DEALING WITH UNKNOWN UNKNOWNS

By

Gaurav Gupta

A Dissertation Presented to the
FACULTY OF THE USC GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(ELECTRICAL ENGINEERING)

May 2022

In the beginning was simplicity.

–*Richard Dawkins*

To my parents.

ACKNOWLEDGEMENTS

Before everything else, I would like to give my sincerest thanks to my Ph.D. advisor, mentor, and friend Paul Bogdan. He has been nothing but supportive since the first meeting I had with him before joining his research group in Sept'15. One of the important qualities which I have admired of Paul and tried to adopt is to always think out of the box and solve problems taking a non-conventional approach. The constant motivation and 'green signal' from Paul have encouraged me to never stop exploring new problems which to large extent is the reason for my inter-disciplinary thesis work. He has always pushed me to go a step further and be the backbone in the process. A remarkable thing about Paul is that he is very approachable. I never had the feeling of communication gap throughout these Ph.D. years. I can always approach him or email him the snapshots of the equations and we just start from there. I was very fortunate to have been advised by such an enthusiastic and hard-working individual. A thing about research work which I have realized is to keep working, and chin-up even and especially when we have constant failures. I have always kept your quote 'they shut the door, we open the window' with me and will live it through my life. In the bad times, Paul has given me emotional support like a friend and a family member. I will always remember the conversations we had while taking walks, casual and comforting talks during lunches, phone calls, and trips.

I would like to thank Sérgio Pequito who happened to jump-start my research work along with Paul in my initial Ph.D. years. Sergio has helped me in shaping the early part of my work, made me practice multiple times during the conferences for the talk, and gave valuable feedback on the writing, and sat patiently over the Skype calls listening to every single step of the equations. He is very energetic and I will always remember the day of May'17 when we aimed to write the paper in a night before the deadline and you were driving but stopping at every gas station to give me feedback on the draft throughout your journey. I would also like to thank my committee members Prof. Jyotirmoy Deshmukh, Prof. Antonio Ortega, and Prof. José M.F. Moura for their valuable feedback on my thesis work. I thank my Quals committee members Prof. Bhaskar Krishnamachari, Prof. Ashutosh Nayyar, and Prof. Ruslan Salakhutdinov for their valuable suggestions which led to improvements in my dissertation.

Before joining grad school, I already had the opportunity to start research work in my undergrad with Prof. Ajit Chaturvedi. He crafted my research potential and motivated me to join grad school and pursue Ph.D. studies. I cannot thank him enough for being a stepping stone towards my research career. I would also like to thank some of the most talented and smart individuals I have met during my Ph.D. studies. I had the opportunity to work with Prof. James Boedicker, Prof. Jyotirmoy Deshmukh, Prof. Justin Rhodes, Prof. Roozbeh Kiani, and Prof. Radu Balan. Apart from the research feedback, I have seen myself evolving in technical writing/speaking

skills throughout these years. I would like to dedicate this learning process to all of the received feedback from my mentors and collaborators as I improved myself by observing their style and being critical of the feedback.

The research group at USC has provided me the opportunity to meet some of the most wonderful and hard-working people. While the group got bigger over time, but Yuankun holds a special place. As a senior to me and a friend, he has been a constant source of inspiration as he set the bar already high. He still is a go-to person to me in all matters of inquiry and concerns. I would like to thank Ridha with whom I have spent numerous nights at our lab, walking the streets at midnight discussing ideas, working on the paper, and catching deadlines. I thank Valeriu for being my labmate, as most of the times only we two were present in the office especially on the weekends. I would like to thank Xiongye, Chenzhong, Yao, and Ruochen with whom I have had the opportunity to work and collaborate. I would also like to thank my other labmates Emily, Mingxi, Panagiotis, and Jayson with whom I had some wonderful discussions over our lengthy group meetings and outside.

I fell in love with the place called USC the moment I first walked it in Sept'15. Throughout these years, USC has been my home as I have spent numerous days and nights over there. I would like to thank all of the workers at Verde, RTC, and Illys with whom I have made a friendly bond across this span of years. In the process of pushing forward my research to the next level, at times I also pushed my physical self resulting in health issues. I would like to thank all of the medical staff at USC as because of them I was able to get back on track quickly. I could not have hoped for a better roommate/friend than Mukul with whom I spent the majority of my free time, either having food, coffee, or walking the campus. I was also fortunate enough to have made friends at USC who have made my time memorable. I would like to thank Kartik, Saurav, Krishna, Thanos, and Martin for being there.

These long years of dedication would not have been possible without the unconditional support from my family. My parents have been nothing but encouraging since my childhood and they have been supportive of my idea of pursuing studies leaving my home country. While I cannot express enough gratitude with words towards my parents but they have been a continuous source of inspiration and were there with me in my ups and downs. I thank my annoying and lovely sister Prachi who has guarded the fort in my absence and brought all of the fun conversations over our family Skype calls. Lastly, I would like to thank Manisha who has been with me all the time, motivated me continuously when I was lacking, shared my sorrows, and taught me the meaning of celebration.

Finally, I would like to express my gratitude to the agencies who have contributed to the funding of my research, including National Science Foundation, Defense Advanced Research Projects Agency, and University of Southern California.

Contents

List of Tables

List of Figures

Abstract

Dealing with Unknown Unknowns

by

Gaurav Gupta

Doctor of Philosophy in Electrical Engineering

University of Southern California

Deciphering patterns from the data to make some inference rely on the assumption of knowledge of the complete system which is seldom true. The lack of complete knowledge could be due to the experimental device limitation, or due to phenomena not known. Even with big data sizes, the presence of unknown contributors may not be neglected due to complex interactions of the observed system with environment. For example, systems like brain, social networks, gene-regulatory networks, and physiological signals have interaction from unknown phenomena which is most of the times not expected. In the first part of this work, we study the incorporation of unknown unknowns (UUs) in the context of non-stationary non-Markovian processes. We then show the applications in the domain of brain electroencephalogram signals, spiking neuron networks, and model-based reinforcement learning. Using the concept of UUs, we also show applications in the brain imagined task prediction problem with a complete redesign of the existing approach.

Next, we will see that the data originating from the complex systems can be efficiently modeled using the partial differential equations (PDEs). In that pursuit, we will explore non-stationary fractional diffusion equations to model a variety of phenomena. As the complexity of underlying phenomena increase and with multi-dimensional data, obtaining solutions to the PDEs can be very time exhaustive and impractical. We propose a data-driven approach to solve PDEs by learning the solution operator with quick inference by a neural network forward pass. The proposed operator architecture is PDE agnostic, thus enables to learn without knowledge of the underlying phenomena. We demonstrate applications for various PDEs spanning domain of complex fluids, traffic flow, shallow water waves, and viscous fluid motion.

Finally, a fusion of complex networks and PDEs is studied in the context of discrete events data. To date, mathematical tools to decode network dynamics and structure from very scarce and partially observed neuronal spiking behavior remain underdeveloped. Large neuronal networks contribute to the intrinsic neuron transfer function

and observed neuronal spike trains encoding complex causal information processing, yet how this emerging causal fractal memory in the spike trains relates to the network topology is not fully understood. Towards this end, we have studied a novel statistical physics inspired neuron particle model that captures the causal information flow and processing features of neuronal spiking activity. Relying on synthetic comprehensive simulations and real-world neuronal spiking activity analysis, the proposed fractional order operators governing the neuronal spiking dynamics provide insights into the memory and scale of the spike trains as well as information about the topological properties of the underlying neuronal networks.

Chapter 1

Introduction

The machine learning methods by treating the data as finite-memory, stationary struggles to learn patterns from complex real-world sources which has non-Markovian, long-range memory components. Instead of assuming the data is random, we show that a lot of real-world signals possess structured patterns. Combining the efficiency of both machine learning and control systems, we discuss new ways to represent data streams in the presence of unknown unknowns (UUs) as well as decode new inferences out of them. Depending on the data format, the modeling strategies change as well as the definition of UUs relevant to the problem. For example, while modeling the continuous data streams recorded from brain with the aim of accurate modeling, the UUs evolve as contribution of artifacts, unobserved parts of the brain for accurate predictions of the observed phenomena. On the other hand, while modeling rare spiking events with the aim of explaining the observed phenomena, the UUs could be the gigantic network acting behind the curtains, and developing models to extract the information, for example, unknown network topology could be useful. Overall, different scenarios demand different modeling tools and the modeling interpretation of UUs change accordingly. In this thesis work, we take two different viewpoints to study various kinds of data-streams: (i) Complex networks and (ii) partial differential equations, and we show that both approaches go hand-in-hand and are useful in utilizing UUs for accurate modeling or obtaining inference.

