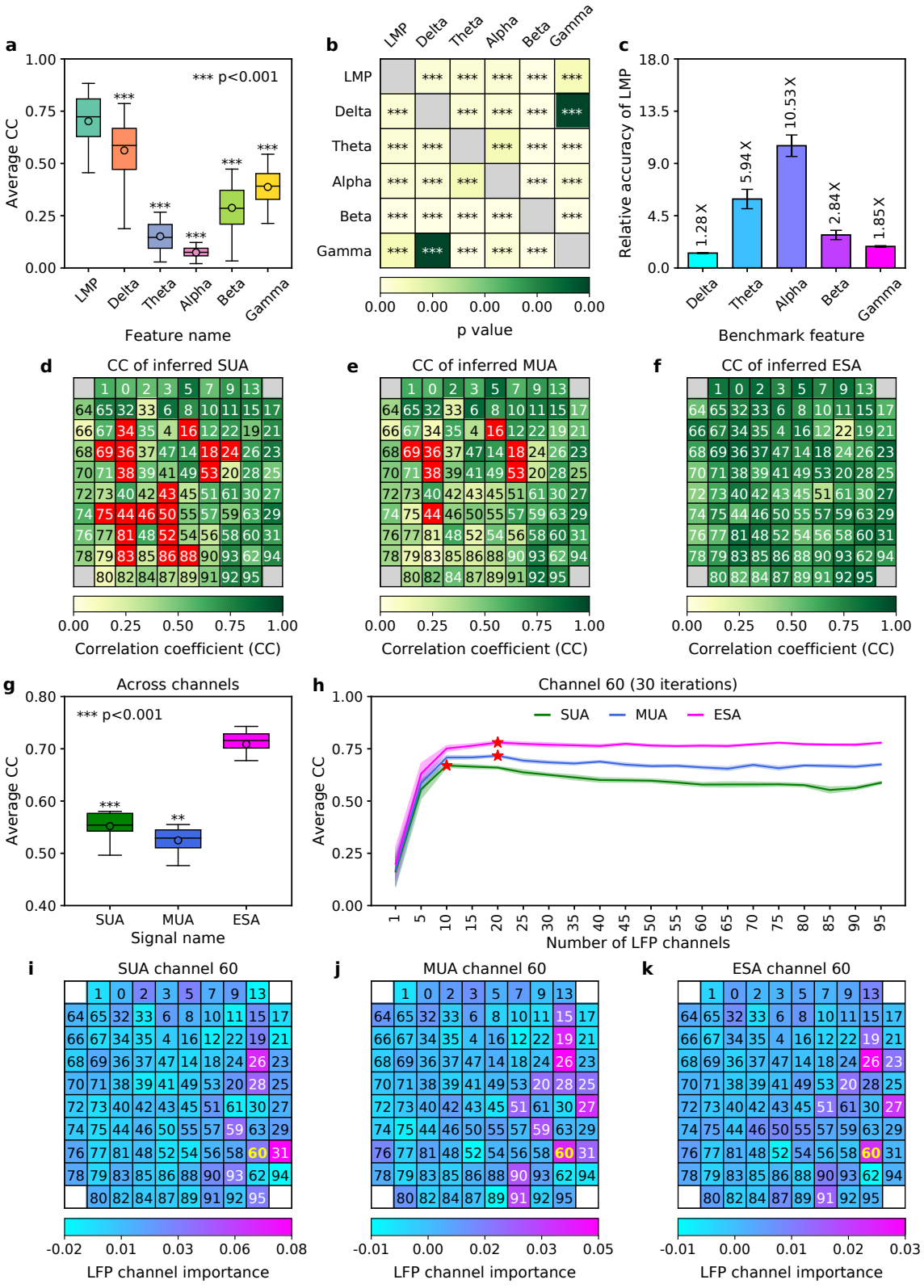

a
Average CC
*** p<0.001
Feature name
LMP
Delta
Theta
Alpha
Beta
Gamma
b
p value
c
Relative accuracy of LMP
1.28 X
5.94 X
10.53 X
2.84 X
1.85 X
Benchmark feature
d
CC of inferred SUA
e
CC of inferred MUA
f
CC of inferred ESA
Correlation coefficient (CC)
g
Across channels
Signal name
SUA
MUA
ESA
h
Channel 60 (30 iterations)
Number of LFP channels
i
SUA channel 60
j
MUA channel 60
k
ESA channel 60
LFP channel importance

Figure 8.6 Performance evaluation on different subject and task (recording session i140703-001) (a) Boxplot comparison of ESA inference (average CC) across LFP features. Asterisks indicate which features were statistically significant difference from that of LMP (*** $p<0.001$). (b) Heatmap of statistical significance matrix among LFP features. Grey boxes in the main diagonal did not have p-value (excluded from statistical test calculation). (c) Relative inference accuracy of LMP with respect to other LFP features being benchmarked against. Larger value indicates better relative performance. Black error bars denote 95% confidence intervals. (d-f) Heatmap of inference accuracy (CC) of 96-channel SUA, MUA, and ESA, respectively mapped onto a 10×10 grid spatially corresponding to Utah electrode array configuration. Black/white numbers inside the grids denote each of the 96-channel numbers. Grey boxes on the grid corners represent unused (unconnected) electrodes. Red boxes indicate channels that were excluded from the inference due to too few number of recorded spikes (≤ 0.5 Hz). (g) Boxplot comparison (average CC across channels) among SUA, MUA, and ESA. Asterisks indicate the statistical significance of other spiking activities being compared to ESA (*** $p<0.001$). (h) The effect of increasing number of LFP channels for the inference accuracy (average CC) of spiking activities in channel 60. The shaded areas represent 95% confidence intervals ($N = 30$). The red star markers indicate the number of LFP channels that corresponds to highest average CC score for each spiking activity. (i-k) Heatmap of LFP channel importance score for the inference of SUA, MUA, and ESA in channel 60 mapped onto a 10×10 grid spatially corresponding to Utah electrode array configuration. Black/white numbers inside the grids denote each of the 96 LFP channel numbers. Bold yellow number (i.e. 60) indicates the channel of spiking activities being inferred. White boxes on the grid corners represent unused (unconnected) electrodes.

8.4 Discussion

The present chapter investigates the relationship between local field potential (LFP) and entire spiking activity (ESA) by asking whether we can infer ESA solely from LFPs. In doing so, first, we examine which feature within LFPs contains the highest information (i.e. most predictive) about ESA. The results reveal that local motor potential (LMP), the smoothed time-domain amplitude of LFP, emerges as the most predictive feature, followed by power feature either from delta or gamma frequency bands. On the other hand, the intermediate frequency bands of LFPs carry only little information about ESA. Considerable number of previous studies have reported that power modulation within low frequency (< 10 Hz) and high-frequency (> 30 Hz) of LFPs are most informative, whereas the intermediate frequency bands are little informative about spiking activity (SUA or MUA) [232], [235], [266], [267], [275]. The low-frequency band LFPs are associated with motor evoked potentials that usually exhibits high correlation with spiking of neuronal population, whereas the high-frequency band LFPs are thought to reflect

synchronised spiking activity arising from local neuronal population [232]. This chapter confirms and extends the previous findings by showing that apart from power within low- and high-frequency LFPs, LMP holds substantial information about spiking activity. Even though LMP has been frequently used as feature for decoding behavioural tasks [22], [37], [52], the use of LMP for inferring spiking activity has not been investigated. A rather similar feature to LMP is the amplitude of low-frequency LFP obtained from low-pass filtering the broadband LFPs and has been used in prior studies [36], [256]. However, this feature differs from LMP in that its frequency band is well separated as low-pass filter results in better roll-off and stopband attenuation. On the other hand, LMP which is obtained via moving average filter contains less random white noise while maintains well the smoothing of LFP signals [277].

Given highly informativeness of LMP, one may ask what the biophysical origin of LMP is. Unfortunately, the answer to this question remains to be established. It has been speculated that LMP is related to firing rate modulation of neuronal population near the recording electrode [233] or distant neuronal population activity that is connected to the recording site [234], [235]. It is also possible that LMP reflects evoked LFPs in the motor cortex in response to movement-related tasks [232], [233]. Prior study suggested that LMP plays important role not only in the execution but also in the preparation of movement (e.g. anticipation) [278]. This may explain the better relative inference accuracy of LMP with respect to other features from dataset II (movement with preparatory delay) than from dataset I (movement without delay interval).

Next, using the LMP feature and LSTM model with optimised hyperparameters, we evaluate the inference accuracy of ESA and compare its performance with that of SUA and MUA. Results across recording sessions, subjects, and tasks show that the inference accuracy of ESA is consistently higher than that of SUA and MUA. Previous studies have only conducted the inference of SUA and MUA from LFPs [36], [232], [256], [266], [267]. Therefore, to the best of our knowledge, this is the first study that performs the inference of ESA from LFPs and systematically compares its performance to that of SUA and MUA. We argue that the high inference accuracy of ESA is attributed to richer spiking information contained within ESA compared to SUA and MUA. Both SUA and MUA are obtained by using threshold-based technique which is

prone to false-positive or false-negative spike detection. If we set the threshold value too high, we could miss true but lower amplitude spikes; on the contrary, if we set the threshold too low, we may detect some background noises as spikes. In contrast to SUA and MUA, ESA does not use thresholding, instead it employs full-rectification and low pass filter which is rather insensitive to random high-frequency noise and is able to preserve full information of spiking activity [251]. In addition, thresholding favours large (pyramidal) neurons with high amplitude spikes and other neurons very close to the electrode tips [249], [251]. The small neurons (with low amplitude spikes) or slightly farther neurons may not be detected. In contrast to that, ESA integrates the contribution from all neurons in the vicinity of recording electrode tip.

To gain more insights into the relationship between LFP and spiking activity, we measure and rank the LFP channel importance/contribution for the inference of SUA, MUA, and ESA at particular channels using feature importance permutation. The results reveal that the LFP importance score in the same channel as the inferred spiking activity on average has high (not necessarily highest) importance score, indicating its rich information about the spiking activity. This is in good agreement with the previous findings [235], [267]. Furthermore, within the same channel, the LFP importance score for ESA is on average higher than SUA and MUA. This may imply that ESA contains additional spiking information that are not detected by thresholding procedure for the cases of SUA and MUA. Lastly, we investigate the effect of different number of LFP channels on the inference accuracy of spiking activity. Our results show that SUA and MUA achieve plateau and peak accuracy in less number of channels compared to ESA. To attain the peak inference accuracy, larger number of LFP channels are needed for the inference of ESA. This could be an indication of larger spatial coverage of ESA (encompassing smaller and farther neurons) compared to SUA and MUA. These results may suggest that LFP represents the total input giving rise to spiking output within larger spatial area (observed through ESA) than previously thought (SUA or MUA).

In this chapter, we infer spiking activity from LFPs recorded from the same electrode. This can confound our analysis since spike waveforms from nearby neurons may contaminate the LFPs. To remove this possible contamination, we use filter with lower frequency cut-off (100 Hz) than the typical frequency cut-off (300 Hz) to obtain LFPs from the raw neural data. Although

slow modulation of spiking activity as low as 10 Hz can contribute to the LFPs [232], its contribution was found to be negligible compared to the other sources such as synaptic activity and membrane potential oscillations [262], [279].

In summary, we have shown that LFPs, in the form of LMP, carry substantial information about spiking activity, particularly ESA. Since spiking activity has been widely used as an input signal for BMI, our finding corroborates the increasingly accumulating evidences that LFPs can be used as an alternative input signal. This is especially relevant for chronic recordings where the spike signals have been found to be unstable or even degrading over long periods of time. In this case, LFP-based BMIs can be implemented using biomimetic or biofeedback approach [36], [163], [218], [280]. Another finding from this chapter is that ESA appears to contain richer information about and larger spatial coverage of spiking activity within an area compared to SUA and MUA. This suggests that ESA could be potentially used in neuroscience research as an alternative neuronal population activity measure for analysing neural responses to various stimuli or behavioural tasks.

8.5 Conclusion

This chapter examines the relationship between LFPs and ESA by investigating whether ESA can be inferred from LFPs alone. We have shown that ESA could be inferred from LFPs with higher accuracy than those of SUA and MUA. We found that LMP carried the highest information about spiking activity (ESA, SUA, and MUA) compared to other LFP features. Overall results suggest the potential use of ESA as an alternative neuronal population activity measure for analysing neural responses to stimuli or behavioural tasks (e.g. BMIs). The results also indicate that LFPs contain substantial information about the spikes, particularly ESA, which could be useful for the development of LFP-based BMIs.

9 | Conclusions and Future Directions

Brain-machine interfaces (BMIs) hold great promise for restoring lost motor functions in patients with neurological disorders. While compelling proof-of-concept BMI systems have been demonstrated in labs and pilot clinical trials, a clinically viable translation of BMIs is hampered by several challenges, such as decoding robustness and accuracy. Within this context, this thesis aims to address both challenges from neural signal processing and decoding points of view. The following sections summarise our original contributions and discuss future directions for further research.

9.1 Original contributions

The work presented in this thesis constitutes a systematic development of neural signal processing and decoding methods towards the ultimate aim of achieving a robust and accurate BMI system. This thesis is composed of three main parts, which corresponds to our consecutive efforts to achieve the ultimate aim. These three parts, along with our original contributions, are briefly described as follows.

Part I - MUA-based BMI. We began from Chapter 3 by developing a new firing rate estimation method based on a kernel smoothing technique with adaptive bandwidth, referred to as Bayesian adaptive kernel smoother (BAKS). We then generated synthetic spike trains based on two biologically plausible models: inhomogeneous Gamma (IG) and inhomogeneous inverse Gaussian (IIG). Using these synthetic data and real data, we showed that BAKS outperformed other methods through an extensive comparison. In Chapter 4, we proposed BAKS as a feature extraction method and long short-term memory (LSTM) as a decoding algorithm for MUA-based BMI. The proposed methods yielded better decoding performance (accuracy

and robustness) than all possible combinations of other two firing rate estimation methods (binning and Gaussian kernel smoother) and two linear decoders (Kalman filter and Wiener filter). Taken together, the first part of this thesis has demonstrated an improved decoding performance of MUA-based BMI.

Part II - LFP-based BMI. Driven by the desirable properties of LFP and inspired by the good decoding performance of LSTM (as described in Part I), we proposed LSTM as a decoding algorithm for LFP-based BMI (Chapter 5). This resulted in decoding performance improvement compared to a commonly used Kalman filter. To improve the decoding performance even further while without requiring a tedious manual feature extraction process as in LSTM decoder, we proposed an end-to-end approach based on temporal convolutional network (TCN) for automatic feature extraction and decoding methods. TCN decoder achieved an improved performance at the expense of increased computational complexity and power consumption. In Chapter 6, we investigated the impact of different referencing schemes on decoding performance using two deep learning algorithms (LSTM and GRU). We also examined which LFP feature contained the highest movement-related information. Results showed that common average reference (CAR) and local motor potential (LMP) yielded better decoding performance than other combinations of referencing schemes and LFP features. Overall results from Part II have successfully demonstrated our attempts at pushing the performance of LFP-based BMIs forward.

Part III - ESA-based BMI. In the last part, we explored the feasibility and viability of using new alternative input signal and decoding algorithm to achieve a robust and accurate BMI system. We proposed entire spiking activity (ESA) as an input signal and quasi-recurrent neural network (QRNN) as a decoding algorithm (Chapter 7). We compared the proposed ESA-driven QRNN decoder against all combinations of other three input signals (SUA, MUA, and LFP) and seven decoding algorithms (WF, WCF, KF, UKF, SRNN, LSTM, and GRU). Extensive results revealed that our proposed method significantly outperformed all these combinations. We also investigated the impact of different training schemes and the amount of training data on decoding performance of ESA-driven QRNN decoder. In addition, we examined the long-term characteristics of ESA and compared them against those of SUA, MUA, and LFP. Furthermore,

to understand the relationship between ESA and LFP, we conducted experiments to infer ESA from LFPs as described in Chapter 8. We compared the inference accuracy of ESA to that of SUA and MUA. We analysed the contribution of different LFP features and individual LFP channel to inference accuracy. Lastly, we assessed the impact of different numbers of LFP channels on inference accuracy. The whole results from Part III suggest the potential use of ESA as input signal and deep learning as a decoding algorithm, particularly QRNN, towards accurate and robust BMIs.

9.2 Future directions

This thesis has focused on the development of neural signal processing and decoding methods for offline decoding task. Offline decoding is useful since it offers a quick and low cost for exploring a new decoding algorithm. It allows the quantification of decoding performance across different decoders. However, to assess the real performance of a decoder, an online (closed-loop) decoding where the decoder is within the loop must be ultimately performed. Therefore, one possible direction for future work is to investigate the efficacy of ESA signal and deep learning algorithm on closed-loop experiments.

We used a total of 26 neural recording sessions spanning from 3.6 to 10.9 months since the implantation date. Despite a gradual declining trend, the number of spikes detected in the last session was still relatively high. Another possible future direction is to examine whether the performance ESA-based deep learning decoder could sustain on long-term neural data where spikes have been severely diminished or even lost. Although our experiments based on spike removal from the raw data (i.e. despiked ESA; Chapter 7), it remains unclear whether such results hold on naturally-lost-spike data.

We developed and implemented neural signal processing and decoding methods in software running on a desktop PC. Despite offering simple and fast development workflow, a PC-based BMI system does not provide desirable properties for a clinically viable BMI, such as compact, portable, and low power. Therefore, future work should be directed towards the development and implementation of the proposed methods on low power, embedded platforms based on ASICs, FPGAs, embedded GPUs, or RISC-V [281]. One possible approach is to implement

ESA computation on on-chip processing (implantable) and deep learning algorithm on off-chip processing (wearable) platforms.

9.3 Concluding remarks

When we started this research project in 2016, the initial objective was to develop neural signal processing and decoding methods for LFP-based BMIs. Signal stability and low bandwidth of LFPs —properties which spike signals are lacking— have been attracting a lot of interest among researchers (including us) to extract informative features within LFP and utilise them for BMIs. Before delving deeper into the development of LFP signal processing and decoding methods, we needed a good baseline spike-based BMI for performance benchmark. Therefore, firstly, we developed a new feature extraction method (BAKS; Chapter 3) and decoding algorithm (LSTM; Chapter 4) to push forward the decoding performance of spike-based BMI. We then investigated the most informative LFP feature and developed deep learning based decoders (Chapters 5 and 5). It turned out that the decoding accuracy of LFPs could not outperform those of spikes (SUA or MUA) under the same decoders. This motivated us to explore a new alternative input signal and decoding algorithm to achieve better decoding robustness and accuracy than those of spikes and LFPs. In the end, we proposed ESA as the input signal and QRNN as the decoding algorithm (Chapter 7 and 8). Extensive comparison results revealed that our proposed ESA-driven QRNN decoder significantly outperformed other possible combinations between other input signals and decoding algorithms. Taken together, we believe that the consecutive efforts and systematic works presented in this thesis have demonstrated solid progress towards robust and accurate BMIs. While significant performance improvements have been shown in this thesis, several interesting research questions and challenges lie ahead (Section 9.2). It is important to note that a successful translation from BMI research into a clinically viable device is not only related to scientific and technological aspects. It requires a good ecosystem and collaborative network to bring together scientists, clinicians, funders, investors, industries, policymakers, regulators, and users, among others [282]. This thesis contributes mainly to addressing two scientific and technological challenges, accuracy and robustness, by providing insights into signal processing and decoding design consideration towards clinically viable translation of BMIs.