1.1 Complex Networks for Time-varying Data

Time-varying complex networks (TVCNs) provide a comprehensive mathematical framework for modeling complex biological, social and technological systems. In this

work, we interpret complex systems (networks) (Barrat, Barthélemy, and Vespignani, 2008; Gao et al., 2014) as graphs comprising of nodes interacting spatially and temporally, i.e., both inter and intra dependence, with node activities available in the form of time-series. Examples include brain dynamic activity (Xue, Rodriguez, and Bogdan, 2016; Gupta, Pequito, and Bogdan, 2018b; Gupta, Pequito, and Bogdan, 2018d), the gene expression and interactions (Wang et al., 2016; Lopes, Cesar, and Costa, 2011), the bacteria dynamics (Arafa, 2013; Rihan et al., 2014; Koorehdavoudi et al., 2017), or the swarm robotics (Couceiro et al., 2012; Couceiro and Sivasundaram, 2016). Many such time varying complex (biological) networks exhibit complex spatio-temporal interactions. For instance, the short- and long-range interactions among neurons contribute to the emergence of long-range memory and fractional dynamics at macroscopic brain regions. Moreover, the non-stationarity which arises in most of the bio-physical processes require modeling techniques supporting interaction among variables in space and time. A computationally efficient strategy for constructing compact yet accurate mathematical models of TVCNs relies on describing the self-activity of nodes in TVCNs through fractional order operators (Moon, 1992; Lundstrom et al., 2008; Werner, 2010; Turcott and Teich, 1996; Thurner et al., 2003).

For the first part of this thesis, we are motivated by the recent success of fractional order dynamics in modeling spatiotemporal properties of physiological signals, such as electrocardiogram (ECG), electromyogram (EMG), electroencephalogram (EEG) and blood oxygenation level dependent (BOLD) just to mention a few (Magin, 2012; Baleanu, Machado, and Luo, 2011). Despite these modeling capabilities, there is one main limitation that continues to elude scientists and engineers alike. Specifically, complex networks such as the brain, whose nodes will dynamically evolve using fractal order dynamics, are often observed locally. Meaning that some of the dynamics assessed by the models are not only due to the local interaction, but might be constrained by unknown sources, i.e., stimuli that are external to the considered network. However, due to the experimental limitations, (e.g., resource constraint, limiting probing capabilities[1]), only a part of the complete CN is available at most of the times. Consequently, we propose a series of models that enable us to account for the existence of such unknown stimuli as well as latent nodes with some known properties, and determine the model that best captures the local dynamics under such stimuli. To be able to successfully model the limited finite-dimensional observed phenomena, we fix the degrees-of-freedom (DOFs) of the unknown contributors beforehand, and set it to a value less than the observed dimensions, as we discuss in Chapter 2. Observe that this enhances the analysis of these systems once we have an additional feature (i.e., the stimuli) that can be the main driver of a possible abnormal

[1] As Wigner argued once: 'It is the skill and ingenuity of the experimenter which show him phenomena which depend on a relatively narrow set of relatively easily realizable and reproducible conditions' (Wigner, 1960).

behavior of the network (Norden and Blumenfeld, 2002). In all such conditions, we have only partial information regarding the complete TVCN, and we wish to better predict the observed data in the presence of latent fractional dynamics as well as unknown drivers.

There is a rich literature regarding study of networks with latent nodes. To list a few, the importance of realizing the inclusion of latent nodes in the context of linear time invariant (LTI) systems has been explored in (Jalali and Sanghavi, 2012; Anandkumar et al., 2013; Geiger et al., 2015), for Bayesian networks in (Elidan, Nachman, and Friedman, 2007), and graphical models with Gaussian distribution of nodes in (Chandrasekaran, Parrilo, and Willsky, 2012), complex systems (Marsili, Mastromatteo, and Roudi, 2013). The Markovian assumption helps in the case of LTI systems, and in some sense to uncover the latent nodes. However, the LTI systems are not sufficient to accurately model physiological signals, such as EEG, ECG and BOLD (just to mention a few) due to their inability in capturing the long-range memory property of the biological signals. Continuing with the assumption of limited DOFs of the UUs, we further improve the model by taking a few fixed number of 'latent' nodes as fractional dynamics with only the knowledge of their fractional derivative coefficients. Through our experiments we show that this further improves the modeling prediction accuracy of the observed system. Next, we acknowledge that sensor placement, either in complete knowledge or in the latent setup, is crucial for the model estimation.

Apart from the electrical recording available in the form of EEG, understanding the microscopic brain activity (mechanisms) in action and in context is crucial for learning and control of the neuron behavior. Towards this end, the main purpose of studying neuron activities is to identify neuron history process and their inter-dependence in the ensemble neural system. The technique of multiple electrodes makes it possible to record and study the spiking activity of each neuron in a large ensemble system simultaneously (Wilson and McNaughton, 1993; Brown, 2005; Lewicki, 1998). At first, the study was mainly focusing on single neuron behavior and the bivariate relationship of neuron pairs or triplets, while ignoring the possible ensemble neuron effect (Krumin and Shoham, 2010; Brown, Kass, and Mitra, 2004; Okatan, Wilson, and Brown, 2005). Later, the multivariate auto-regressive framework was introduced with more complex neuronal connections. Multiple covariates affect the spiking activity of each neuron in the system (Kim et al., 2011). The most common covariates are the intrinsic, extrinsic and other parameters related to inputs and the environmental stimuli. Previous attempts have been made to analyze the impact of spiking history and the interaction with other neurons (Truccolo et al., 2005; Okatan, Wilson, and Brown, 2005; Kim et al., 2011).

From the mathematical perspective, the neuron activity and their spike trains are

stochastic in nature and their statistics are time-varying or non-stationary. Traditionally, the neuron activity is modeled by a discrete time series event of point processes (Karr, 2017; Brown, 2005; Brown, Kass, and Mitra, 2004). The likelihood method for multivariate point process is a central tool of statistical learning to model neuron spiking behaviors. The likelihood function is a variate of the parameters for the point process. These parameters are estimated from the experimental data with statistical tools (Brown, Kass, and Mitra, 2004). However, these models are concerned with the closed system assumption and does not involve the effects of the unknown external sources. The adoption of UUs is also applicable to the neural spiking system. Prior work on the neural activity modeling assume that all neurons in the system can be monitored and their activities are available for mathematical modeling (Okatan, Wilson, and Brown, 2005). In addition to the intrinsic and external covariates, we add the contribution of UUs covariates and provide a joint estimation method of the system parameters as well as UUs.

Finally, in the current work, we note that the sensor selection helps in not only reducing the deployed sensors but also helps in placing the sensors in the environment for better model estimation. The set functions chosen as objectives can have arbitrary structures that are problem-dependent. Nonetheless, in a quest to quantify sub-optimality in such discrete optimization problems, we often try to unveil structures (i.e., subclasses of functions with specific properties) that enable us to either develop efficient algorithms to determine the optimal solution, or to approximate the solutions when the problem is NP-hard with sub-optimality guarantees. Within this class, there was a surge for *submodular* functions, whose approximate solution can be determined by a *greedy algorithm* that determines a sub-optimal solution by recursively adding the element which maximizes the objective. Also, this algorithm is known to achieve a constant performance bound (Feige, Mirrokni, and Vondrak, 2011; Nemhauser et al., 1978; Sviridenko, 2004; Buchbinder et al., 2012), which can be improved by using the concept of curvature (Conforti and Cornuejols, 1984; Iyer, Jegelka, and Bilmes, 2013). Notwithstanding, some objectives of interest do not possess such properties (Bian et al., 2017), e.g., Bayesian A-optimality, determinantal function, subset selection in R^2 objective (Das and Kempe, 2011), and sparse approximation (Das and Kempe, 2008; Krause and Cevher, 2010). Regardless, it has been proposed to use the greedy algorithm, which empirically leads to a good performance. *Is this a coincidence, or are there implicit features that justify some of the empirical performances obtained?* This is the quest we also pursue in the present work by defining a class of approximate submodular functions, and then quantifying the performance guarantees using a deviation parameter. Next, we discuss the other possible modeling aspects where the inference about the UUs could be obtained by observing a limited phenomena.

1.2 Partial Differential Equations for Time-varying Data

In the next part of the thesis work, we will be exploring the partial differential equations (PDEs) frameworks for modeling and solving real-world problems. We discuss two aspects of it, namely, (i) modeling PDEs, and (ii) solving PDEs. In the modeling part, we developed a set of techniques to represent the observed phenomena using PDEs and then obtain inference about the underlying mechanism. Modeling the data using PDEs also enable us to proceed in the direction of UUs contribution, as we discuss in Section 1.3. For the solution part, we will present the novel deep learning techniques to solve the PDEs efficiently.

Many natural and human-built systems (e.g., aerospace, complex fluids, neuro-glia information processing) exhibit complex dynamics characterized by partial differential equations (PDEs) (Shlesinger, West, and Klafter, 1987; McKeown et al., 2020). For example, the design of wings and airplanes robust to turbulence, requires to learn complex PDEs. Along the same lines, complex fluids (gels, emulsions) are multiphasic materials characterized by a macroscopic behavior (Ovarlez et al., 2020) modeled by non-linear PDEs. Understanding their variations in viscosity as a function of the shear rate is critical for many engineering projects.

Despite the tremendous boost that is provided by the traditional machine learning (ML) and artificial intelligence (AI) for the analysis of static data through identifying the statistical interdependence between components of a system of interest, there is little to say about analyzing dynamical processes from big data and uncertainty quantification for large-scale complex systems. Specifically, ML has a limiting ability in deciphering the physical driving laws and governing equations from multimodal heterogeneous, scarce, and/or noisy time-series data associated with complex systems exhibiting multi-scale and multi-physics spatiotemporal evolution. These multi-scale and multi-physics spatiotemporal characteristics that occur in physics, biology, chemistry, neuroscience, and even geology, are usually encoded through (fractional or integer order) PDEs with possibly uncertain parameters. These PDEs are derived from conservation laws on energy, momentum, or electric charge (e.g., diffusion equation, Maxwell's equations, Navier-Stokes equations, Schrödinger equations). However, a plethora of complex systems from biology, neuroscience, or finance have numerous hidden interaction mechanisms, and the derivation of the PDEs describing their evolution is unknown. In the big data era, we witness new opportunities for data-driven discoveries of potentially new physical phenomena and new physics laws (or rules). Consequently, one may ask the following fundamental question: *Can we learn a PDE model from a given set of time-series measurements and perform accurate, efficient, and robust predictions using this learned model?* This question has

motivated researchers to develop methods for estimating PDE parameters using numerical solutions of PDEs (Fox and Nicholls, 2001; Müller and Timmer, 2002) (which requires careful parameterizations and high computational cost), Bayesian approaches (Xun et al., 2013), and a two-state approach (Liang and Wu, 2008; Bär, Hegger, and Kantz, 1999; Müller and Timmer, 2004; Voss et al., 1999) where the parameters of the PDEs are estimated via least squares. However, it has not been addressed that how to exploit the higher-order statistics of the measurements? Such measurements can characterize the rare events for a robust understanding of complex systems, and determining whether fractional or integer order PDEs together with their corresponding parameters govern the observations.

1.2.1 Modeling PDEs

Regarding modeling, diffusion is one of the fundamental mechanisms used for analyzing the transport of particles, and a common example of a diffusion process is the Brownian motion. Chaotic motion of a particle characterizes the latter process, and it can be modeled by a random walk such that the mean square displacement follows the diffusing scaling $\langle(\Delta X)^2\rangle \sim t$ (where $\langle . \rangle$ designates the mean). Furthermore, diffusion is a principal concept that explains many natural and scientific / technological phenomena (e.g., particles motion (Einstein, 1905), DNA and cellular processing (Barkai, Garini, and Metzler, 2012; Tolić-Nørrelykke et al., 2004; Jeon et al., 2011; Goychuk, 2015), microbial communities (Koorehdavoudi et al., 2017), brain activity (Papo, 2014), physiological complexity and cyber-physical systems modeling (Gupta, Pequito, and Bogdan, 2018b; Gupta, Pequito, and Bogdan, 2018c; Xue and Bogdan, 2017)), neuron spikes (Yang, Gupta, and Bogdan, 2019). The focus on analyzing complex systems led to studying anomalous diffusion (Gefen, Aharony, and Alexander, 1983; Klafter, Blumen, and Shlesinger, 1987; Bouchaud and Georges, 1990; Metzler and Klafter, 2000; Metzler, Glöckle, and Nonnenmacher, 1994; Metzler and Klafter, 2004; Klafter and Sokolov, 2005; Thiel, Flegel, and Sokolov, 2013; McKinley and Nguyen, 2018; Oliveira et al., 2019; Morgado et al., 2002; Metzler et al., 2014; Sokolov, 2012) to decipher complex system properties (e.g., long-range memory, higher-order correlations, ergodicity breaking measured as a discrepancy between the long time-averaged mean squared displacement and the ensemble-averaged mean squared displacement).

The anomalous diffusion has been shown to be able to describe complex fluid dynamics (Grmela and Öttinger, 1997; Cabreira et al., 2018), biological systems (Brangwynne et al., 2009; Palmieri et al., 2015; Lomholt, Ambjörnsson, and Metzler, 2005), transport (Klages, Radons, and Sokolov, 2008), dynamics in fractal structures (Mandelbrot, Freeman, and Company, 1983; Balankin, 2018; Bogdan and Marculescu, 2011), and economics (Scalas, 2006). Contrary to random walks processes describing classical

diffusion (e.g., Brownian motion), the particle possesses an internal memory that leads to a non-stationary motion, where the mean square displacement is heavy-tailed $\langle(\Delta X)^2\rangle \sim t^\beta$ (β is a parameter that is related to the memory of a particle).

1.2.2 Solving PDEs

From the solution side, we discuss an operator learning framework for solving the multi-dimensional, multi-disciplinary PDEs. Recent efforts on learning PDEs (i.e., mappings between infinite-dimensional spaces of functions), from trajectories of variables, focused on developing machine learning and in particular deep neural networks (NNs) techniques. Towards this end, a stream of work aims at parameterizing the solution map as deep NNs (Guo, Li, and Iorio, 2016; Zhu and Zabaras, 2018; Bhatnagar et al., 2019; Adler and Öktem, 2017; Khoo, Lu, and Ying, 2020). One issue, however, is that the NNs are tied to a specific resolution during training, and therefore, may not generalize well to other resolutions, thus, requiring retraining (and possible modifications of the model) for every set of discretizations. In parallel, another stream of work focuses on constructing the PDE solution function as a NN architecture (Raissi, Perdikaris, and Karniadakis, 2019; Greenfeld et al., 2019; Kochkov et al., 2021; Wang, Wang, and Perdikaris, 2021). This approach, however, is designed to work with one instance of a PDE and, therefore, upon changing the coefficients associated with the PDE, the model has to be re-trained. Additionally, the approach is not a complete data-dependent one, and hence, cannot be made oblivious to the knowledge of the underlying PDE structure. Finally, the closest stream of work to the problem we investigate is represented by the "Neural Operators" (Li et al., 2020b; Li et al., 2020c; Li et al., 2020a; Bhattacharya et al., 2020; Patel et al., 2021). Being a complete *data-driven* approach, the neural operators method aims at learning the operator map without having knowledge of the underlying PDEs. The neural operators have also demonstrated the capability of discretization-independence. Obtaining the data for learning the operator map could be prohibitively expensive or time consuming (e.g., aircraft performance to different initial conditions). To be able to better solve the problem of learning the PDE operators from scarce and noisy data, we would ideally explore fundamental properties of the operators that have implication in data-efficient representation.