Appendices

A | Mathematical Derivation of BAKS

A.1 Posterior distribution of bandwidth

In this appendix, we derive a closed-form expression of the posterior density of bandwidth as given in Eq (3.9). According to Bayes' theorem, the posterior density is formulated as:

$$\pi(h(t)|t) = \frac{\hat{f}(t|h(t))\pi(h(t))}{\int \hat{f}(t|h(t))\pi(h(t))dh(t)} \tag{A.1}$$

By using likelihood function as in Eq (3.7) and prior distribution of bandwidth as in Eq (3.6), we can obtain:

$$\pi(h(t)|t) = \frac{\frac{1}{n}\sum_{i=1}^{n} K_{h(t)}(t-t_i)\frac{2h(t)^{-2\alpha-1}}{\Gamma(\alpha)\beta^{\alpha}}\exp\left\{-\frac{1}{\beta h(t)^2}\right\}}{\int \frac{1}{n}\sum_{i=1}^{n} K_{h(t)}(t-t_i)\frac{2h(t)^{-2\alpha-1}}{\Gamma(\alpha)\beta^{\alpha}}\exp\left\{-\frac{1}{\beta h(t)^2}\right\}dh(t)} \tag{A.2}$$

By substituting Gaussian kernel into the likelihood function and removing the same constants in both numerator and denominator, Eq (A.2) then becomes:

$$\pi(h(t)|t) = \frac{\sum_{i=1}^{n} h(t)^{2\alpha-2}\exp\left\{-\frac{1}{h(t)^2}\left[\frac{(t-t_i)^2}{2}+\frac{1}{\beta}\right]\right\}}{\int \sum_{i=1}^{n} h(t)^{2\alpha-2}\exp\left\{-\frac{1}{h(t)^2}\left[\frac{(t-t_i)^2}{2}+\frac{1}{\beta}\right]\right\}dh(t)} \tag{A.3}$$

Let now consider the denominator of Eqs (A.1) and (A.3), which we can rewrite as:

$$\int \hat{f}(t|h(t))\pi(h(t))dh(t) = \int \sum_{i=1}^{n} h(t)^{2\alpha-2}\exp\left\{-\frac{1}{h(t)^2}\left[\frac{(t-t_i)^2}{2}+\frac{1}{\beta}\right]\right\}dh(t) \tag{A.4}$$

To simplify the calculation, let us define variables as follows:

$$\sigma = \frac{1}{h(t)^2}, \quad h(t) = \sigma^{-\frac{1}{2}}, \quad dh(t) = -\frac{1}{2}\sigma^{-\frac{3}{2}}d\sigma \tag{A.5}$$

$$\theta = \left[\frac{(t-t_i)^2}{2} + \frac{1}{\beta}\right]^{-1} \tag{A.6}$$

By substituting Eqs (A.5) and (A.6) into Eq (A.4), the integral function can be represented as:

$$\int \hat{f}(t|h(t))\pi(h(t))dh(t) = \frac{1}{2}\sum_{i=1}^{n}\int \sigma^{\alpha-\frac{1}{2}}\exp\left\{-\frac{\sigma}{\theta}\right\}d\sigma \tag{A.7}$$

Eq (A.7) can be simplified so that the integral part forms Gamma probability density as follows:

$$\begin{aligned}\int \hat{f}(t|h(t))\pi(h(t))dh(t) &= \frac{1}{2}\sum_{i=1}^{n}\Gamma(\alpha+\frac{1}{2})\theta^{(\alpha+\frac{1}{2})} \\ &\times \int \frac{\sigma^{(\alpha+\frac{1}{2})-1}}{\Gamma(\alpha+\frac{1}{2})\theta^{(\alpha+\frac{1}{2})}}\exp\left\{-\frac{\sigma}{\theta}\right\}d\sigma\end{aligned} \tag{A.8}$$

Since the integration of Gamma probability density function is equal to 1,

$$\int \frac{\sigma^{(\alpha+\frac{1}{2})-1}}{\Gamma(\alpha+\frac{1}{2})\theta^{(\alpha+\frac{1}{2})}}\exp\left\{-\frac{\sigma}{\theta}\right\}d\sigma = 1 \tag{A.9}$$

Eq A.8 can then be analytically expressed as:

$$\begin{aligned}\int \hat{f}(t|h(t))\pi(h(t))dh(t) &= \frac{1}{2}\sum_{i=1}^{n}\Gamma(\alpha+\frac{1}{2})\theta^{(\alpha+\frac{1}{2})} \\ &= \frac{1}{2}\Gamma(\alpha+\frac{1}{2})\sum_{i=1}^{n}\left[\frac{(t-t_i)^2}{2}+\frac{1}{\beta}\right]^{(-\alpha-\frac{1}{2})}\end{aligned} \tag{A.10}$$

Finally, by substituting Eq (A.10) back to the original equation of posterior density of bandwidth in Eq (A.1), we can obtain the closed-form solution as in Eq (3.9):

$$\pi(h(t)|t) = \frac{\sum_{i=1}^{n} h(t)^{-2\alpha-2}\exp\left\{-\frac{1}{h(t)^2}\left[\frac{(t-t_i)^2}{2}+\frac{1}{\beta}\right]\right\}}{\frac{1}{2}\Gamma(\alpha+\frac{1}{2})\sum_{i=1}^{n}\left[\frac{(t-t_i)^2}{2}+\frac{1}{\beta}\right]^{(-\alpha-\frac{1}{2})}} \tag{A.11}$$

A.2 Adaptive bandwidth estimate

Under squared error loss function, the adaptive bandwidth can be estimated by using the posterior mean as given by:

$$\hat{h}(t) = \int h(t)\pi(h(t)|t)dh(t) \tag{A.12}$$

By substituting Eq (A.11) into Eq (A.12), we can obtain:

$$\hat{h}(t) = \frac{\int \sum_{i=1}^{n} h(t)^{-2\alpha-1} \exp\left\{-\frac{1}{h^2}\left[\frac{(t-t_i)^2}{2} + \frac{1}{\beta}\right]\right\} dh(t)}{\frac{1}{2}\Gamma(\alpha+\frac{1}{2}) \sum_{i=1}^{n} \left[\frac{(t-t_i)^2}{2} + \frac{1}{\beta}\right]^{(-\alpha-\frac{1}{2})}} \tag{A.13}$$

Similar to the derivation procedure for the posterior distribution of bandwidth (A.1), by the change-of-variables rule using Eqs (A.5) and (A.6), Eq (A.13) can be written as:

$$\hat{h}(t) = \frac{\frac{1}{2}\sum_{i=1}^{n} \int \sigma^{\alpha-1} \exp\left\{-\frac{\sigma}{\theta}\right\} d\sigma}{\frac{1}{2}\Gamma(\alpha+\frac{1}{2}) \sum_{i=1}^{n} \left[\frac{(t-t_i)^2}{2} + \frac{1}{\beta}\right]^{(-\alpha-\frac{1}{2})}} \tag{A.14}$$

By modifying the integral part of numerator to be an integration of Gamma probability density function (which is equal to 1) as in Eq (A.8), we can obtain the final closed-form solution as in Eq (3.11):

$$\hat{h}(t) = \frac{\Gamma(\alpha) \sum_{i=1}^{n} \left[\frac{(t-t_i)^2}{2} + \frac{1}{\beta}\right]^{-\alpha}}{\Gamma(\alpha+\frac{1}{2}) \sum_{i=1}^{n} \left[\frac{(t-t_i)^2}{2} + \frac{1}{\beta}\right]^{-\alpha-\frac{1}{2}}} \tag{A.15}$$

B | Supplementary Figures and Tables

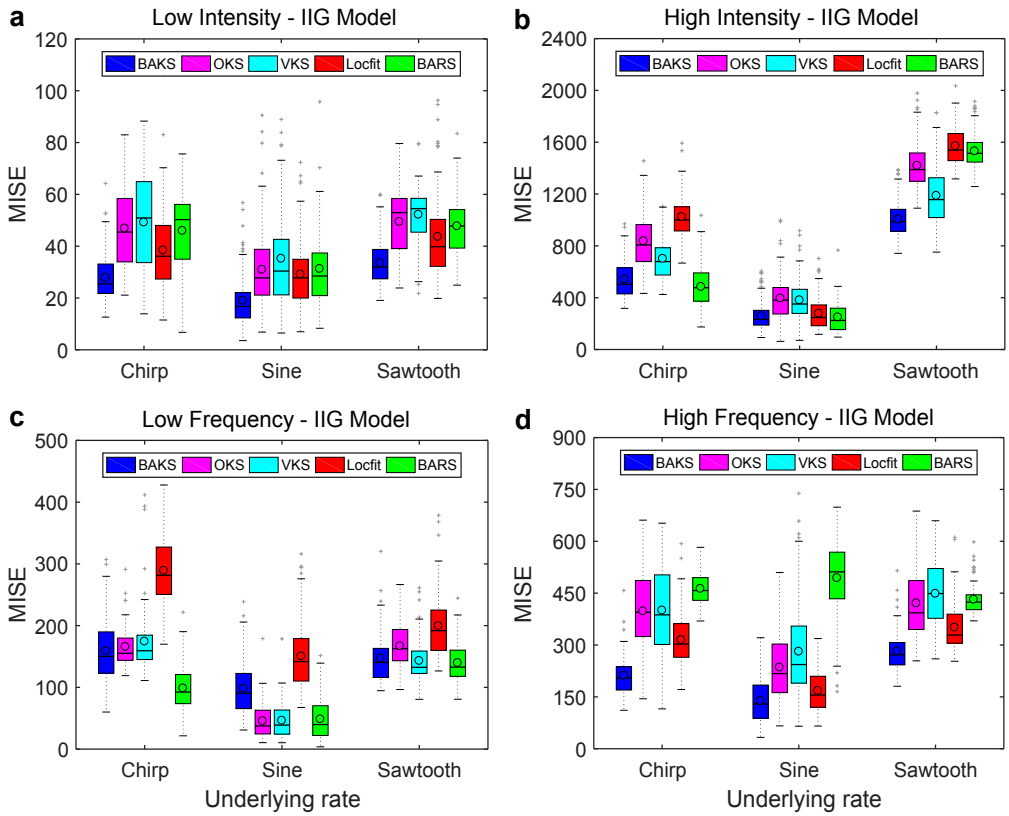


Figure B.1 MISE comparison under different values of intensity and frequency for IIG model. (a) MISE comparison for the case of low intensity. (b) MISE comparison for the case of high intensity. (c) MISE comparison for the case of low frequency. (d) MISE comparison for the case of high frequency. In each boxplot, black lines show the medians; black circles indicate the means; colored solid boxes represent interquartile ranges; whiskers extend 1.5× from upper and lower quartiles; grey crosses represent outliers.

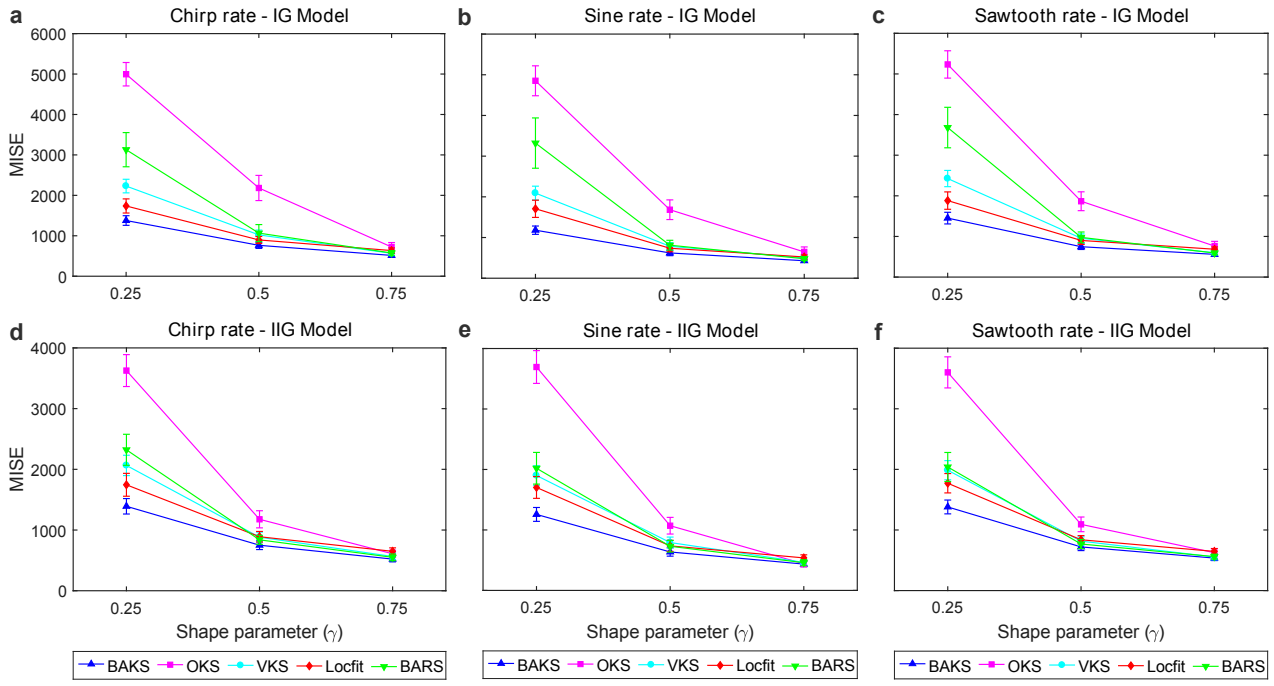


Figure B.2 MISE comparison under different values of shape parameter ($\gamma = \{0.25, 0.5, 0.75\}$). (a)-(c) MISE comparison for chirp, sine, and sawtooth rate functions, respectively, from IG model. (d)-(f) MISE comparison for chirp, sine, and sawtooth rate functions, respectively, from IIG model. Vertical bars represent the 95% confidence intervals.

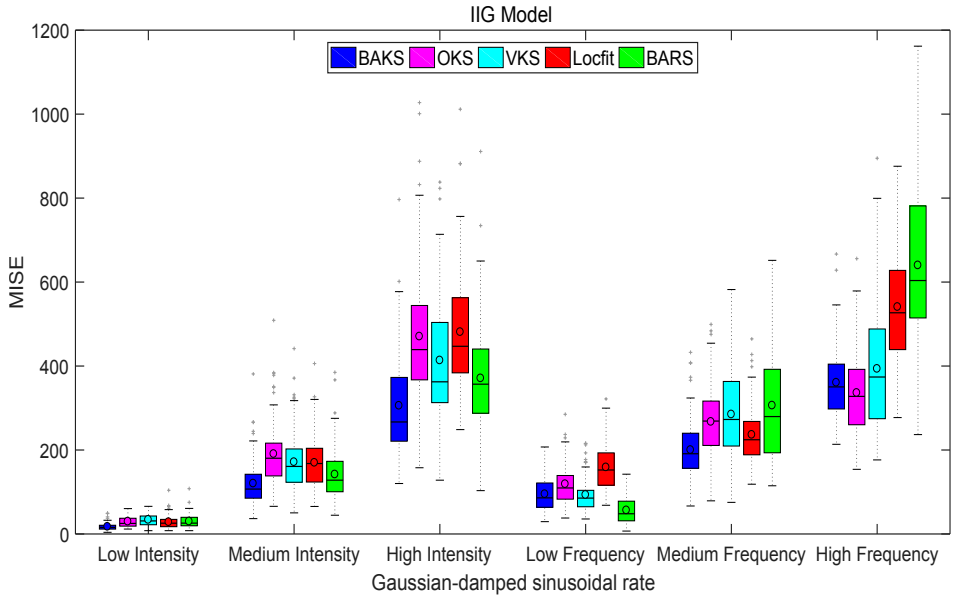


Figure B.3 MISE comparison for IIG model with Gaussian-damped sinusoidal rate function. In each boxplot, black lines show the medians; black circles indicate the means; colored solid boxes represent interquartile ranges; whiskers extend 1.5× from upper and lower quartiles; grey crosses represent outliers.

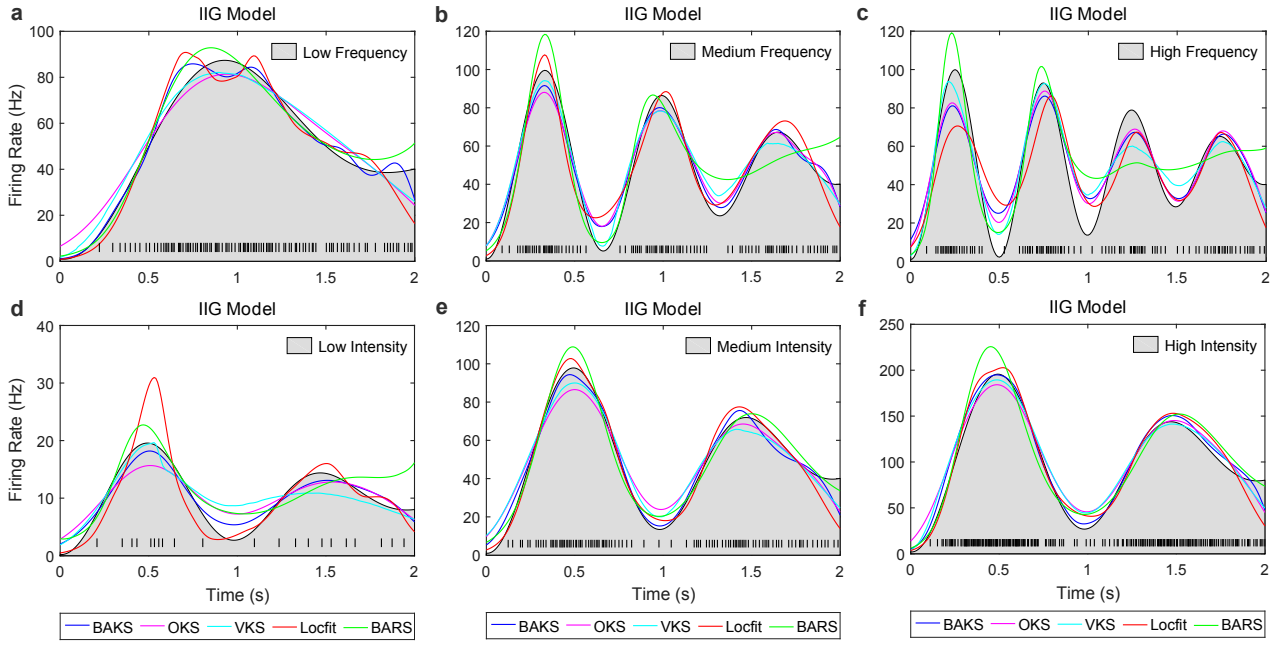


Figure B.4 Firing rate estimate comparison for IIG model with Gaussian-damped sinusoidal rate function. (a)-(c) Firing rate estimates for the cases of low, medium and high frequency, respectively. (d)-(f) Firing rate estimates for the cases of low, medium and high intensity, respectively. Black lines with grey-shaded regions indicate the underlying rate functions. Black raster in the bottom of each plot represents a spike train generated from the associated underlying rate function.

Table B.1 Hyperparameter search spaces during the optimisation of deep learning decoders.

Hyperparameter	SRNN	GRU	LSTM	QRNN
# of units	$\{100, 150, \cdots, 450\}$	$\{50, 75, \cdots, 225\}$	$\{50, 75, \cdots, 200\}$	$\{50, 100, \cdots, 400\}$
# of epochs	$\{2, 3, \cdots, 8\}$			
Batch size	$\{32, 64, 96\}$			
Dropout rate	$\{0, 0.1, \cdots, 0.5\}$			
Learning rate	$\{5, 10, \cdots, 50\} \times 10^{-4}$			

Table B.2 Hyperparameter configurations for SUA-, MUA-, and LFP-driven DL decoders.

Hyperparameter	SUA Signal				MUA Signal				LFP Signal			
	SRNN	GRU	LSTM	QRNN	SRNN	GRU	LSTM	QRNN	SRNN	GRU	LSTM	QRNN
# of units	400	200	200	400	450	200	150	400	450	175	125	350
# of epochs	8	3	6	4	5	4	6	4	7	3	5	2
Batch size	64	96	64	96	64	64	64	64	64	32	64	64
Dropout rate	0.1	0.4	0	0.4	0.1	0.3	0	0.4	0.1	0.3	0.2	0.1
Learning rate	0.0025	0.0045	0.002	0.003	0.0035	0.003	0.0035	0.005	0.0002	0.0035	0.004	0.0005

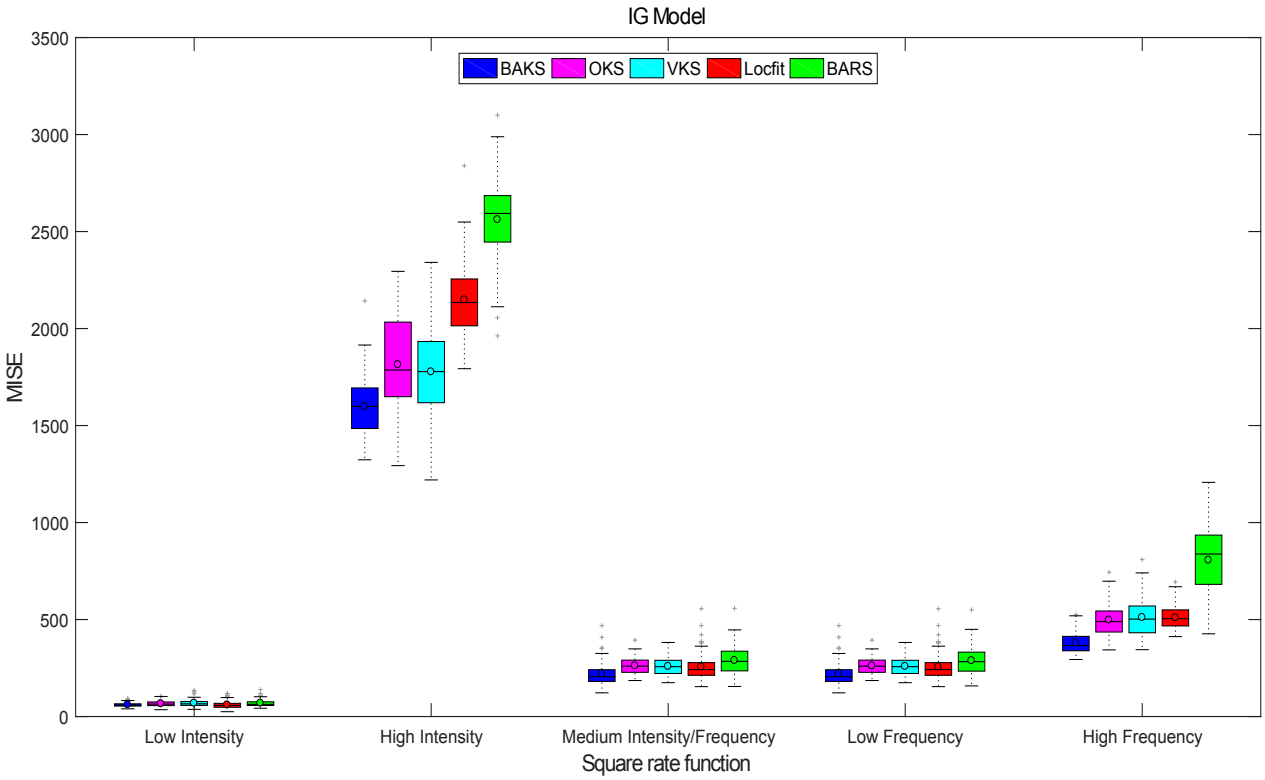


Figure B.5 MISE comparison for IG model with square rate function. In each boxplot, black lines show the medians; black circles indicate the means; colored solid boxes represent interquartile ranges; whiskers extend 1.5× from upper and lower quartiles; grey crosses represent outliers.

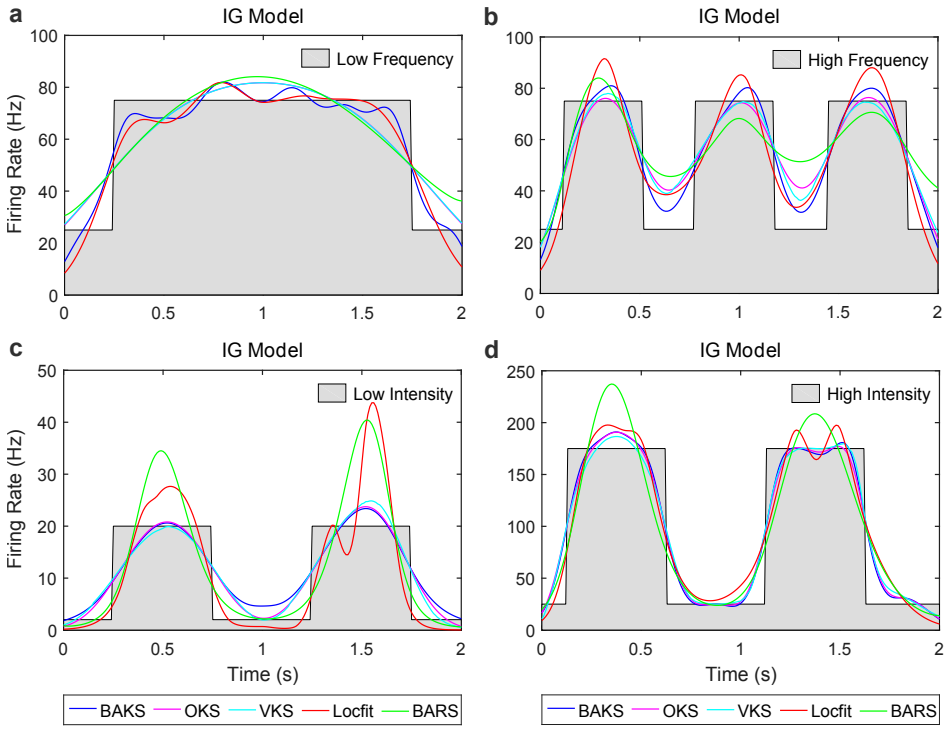


Figure B.6 Firing rate estimate comparison for IG model with square rate function. (a)-(b) Firing rate estimates for the cases of low and high frequency, respectively. (c)-(d) Firing rate estimates for the cases of low and high intensity, respectively. Black lines with grey-shaded regions indicate the underlying rate functions.

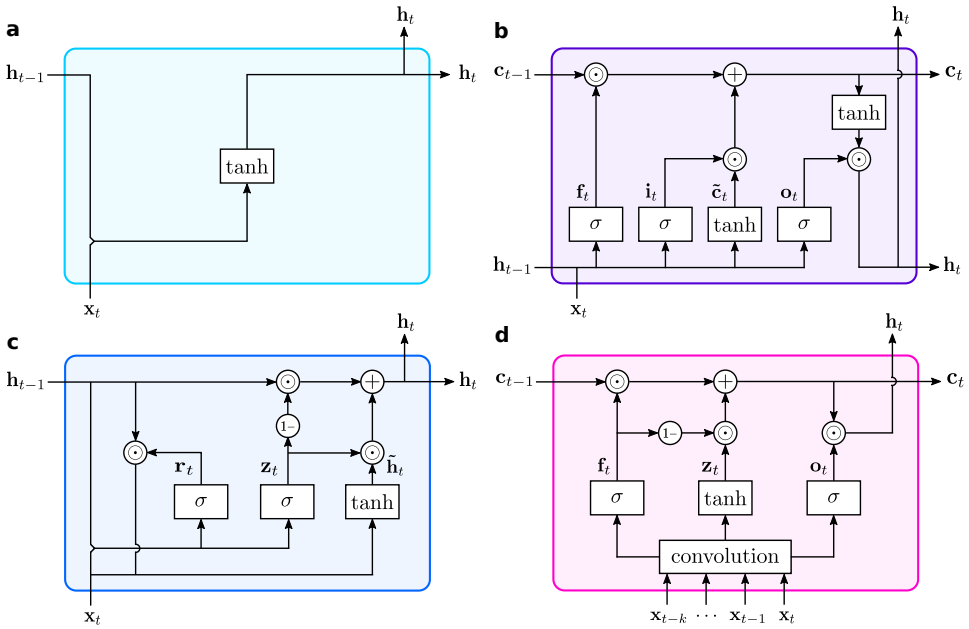


Figure B.7 Graphical illustration of the deep learning decoders. (a) Standard recurrent neural network (SRNN). (b) Long short-term memory (LSTM). (c) Gated recurrent unit (GRU). (d) Quasi-recurrent neural network (QRNN).

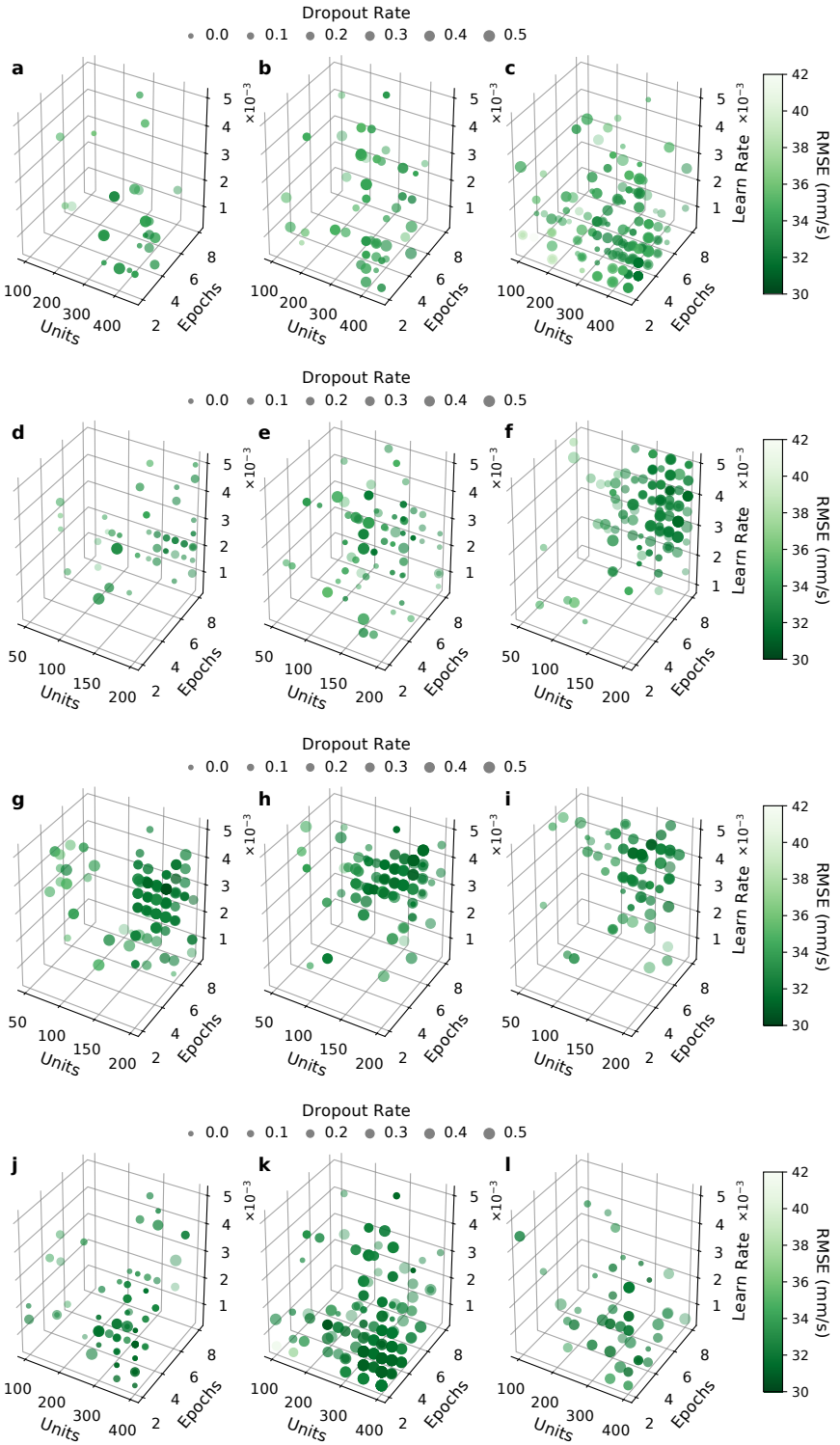


Figure B.8 Visualisation of hyperparameter optimisation result for ESA-driven deep learning decoders. (a)-(c) 3D scatter plots correspond to RNN decoder with the batch size of 32, 64, and 96, respectively. (d)-(f) 3D scatter plots correspond to GRU decoder with the batch size of 32, 64, and 96, respectively. (g)-(i) 3D scatter plots correspond to LSTM decoder with the batch size of 32, 64, and 96, respectively. (j)-(l) 3D scatter plots correspond to QRNN decoder with the batch size of 32, 64, and 96, respectively. The x-axes represent the number of units; y-axes represent the number of epochs; z-axes represent the learning rate. The size of circles denotes the dropout rate while the color of circles denotes the loss value (RMSE).

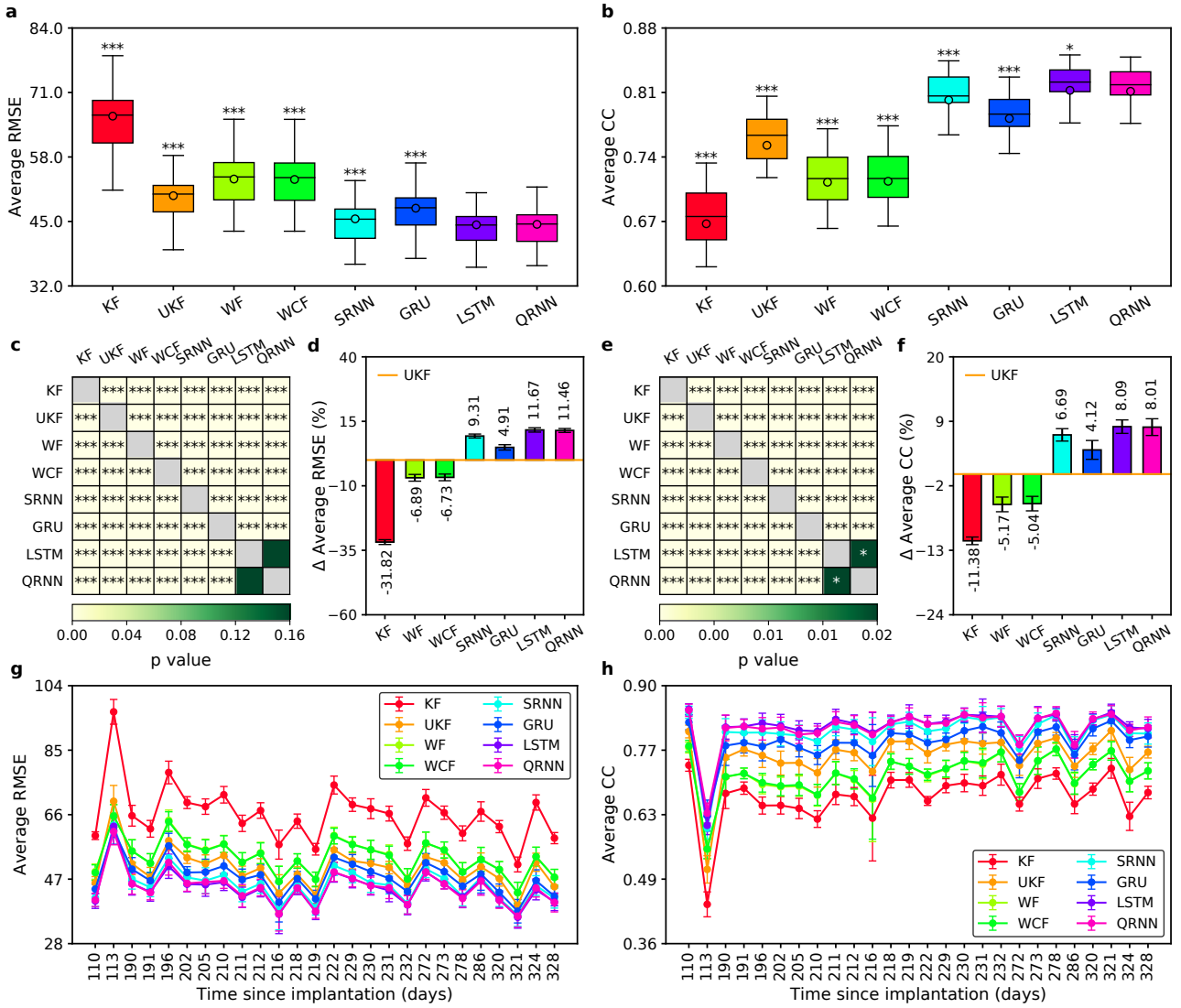


Figure B.9 Long-term decoding performance comparison across different decoders driven by SUA signal. (a),(b) Boxplot comparison of long-term decoding performance across decoders as measured by average RMSE and CC, respectively. The horizontal lines inside the boxes denote the medians whereas the circles denote the means. Asterisks represent decoders whose decoding performance differed significantly from that of QRNN (two-tailed paired Wilcoxon signed-rank test; * $p < 0.05$, *** $p < 0.001$). (c),(e) Heatmap of statistical significant matrix of decoding performance between all possible pairs of decoders as measured by average RMSE and CC, respectively. Each cell is coloured based on the p-value except cells in the main diagonal (grey; excluded from statistical test calculation). Asterisks indicate whether there were statistical significant differences between the pairs (two-tailed paired Wilcoxon signed-rank test; * $p < 0.05$, *** $p < 0.001$). (d),(f) Average performance improvement/degradation (in %) of decoders with respect to that of UKF as measured by RMSE and CC metrics, respectively. Positive (negative) value indicates performance improvement (degradation). Black numeral texts represent the means whereas black error bars represent the 95% confidence intervals ($n = 26$). (g),(h) Decoding performance comparison over time across decoders in terms of average RMSE and CC, respectively. Coloured circles represent the means whereas coloured error bars represent the 95% confidence intervals ($n = 10$) for each session. The x-axes represent the number of days since electrode implantation. The x-axes' ticks correspond to different sessions having irregular gaps between the consecutive sessions.