Our intuition is to transform the problem of learning a PDE to a domain where a compact representation of the operator exists. With a mild assumption regarding the smoothness of the operator's kernel, except finitely many singularities, the multiwavelets (Alpert, 1993), with their *vanishing moments property*, sparsify the kernel in their projection with respect to (w.r.t.) a measure. Therefore, learning an operator kernel in the multiwavelet domain is feasible and data efficient. The wavelets have a rich history in signal processing (Daubechies, 1988; Daubechies, 1992), and

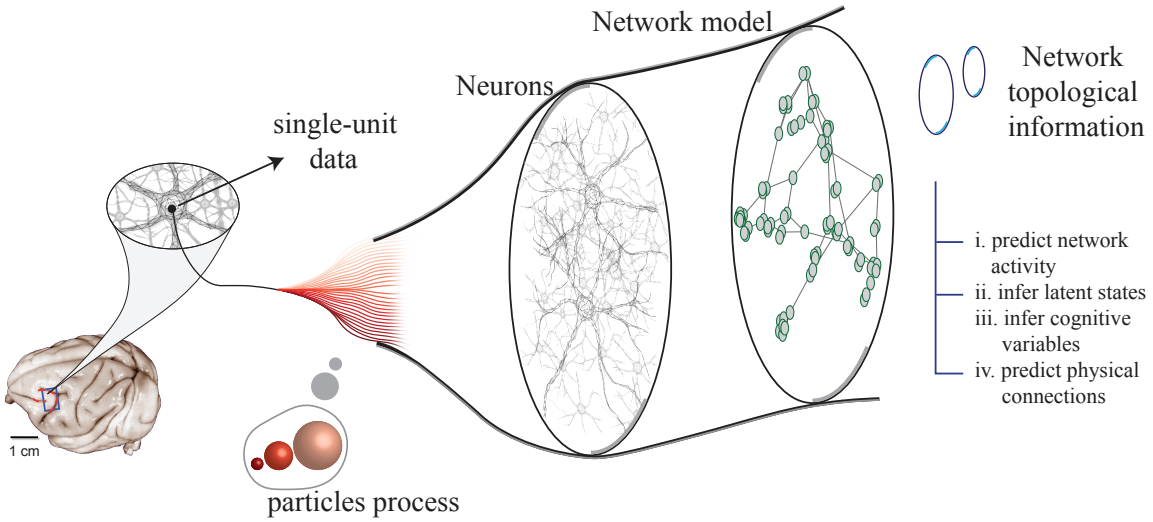


Figure 1.1: The particles process aims at extracting the high-dimensional information from a single-unit events data. The high-dimensional information is used to identify the network topological information. Additionally, the particles process can be used to perform the tasks as mentioned in (i)-(iv). In the present thesis work, the task (iii) is discussed as Monkey's choice prediction in Chapter 8.

are popular in audio, image compression (Balan et al., 2009; Silverman, Vassilicos, and Kingsbury, 1999). For multiwavelets, the orthogonal polynomial (OP) w.r.t. a measure emerges as a natural basis for the multiwavelet subspace, and an appropriate scale/shift provides a sequence of subspaces which captures the locality at various resolutions. We generalize and exploit the multiwavelets concept to work with arbitrary measures which opens-up new possibilities to design a series of models for the operator learning from complex data streams.

1.3 Intertwine of Complex Networks and PDEs

The concept of UUs can be directly embedded into the complex networks for the purpose of accurate modeling (see Chapter 2), and similarly PDEs can be used to extract information about the underlying unknown phenomena (see Chapter 4). In this section, we discuss the cross-relation of these two mathematical directions. We show that PDE modeling of the observed data can effectively be used to decipher some properties of the UUs in the complex networks format.

One of the general principles of most complex systems is that they consist of numerous structures or units with dynamic inter-dependence and interactions among them. The complex network interactions can yield unique output patterns, that can result in higher-order functional responses. For example, neural population interactions leading to thoughts and movements (Trousdale et al., 2012), social interactions in

cyber-terrorist networks (Johnson et al., 2019), ecological interactions between species (Fricke and Svenning, 2020), interactions between climate features (Réjou-Méchain et al., 2021). In practical settings, we only have partial knowledge of the structure, e.g., knowledge of only a fraction of the nodes in the complex network. Herein, we address the question of inferring the underlying network topology (a property of UUs to the limited observed spiking activities) using events recorded from multiple single nodes that make up only a small fraction of the total nodes in the network. In this pursuit, the network of neurons is a prominent example which is the primary case study in the current thesis.

The human brain functions through networks of interactions between billions of neurons, glia and vascular systems connected through a complex architecture (Bassett and Gazzaniga, 2011; Sporns, 2011; Bartheld, Bahney, and Herculano-Houzel, 2016). Together, these systems enable shifts in electrical activation patterns across brain networks, depending on sensory stimulation and intrinsic states which are shaped by past experience (Sporns, 2002). The view that all our perceptions, thoughts, emotions, and goal directed behaviors can ultimately be traced to the dynamic changes in the network activity within the brain is widely held (Bassett and Gazzaniga, 2011; Sporns, 2002; Bargmann and Marder, 2013; Borst and Theunissen, 1999; Kopell et al., 2014; Rolls and Treves, 2011; Taherkhani et al., 2020).

However, to date, no method is capable of measuring all the relevant features to decode the network dynamics into units of behavior. While technologies that analyze anatomical connections and proxies of neuronal activity across the entire brain such as magnetic resonance imaging (MRI) and functional MRI (fMRI) can be used to infer certain aspects of macroscopic (brain region) network dynamics, information is lost during the summation (fusion) that could be crucial, such as timing of intervals between spikes (Rolls and Treves, 2011; Petersen, Panzeri, and Diamond, 2001; Srivastava et al., 2017). While technologies for measuring action potentials or spikes from single neurons has advanced considerably in recent years (Pal and Tian, 2020; Banstola et al., 2020), even the best technologies can only measure hundreds to thousands of units usually only in a few restricted regions of the brain (Vázquez-Guardado et al., 2020) and providing very scarce information about the neuronal networks involved in the respective cognitive functions. Hence, there is a need for developing new mathematical methods capable of inferring features of the neuronal network from measuring only a small sample of single units (from a few to thousands) and relate the statistical characteristics of neuronal spiking activity with the overall brain functionality and behavior.

One of the major limitations of current statistical models of neuron spike data is that they do not account for the heterogeneity and non-Markovianity in the spike trains from individual units. We hypothesize that *neuronal spike trains could be modeled*

more precisely by fractional order partial differential equations. Further, we hypothesize that a more precise statistical representation of the spiking patterns that captures the fractal properties and long-range memory of the inter-spike intervals could uncover features about the underlying network topology and dynamics, and thereby provide new insight into how cognitive and sensory processes are represented in the brain. The need for non-Markovian fractal methods is recognized as an important open-problem in the neuroscience community (Breakspear, 2017). The current thesis work, by taking a novel fractional diffusion-based analogy for the neural spike trains, is a step towards addressing this challenge.

1.4 Thesis Contributions

The main problem considered in this thesis work is how unknown sources contribute towards the observed data, and how to observe their presence with minimal observations. Instead of treating them as noise, we show that under certain assumptions, the acknowledgement of their presence can be very helpful. By using some existing and then developing several new mathematical tools, we show that modeling efficiency improves a lot and a variety of new inferences can be made. The thesis contributions are divided as follows:

- In Chapter 2, we discuss the time-varying complex networks framework for multi-dimensional data streams with unknown unknowns. We take the path of most restrictive assumptions and then see how modeling evolves after relaxing them one by one. This work has appeared as Gupta, Pequito, and Bogdan (2018b), Gupta, Pequito, and Bogdan (2018d), and Gupta, Pequito, and Bogdan (2018c).
- In Chapter 3, the modeling of multi-dimensional discrete event-type data streams is considered. Without the complete knowledge assumption, a unknown sources version of modeling event streams is considered. The work has previously appeared as Yang, Gupta, and Bogdan (2019).
- In Chapter 4, a partial differential equations viewpoint is taken for the data. The anomalous diffusion provides generalization to model several complex phenomena. We discuss two efficient algorithms to estimate the fractional diffusion equation parameters from fewer data streams. The work has appeared as Znaidi et al. (2020).

- In Chapter 5, we take a new approach of 'Neural Operators' to solve the PDEs which is time-efficient. By learning the operator map in the multiwavelet domain, we show that the operator learning is efficient and more accurate. The work has appeared as Gupta, Xiao, and Bogdan (2021).

- In Chapter 6, we study an extension of the work in Chapter 2 of sensor selection. A new way of understanding the non-submodular cost functions is studied with performance guarantees trade-offs. The work has appeared as Gupta, Pequito, and Bogdan (2018a).

- In Chapter 7, a new model-based reinforcement learning method is considered using the developed fractional dynamics in Chapter 2. We show performance bounds of the non-Markovian RL and applications on diabetes simulator. The work has appeared as Gupta et al. (2021c).

- In Chapter 8, we discuss a fusion of complex networks and PDEs through a novel concept of neuron particles. The network topological inference is made using PDE parameters. A part of the ongoing work has appeard in Gupta et al. (2021b).

- Finally, we conclude the thesis and propose some future directions in Chapter 9.

Part I

Complex Networks

Chapter 2

Time-Varying Complex Networks

We first describe the time-varying complex network (TVCN) model having fractional order dynamical growth under unknown excitations. Next, building upon this model, we propose TVCN with latent fractional nodes and unknown unknowns.