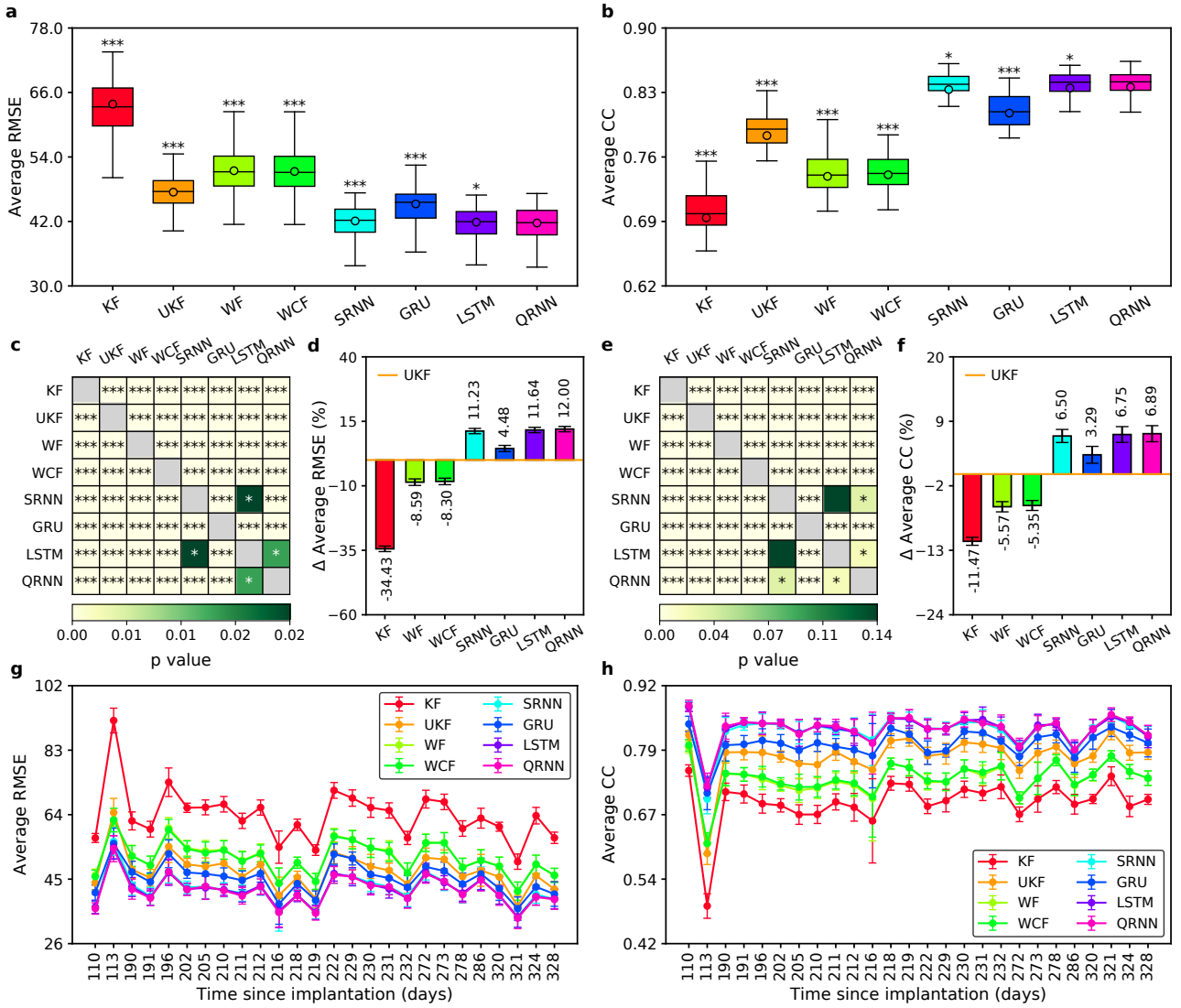


Figure B.10 Long-term decoding performance comparison across different decoders driven by MUA signal. (a),(b) Boxplot comparison of long-term decoding performance across decoders as measured by average RMSE and CC, respectively. The horizontal lines inside the boxes denote the medians whereas the circles denote the means. Asterisks represent decoders whose decoding performance differed significantly from that of QRNN (two-tailed paired Wilcoxon signed-rank test; * $p < 0.05$, *** $p < 0.001$). (c),(e) Heatmap of statistical significant matrix of decoding performance between all possible pairs of decoders as measured by average RMSE and CC, respectively. Each cell is coloured based on the p-value except cells in the main diagonal (grey; excluded from statistical test calculation). Asterisks indicate whether there were statistical significant differences between the pairs (two-tailed paired Wilcoxon signed-rank test; * $p < 0.05$, *** $p < 0.001$). (d),(f) Average performance improvement/degradation (in %) of decoders with respect to that of UKF as measured by RMSE and CC metrics, respectively. Positive (negative) value indicates performance improvement (degradation). Black numeral texts represent the means whereas black error bars represent the 95% confidence intervals ($n = 26$). (g),(h) Decoding performance comparison over time across decoders in terms of average RMSE and CC, respectively. Coloured circles represent the means whereas coloured error bars represent the 95% confidence intervals ($n = 10$) for each session. The x-axes represent the number of days since electrode implantation. The x-axes' ticks correspond to different sessions having irregular gaps between the consecutive sessions.

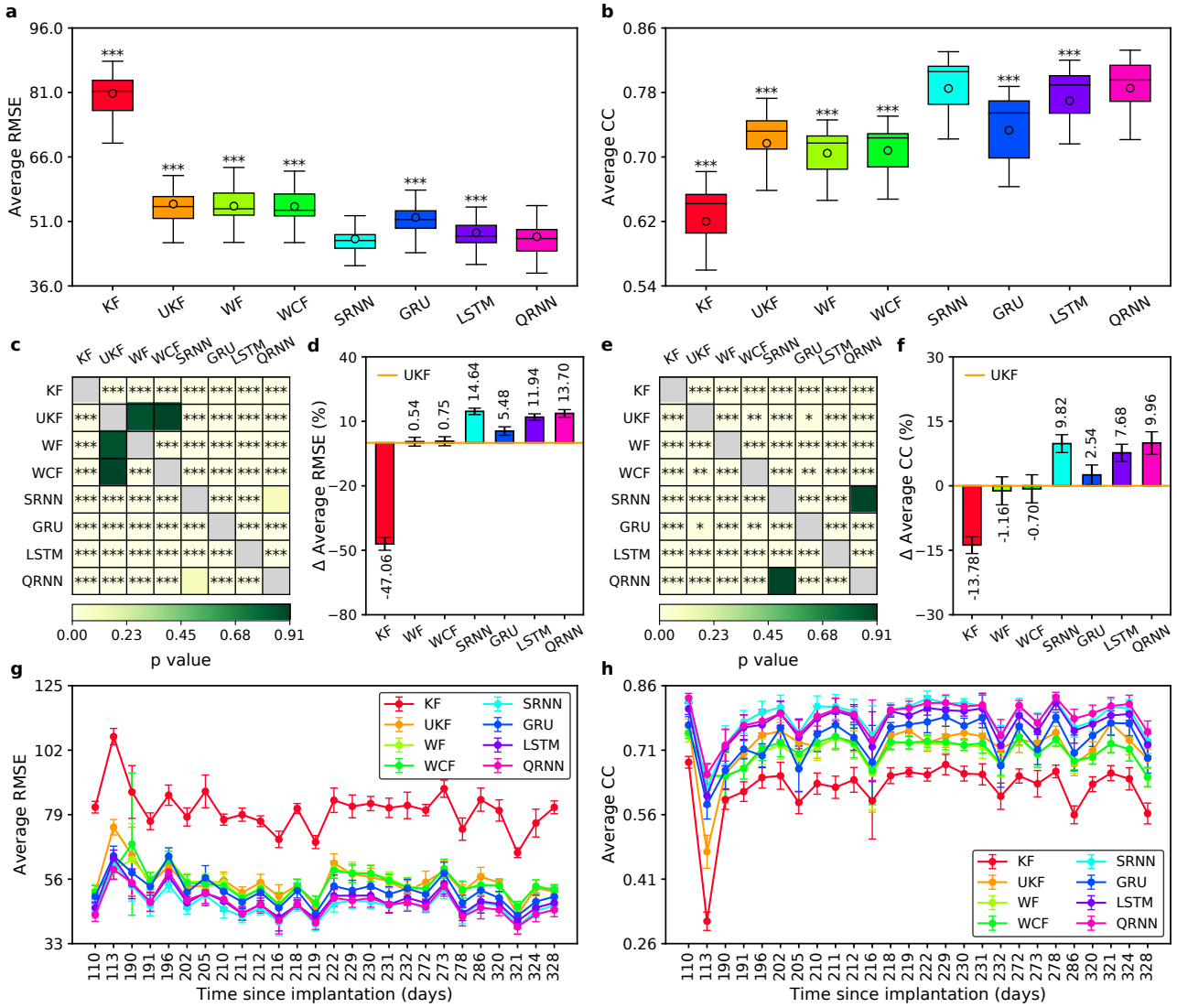


Figure B.11 Long-term decoding performance comparison across different decoders driven by LFP signal. (a),(b) Boxplot comparison of long-term decoding performance across decoders as measured by average RMSE and CC, respectively. The horizontal lines inside the boxes denote the medians whereas the circles denote the means. Asterisks represent decoders whose decoding performance differed significantly from that of QRNN (two-tailed paired Wilcoxon signed-rank test; *** $p < 0.001$). (c),(e) Heatmap of statistical significant matrix of decoding performance between all possible pairs of decoders as measured by average RMSE and CC, respectively. Each cell is coloured based on the p-value except cells in the main diagonal (grey; excluded from statistical test calculation). Asterisks indicate whether there were statistical significant differences between the pairs (two-tailed paired Wilcoxon signed-rank test; * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$). (d),(f) Average performance improvement/degradation (in %) of decoders with respect to that of UKF as measured by RMSE and CC metrics, respectively. Positive (negative) value indicates performance improvement (degradation). Black numeral texts represent the means whereas black error bars represent the 95% confidence intervals ($n = 26$). (g),(h) Decoding performance comparison over time across decoders in terms of average RMSE and CC, respectively. Coloured circles represent the means whereas coloured error bars represent the 95% confidence intervals ($n = 10$) for each session. The x-axes represent the number of days since electrode implantation. The x-axes' ticks correspond to different sessions having irregular gaps between the consecutive sessions.

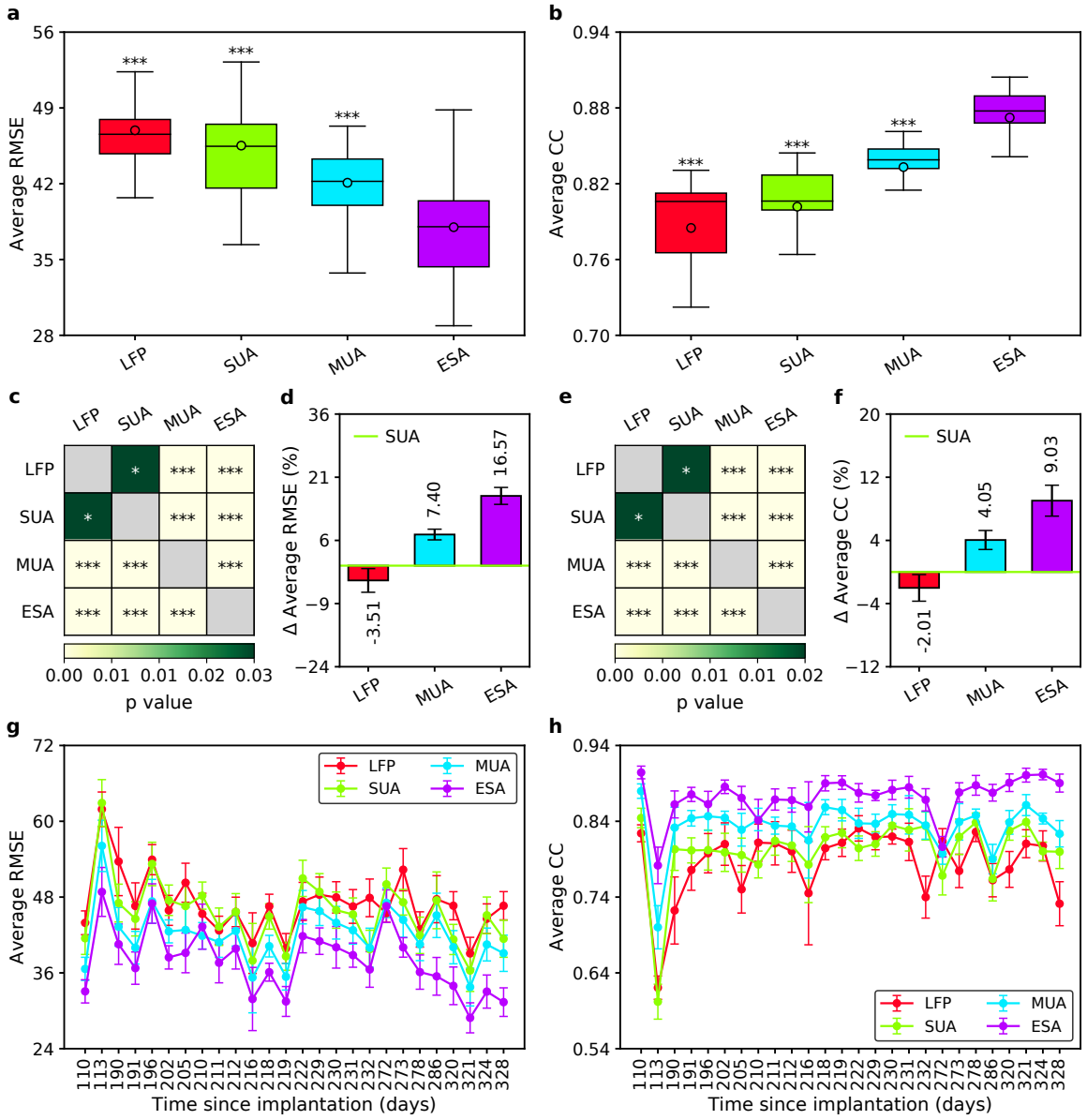


Figure B.12 Long-term decoding performance comparison across SRNN decoders driven by different neural signals. (a),(b) Boxplot comparison of long-term decoding performance across different neural signals as measured by average RMSE and CC, respectively. The horizontal lines inside the boxes denote the medians whereas the circles denote the means. Asterisks represent neural signals whose decoding performance differed significantly from that of ESA (two-tailed paired Wilcoxon signed-rank test; *** $p < 0.001$). (c),(e) Heatmap of statistical significant matrix of decoding performance between all possible pairs of neural signals as measured by average RMSE and CC, respectively. Each cell is coloured based on the p-value except cells in the main diagonal (grey; excluded from statistical test calculation). Asterisks indicate whether there were statistical significant differences between the pairs (two-tailed paired Wilcoxon signed-rank test; * $p < 0.05$, *** $p < 0.001$). (d),(f) Average performance improvement/degradation (in %) of neural signals with respect to that of SUA as measured by RMSE and CC metrics, respectively. Positive (negative) value indicates performance improvement (degradation). Black numeral texts represent the means whereas black error bars represent the 95% confidence intervals ($n = 26$). (g),(h) Decoding performance comparison over time across neural signals in terms of average RMSE and CC, respectively. Coloured circles represent the means whereas coloured error bars represent the 95% confidence intervals ($n = 10$) for each session. The x-axes represent the number of days since electrode implantation. The x-axes' ticks correspond to different sessions having irregular gaps between the consecutive sessions.

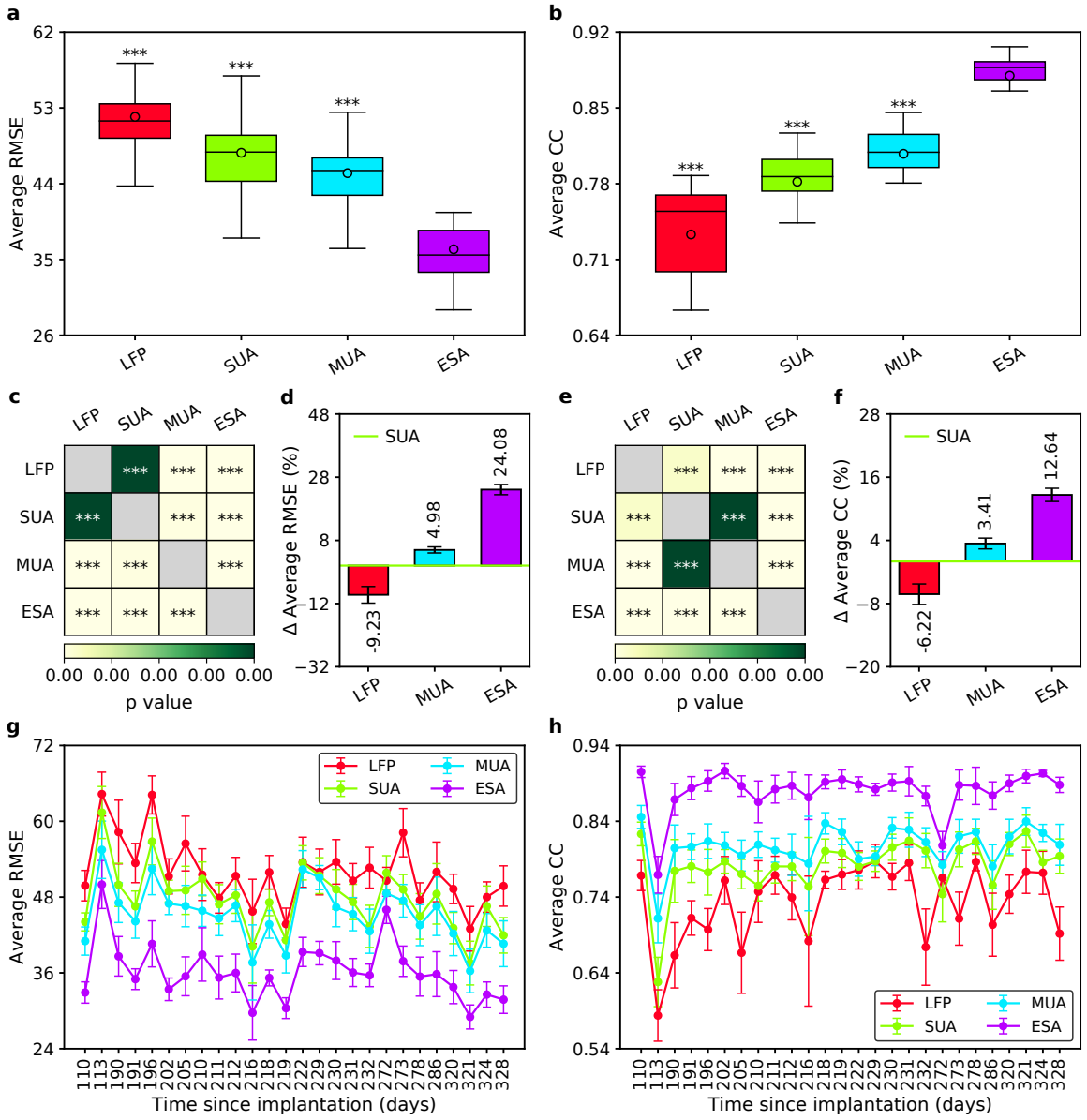


Figure B.13 Long-term decoding performance comparison across GRU decoders driven by different neural signals. (a),(b) Boxplot comparison of long-term decoding performance across different neural signals as measured by average RMSE and CC, respectively. The horizontal lines inside the boxes denote the medians whereas the circles denote the means. Asterisks represent neural signals whose decoding performance differed significantly from that of ESA (two-tailed paired Wilcoxon signed-rank test; *** $p < 0.001$). (c),(e) Heatmap of statistical significant matrix of decoding performance between all possible pairs of neural signals as measured by average RMSE and CC, respectively. Each cell is coloured based on the p-value except cells in the main diagonal (grey; excluded from statistical test calculation). Asterisks indicate whether there were statistical significant differences between the pairs (two-tailed paired Wilcoxon signed-rank test; *** $p < 0.001$). (d),(f) Average performance improvement/degradation (in %) of neural signals with respect to that of SUA as measured by RMSE and CC metrics, respectively. Positive (negative) value indicates performance improvement (degradation). Black numeral texts represent the means whereas black error bars represent the 95% confidence intervals ($n = 26$). (g),(h) Decoding performance comparison over time across neural signals in terms of average RMSE and CC, respectively. Coloured circles represent the means whereas coloured error bars represent the 95% confidence intervals ($n = 10$) for each session. The x-axes represent the number of days since electrode implantation. The x-axes' ticks correspond to different sessions having irregular gaps between the consecutive sessions.