2.1 TVCN with Unknown Unknowns

We consider a linear discrete time fractional-order dynamical model described as follows:

$$\begin{aligned} \Delta^{\alpha} x[k+1] &= Ax[k] + Bu[k] \\ y[k] &= Cx[k], \end{aligned} \tag{2.1}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^p$ is the unknown input and $y \in \mathbb{R}^n$ is the output vector. Also, we can describe the system by its matrices tuple (α, A, B, C) of appropriate dimensions. In what follows, we assume that the input size is always strictly less than the size of state vector, i.e., $p < n$. The difference between a classic linear time-invariant and the above model is the inclusion of fractional-order derivative whose expansion and discretization (Dzielinski and Sierociuk, 2005) for any ith state $(1 \leq i \leq n)$ can be written as

$$\Delta^{\alpha_i} x_i[k] = \sum_{j=0}^{k} \psi(\alpha_i, j) x_i[k-j], \tag{2.2}$$

where α_i is the fractional order corresponding to the ith state and $\psi(\alpha_i, j) = \frac{\Gamma(j-\alpha_i)}{\Gamma(-\alpha_i)\Gamma(j+1)}$ with $\Gamma(.)$ denoting the gamma function.

Having defined the system model, the system identification i.e. estimation of model parameters from the given data is an important step. It becomes nontrivial when we have unknown inputs since one has to be able to differentiate which part of the evolution of the complex network is due to its intrinsic dynamics and what is due to the unknown inputs. Subsequently, one of the first problems we need to address is that of system identification from the data, as described next.

2.1.1 Problem Formulation

The fractional-order dynamical model takes care of long-range memory which often is the property of many physiological signals. The estimation of the spatiotemporal parameters when there are no inputs to the system was addressed in (Xue, Rodriguez, and Bogdan, 2016). But as it happens, ignoring the inputs inherently assume that the system is isolated from the external surrounding. Hence, for a model to be able to capture the system dynamics, the inclusion of unknown inputs is necessary. Therefore, the first problem that we consider is as follows.

Problem-1: Given the measurements of all states across a time horizon $[t, t+T-1]$ of length T, we aim to estimate the model parameters (α, A) and the unknown inputs $\{u[k]\}_t^{t+T-2}$.

Notice that this would extend the work in (Xue, Rodriguez, and Bogdan, 2016) to include the case of unknown inputs. In fact, we will see in Section 2.1.2 that the proposed solution would result in a different set of model parameters.

2.1.2 Model Estimation

We consider the problem of estimating α, A, B and inputs $\{u[k]\}_t^{t+T-2}$ from the given limited observations $y[k]$, $k = [t, t+T-1]$, which due to the dedicated nature of sensing mechanism is same as $x[k]$. The realization of B can be application dependent, and hence instead of estimating, can also be provided using some experimental data. For the simplicity of notation, let us denote $z[k] = \Delta^{\alpha} x[k+1]$ with k chosen appropriately. The pre-factors in the summation in (2.2) grows as $\psi(\alpha_i, j) \sim \mathcal{O}(j^{-\alpha_i - 1})$ and, therefore, for the purpose of computational ease we have limited the summation in (2.2) to the first J values, where $J > 0$ is sufficiently large. Therefore, $z_i[k]$ can be written as

$$z_i[k] = \sum_{j=0}^{J-1} \psi(\alpha_i, j) x[k+1-j], \tag{2.3}$$

with the assumption that $x[k], u[k] = 0$ for $k \leq t-1$. Using the above introduced notations and the model definition in (2.1), the given observations are written as

$$z[k] = Ax[k] + Bu[k] + e[k], \tag{2.4}$$

where $e \sim \mathcal{N}(0, \Sigma)$ is assumed to be Gaussian noise independent across space and time. For simplicity, we have assumed that each noise component has same variance, i.e., $\Sigma = \sigma^2 I$. Also, let us denote the system matrices as $A = [a_1, a_2, \ldots, a_n]^T$ and $B = [b_1, b_2, \ldots, b_n]^T$. The vertical concatenated states and inputs during an arbitrary window of time as $X_{[t-1,t+T-2]} = [x[t-1], x[t], \ldots, x[t+T-2]]^T$, $U_{[t-1,t+T-2]} = [u[t-1], u[t], \ldots, u[t+T-2]]^T$ respectively, and for any ith state we have $Z_{i,[t-1,t+T-2]} = [z_i[t-1], z_i[t], \ldots, z_i[t+T-2]]^T$. For the sake of brevity, we would be dropping the time horizon subscript from the above matrices as it is clear from the context.

Since the problem of joint estimation of the different parameters is highly nonlinear, we proceed as follows: (i) we estimate the fractional order α using the wavelet technique described in (Flandrin, 1992); and (ii) with α known, the z in (2.3) can be computed. Therefore, the problem now reduces to estimate A, B and the inputs $\{u[k]\}_t^{t+T-2}$. Towards this goal, we propose the following sequential optimization algorithm similar to an expectation-maximization (EM) algorithm (McLachlan and Krishnan, 1996). Briefly, the EM algorithm is used for maximum likelihood estimation (MLE) of parameters subject to hidden variables. Intuitively, in our case, in Algorithm 1, we estimate A and B in the presence of hidden variables or *unknown unknowns* $\{u[k]\}_t^{t+T-2}$. Therefore, the 'E-step' is performed to average out the effects of unknown unknowns and obtain an estimate of u, where due to the diversity of solutions, we control the sparsity of the inputs (using the parameter λ'). Subsequently, the 'M-step' can then accomplish MLE estimation to obtain an estimate of A and B. The solution provided in (Xue, Rodriguez, and Bogdan, 2016) can be related to the proposed technique as follows.

Remark 2.1. *The solution to the system parameters (α, A, B) estimation without inputs (Xue, Rodriguez, and Bogdan, 2016) is a special case of the EM like approach proposed in the Algorithm 1.*

Proof. Upon setting $\{u[k]\}_t^{t+T-2} = 0$ in the E-step of the Algorithm 1, M-step would be the same at each iteration. Hence the algorithm stays at the initial point which is the solution in (Xue, Rodriguez, and Bogdan, 2016). □

It is worthwhile to mention the following result regarding EM algorithm.

Theorem 2.1 ((Dempster, Laird, and Rubin, 1977))**.** *The incomplete data (without unknown unknowns) likelihood $\mathbb{P}(z, x; A^{(l)}, B^{(l)})$ is non-decreasing after an EM iteration.*

Algorithm 1: EM algorithm

Input: $x[k], k \in [t, t+T-1]$

Output: A, B and $\{u[k]\}_t^{t+T-2}$

Initialize compute α using (Flandrin, 1992) and then $z[k]$. For $l = 0$, initialize $B^{(l)}$, and $A^{(l)}$ as

$$a_i^{(l)} = \arg\min_a ||Z_i - Xa||_2^2$$

repeat

(i) **'E-step'**: For $k \in [t, t+T-2]$ obtain $u[k]$ as

$$u[k] = \arg\min_u ||z[k] - A^{(l)}x[k] - B^{(l)}u||_2^2 + \lambda'||u||_1,$$

where $\lambda' = 2\sigma^2\lambda$;

(ii) **'M-step'**:

obtain $A^{(l+1)} = [a_1^{(l+1)}, a_2^{(l+1)}, \ldots, a_n^{(l+1)}]^T$ and $B^{(l+1)} = [b_1^{(l+1)}, b_2^{(l+1)}, \ldots, b_n^{(l+1)}]^T$ where

$$[a_i^{(l+1)\,T}, b_i^{(l+1)\,T}]^T = \arg\min_{a,b} ||Z_i - Xa - Ub||_2^2,$$

$l \leftarrow l + 1$;

until until converge;

Hence, the proposed algorithm being EM (detailed formulation in the Appendix A.1) has non-decreasing likelihood. Additionally, we have the following result about the incomplete data likelihood.

Proposition 1. *The incomplete data likelihood* $\mathbb{P}(z, x; A^{(l)}, B^{(l)})$ *is bounded at each iteration* l*.*

We can comment about the convergence of the Algorithm 1 as follows.

Lemma 2.1. *The Algorithm 1 is convergent in the likelihood sense.*

Proof. Using Theorem 2.1, Proposition 1 and Monotone Convergence Theorem, we can claim that the likelihood $\mathbb{P}(z, x; A^{(l)}, B^{(l)})$ will converge. □

It should be noted that convergence in likelihood does not always guarantee convergence of the parameters. But as emphasized in (Wu, 1983), from numerical viewpoint the convergence of parameters is not as important as convergence of the likelihood. Also the EM algorithm can converge to saddle points as exemplified in (McLachlan and Krishnan, 1996).