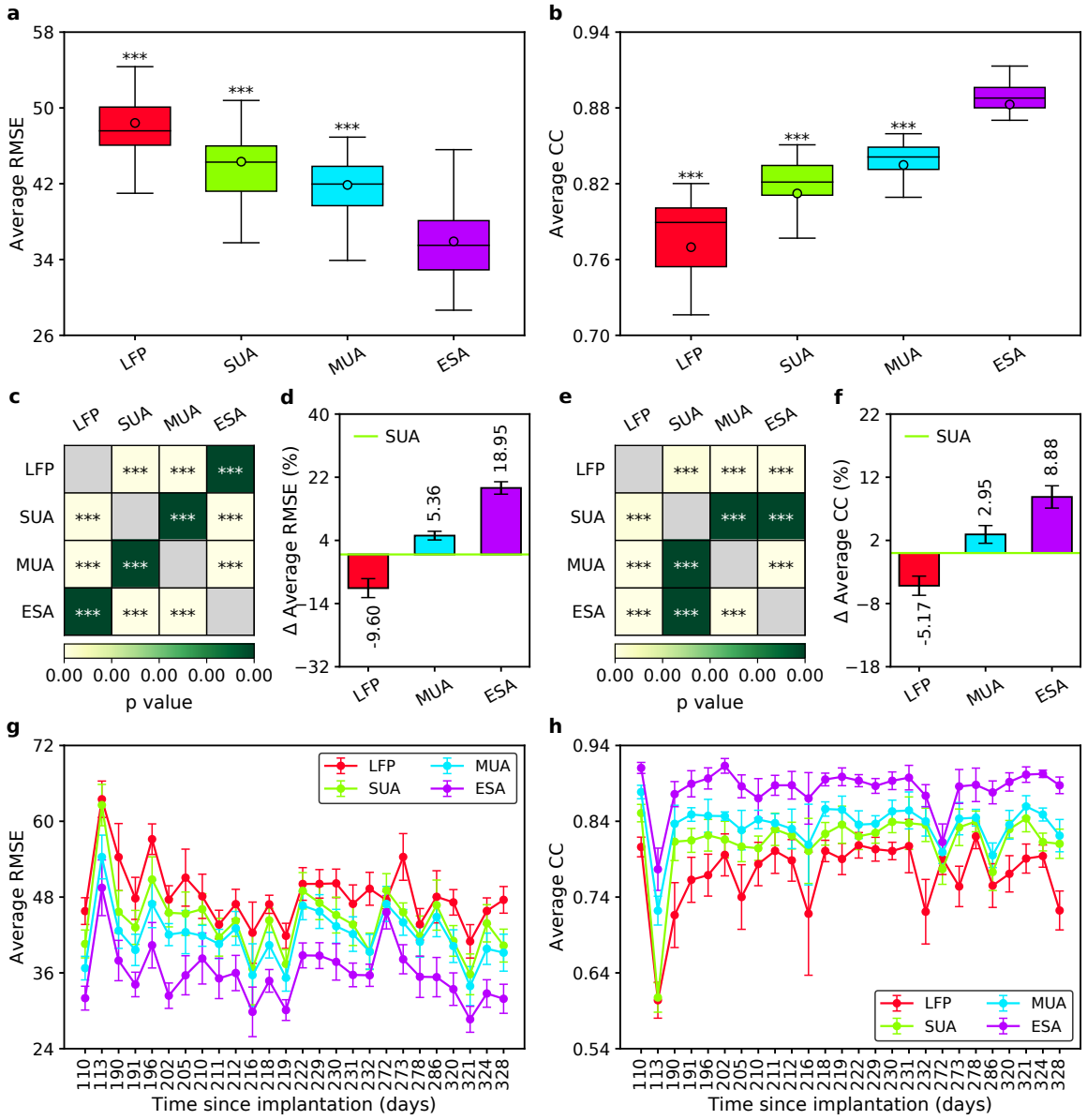


Figure B.14 Long-term decoding performance comparison across LSTM decoders driven by different neural signals. (a),(b) Boxplot comparison of long-term decoding performance across different neural signals as measured by average RMSE and CC, respectively. The horizontal lines inside the boxes denote the medians whereas the circles denote the means. Asterisks represent neural signals whose decoding performance differed significantly from that of ESA (two-tailed paired Wilcoxon signed-rank test; *** $p < 0.001$). (c),(e) Heatmap of statistical significant matrix of decoding performance between all possible pairs of neural signals as measured by average RMSE and CC, respectively. Each cell is coloured based on the p-value except cells in the main diagonal (grey; excluded from statistical test calculation). Asterisks indicate whether there were statistical significant differences between the pairs (two-tailed paired Wilcoxon signed-rank test; *** $p < 0.001$). (d),(f) Average performance improvement/degradation (in %) of neural signals with respect to that of SUA as measured by RMSE and CC metrics, respectively. Positive (negative) value indicates performance improvement (degradation). Black numeral texts represent the means whereas black error bars represent the 95% confidence intervals ($n = 26$). (g),(h) Decoding performance comparison over time across neural signals in terms of average RMSE and CC, respectively. Coloured circles represent the means whereas coloured error bars represent the 95% confidence intervals ($n = 10$) for each session. The x-axes represent the number of days since electrode implantation. The x-axes' ticks correspond to different sessions having irregular gaps between the consecutive sessions.

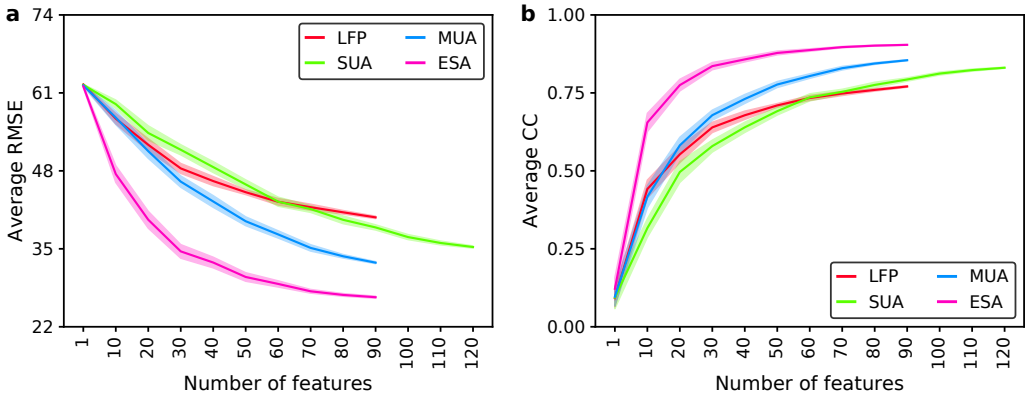


Figure B.15 Decoding performance comparison across different types of neural signals. (a),(b) Impact of different number of features (units or channels) on the decoding comparison of QRNN decoders driven by different neural signals as measured by average RMSE and CC, respectively. The shaded areas represent the 95% confidence intervals ($n = 30$). Data are taken from the last session (I20170131_02).

C | List of Publications

[1] **N. Ahmadi**, T. G. Constandinou, and C.-S. Bouganis, "Estimation of neuronal firing rate using Bayesian adaptive kernel smoother (BAKS)", *PLoS One*, vol. 13, no. 11, e0206794, 2018. DOI: 10.1371/journal.pone.0206794.

[2] **N. Ahmadi**, T. G. Constandinou, and C.-S. Bouganis, "Spike rate estimation using Bayesian adaptive kernel smoother (BAKS) and its application to brain-machine interfaces", in *40th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, IEEE, 2018, pp. 2547–2550. DOI: 10.1109/EMBC.2018.8512830.

[3] **N. Ahmadi**, T. G. Constandinou, and C.-S. Bouganis, "Improved spike-based brain-machine interface using bayesian adaptive kernel smoother and deep learning", *techRxiv*, 2020. DOI: 10.36227/techrxiv.12383600.v1.

[4] **N. Ahmadi**, M. L. Cavuto, P. Feng, L. B. Leene, M. Maslik, F. Mazza, O. Savolainen, K. M. Szostak, C.-S. Bouganis, J. Ekanayake, *et al.*, "Towards a distributed, chronically-implantable neural interface", in *9th Int. IEEE/EMBS Conf. Neural Eng. (NER)*, IEEE, 2019, pp. 719–724. DOI: 10.1109/NER.2019.8716998.

[5] **N. Ahmadi**, T. G. Constandinou, and C.-S. Bouganis, "Decoding hand kinematics from local field potentials using long short-term memory (LSTM) network", in *9th Int. IEEE/EMBS Conf. Neural Eng. (NER)*, IEEE, 2019, pp. 415–419. DOI: 10.1109/NER.2019.8717045.

[6] **N. Ahmadi**, T. G. Constandinou, and C.-S. Bouganis, "End-to-end hand kinematic decoding from LFPs using temporal convolutional network", in *IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, IEEE, 2019, pp. 1–4. DOI: 10.1109/BIOCAS.2019.8919131.

[7] **N. Ahmadi**, T. G. Constandinou, and C.-S. Bouganis, "Impact of referencing scheme on decoding performance of LFP-based brain-machine interface", *bioRxiv*, 2020. DOI: 10.1101/2020.05.03.075218.

[8] **N. Ahmadi**, T. G. Constandinou, and C.-S. Bouganis, "Robust and accurate decoding of hand kinematics from entire spiking activity using deep learning", *bioRxiv*, 2020. DOI: 10.1101/2020.05.07.083063.

[9] **N. Ahmadi**, T. G. Constandinou, and C.-S. Bouganis, "Inferring entire spiking activity from local field potentials with deep learning", *bioRxiv*, 2020. DOI: 10.1101/2020.05.02.074104.

Bibliography

[1] B. S. Armour, E. A. Courtney-Long, M. H. Fox, H. Fredine, and A. Cahill, "Prevalence and causes of paralysis–united states, 2013", *Am. J. Public Health*, vol. 106, no. 10, pp. 1855–1857, 2016.

[2] D. McDaid, A.-L. Park, A. Gall, M. Purcell, and M. Bacon, "Understanding and modelling the economic impact of spinal cord injuries in the united kingdom", *Spinal Cord*, vol. 57, no. 9, pp. 778–788, 2019.

[3] N. I. for Clinical Excellence, *Spinal injury: Assessment and initial management*, `https://www.spinal.co.uk`, 2019.

[4] K. D. Anderson, "Targeting recovery: Priorities of the spinal cord-injured population", *J. Neurotrauma*, vol. 21, no. 10, pp. 1371–1383, 2004.

[5] C. H. Blabe, V. Gilja, C. A. Chestek, K. V. Shenoy, K. D. Anderson, and J. M. Henderson, "Assessment of brain–machine interfaces from the perspective of people with paralysis", *J. Neural Eng.*, vol. 12, no. 4, p. 043 002, 2015.

[6] J. Lahr, C. Schwartz, B. Heimbach, A. Aertsen, J. Rickert, and T. Ball, "Invasive brain–machine interfaces: A survey of paralyzed patients' attitudes, knowledge and methods of information retrieval", *J. Neural Eng.*, vol. 12, no. 4, p. 043 001, 2015.

[7] J. Wessberg, C. R. Stambaugh, J. D. Kralik, P. D. Beck, M. Laubach, J. K. Chapin, J. Kim, S. J. Biggs, M. A. Srinivasan, and M. A. Nicolelis, "Real-time prediction of hand trajectory by ensembles of cortical neurons in primates", *Nature*, vol. 408, no. 6810, p. 361, 2000.

[8] M. D. Serruya, N. G. Hatsopoulos, L. Paninski, M. R. Fellows, and J. P. Donoghue, "Instant neural control of a movement signal", *Nature*, vol. 416, no. 6877, pp. 141–142, 2002.

[9] D. M. Taylor, S. I. H. Tillery, and A. B. Schwartz, "Direct cortical control of 3d neuroprosthetic devices", *Science*, vol. 296, no. 5574, pp. 1829–1832, 2002.

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

[11] S.-P. Kim, J. D. Simeral, L. R. Hochberg, J. P. Donoghue, and M. J. Black, "Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia", *J. Neural Eng.*, vol. 5, no. 4, p. 455, 2008.

[12] J. Simeral, S.-P. Kim, M. Black, J. Donoghue, and L. Hochberg, "Neural control of cursor trajectory and click by a human with tetraplegia 1000 days after implant of an intracortical microelectrode array", *J. Neural Eng.*, vol. 8, no. 2, p. 025 027, 2011.

[13] D. Bacher, B. Jarosiewicz, N. Y. Masse, S. D. Stavisky, J. D. Simeral, K. Newell, E. M. Oakley, S. S. Cash, G. Friehs, and L. R. Hochberg, "Neural point-and-click communication by a person with incomplete locked-in syndrome", *Neurorehabil. Neural Repair*, vol. 29, no. 5, pp. 462–471, 2015.

[14] J. M. Carmena, M. A. Lebedev, R. E. Crist, J. E. O'Doherty, D. M. Santucci, D. F. Dimitrov, P. G. Patil, C. S. Henriquez, and M. A. Nicolelis, "Learning to control a brain–machine interface for reaching and grasping by primates", *PLOS Biol.*, vol. 1, no. 2, e42, 2003.

[15] L. R. Hochberg, M. D. Serruya, G. M. Friehs, J. A. Mukand, M. Saleh, A. H. Caplan, A. Branner, D. Chen, R. D. Penn, and J. P. Donoghue, "Neuronal ensemble control of prosthetic devices by a human with tetraplegia", *Nature*, vol. 442, no. 7099, p. 164, 2006.

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

[17] B Wodlinger, J. Downey, E. Tyler-Kabara, A. Schwartz, M. Boninger, and J. Collinger, "Ten-dimensional anthropomorphic arm control in a human brain- machine interface: Difficulties, solutions, and limitations", *J. Neural Eng.*, vol. 12, no. 1, p. 016 011, 2014.

[18] T. Aflalo, S. Kellis, C. Klaes, B. Lee, Y. Shi, K. Pejsa, K. Shanfield, S. Hayes-Jackson, M. Aisen, C. Heck, *et al.*, "Decoding motor imagery from the posterior parietal cortex of a tetraplegic human", *Science*, vol. 348, no. 6237, pp. 906–910, 2015.

[19] C. A. Chestek, V. Gilja, P. Nuyujukian, J. D. Foster, J. M. Fan, M. T. Kaufman, M. M. Churchland, Z. Rivera-Alvidrez, J. P. Cunningham, S. I. Ryu, *et al.*, "Long-term stability of neural prosthetic control signals from silicon cortical arrays in rhesus macaque motor cortex", *J. Neural Eng.*, vol. 8, no. 4, p. 045 005, 2011.

[20] J. A. Perge, M. L. Homer, W. Q. Malik, S. Cash, E. Eskandar, G. Friehs, J. P. Donoghue, and L. R. Hochberg, "Intra-day signal instabilities affect decoding performance in an intracortical neural interface system", *J. Neural Eng.*, vol. 10, no. 3, p. 036 004, 2013.

[21] R. A. Andersen, S. Kellis, C. Klaes, and T. Aflalo, "Toward more versatile and intuitive cortical brain–machine interfaces", *Curr. Biol.*, vol. 24, no. 18, R885–R897, 2014.

[22] D. Wang, Q. Zhang, Y. Li, Y. Wang, J. Zhu, S. Zhang, and X. Zheng, "Long-term decoding stability of local field potentials from silicon arrays in primate motor cortex during a 2D center out task", *J. Neural Eng.*, vol. 11, no. 3, p. 036 009, 2014.

[23] M. D. Murphy, D. J. Guggenmos, D. T. Bundy, and R. J. Nudo, "Current challenges facing the translation of brain computer interfaces from preclinical trials to use in human patients", *Front. Cell. Neurosci.*, vol. 9, p. 497, 2016.

[24] M. D. Serruya, "Bottlenecks to clinical translation of direct brain-computer interfaces", *Frontiers in systems neuroscience*, vol. 8, p. 226, 2014.

[25] A. Jackson and T. M. Hall, "Decoding local field potentials for neural interfaces", *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 10, pp. 1705–1714, 2017.

[26] G. W. Fraser, S. M. Chase, A. Whitford, and A. B. Schwartz, "Control of a brain–computer interface without spike sorting", *J. Neural Eng.*, vol. 6, no. 5, p. 055 004, 2009.

[27] B. P. Christie, D. M. Tat, Z. T. Irwin, V. Gilja, P. Nuyujukian, J. D. Foster, S. I. Ryu, K. V. Shenoy, D. E. Thompson, and C. A. Chestek, "Comparison of spike sorting and thresholding of voltage waveforms for intracortical brain–machine interface performance", *J. Neural Eng.*, vol. 12, no. 1, p. 016 009, 2014.

[28] H. Lindén, T. Tetzlaff, T. C. Potjans, K. H. Pettersen, S. Grün, M. Diesmann, and G. T. Einevoll, "Modeling the spatial reach of the LFP", *Neuron*, vol. 72, no. 5, pp. 859–872, 2011.

[29] G. Buzsáki, C. A. Anastassiou, and C. Koch, "The origin of extracellular fields and currents—EEG, ECoG, LFP and spikes", *Nat. Rev. Neurosci.*, vol. 13, no. 6, p. 407, 2012.

[30] G. T. Einevoll, C. Kayser, N. K. Logothetis, and S. Panzeri, "Modelling and analysis of local field potentials for studying the function of cortical circuits", *Nat. Rev. Neurosci.*, vol. 14, no. 11, p. 770, 2013.

[31] C. Mehring, J. Rickert, E. Vaadia, S. C. de Oliveira, A. Aertsen, and S. Rotter, "Inference of hand movements from local field potentials in monkey motor cortex", *Nat. Neurosci.*, vol. 6, no. 12, p. 1253, 2003.

[32] J. Zhuang, W. Truccolo, C. Vargas-Irwin, and J. P. Donoghue, "Decoding 3-D reach and grasp kinematics from high-frequency local field potentials in primate primary motor cortex", *IEEE Trans. Biomed. Eng.*, vol. 57, no. 7, pp. 1774–1784, 2010.

[33] A. K. Bansal, W. Truccolo, C. E. Vargas-Irwin, and J. P. Donoghue, "Decoding 3D reach and grasp from hybrid signals in motor and premotor cortices: Spikes, multiunit activity, and local field potentials", *J. Neurophysiol.*, vol. 107, no. 5, pp. 1337–1355, 2012.