2.2 TVCN with Latent and Unknowns

We consider a time-varying complex network (TVCN) described by a linear discrete-time fractional-order model with latent nodes, which can be mathematically written as

$$\Delta^{\alpha} \begin{bmatrix} x[k+1] \\ z[k+1] \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x[k] \\ z[k] \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u[k] + \begin{bmatrix} e_1[k] \\ e_2[k] \end{bmatrix}, \tag{2.5}$$

where $x \in \mathbb{R}^n$ are the observed state variables, $z \in \mathbb{R}^m$ are the latent states, and $u \in \mathbb{R}^p$ are the unknown excitations. The system matrices $(\alpha, A_{11}, A_{12}, A_{21}, A_{22}, B_1, B_2)$ are of appropriate dimensions. The noise variables are assumed to be uncorrelated across observed and latent nodes with $e_1 \sim \mathcal{N}(0, \Sigma_1)$ and $e_2 \sim \mathcal{N}(0, \Sigma_2)$. The fractional-order derivative in equation (2.5) obeys the discrete form for every node, either observed or latent, as follows (Oldham and Spanier, 2006):

$$\Delta^{\alpha} x[k] = \sum_{j=0}^{k} \Psi_j^1 x[k-j], \quad \Delta^{\alpha} z[k] = \sum_{j=0}^{k} \Psi_j^2 z[k-j], \tag{2.6}$$

where the matrices $\Psi_j^1 = \text{diag}(\psi(\alpha_1^o, j), \ldots, \psi(\alpha_n^o, j))$ and $\Psi_j^2 = \text{diag}(\psi(\alpha_1^l, j), \ldots, \psi(\alpha_m^l, j))$ with $\psi(\alpha, j) = \frac{\Gamma(j-\alpha)}{\Gamma(-\alpha)\Gamma(j+1)}$, and $\Gamma(.)$ denotes the gamma function. The fractional-order coefficients corresponding to the ith node of observed and latent variables are denoted by α_i^o and α_i^l, respectively.

2.2.1 System Identification

The physiological signals, e.g. EEG and ECG, display spatio-temporal behavior, where the temporal component shows long-range memory dependence as realized in (Xue, Rodriguez, and Bogdan, 2016; Gupta, Pequito, and Bogdan, 2018b). As a consequence, properly modeled by the systems described in Section 2.1.2. When trying to obtain the system representation, i.e., to identify the system's parameters, the model estimation is contingent on the complete knowledge of the assumed complex-network dynamics; specifically, their nodes' activities in terms of time-series. Such assumptions may not hold in most of the cases where we are provided only with partial data. In such cases, to better predict the complex systems dynamics, it is beneficial to incorporate latent nodes. In this regard, the problem considered in this work can be stated as follows.

Problem Statement: *Given* the partial data (observed) $x[k]$ in terms of time-series across a time-horizon $k \in \{1, \ldots, N\}$, and knowledge of the fractional orders of latent nodes $\alpha_i^l (1 \leq i \leq m)$. *Estimate* the model parameters $(\alpha^o, A_{11}, A_{12}, A_{21}, A_{22}, B_1, B_2)$ and latent states $\{z[k]\}_1^{N-1}$, and the unknown inputs $\{u[k]\}_1^{N-1}$.

This problem will build upon the models in (Gupta, Pequito, and Bogdan, 2018b; Xue, Rodriguez, and Bogdan, 2016), but with striking difference of availability of only the partial data, and presence of latent nodes in the model. In contrast to (Gupta, Pequito, and Bogdan, 2018b), this work also relax the assumption of knowledge of the input matrices, and they will be computed as part of the system's parameters. Notice that we are implicitly assuming that the fractional-order coefficients are constant over time since these have been shown to be empirically slowly time-varying. We will see in the Section 2.3.2.2 that by considering latent nodes we can improve the prediction accuracy of the observed data. In the next section, we detail the assumptions required, and solution to this estimation problem.

2.2.2 Model Estimation

Due to notational convenience, we denote $x[k]$ as x_k. The model estimation procedure begins with, first making an estimate of the latent node activities with assumptions that some approximation of the system's parameters are known. A fractional Kalman filtering approach similar to (Sierociuk and Dzieliński, 2006) under Bayesian approximation is used for this purpose in Section 2.2.2.1. Subsequently, we will use the new data (estimated latent and the available data) to perform the system identification. In Section 2.2.2.2, we present an iterative algorithm to jointly estimate the system's parameters and the unknown unknowns.

2.2.2.1 Fractional Kalman filtering

The fractional-order Kalman filtering aims at estimation of the latent states at each kth time step with the available data. Using standard notations in the Kalman filtering (for the linear systems), we define the following estimates

$$\begin{aligned} \hat{z}_k &= \mathbb{E}[z_k | x_1, x_2, \ldots, x_{k+1}, u_1, \ldots, u_k], \text{and} \\ \tilde{z}_k &= \mathbb{E}[z_k | x_1, x_2, \ldots, x_k, u_1, \ldots, u_{k-1}]. \end{aligned} \tag{2.7}$$

In contrast to the Kalman filtering for classical linear system, we have the $\hat{z}_k$'s conditioned on the observed data from x_1 till x_{k+1}. The reasoning behind this can be quickly seen in the definition of system model in equation (2.5). In the classical linear system, the observations and latent activities are indexed at the same time. While in

$$\begin{bmatrix} A_{11}^{(t+1)\,T} \\ A_{12}^{(t+1)\,T} \\ B_{1}^{(t+1)\,T} \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{N} x_{k-1}x_{k-1}^{T} & \sum_{k=1}^{N} x_{k-1}\hat{z}_{k-1}^{(t)\,T} & \sum_{k=1}^{N} x_{k-1}u_{k-1}^{(t)\,T} \\ \sum_{k=1}^{N} \hat{z}_{k-1}^{(t)}x_{k-1}^{T} & \sum_{k=1}^{N} (\hat{P}_{k-1}^{(t)} + \hat{z}_{k-1}^{(t)}\hat{z}_{k-1}^{(t)\,T}) & \sum_{k=1}^{N} \hat{z}_{k-1}^{(t)}u_{k-1}^{T} \\ \sum_{k=1}^{N} u_{k-1}^{(t)}x_{k-1}^{T} & \sum_{k=1}^{N} u_{k-1}^{(t)}\hat{z}_{k-1}^{(t)\,T} & \sum_{k=1}^{N} u_{k-1}^{(t)}u_{k-1}^{(t)\,T} \end{bmatrix}^{-1} \times \begin{bmatrix} \sum_{k=1}^{N} x_{k-1}\mathring{x}_{k}^{T} \\ \sum_{k=1}^{N} \hat{z}_{k-1}^{(t)}\mathring{x}_{k}^{T} \\ \sum_{k=1}^{N} u_{k-1}^{(t)}\mathring{x}_{k}^{T} \end{bmatrix} \tag{2.8}$$

$$\begin{bmatrix} A_{21}^{(t+1)\,T} \\ A_{22}^{(t+1)\,T} \\ B_{2}^{(t+1)\,T} \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{N-1} x_{k-1}x_{k-1}^{T} & \sum_{k=1}^{N-1} x_{k-1}\hat{z}_{k-1}^{(t)\,T} & \sum_{k=1}^{N-1} x_{k-1}u_{k-1}^{(t)\,T} \\ \sum_{k=1}^{N-1} \hat{z}_{k-1}^{(t)}x_{k-1}^{T} & \sum_{k=1}^{N-1} (\hat{P}_{k-1}^{(t)} + \hat{z}_{k-1}^{(t)}\hat{z}_{k-1}^{(t)\,T}) & \sum_{k=1}^{N-1} \hat{z}_{k-1}^{(t)}u_{k-1}^{(t)\,T} \\ \sum_{k=1}^{N-1} u_{k-1}^{(t)}x_{k-1}^{T} & \sum_{k=1}^{N-1} u_{k-1}^{(t)}\hat{z}_{k-1}^{T\,(t)} & \sum_{k=1}^{N-1} u_{k-1}^{(t)}u_{k-1}^{T\,(t)} \end{bmatrix}^{-1} \times \begin{bmatrix} \sum_{k=1}^{N-1} x_{k-1}\mathring{z}_{k}^{(t)\,T} \\ \sum_{k=1}^{N-1} (\hat{P}_{k-1}^{(t)}\Psi_{1}^{2\,T} + \hat{z}_{k-1}^{(t)}\mathring{z}_{k}^{(t)\,T}) \\ \sum_{k=1}^{N-1} u_{k-1}^{(t)}\mathring{z}_{k}^{(t)\,T} \end{bmatrix} \tag{2.9}$$

the considered system model, with the observations being x_k and latent nodes being z_k, we can witness that the equation (2.5) relate the latent node activity z_k with observations till x_{k+1}. The Kalman filtering solutions can be mathematically intractable due to the complexities introduced by long-range dependence of the fractional operator Δ^{α}. We will resort to the Bayesian network assumption (as in (Gupta, Pequito, and Bogdan, 2018c)) for the latent state estimates which is described in the following lemma.

Lemma 2.2. *The Fractional-order Kalman filtering solution for the system described in (2.5) with the Bayesian Network assumption is written as*

$$\begin{aligned} \hat{z}_k &= \tilde{z}_k + K_k(y_k - A_{12}^{T}\tilde{z}_k), \\ \hat{P}_k &= (A_{12}^{T}\Sigma_1^{-1}A_{12} + \tilde{P}_k^{-1})^{-1}, \\ K_k &= \tilde{P}_k A_{12}^{T}(\Sigma_1 + A_{12}\tilde{P}_k A_{12}^{T})^{-1}, \\ y_k &= x_{k+1} + \sum_{j=0}^{k+1} \Psi_j^1 x_{k+1-j} - A_{11}x_k - B_1 u_k, \end{aligned}$$

$$
\begin{aligned}
\tilde{z}_k &= A_{22}\hat{z}_{k-1} + A_{21}x_{k-1} + B_2 u_{k-1} - \sum_{j=0}^{k} \Psi_j^2 \hat{z}_{k-j}, \\
\tilde{P}_k &= (A_{22} - \Psi_1^2)\hat{P}_{k-1}(A_{22} - \Psi_1^2)^T + \sum_{j=2}^{k} \Psi_j^2 \hat{P}_{k-j} \Psi_j^{2\,T} + \Sigma_2,
\end{aligned}
$$

where the conditional covariances are defined as $\hat{P}_k = \mathbb{E}[(z_k - \hat{z}_k)(z_k - \hat{z}_k)^T | x_1, \ldots, x_{k+1}, u_1, \ldots, u_k]$ and $\tilde{P}_k = \mathbb{E}[(z_k - \tilde{z}_k)(z_k - \hat{z}_k)^T | x_1, \ldots, x_k, u_1, \ldots, u_{k-1}]$.

Next, we present an algorithm to determine the system's parameters attaining maximum likelihood estimation.

2.2.2.2 Maximum Likelihood Estimation

The MLE estimate of the system's parameters has to be performed in the presence of latent variables. We propose to use an Expectation-Maximization (EM) like algorithm. The conditional distributions of the latent fractional nodes considered are as described in the Section 2.2.2.1. Moreover, we have unknown unknowns in our system u_k, and this work in contrast to (Gupta, Pequito, and Bogdan, 2018b) will make an estimation of u_k as well as the input matrices B_i. We define the EM algorithm to make an estimate of the latent fractional nodes activities and unknown unknowns jointly.

The EM update of the system's parameters at each iteration is performed via the following result.

Theorem 2.2. *An update of the system parameters used in (2.5) with given* $\{x_k\}_1^N$, $\{u_k^{(t)}\}_1^{N-1}$ *and the initial conditions* $x_0, z_0, u_0, \hat{P}_0$, *at each iteration index* t *is (2.8), (2.9) and*

$$
\Sigma_1^{(t+1)} = \frac{1}{N}\sum_{k=1}^{N}\left[(\mathring{x}_k - A_{11}^{(t+1)}x_{k-1} - A_{12}^{(t+1)}\hat{z}_{k-1}^{(t)} - B_1^{(t+1)}u_{k-1}^{(t)})\mathring{x}_k^T\right], \tag{2.10}
$$

$$
\begin{aligned}
\Sigma_2^{(t+1)} = \frac{1}{N-1}\sum_{k=1}^{N-1}\Bigg[&\hat{P}_k^{(t)} + \sum_{j=1}^{k}\Psi_j^2\hat{P}_{k-j}^{(t)}\Psi_j^{2\,T} + \mathring{z}_k^{(t)}\mathring{z}_k^{(t)\,T} - A_{21}^{(t+1)}x_{k-1}\mathring{z}_k^{(t)\,T} - A_{22}^{(t+1)}(\\
&\hat{P}_{k-1}^{(t)}\Psi_1^{2\,T} + \hat{z}_{k-1}^{(t)}\mathring{z}_k^{(t)\,T}) - B_2^{(t+1)}u_{k-1}^{(t)}\mathring{z}_k^{(t)\,T}\Bigg],
\end{aligned} \tag{2.11}
$$

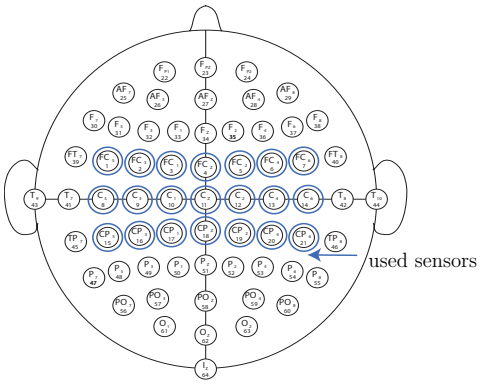


Figure 2.1: Sensor distribution for the measurement of EEG. The channel labels are shown with their corresponding number. For the model estimation with only unknown unknowns we use the full 64 sensors, while for latent nodes+unknown unknowns we used the indicated sensors in blue to reduce the dimensionality of the problem.

where $\mathring{x}_k = \sum_{j=0}^{k} \Psi_j^1 x_{k-j}$ and $\mathring{z}_k^{(t)} = \sum_{j=0}^{k} \Psi_j^1 \hat{z}_{k-j}^{(t)}$.

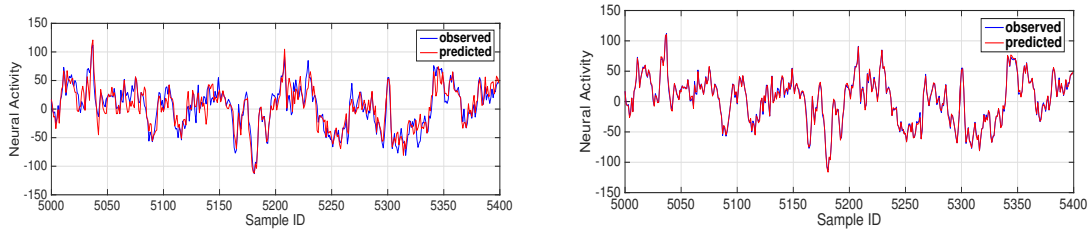
Now, using these results, we write the iterative algorithm for system's parameter estimation. Intuitively, the algorithm starts with an initial guess of the system parameters, and then at each step, obtain the latent node activities using the approach described in Section 2.2.2.1. Further, upon estimating the incurred error through unknown unknowns, an update of the system's parameters is obtained and the process is repeated until convergence. The mentioned steps are formally detailed as Algorithm 1.

2.3 Estimation Results

We demonstrate the application of the results derived in the previous sections on physiological signals. In particular we have taken a 64-channel EEG signal which records the brain activity of 109 subjects. The 64-channel/electrode distribution with the corresponding labels and numbers are shown in Figure 2.1. The subjects were asked to perform motor and imagery tasks. The data was collected by BCI2000 system with sampling rate of 160Hz (Schalk et al., 2004; Goldberger et al., 2000).