[34] E. J. Hwang and R. A. Andersen, "The utility of multichannel local field potentials for brain–machine interfaces", *J. Neural Eng.*, vol. 10, no. 4, p. 046 005, 2013.

[35] R. D. Flint, E. W. Lindberg, L. R. Jordan, L. E. Miller, and M. W. Slutzky, "Accurate decoding of reaching movements from field potentials in the absence of spikes", *J. Neural Eng.*, vol. 9, no. 4, p. 046 006, 2012.

[36] T. M. Hall, K. Nazarpour, and A. Jackson, "Real-time estimation and biofeedback of single-neuron firing rates using local field potentials", *Nat. Commun.*, vol. 5, p. 5462, 2014.

[37] S. D. Stavisky, J. C. Kao, P. Nuyujukian, S. I. Ryu, and K. V. Shenoy, "A high performing brain–machine interface driven by low-frequency local field potentials alone and together with spikes", *J. Neural Eng.*, vol. 12, no. 3, p. 036 009, 2015.

[38] S. Todorova, P. Sadtler, A. Batista, S. Chase, and V. Ventura, "To sort or not to sort: The impact of spike-sorting on neural decoding performance", *J. Neural Eng.*, vol. 11, no. 5, p. 056 005, 2014.

[39] S. Perel, P. T. Sadtler, E. R. Oby, S. I. Ryu, E. C. Tyler-Kabara, A. P. Batista, and S. M. Chase, "Single-unit activity, threshold crossings, and local field potentials in motor

cortex differentially encode reach kinematics", *J. Neurophysiol.*, vol. 114, no. 3, pp. 1500–1512, 2015.

[40] N. F. Ince, R. Gupta, S. Arica, A. H. Tewfik, J. Ashe, and G. Pellizzer, "High accuracy decoding of movement target direction in non-human primates based on common spatial patterns of local field potentials", *PLoS ONE*, vol. 5, no. 12, 2010.

[41] V. Aggarwal, M. Mollazadeh, A. G. Davidson, M. H. Schieber, and N. V. Thakor, "State-based decoding of hand and finger kinematics using neuronal ensemble and lfp activity during dexterous reach-to-grasp movements", *J. Neurophysiol.*, vol. 109, no. 12, pp. 3067–3081, 2013.

[42] N. Hatsopoulos, J. Joshi, and J. G. O'Leary, "Decoding continuous and discrete motor behaviors using motor and premotor cortical ensembles", *J. Neurophysiol.*, vol. 92, no. 2, pp. 1165–1174, 2004.

[43] S.-P. Kim, J. C. Sanchez, Y. N. Rao, D. Erdogmus, J. M. Carmena, M. A. Lebedev, M. Nicolelis, and J. Principe, "A comparison of optimal MIMO linear and nonlinear models for brain–machine interfaces", *J. Neural Eng.*, vol. 3, no. 2, p. 145, 2006.

[44] W. Wu, M. J. Black, Y. Gao, M Serruya, A Shaikhouni, J. Donoghue, and E. Bienenstock, "Neural decoding of cursor motion using a kalman filter", in *Adv. Neural Inf. Process. Syst.*, 2003, pp. 133–140.

[45] W. Wu, M. J. Black, D. Mumford, Y. Gao, E. Bienenstock, and J. P. Donoghue, "Modeling and decoding motor cortical activity using a switching kalman filter", *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 933–942, 2004.

[46] W. Wu, Y. Gao, E. Bienenstock, J. P. Donoghue, and M. J. Black, "Bayesian population decoding of motor cortical activity using a kalman filter", *Neural Comput.*, vol. 18, no. 1, pp. 80–118, 2006.

[47] W. Q. Malik, W. Truccolo, E. N. Brown, and L. R. Hochberg, "Efficient decoding with steady-state kalman filter in neural interface systems", *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 19, no. 1, pp. 25–34, 2010.

[48] S.-P. Kim, J. D. Simeral, L. R. Hochberg, J. P. Donoghue, G. M. Friehs, and M. J. Black, "Point-and-click cursor control with an intracortical neural interface system by humans with tetraplegia", *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 19, no. 2, pp. 193–203, 2011.

[49] M. Paluš, "Nonlinearity in normal human EEG: Cycles, temporal asymmetry, nonstationarity and randomness, not chaos", *Biol. Cybern.*, vol. 75, no. 5, pp. 389–396, 1996.

[50] C. Ethier, E. R. Oby, M. J. Bauman, and L. E. Miller, "Restoration of grasp following paralysis through brain-controlled stimulation of muscles", *Nature*, vol. 485, no. 7398, pp. 368–371, 2012.

[51] R. D. Flint, C. Ethier, E. R. Oby, L. E. Miller, and M. W. Slutzky, "Local field potentials allow accurate decoding of muscle activity", *J. Neurophysiol.*, vol. 108, no. 1, pp. 18–24, 2012.

[52] R. D. Flint, Z. A. Wright, M. R. Scheid, and M. W. Slutzky, "Long term, stable brain machine interface performance using local field potentials and multiunit spikes", *J. Neural Eng.*, vol. 10, no. 5, p. 056 005, 2013.

[53] R. D. Flint, M. R. Scheid, Z. A. Wright, S. A. Solla, and M. W. Slutzky, "Long-term stability of motor cortical activity: Implications for brain machine interfaces and optimal feedback control", *J. Neurosci.*, vol. 36, no. 12, pp. 3623–3632, 2016.

[54] Z. Li, J. E. O'Doherty, T. L. Hanson, M. A. Lebedev, C. S. Henriquez, and M. A. Nicolelis, "Unscented kalman filter for brain-machine interfaces", *PLOS ONE*, vol. 4, no. 7, e6243, 2009.

[55] Z. Li, J. E. O'Doherty, M. A. Lebedev, and M. A. Nicolelis, "Adaptive decoding for brain-machine interfaces through bayesian parameter updates", *Neural Comput.*, vol. 23, no. 12, pp. 3162–3204, 2011.

[56] P. J. Ifft, S. Shokur, Z. Li, M. A. Lebedev, and M. A. Nicolelis, "A brain-machine interface enables bimanual arm movements in monkeys", *Sci. Transl. Med.*, vol. 5, no. 210, 210ra154–210ra154, 2013.

[57] S. Li, J. Li, and Z. Li, "An improved unscented kalman filter based decoder for cortical brain-machine interfaces", *Front. Neurosci.*, vol. 10, p. 587, 2016.

[58] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning", *Nature*, vol. 521, no. 7553, p. 436, 2015.

[59] J. Schmidhuber, "Deep learning in neural networks: An overview", *Neural networks*, vol. 61, pp. 85–117, 2015.

[60] J. I. Glaser, R. H. Chowdhury, M. G. Perich, L. E. Miller, and K. P. Kording, "Machine learning for neural decoding", *arXiv preprint*, arXiv:1708.00909, 2017.

[61] M. A. Schwemmer, N. D. Skomrock, P. B. Sederberg, J. E. Ting, G. Sharma, M. A. Bockbrader, and D. A. Friedenberg, "Meeting brain–computer interface user performance expectations using a deep neural network decoding framework", *Nat. Med.*, vol. 24, no. 11, p. 1669, 2018.

[62] P.-H. Tseng, N. A. Urpi, M. Lebedev, and M. Nicolelis, "Decoding movements from cortical ensemble activity using a long short-term memory recurrent network", *Neural Comput.*, vol. 31, no. 6, pp. 1085–1113, 2019.

[63] E. V. Evarts, "Relation of pyramidal tract activity to force exerted during voluntary movement", *J. Neurophysiol.*, vol. 31, no. 1, pp. 14–27, 1968.

[64] E. E. Fetz, "Operant conditioning of cortical unit activity", *Science*, vol. 163, no. 3870, pp. 955–958, 1969.

[65] D. R. Humphrey, E. Schmidt, and W. Thompson, "Predicting measures of motor performance from multiple cortical spike trains", *Science*, vol. 170, no. 3959, pp. 758–762, 1970.

[66] J. J. Vidal, "Real-time detection of brain events in eeg", *Proc. IEEE*, vol. 65, no. 5, pp. 633–641, 1977.

[67] A. P. Georgopoulos, J. F. Kalaska, R. Caminiti, and J. T. Massey, "On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex", *J. Neurosci.*, vol. 2, no. 11, pp. 1527–1537, 1982.

[68] A. P. Georgopoulos, A. B. Schwartz, and R. E. Kettner, "Neuronal population coding of movement direction", *Science*, vol. 233, no. 4771, pp. 1416–1419, 1986.

[69] K. L. Drake, K. D. Wise, J. Farraye, D. J. Anderson, and S. L. BeMent, "Performance of planar multisite microprobes in recording extracellular single-unit intracortical activity", *IEEE Trans. Biomed. Eng.*, vol. 35, no. 9, pp. 719–732, 1988.

[70] K Wise and J Angell, "A microprobe with integrated amplifiers for neurophysiology", in *IEEE Int. Solid-State Circuits Conf. (ISSCC)*, IEEE, vol. 14, 1971, pp. 100–101.

[71] P. K. Campbell, K. E. Jones, R. J. Huber, K. W. Horch, and R. A. Normann, "A silicon-based, three-dimensional neural interface: Manufacturing processes for an intracortical electrode array", *IEEE Trans. Biomed. Eng.*, vol. 38, no. 8, pp. 758–768, 1991.

[72] P. R. Kennedy and R. A. Bakay, "Restoration of neural output from a paralyzed patient by a direct brain connection", *Neuroreport*, vol. 9, no. 8, pp. 1707–1711, 1998.

[73] J. K. Chapin, K. A. Moxon, R. S. Markowitz, and M. A. Nicolelis, "Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex", *Nat. Neurosci.*, vol. 2, no. 7, p. 664, 1999.

[74] N. G. Hatsopoulos and J. P. Donoghue, "The science of neural interface systems", *Annu. Rev. Neurosci.*, vol. 32, pp. 249–266, 2009.

[75] L. R. Hochberg, D. Bacher, B. Jarosiewicz, N. Y. Masse, J. D. Simeral, J. Vogel, S. Haddadin, J. Liu, S. S. Cash, P. van der Smagt, *et al.*, "Reach and grasp by people with tetraplegia using a neurally controlled robotic arm", *Nature*, vol. 485, no. 7398, p. 372, 2012.

[76] A. B. Ajiboye, F. R. Willett, D. R. Young, W. D. Memberg, B. A. Murphy, J. P. Miller, B. L. Walter, J. A. Sweet, H. A. Hoyen, M. W. Keith, *et al.*, "Restoration of reaching and grasping movements through brain-controlled muscle stimulation in a person with tetraplegia: A proof-of-concept demonstration", *The Lancet*, vol. 389, no. 10081, pp. 1821–1830, 2017.

[77] M. A. Lebedev and M. A. Nicolelis, "Brain-machine interfaces: From basic science to neuroprostheses and neurorehabilitation", *Physiol. Rev.*, vol. 97, no. 2, pp. 767–837, 2017.

[78] D. M. Brandman, S. S. Cash, and L. R. Hochberg, "Human intracortical recording and neural decoding for brain–computer interfaces", *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 10, pp. 1687–1696, 2017.

[79] R. A. Andersen, S. Musallam, and B. Pesaran, "Selecting the signals for a brain–machine interface", *Curr. Opin. Neurobiol.*, vol. 14, no. 6, pp. 720–726, 2004.

[80] J. J. Jun, N. A. Steinmetz, J. H. Siegle, D. J. Denman, M. Bauza, B. Barbarits, A. K. Lee, C. A. Anastassiou, A. Andrei, Ç. Aydın, *et al.*, "Fully integrated silicon probes for high-density recording of neural activity", *Nature*, vol. 551, no. 7679, pp. 232–236, 2017.

[81] E. Musk *et al.*, "An integrated brain-machine interface platform with thousands of channels", *J. Med. Internet Res.*, vol. 21, no. 10, e16194, 2019.

[82] D. Seo, J. M. Carmena, J. M. Rabaey, M. M. Maharbiz, and E. Alon, "Model validation of untethered, ultrasonic neural dust motes for cortical recording", *J. Neurosci. Methods*, vol. 244, pp. 114–122, 2015.

[83] N. Ahmadi, M. L. Cavuto, P. Feng, L. B. Leene, M. Maslik, F. Mazza, O. Savolainen, K. M. Szostak, C.-S. Bouganis, J. Ekanayake, *et al.*, "Towards a distributed, chronically-implantable neural interface", in *Proc. Int. IEEE/EMBS Conf. Neural Eng. (NER)*, IEEE, 2019, pp. 719–724.

[84] R. Chen, A. Canales, and P. Anikeeva, "Neural recording and modulation technologies", *Nat. Rev. Mater.*, vol. 2, no. 2, pp. 1–16, 2017.

[85] G. Hong and C. M. Lieber, "Novel electrode technologies for neural recordings", *Nat. Rev. Neurosci.*, vol. 20, no. 6, pp. 330–345, 2019.

[86] H. G. Rey, C. Pedreira, and R. Q. Quiroga, "Past, present and future of spike sorting techniques", *Brain Res. Bull.*, vol. 119, pp. 106–117, 2015.

[87] R. Bestel, A. W. Daus, and C. Thielemann, "A novel automated spike sorting algorithm with adaptable feature extraction", *J. Neurosci. Methods*, vol. 211, no. 1, pp. 168–178, 2012.

[88] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul, "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering", *Neural Comput.*, vol. 16, no. 8, pp. 1661–1687, 2004.

[89] B. Lefebvre, P. Yger, and O. Marre, "Recent progress in multi-electrode spike sorting methods", *Jo. Physiol. Paris*, vol. 110, no. 4, pp. 327–335, 2016.

[90] V. Gilja, C. Pandarinath, C. H. Blabe, P. Nuyujukian, J. D. Simeral, A. A. Sarma, B. L. Sorice, J. A. Perge, B. Jarosiewicz, L. R. Hochberg, *et al.*, "Clinical translation of a high-performance neural prosthesis", *Nat. Med.*, vol. 21, no. 10, p. 1142, 2015.

[91] J. P. Cunningham, V. Gilja, S. I. Ryu, and K. V. Shenoy, "Methods for estimating neural firing rates, and their application to brain–machine interfaces", *Neural Networks*, vol. 22, no. 9, pp. 1235–1246, 2009. DOI: 10.1016/j.neunet.2009.02.004.

[92] J. A. Perge, S. Zhang, W. Q. Malik, M. L. Homer, S. Cash, G. Friehs, E. N. Eskandar, J. P. Donoghue, and L. R. Hochberg, "Reliability of directional information in unsorted spikes and local field potentials recorded in human motor cortex", *J. Neural Eng.*, vol. 11, no. 4, p. 046 007, 2014.

[93] K. So, S. Dangi, A. L. Orsborn, M. C. Gastpar, and J. M. Carmena, "Subject-specific modulation of local field potential spectral power during brain–machine interface control in primates", *J. Neural Eng.*, vol. 11, no. 2, p. 026 002, 2014.

[94] C. Magri, A. Mazzoni, N. K. Logothetis, and S. Panzeri, "Optimal band separation of extracellular field potentials", *J. Neurosci. Methods*, vol. 210, no. 1, pp. 66–78, 2012.

[95] D. J. Warren, S. Kellis, J. G. Nieveen, S. M. Wendelken, H. Dantas, T. S. Davis, D. T. Hutchinson, R. A. Normann, G. A. Clark, and V. J. Mathews, "Recording and decoding for neural prostheses", *Proc. IEEE*, vol. 104, no. 2, pp. 374–391, 2016.

[96] M. M. Shanechi, "Brain–machine interface control algorithms", *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 10, pp. 1725–1734, 2017.

[97] L. Srinivasan, U. T. Eden, A. S. Willsky, and E. N. Brown, "A state-space analysis for reconstruction of goal-directed movements using neural signals", *Neural Comput.*, vol. 18, no. 10, pp. 2465–2494, 2006.

[98] B. M. Yu, C. Kemere, G. Santhanam, A. Afshar, S. I. Ryu, T. H. Meng, M. Sahani, and K. V. Shenoy, "Mixture of trajectory models for neural decoding of goal-directed movements", *J. Neurophysiol.*, vol. 97, no. 5, pp. 3763–3780, 2007.

[99] G. H. Mulliken, S. Musallam, and R. A. Andersen, "Decoding trajectories from posterior parietal cortex ensembles", *J. Neurosci.*, vol. 28, no. 48, pp. 12 913–12 926, 2008.

[100] M. M. Shanechi, G. W. Wornell, Z. M. Williams, and E. N. Brown, "Feedback-controlled parallel point process filter for estimation of goal-directed movements from neural signals", *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 21, no. 1, pp. 129–140, 2013.

[101] M. M. Shanechi, Z. M. Williams, G. W. Wornell, R. C. Hu, M. Powers, and E. N. Brown, "A real-time brain-machine interface combining motor target and trajectory intent using an optimal feedback control design", *PLOS ONE*, vol. 8, no. 4, e59049, 2013.

[102] J. C. Kao, P. Nuyujukian, S. I. Ryu, and K. V. Shenoy, "A high-performance neural prosthesis incorporating discrete state selection with hidden markov models", *IEEE Trans. Biomed. Eng.*, vol. 64, no. 4, pp. 935–945, 2017.

[103] C. Pandarinath, P. Nuyujukian, C. H. Blabe, B. L. Sorice, J. Saab, F. R. Willett, L. R. Hochberg, K. V. Shenoy, and J. M. Henderson, "High performance communication by people with paralysis using an intracortical brain-computer interface", *Elife*, vol. 6, 2017.

[104] M. Parastarfeizabadi and A. Z. Kouzani, "Advances in closed-loop deep brain stimulation devices", *J. Neuroeng. Rehabil.*, vol. 14, no. 1, p. 79, 2017.

[105] S. A. Shah, H. Tan, G. Tinkhauser, and P. Brown, "Towards real-time, continuous decoding of gripping force from deep brain local field potentials", *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 7, pp. 1460–1468, 2018.

[106] B. Houston, M. Thompson, A. Ko, and H. Chizeck, "A machine-learning approach to volitional control of a closed-loop deep brain stimulation system", *J. Neural Eng.*, vol. 16, no. 1, p. 016 004, 2018.

[107] M. J. Vansteensel, E. G. Pels, M. G. Bleichner, M. P. Branco, T. Denison, Z. V. Freudenburg, P. Gosselaar, S. Leinders, T. H. Ottens, M. A. Van Den Boom, *et al.*, "Fully implanted brain–computer interface in a locked-in patient with ALS", *New England Journal of Medicine*, vol. 375, no. 21, pp. 2060–2066, 2016.

[108] K. Volkova, M. A. Lebedev, A. Kaplan, and A. Ossadtchi, "Decoding movement from electrocorticographic activity: A review", *Front. Neuroinform.*, vol. 13, 2019.

[109] M. Jorfi, J. L. Skousen, C. Weder, and J. R. Capadona, "Progress towards biocompatible intracortical microelectrodes for neural interfacing applications", *J. Neural Eng.*, vol. 12, no. 1, p. 011 001, 2014.

[110] G. R. Müller-Putz, R. Scherer, G. Pfurtscheller, and R. Rupp, "Eeg-based neuroprosthesis control: A step towards clinical practice", *Neurosci. Lett.*, vol. 382, no. 1-2, pp. 169–174, 2005.

[111] S. Waldert, H. Preissl, E. Demandt, C. Braun, N. Birbaumer, A. Aertsen, and C. Mehring, "Hand movement direction decoded from meg and eeg", *J. Neurosci.*, vol. 28, no. 4, pp. 1000–1008, 2008.

[112] K Jerbi, J. Vidal, J Mattout, E Maby, F Lecaignard, T Ossandon, C. Hamamé, S. Dalal, R Bouet, J.-P. Lachaux, *et al.*, "Inferring hand movement kinematics from meg, eeg and intracranial eeg: From brain-machine interfaces to motor rehabilitation", *IRBM*, vol. 32, no. 1, pp. 8–18, 2011.

[113] D. Huang, P. Lin, D.-Y. Fei, X. Chen, and O. Bai, "Decoding human motor activity from eeg single trials for a discrete two-dimensional cursor control", *J. Neural Eng.*, vol. 6, no. 4, p. 046 005, 2009.

[114] D. J. McFarland, W. A. Sarnacki, and J. R. Wolpaw, "Electroencephalographic (eeg) control of three-dimensional movement", *J. Neural Eng.*, vol. 7, no. 3, p. 036 007, 2010.

[115] T. J. Bradberry, R. J. Gentili, and J. L. Contreras-Vidal, "Reconstructing three-dimensional hand movements from noninvasive electroencephalographic signals", *J. Neurosci.*, vol. 30, no. 9, pp. 3432–3437, 2010.

[116] J.-H. Kim, F. Bießmann, and S.-W. Lee, "Decoding three-dimensional trajectory of executed and imagined arm movements from electroencephalogram signals", *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 23, no. 5, pp. 867–876, 2015.

[117] G. R. Holt and C. Koch, "Electrical interactions via the extracellular potential near cell bodies", *J. Comput. Neurosci.*, vol. 6, no. 2, pp. 169–184, 1999.

[118] D. A. Henze, Z. Borhegyi, J. Csicsvari, A. Mamiya, K. D. Harris, and G. Buzsaki, "Intracellular features predicted by extracellular recordings in the hippocampus in vivo", *J. Neurophysiol.*, vol. 84, no. 1, pp. 390–400, 2000.

[119] G. Buzsáki, "Large-scale recording of neuronal ensembles", *Nat. Neurosci.*, vol. 7, no. 5, p. 446, 2004.

[120] C. Gold, D. A. Henze, C. Koch, and G. Buzsaki, "On the origin of the extracellular action potential waveform: A modeling study", *J. Neurophysiol.*, vol. 95, no. 5, pp. 3113–3128, 2006.

[121] U. Mitzdorf, "Current source-density method and application in cat cerebral cortex: Investigation of evoked potentials and eeg phenomena", *Physiol. Rev.*, vol. 65, no. 1, pp. 37–100, 1985.

[122] M. W. Slutzky and R. D. Flint, "Physiological properties of brain-machine interface input signals", *J. Neurophysiol.*, vol. 118, no. 2, pp. 1329–1343, 2017.

[123] E. V. Evarts, "Activity of motor cortex neurons in association with learned movement", *Int. J. Neurosci.*, vol. 3, no. 3, pp. 113–124, 1972.

[124] D. W. Moran and A. B. Schwartz, "Motor cortical representation of speed and direction during reaching", *J. Neurophysiol.*, vol. 82, no. 5, pp. 2676–2692, 1999.

[125] F. R. Willett, A. J. Suminski, A. H. Fagg, and N. G. Hatsopoulos, "Improving brain–machine interface performance by decoding intended future movements", *J. Neural Eng.*, vol. 10, no. 2, p. 026 011, 2013.

[126] S. Koyama, S. M. Chase, A. S. Whitford, M. Velliste, A. B. Schwartz, and R. E. Kass, "Comparison of brain–computer interface decoding algorithms in open-loop and closed-loop control", *J. Comput. Neurosci.*, vol. 29, no. 1-2, pp. 73–87, 2010.

[127] K. V. Shenoy and J. M. Carmena, "Combining decoder design and neural adaptation in brain-machine interfaces", *Neuron*, vol. 84, no. 4, pp. 665–680, 2014.

[128] D. M. Brandman, T. Hosman, J. Saab, M. C. Burkhart, B. E. Shanahan, J. G. Ciancibello, A. A. Sarma, D. J. Milstein, C. E. Vargas-Irwin, B. Franco, *et al.*, "Rapid calibration of an intracortical brain–computer interface for people with tetraplegia", *J. Neural Eng.*, vol. 15, no. 2, p. 026 007, 2018.

[129] K. Nazarpour and A. Jackson, "Biomimetic and biofeedback approaches for brain machine interface", *Proc. APSIPA ASC, Biopolis, Singapore*, 2010.

[130] A. H. Fagg, N. G. Hatsopoulos, V. De Lafuente, K. A. Moxon, S. Nemati, J. M. Rebesco, R. Romo, S. A. Solla, J. Reimer, D. Tkach, *et al.*, "Biomimetic brain machine interfaces for the control of movement", *J. Neurosci.*, vol. 27, no. 44, pp. 11 842–11 846, 2007.

[131] V. Gilja, P. Nuyujukian, C. A. Chestek, J. P. Cunningham, M. Y. Byron, J. M. Fan, M. M. Churchland, M. T. Kaufman, J. C. Kao, S. I. Ryu, *et al.*, "A high-performance neural prosthesis enabled by control algorithm design", *Nat. Neurosci.*, vol. 15, no. 12, p. 1752, 2012.

[132] E. Fetz and D. Finocchio, "Correlations between activity of motor cortex cells and arm muscles during operantly conditioned response patterns", *Exp. Brain Res.*, vol. 23, no. 3, pp. 217–240, 1975.

[133] A. Jackson and E. E. Fetz, "Interfacing with the computational brain", *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 19, no. 5, pp. 534–541, 2011.

[134] C. T. Moritz, S. I. Perlmutter, and E. E. Fetz, "Direct control of paralysed muscles by cortical neurons", *Nature*, vol. 456, no. 7222, p. 639, 2008.

[135] C. T. Moritz and E. E. Fetz, "Volitional control of single cortical neurons in a brain–machine interface", *J. Neural Eng.*, vol. 8, no. 2, p. 025 017, 2011.

[136] B. Engelhard, N. Ozeri, Z. Israel, H. Bergman, and E. Vaadia, "Inducing gamma oscillations and precise spike synchrony by operant conditioning via brain-machine interface", *Neuron*, vol. 77, no. 2, pp. 361–375, 2013.

[137] Y. Sakurai, K. Song, S. Tachibana, and S. Takahashi, "Volitional enhancement of firing synchrony and oscillation by neuronal operant conditioning: Interaction with neurorehabilitation and brain-machine interface", *Front. Syst. Neurosci.*, vol. 8, p. 11, 2014.

[138] R. Wahnoun, J. He, and S. I. H. Tillery, "Selection and parameterization of cortical neurons for neuroprosthetic control", *J. Neural Eng.*, vol. 3, no. 2, p. 162, 2006.

[139] S. M. Chase, A. B. Schwartz, and R. 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*, vol. 22, no. 9, pp. 1203–1213, 2009.

[140] R. D. Flint, J. M. Rosenow, M. C. Tate, and M. W. Slutzky, "Continuous decoding of human grasp kinematics using epidural and subdural signals", *J. Neural Eng.*, vol. 14, no. 1, p. 016 005, 2016.

[141] E. A. Pohlmeyer, S. A. Solla, E. J. Perreault, and L. E. Miller, "Prediction of upper limb muscle activity from motor cortical discharge during reaching", *J. Neural Eng.*, vol. 4, no. 4, p. 369, 2007.

[142] R. E. Kalman, "A new approach to linear filtering and prediction problems", *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, 1960.

[143] W. Wu and N. G. Hatsopoulos, "Real-time decoding of nonstationary neural activity in motor cortex", *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 16, no. 3, pp. 213–222, 2008.

[144] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation", *Proc. IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[145] A. E. Brockwell, A. L. Rojas, and R. Kass, "Recursive bayesian decoding of motor cortical signals by particle filtering", *J. Neurophysiol.*, vol. 91, no. 4, pp. 1899–1907, 2004.

[146] S. Shoham, L. M. Paninski, M. R. Fellows, N. G. Hatsopoulos, J. P. Donoghue, and R. A. Normann, "Statistical encoding model for a primary motor cortical brain-machine interface", *IEEE Trans. Biomed. Eng.*, vol. 52, no. 7, pp. 1312–1322, 2005.

[147] W. Truccolo, U. T. Eden, M. R. Fellows, J. P. Donoghue, and E. N. Brown, "A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects", *J. Neurophysiol.*, vol. 93, no. 2, pp. 1074–1089, 2005.

[148] U. T. Eden, L. M. Frank, R. Barbieri, V. Solo, and E. N. Brown, "Dynamic analysis of neural encoding by point process adaptive filtering", *Neural Comput.*, vol. 16, no. 5, pp. 971–998, 2004.

[149] M. M. Shanechi, A. L. Orsborn, and J. M. Carmena, "Robust brain-machine interface design using optimal feedback control modeling and adaptive point process filtering", *PLOS Comput. Biol.*, vol. 12, no. 4, e1004730, 2016.

[150] M. M. Shanechi, A. L. Orsborn, H. G. Moorman, S. Gowda, S. Dangi, and J. M. Carmena, "Rapid control and feedback rates enhance neuroprosthetic control", *Nat. Commun.*, vol. 8, no. 1, pp. 1–10, 2017.

[151] S. Acharya, F. Tenore, V. Aggarwal, R. Etienne-Cummings, M. H. Schieber, and N. V. Thakor, "Decoding individuated finger movements using volume-constrained neuronal ensembles in the m1 hand area", *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 16, no. 1, pp. 15–23, 2008.

[152] D. Sussillo, P. Nuyujukian, J. M. Fan, J. C. Kao, S. D. Stavisky, S. Ryu, and K. Shenoy, "A recurrent neural network for closed-loop intracortical brain–machine interface decoders", *J. Neural Eng.*, vol. 9, no. 2, p. 026 027, 2012.

[153] D. Sussillo, S. D. Stavisky, J. C. Kao, S. I. Ryu, and K. V. Shenoy, "Making brain–machine interfaces robust to future neural variability", *Nat. Commun.*, vol. 7, p. 13 749, 2016.

[154] A. J. Brockmeier and J. C. Príncipe, "Decoding algorithms for brain–machine interfaces", in *Neural engineering*, Springer, 2013, pp. 223–257.

[155] J. C. Kao, S. D. Stavisky, D. Sussillo, P. Nuyujukian, and K. V. Shenoy, "Information systems opportunities in brain–machine interface decoders", *Proc. IEEE*, vol. 102, no. 5, pp. 666–682, 2014.

[156] Y. Gao, M. J. Black, E. Bienenstock, S. Shoham, and J. P. Donoghue, "Probabilistic inference of hand motion from neural activity in motor cortex", in *Adv. Neural Inf. Process. Syst.*, 2002, pp. 213–220.

[157] P. Nuyujukian, J. C. Kao, J. M. Fan, S. D. Stavisky, S. I. Ryu, and K. V. Shenoy, "Performance sustaining intracortical neural prostheses", *J. Neural Eng.*, vol. 11, no. 6, p. 066 003, 2014.

[158] P. Nuyujukian, J. M. Fan, J. C. Kao, S. I. Ryu, and K. V. Shenoy, "A high-performance keyboard neural prosthesis enabled by task optimization", *IEEE Trans. Biomed. Eng.*, vol. 62, no. 1, pp. 21–29, 2014.

[159] B. Jarosiewicz, A. A. Sarma, D. Bacher, N. Y. Masse, J. D. Simeral, B. Sorice, E. M. Oakley, C. Blabe, C. Pandarinath, V. Gilja, *et al.*, "Virtual typing by people with tetraplegia using a self-calibrating intracortical brain-computer interface", *Sci. Transl. Med.*, vol. 7, no. 313, 313ra179–313ra179, 2015.

[160] B. Jarosiewicz, A. A. Sarma, J. Saab, B. Franco, S. S. Cash, E. N. Eskandar, and L. R. Hochberg, "Retrospectively supervised click decoder calibration for self-calibrating point-and-click brain–computer interfaces", *J. Physiol. Paris*, vol. 110, no. 4, pp. 382–391, 2016.

[161] P. Nuyujukian, J. A. Sanabria, J. Saab, C. Pandarinath, B. Jarosiewicz, C. H. Blabe, B. Franco, S. T. Mernoff, E. N. Eskandar, J. D. Simeral, *et al.*, "Cortical control of a tablet computer by people with paralysis", *PloS one*, vol. 13, no. 11, 2018.

[162] A. B. Ajiboye, J. D. Simeral, J. P. Donoghue, L. R. Hochberg, and R. F. Kirsch, "Prediction of imagined single-joint movements in a person with high-level tetraplegia", *IEEE Trans. Biomed. Eng.*, vol. 59, no. 10, pp. 2755–2765, 2012.

[163] T. Milekovic, D. Bacher, A. A. Sarma, J. D. Simeral, J. Saab, C. Pandarinath, B. Yvert, B. L. Sorice, C. Blabe, E. M. Oakley, *et al.*, "Volitional control of single-electrode high gamma local field potentials by people with paralysis", *J. Neurophysiol.*, vol. 121, no. 4, pp. 1428–1450, 2019.

[164] S. Cherif, K. E. Cullen, and H. L. Galiana, "An improved method for the estimation of firing rate dynamics using an optimal digital filter", *J. Neurosci. Methods*, vol. 173, no. 1, pp. 165–181, 2008.

[165] P. Lánskỳ, R. Rodriguez, and L. Sacerdote, "Mean instantaneous firing frequency is always higher than the firing rate", *Neural Comput.*, vol. 16, no. 3, pp. 477–489, 2004.

[166] B. Yu, J. Cunningham, G Santhanam, S. Ryu, K. Shenoy, and M Sahani, "Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity", *J. Neurophysiol.*, vol. 102, no. 1, pp. 614–635, 2009.

[167] M. Nawrot, A. Aertsen, and S. Rotter, "Single-trial estimation of neuronal firing rates: From single-neuron spike trains to population activity", *J. Neurosci. Methods*, vol. 94, no. 1, pp. 81–92, 1999.

[168] M. P. Nawrot, C. Boucsein, V. R. Molina, A. Riehle, A. Aertsen, and S. Rotter, "Measurement of variability dynamics in cortical spike trains", *J. Neurosci. Methods*, vol. 169, no. 2, pp. 374–390, 2008.

[169] K. W. Latimer, J. L. Yates, M. L. Meister, A. C. Huk, and J. W. Pillow, "Single-trial spike trains in parietal cortex reveal discrete steps during decision-making", *Science*, vol. 349, no. 6244, pp. 184–187, 2015.

[170] A. Afshar, G. Santhanam, M. Y. Byron, S. I. Ryu, M. Sahani, and K. V. Shenoy, "Single-trial neural correlates of arm movement preparation", *Neuron*, vol. 71, no. 3, pp. 555–564, 2011.

[171] H. G. Rey, M. Ahmadi, and R. Q. Quiroga, "Single trial analysis of field potentials in perception, learning and memory", *Curr. Opin. Neurol.*, vol. 31, pp. 148–155, 2015.

[172] S. Shinomoto, "Estimating the firing rate", in *Analysis of Parallel Spike Trains*, Springer, 2010, pp. 21–35.

[173] S. Shinomoto, "Estimation of neuronal firing rate", in *Encyclopedia of Computational Neuroscience*, D. Jaeger and R. Jung, Eds., Springer, 2015, pp. 1148–1152.

[174] H. Shimazaki and S. Shinomoto, "Kernel bandwidth optimization in spike rate estimation", *J. Comput. Neurosci.*, vol. 29, no. 1-2, pp. 171–182, 2010.

[175] C. Loader, *Local regression and likelihood*. Springer, 1999.

[176] I. DiMatteo, C. R. Genovese, and R. E. Kass, "Bayesian curve-fitting with free-knot splines", *Biometrika*, pp. 1055–1071, 2001.

[177] R. Barbieri, M. C. Quirk, L. M. Frank, M. A. Wilson, and E. N. Brown, "Construction and analysis of non-poisson stimulus-response models of neural spiking activity", *J. Neurosci. Methods*, vol. 105, no. 1, pp. 25–37, 2001.

[178] R. E. Kass, V. Ventura, and E. N. Brown, "Statistical issues in the analysis of neuronal data", *J. Neurophysiol.*, vol. 94, no. 1, pp. 8–25, 2005.

[179] R. Kass, "Adaptive spline smoothing of neural data", in *Neural Signal Processing: Quantitative Analysis of Neural Activity*, P. P. Mitra, Ed., Society for Neuroscience, 2008, pp. 35–42.

[180] B. E. Hansen, *Lecture notes on nonparametrics*, 2009.

[181] T. Cheng, J. Gao, and X. Zhang, "Nonparametric localized bandwidth selection for kernel density estimation", *Economet. Rev.*, pp. 1–30, 2018.

[182] B. W. Silverman, *Density estimation for statistics and data analysis*. CRC press, 1986, vol. 26.

[183] S. J. Sheather and M. C. Jones, "A reliable data-based bandwidth selection method for kernel density estimation", *J. R. Stat. Soc. Series B Stat. Methodol.*, pp. 683–690, 1991.

[184] M. Rudemo, "Empirical choice of histograms and kernel density estimators", *Scand. J. Stat.*, pp. 65–78, 1982.

[185] A. W. Bowman, "An alternative method of cross-validation for the smoothing of density estimates", *Biometrika*, vol. 71, no. 2, pp. 353–360, 1984.

[186] N.-B. Heidenreich, A. Schindler, and S. Sperlich, "Bandwidth selection for kernel density estimation: A review of fully automatic selectors", *Adv. Stat. Anal.*, vol. 97, no. 4, pp. 403–433, 2013.

[187] V. Ventura, R. Carta, R. E. Kass, S. N. Gettner, and C. R. Olson, "Statistical analysis of temporal evolution in single-neuron firing rates", *Biostatistics*, vol. 3, no. 1, pp. 1–20, 2002.

[188] O. Kornienko, "Neural representations and decoding with optimized kernel density estimates", PhD thesis, University of Heidelberg, Germany, 2015.

[189] J. P. Cunningham, M. Y. Byron, K. V. Shenoy, and M. Sahani, "Inferring neural firing rates from spike trains using gaussian processes", in *Adv. Neural Inf. Process. Syst.*, 2008, pp. 329–336.

[190] T. Shimokawa and S. Shinomoto, "Estimating instantaneous irregularity of neuronal firing", *Neural Comput.*, vol. 21, no. 7, pp. 1931–1951, 2009.

[191] E. N. Brown, R. Barbieri, V. Ventura, R. E. Kass, and L. M. Frank, "The time-rescaling theorem and its application to neural spike train data analysis", *Neural Comput.*, vol. 14, no. 2, pp. 325–346, 2002.

[192] T. Shimokawa, S. Koyama, and S. Shinomoto, "A characterization of the time-rescaled gamma process as a model for spike trains", *J. Comput. Neurosci.*, vol. 29, no. 1-2, pp. 183–191, 2010.

[193] P. G. Moschopoulos, "The distribution of the sum of independent gamma random variables", *Ann. I. Stat. Math.*, vol. 37, no. 1, pp. 541–544, 1985.

[194] D. Fink, "A compendium of conjugate priors", 1997.

[195] A. Z. Zambom and R. Dias, "A review of kernel density estimation with applications to econometrics", *Int. Econom. Rev.*, vol. 5, no. 1, pp. 20–42, 2013.

[196] R. E. Kass, V. Ventura, and C. Cai, "Statistical smoothing of neuronal data", *Network Comput. Neural Syst.*, vol. 14, no. 1, pp. 5–16, 2003.

[197] R. E. Kass, U. T. Eden, and E. N. Brown, *Analysis of neural data*. Springer, 2014, vol. 491.

[198] D. S. Soteropoulos and S. N. Baker, "Quantifying neural coding of event timing", *J. Neurophysiol.*, vol. 101, no. 1, pp. 402–417, 2009.

[199] E. N. Brown, R. Barbieri, U. T. Eden, and L. M. Frank, "Likelihood methods for neural spike train data analysis", in *Computational neuroscience: A comprehensive approach*, J. Feng, Ed., CRC Press London, 2003, pp. 253–286.

[200] K. Takiyama and M. Okada, "Detection of hidden structures in nonstationary spike trains", *Neural Comput.*, vol. 23, no. 5, pp. 1205–1233, 2011.

[201] Y. W. Teh and V. Rao, "Gaussian process modulated renewal processes", in *Adv. Neural Inf. Process. Syst.*, 2011, pp. 2474–2482.

[202] M. E. Mazurek and M. N. Shadlen, "Limits to the temporal fidelity of cortical spike rate signals", *Nat. Neurosci.*, vol. 5, no. 5, p. 463, 2002.

[203] Y. Mochizuki and S. Shinomoto, "Analog and digital codes in the brain", *Phys. Rev. E*, vol. 89, no. 2, p. 022 705, 2014.

[204] G. Wallstrom, J. Liebner, and R. E. Kass, "An implementation of bayesian adaptive regression splines (bars) in c with s and r wrappers", *J. Stat. Softw.*, vol. 26, no. 1, p. 1, 2008.

[205] K. H. Britten, M. N. Shadlen, W. T. Newsome, and J. A. Movson, *Responses of single neurons in macaque MT/V5 as a function of motion coherence in stochastic dot stimuli*, nsa2004.1, The Neural Signal Archive, http://www.neuralsignal.org/, 2004.

[206] K. H. Britten, M. N. Shadlen, W. T. Newsome, and J. A. Movshon, "The analysis of visual motion: A comparison of neuronal and psychophysical performance", *J. Neurosci.*, vol. 12, no. 12, pp. 4745–4765, 1992.

[207] D. L. Menzer, H. Bokil, J. W. Ryou, N. D. Schiff, K. P. Purpura, and P. P. Mitra, "Characterization of trial-to-trial fluctuations in local field potentials recorded in cerebral cortex of awake behaving macaque", *J. Neurosci. Methods*, vol. 186, no. 2, pp. 250–261, 2010.

[208] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey", *IEEE Trans. Neural Syst. Rehabil. Eng*, vol. 28, no. 10, pp. 2222–2232, 2017.

[209] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "TensorFlow: A system for large-scale machine learning", in *Proc. USENIX Symp. OS Design Impl. (OSDI)*, 2016, pp. 265–283.

[210] J. E. O'doherty, M. M. B. Cardoso, J. G. Makin, and P. N. Sabes, *Nonhuman primate reaching with multichannel sensorimotor cortex electrophysiology*, Zenodo `http://doi.org/10.5281/zenodo.583331`, 2017.

[211] J. G. Makin, J. E. O'Doherty, M. M. Cardoso, and P. N. Sabes, "Superior arm-movement decoding from cortex with a new, unsupervised-learning algorithm", *J. Neural Eng.*, vol. 15, no. 2, p. 026 010, 2018.

[212] S. Hochreiter and J. Schmidhuber, "Long short-term memory", *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[213] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, and D. D. Cox, "Hyperopt: A python library for model selection and hyperparameter optimization", *Comput. Sci. Discovery*, vol. 8, no. 1, p. 014 008, 2015.

[214] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling", *arXiv preprint arXiv:1803.01271*, 2018.

[215] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions", in *arXiv preprint arXiv:1511.07122*, 2016, pp. 1–13.

[216] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio", *arXiv preprint arXiv:1609.03499*, 2016.

[217] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", in *Proc. Int. Conf. Learn. Repr. (ICLR)*, 2015, pp. 448–456.

[218] T. Milekovic, A. A. Sarma, D. Bacher, J. D. Simeral, J. Saab, C. Pandarinath, B. L. Sorice, C. Blabe, E. M. Oakley, K. R. Tringale, *et al.*, "Stable long-term BCI-enabled communication in ALS and locked-in syndrome using LFP signals", *J. Neurophysiol.*, vol. 120, no. 7, pp. 343–360, 2018.

[219] A. Trongnetrpunya, B. Nandi, D. Kang, B. Kocsis, C. E. Schroeder, and M. Ding, "Assessing granger causality in electrophysiological data: Removing the adverse effects of common signals via bipolar derivations", *Front. Syst. Neurosci.*, vol. 9, p. 189, 2016.

[220] N. W. Whitmore and S.-C. Lin, "Unmasking local activity within local field potentials (LFPs) by removing distal electrical signals using independent component analysis", *Neuroimage*, vol. 132, pp. 79–92, 2016.

[221] P. L. Nunez, R. Srinivasan, *et al.*, *Electric fields of the brain: the neurophysics of EEG*. Oxford University Press, USA, 2006.

[222] R. Hindriks, X. D. Arsiwalla, T. Panagiotaropoulos, M. Besserve, P. F. Verschure, N. K. Logothetis, and G. Deco, "Discrepancies between multi-electrode LFP and CSD phase-patterns: A forward modeling study", *Front. Neural Circuits*, vol. 10, p. 51, 2016.

[223] V. Shirhatti, A. Borthakur, and S. Ray, "Effect of reference scheme on power and phase of the local field potential", *Neural Comput.*, vol. 28, no. 5, pp. 882–913, 2016.

[224] D. Cohen and N. Tsuchiya, "The effect of common signals on power, coherence and granger causality: Theoretical review, simulations, and empirical analysis of fruit fly lfps data", *Front. Syst. Neurosci.*, vol. 12, p. 30, 2018.

[225] J. Osselton, "Bipolar, unipolar and average reference recording methods i: Mainly theoretical considerations", *Am. J. EEG Tech.*, vol. 6, no. 4, pp. 129–144, 1966.

[226] B. Hjorth, "An on-line transformation of EEG scalp potentials into orthogonal source derivations", *Electroencephalogr. Clin. Neurophysiol.*, vol. 39, no. 5, pp. 526–530, 1975.

[227] F. N. Wilson, F. D. Johnston, A. G. Macleod, and P. S. Barker, "Electrocardiograms that represent the potential variations of a single electrode", *Am. Heart J.*, vol. 9, no. 4, pp. 447–458, 1934.

[228] F. F. Offner, "The EEG as potential mapping: The value of the average monopolar reference", *Electroencephalogr. Clin. Neurophysiol.*, vol. 2, no. 2, pp. 213–214, 1950.

[229] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation", in *Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, 2014, 1724—1734.

[230] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling", in *NIPS Workshop on Deep Learning*, 2014.

[231] S. Ray and J. H. Maunsell, "Different origins of gamma rhythm and high-gamma activity in macaque visual cortex", *PLoS Biol.*, vol. 9, no. 4, 2011.

[232] M. E. Rule, C. Vargas-Irwin, J. P. Donoghue, and W. Truccolo, "Contribution of LFP dynamics to single-neuron spiking variability in motor cortex during movement execution", *Front. Syst. Neurosci.*, vol. 9, p. 89, 2015.

[233] G Schalk, J Kubanek, K. Miller, N. Anderson, E. Leuthardt, J. Ojemann, D. Limbrick, D Moran, L. Gerhardt, and J. Wolpaw, "Decoding two-dimensional movement trajectories using electrocorticographic signals in humans", *J. Neural Eng.*, vol. 4, no. 3, p. 264, 2007.

[234] A. Belitski, A. Gretton, C. Magri, Y. Murayama, M. A. Montemurro, N. K. Logothetis, and S. Panzeri, "Low-frequency local field potentials and spikes in primary visual cortex convey independent visual information", *J. Neurosci.*, vol. 28, no. 22, pp. 5696–5709, 2008.

[235] S. Zanos, T. P. Zanos, V. Z. Marmarelis, G. A. Ojemann, and E. E. Fetz, "Relationships between spike-free local field potentials and spike timing in human temporal cortex", *J. Neurophysiol.*, vol. 107, no. 7, pp. 1808–1821, 2011.

[236] S. Katzner, I. Nauhaus, A. Benucci, V. Bonin, D. L. Ringach, and M. Carandini, "Local origin of field potentials in visual cortex", *Neuron*, vol. 61, no. 1, pp. 35–41, 2009.

[237] D. Xing, C.-I. Yeh, and R. M. Shapley, "Spatial spread of the local field potential and its laminar variation in visual cortex", *J. Neurosci.*, vol. 29, no. 37, pp. 11 540–11 549, 2009.

[238] A. Dubey and S. Ray, "Spatial spread of local field potential is band-pass in the primary visual cortex", *J. Neurophysiol.*, vol. 116, no. 4, pp. 1986–1999, 2016.

[239] P. Berens, G. A. Keliris, A. S. Ecker, N. K. Logothetis, and A. S. Tolias, "Feature selectivity of the gamma-band of the local field potential in primate primary visual cortex", *Front. Neurosci.*, vol. 2, p. 37, 2008.

[240] Y. Kajikawa and C. E. Schroeder, "How local is the local field potential?", *Neuron*, vol. 72, no. 5, pp. 847–858, 2011.

[241] D. J. McFarland, L. M. McCane, S. V. David, and J. R. Wolpaw, "Spatial filter selection for eeg-based communication", *Electroencephalogr. Clin. Neurophysiol.*, vol. 103, no. 3, pp. 386–394, 1997.

[242] D. J. McFarland, "The advantages of the surface laplacian in brain–computer interface research", *Int. J. Psychophysiol.*, vol. 97, no. 3, pp. 271–276, 2015.

[243] D. Rathee, H. Raza, G. Prasad, and H. Cecotti, "Current source density estimation enhances the performance of motor-imagery-related brain–computer interface", *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 12, pp. 2461–2471, 2017.

[244] K. A. Ludwig, R. M. Miriani, N. B. Langhals, M. D. Joseph, D. J. Anderson, and D. R. Kipke, "Using a common average reference to improve cortical neuron recordings from microelectrode arrays", *J. Neurophysiol.*, vol. 101, no. 3, p. 1679, 2009.

[245] E. Drebitz, M. Haag, I. Grothe, S. Mandon, and A. K. Kreiter, "Attention configures synchronization within local neuronal networks for processing of the behaviorally relevant stimulus", *Front. Neural circuits*, vol. 12, 2018.

[246] A. D. Legatt, J. Arezzo, and H. G. Vaughan Jr, "Averaged multiple unit activity as an estimate of phasic changes in local neuronal activity: Effects of volume-conducted potentials", *J. Neurosci. Methods*, vol. 2, no. 2, pp. 203–217, 1980.

[247] R. Eckhorn, R. Bauer, W. Jordan, M. Brosch, W. Kruse, M. Munk, and H. Reitboeck, "Coherent oscillations: A mechanism of feature linking in the visual cortex?", *Biol. Cybern.*, vol. 60, no. 2, pp. 121–130, 1988.

[248] R. Eckhorn, A. M. Gail, A. Bruns, A. Gabriel, B. Al-Shaikhli, and M. Saam, "Different types of signal coupling in the visual cortex related to neural mechanisms of associative processing and perception", *IEEE Trans. Neural Netw.*, vol. 15, no. 5, pp. 1039–1052, 2004.

[249] H. Super and P. R. Roelfsema, "Chronic multiunit recordings in behaving animals: Advantages and limitations", *Prog. Brain Res.*, vol. 147, pp. 263–282, 2005.

[250] Y.-S. Choi, M. A. Koenig, X. Jia, and N. V. Thakor, "Quantifying time-varying multiunit neural activity using entropy-based measures", *IEEE Trans. Biomed. Eng.*, vol. 57, no. 11, pp. 2771–2777, 2010.

[251] E. Drebitz, B. Schledde, A. K. Kreiter, and D. Wegener, "Optimizing the yield of multiunit activity by including the entire spiking activity", *Front. Neurosci.*, vol. 13, p. 83, 2019.

[252] M Brosch, R Bauer, and R Eckhorn, "Stimulus-dependent modulations of correlated high-frequency oscillations in cat visual cortex.", *Cerebral cortex (New York, NY: 1991)*, vol. 7, no. 1, pp. 70–76, 1997.

[253] J. Bradbury, S. Merity, C. Xiong, and R. Socher, "Quasi-recurrent neural networks", in *Int. Conf. Learn. Repr. (ICLR)*, 2017, pp. 1–12.

[254] E. Stark and M. Abeles, "Predicting movement from multiunit activity", *J. Neurosci.*, vol. 27, no. 31, pp. 8387–8394, 2007.

[255] G. Sharma, N. Annetta, D. Friedenberg, T. Blanco, D. Vasconcelos, A. Shaikhouni, A. R. Rezai, and C. Bouton, "Time stability and coherence analysis of multiunit, single-unit and local field potential neuronal signals in chronically implanted brain electrodes", *Bioelectron. Med.*, vol. 2, no. 1, p. 63, 2015.

[256] A. K. Bansal, C. E. Vargas-Irwin, W. Truccolo, and J. P. Donoghue, "Relationships among low-frequency local field potentials, spiking activity, and three-dimensional reach and grasp kinematics in primary motor and ventral premotor cortices", *J. Neurophysiol.*, vol. 105, no. 4, pp. 1603–1619, 2011.

[257] K. D. Harris, R. Q. Quiroga, J. Freeman, and S. L. Smith, "Improving data quality in neuronal population recordings", *Nat. Neurosci.*, vol. 19, no. 9, p. 1165, 2016.

[258] Y. G. Maguire, M. G. Shapiro, T. R. Cybulski, J. I. Glaser, D. Amodei, P. B. Stranges, R. Kalhor, D. A. Dalrymple, D. Seo, E. Alon, *et al.*, "Physical principles for scalable neural recording", *Front. Comput. Neurosci.*, vol. 7, p. 137, 2013.

[259] P. Zhang, X. Ma, L. Chen, J. Zhou, C. Wang, W. Li, and J. He, "Decoder calibration with ultra small current sample set for intracortical brain–machine interface", *J. Neural Eng.*, vol. 15, no. 2, p. 026 019, 2018.

[260] A. Gibiansky, S. Arik, G. Diamos, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, "Deep voice 2: Multi-speaker neural text-to-speech", in *Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 2962–2970.

[261] M. Zhang, M. A. Schwemmer, J. E. Ting, C. E. Majstorovic, D. A. Friedenberg, M. A. Bockbrader, W. J. Mysiw, A. R. Rezai, N. V. Annetta, C. E. Bouton, *et al.*, "Extracting wavelet based neural features from human intracortical recordings for neuroprosthetics applications", *Bioelectron. Med.*, vol. 4, no. 1, p. 11, 2018.

[262] A Moran and I Bar-Gad, "Revealing neuronal functional organization through the relation between multi-scale oscillatory extracellular signals", *J. Neurosci. Methods*, vol. 186, no. 1, pp. 116–129, 2010.

[263] G. Buzsáki, "Neural syntax: Cell assemblies, synapsembles, and readers", *Neuron*, vol. 68, no. 3, pp. 362–385, 2010.

[264] R. Storchi, A. G. Zippo, G. C. Caramenti, M. Valente, and G. E. Biella, "Predicting spike occurrence and neuronal responsiveness from LFPs in primary somatosensory cortex", *PloS one*, vol. 7, no. 5, e35850, 2012.

[265] S. P. Burns, D. Xing, and R. M. Shapley, "Comparisons of the dynamics of local field potential and multiunit activity signals in macaque visual cortex", *J. Neurosci.*, vol. 30, no. 41, pp. 13 739–13 749, 2010.

[266] J. R. Manning, J. Jacobs, I. Fried, and M. J. Kahana, "Broadband shifts in local field potential power spectra are correlated with single-neuron spiking in humans", *J. Neurosci.*, vol. 29, no. 43, pp. 13 613–13 620, 2009.

[267] M. J. Rasch, A. Gretton, Y. Murayama, W. Maass, and N. K. Logothetis, "Inferring spike trains from local field potentials", *J. Neurophysiol.*, vol. 99, no. 3, pp. 1461–1476, 2008.

[268] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library", in *Adv. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.

[269] T. Brochier, L. Zehl, Y. Hao, M. Duret, J. Sprenger, M. Denker, S. Grün, and A. Riehle, *Massively parallel recordings in macaque motor cortex during an instructed delayed reach-to-grasp task*, G-Node `http://doi.org/10.12751/g-node.f83565`, 2018.

[270] T. Brochier, L. Zehl, Y. Hao, M. Duret, J. Sprenger, M. Denker, S. Grün, and A. Riehle, "Massively parallel recordings in macaque motor cortex during an instructed delayed reach-to-grasp task", *Scientific data*, vol. 5, p. 180 055, 2018.

[271] L. Breiman, "Random forests", *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[272] A. Fisher, C. Rudin, and F. Dominici, "All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously", *J. Mach. Learn. Res.*, vol. 20, no. 177, pp. 1–81, 2019.

[273] S. Raschka, "MLxtend: Providing machine learning and data science utilities and extensions to python's scientific computing stack", *J. Open Source Softw.*, vol. 3, no. 24, p. 638, 2018.

[274] A. Belitski, S. Panzeri, C. Magri, N. K. Logothetis, and C. Kayser, "Sensory information in local field potentials and spikes from visual and auditory cortices: Time scales and frequency bands", *J. Comput. Neurosci.*, vol. 29, no. 3, pp. 533–545, 2010.

[275] E. E. Galindo-Leon and R. C. Liu, "Predicting stimulus-locked single unit spiking from cortical local field potentials", *J. Comput. Neurosci.*, vol. 29, no. 3, pp. 581–597, 2010.

[276] A. Mazzoni, K. Whittingstall, N. Brunel, N. K. Logothetis, and S. Panzeri, "Understanding the relationships between spike rate and delta/gamma frequency bands of LFPs and EEGs using a local cortical network model", *Neuroimage*, vol. 52, no. 3, pp. 956–972, 2010.

[277] S. W. Smith *et al.*, *The scientist and engineer's guide to digital signal processing*. California Technical Publishing, 1997.

[278] A. Gunduz, P. Brunner, M. Sharma, E. C. Leuthardt, A. L. Ritaccio, B. Pesaran, and G. Schalk, "Differential roles of high gamma and local motor potentials for movement preparation and execution", *Brain-Computer Interfaces*, vol. 3, no. 2, pp. 88–102, 2016.

[279] N. K. Logothetis, "The neural basis of the blood–oxygen–level–dependent functional magnetic resonance imaging signal", *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, vol. 357, no. 1424, pp. 1003–1037, 2002.

[280] A. Jackson and T. M. Hall, "Decoding local field potentials for neural interfaces", *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 10, pp. 1705–1714, 2016.

[281] M. Murshed, C. Murphy, D. Hou, N. Khan, G. Ananthanarayanan, and F. Hussain, "Machine learning at the network edge: A survey", *arXiv preprint arXiv:1908.00080*, 2019.

[282] R. Society, *iHuman: blurring lines between mind and machine*. Royal Society, 2019.