2.3.1 Unknown Unknowns

The parameters of the system model α, A and B, are estimated by the application of Algorithm 1. The performance of EM algorithm like any iterative algorithm is crucially dependent on its initial conditions. For the considered example of EEG dataset, it was observed that convergence of the algorithm is fast. Further, even few iterations, for example 5, was sufficient to reach the point of local maxima of



(a) (b)

Figure 2.2: Comparison of predicted EEG state for the channel C_1 using fractional-order dynamical model. The five step and one step predictions are shown in (2.2a) and (2.2b) respectively.

the likelihood. This shows that the choice of the initial point for EM algorithm is considerably good. The values predicted by the model in comparison with actual data are shown in Figure 2.2. The one step prediction follows very closely the actual data, but there is small mismatch in the five step prediction. The ratio of square root of mean squared error (RMSE) of the prediction by model with and without inputs (Xue et al., 2016) is shown in Figure 2.3 for total of 109 subjects. The RMSE is defined as

$$e_i = \sqrt{\sum_{k=1}^{N}(x_i[k] - \hat{x}_i[k])^2 \,/\, \sum_{k=1}^{N} x_i^2[k]}\,, \tag{2.8}$$

As observed, the error ratio is less than one-tenth in the case when unknown inputs is considered. It should be noted that the subjects for which the ratio is high (for example ID$-52, 72$) are the ones which have low RMSE for both the cases of with and without inputs, as shown in Figure 2.4. Hence, the ratio approaches closer to 0.5 for these cases. Therefore, the fractional-order dynamical model with unknown inputs fits the EEG data much better than the case of no inputs. In the next part, we will show the results for estimation procedure working with latent fractional nodes as well as unknown unknowns.

2.3.2 Latent fractional and Unknown Unknowns

2.3.2.1 Simulated Data

We consider a pedagogical fractional-order system with three nodes, and without unknown inputs with the following parameters:

$$A = \begin{bmatrix} 0 & 0.1 & 0.2 \\ -0.01 & -0.02 & 0.3 \\ 0.01 & -0.03 & -0.05 \end{bmatrix}, \alpha = \{0.7, 1.1, 0.8\}.$$

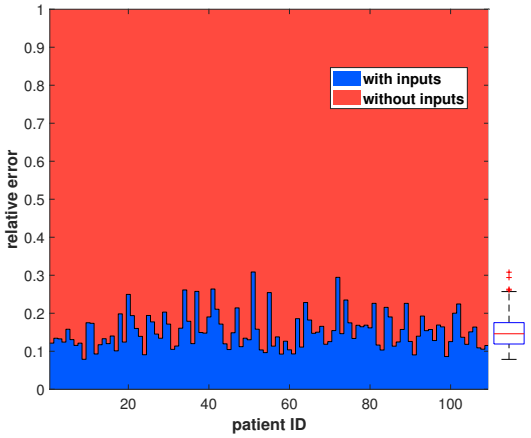


Figure 2.3: Error ratio for prediction using fractional-order dynamical model with and without inputs, and box plot for statistics of the ratio.

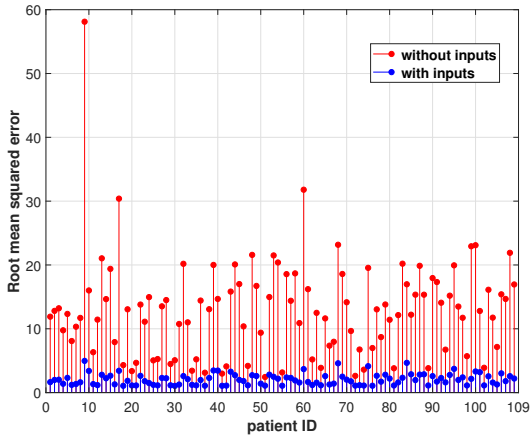


Figure 2.4: Root mean squared error for prediction using fractional-order dynamical model with and without inputs.

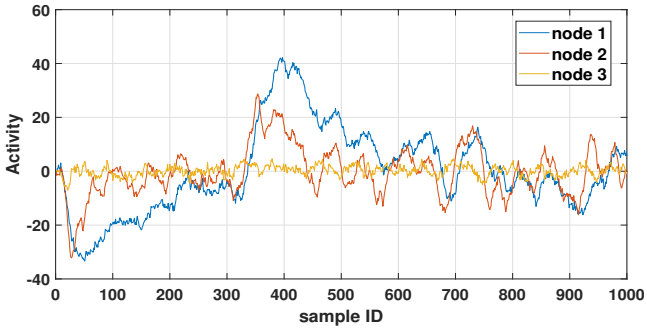


Figure 2.5: Simulated activities for all three nodes generated by selected model parameters.

For studying the latent node behavior, we remove one node and observe the rest, $n = 2$ and $m = 1$, from which we use to recover the time-series generated by the accessible nodes (i.e., the nodes that are not latent) using our proposed method (i.e., with latent variables) versus those previously used in the literature (Gupta, Pequito, and Bogdan, 2018b) (i.e., without latent variables). The five-step prediction error results are summarized in Table 2.1, where the relative error values are computed as in (2.8).

Observed		without latent		with latent	
2	3	13.40%	71.14%	7.94%	68.22%
1	3	22.12%	69.26%	21.12%	65.31%
1	2	11.44 %	28.92%	8.37%	21.03%

Table 2.1: A comparison of mean squared prediction error with and without using latent node model for various possibilities of observed nodes. The ith row corresponds to making node i as latent.

Observed (1→12)+	ϕ	13	13→14	13→15	13→16	13→17	13→18	13→19	13→20
Hidden	13→21	14→21	15→21	16→21	17→21	18→21	19→21	20→21	21
Without latent	10.51	10.44	10.97	11.71	13.51	13.77	15.29	16.03	14.56
With latent	6.07	7.20	6.53	8.49	11.36	10.50	13.18	13.18	13.16

(a)

Observed (1→12) +	ϕ	56	56→57	56→58	56→59	56→60	56→61	56→62	56→63
Hidden	56→64	57→64	58→64	59→64	60→64	61→64	62→64	63→64	64
Without latent	10.51	12.09	15.99	13.45	15.29	16.59	20.40	20.89	21.52
With latent	6.31	7.92	11.27	9.25	9.90	9.69	13.28	19.87	20.07

(b)

Table 2.2: Average prediction error (in %) with and without using latent model for two different set of sensors in (a) and (b). Each column has labeled hidden nodes, and observed nodes are union of $(1 \rightarrow 12)$ and the corresponding column entry. The total number of observed and latent nodes change by $n+1$ and $m-1$ from left to right.

where $\hat{x}_i[k]$ is the predicted value of the ith node at time k. The error percentage is consistently high for node 3, and the reason for this lies in the actual behavior of the node activity as seen in Figure 2.5. Specifically, node 3 activity –unlike other two nodes– stays very close to zero and vary frequently, which makes it difficult to use for accurate predictions using the proposed model. Next, we see the application on real-world EEG dataset.

2.3.2.2 Real-world data

The estimation of model's parameters, such as the coupling matrices A_{ij}, the input matrices B_i, and the latent states, is performed for the EEG data using Algorithm 1. The log likelihood converges with iterations as we observe in Figure 2.6. We notice that the choice of initial conditions can play a great role in fast convergence, as well as the accuracy of the results. If some previous knowledge is available (for example, through experiments) about the coupling matrices, then it can be used to achieve better results. In this work, the matrices are initialized to entries selected uniformly at random between -1 and 1. In the current experiment, we have used the data when the subject has executed 'both feet movement'. While performing model estimation, the use of relevant EEG sensors are required for having accurate predictions. Therefore, we used neuro-physiological evidence-based sensors that capture the behavior associated with the peripheral nervous system (i.e., the motor cortex) and labeled from 1 to 21, –see Figure 2.1. Specifically, different subregions in the motor region are activated when the feet move, so we have used this information to carefully reduce the number of sensors/nodes in our study.

The proposed latent model is tested in a comprehensive manner by performing the following steps: *(i)* first fixing the nodes to make prediction from sensor IDs 1 to 12 (denoted by $1 \rightarrow 12$); *(ii)* second, consecutively reveal new nodes to increase the total observed nodes dimension n (originally from 12) by one and decrease total latent node dimension m (from 9) by one, in each step. The reported error values are computed

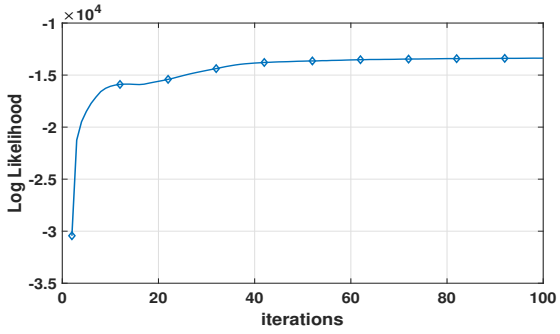


Figure 2.6: Log likelihood vs number of iterations for observed indices ($1 \rightarrow 12$) and hidden indices ($13 \rightarrow 21$) using Algorithm 1.

from equation (2.8), and are averaged across the fixed twelve observed nodes. In Table 2.2a we provide some evidence that the latent model with minimal information concerning fractional orders of the latent nodes perform better than without using any model on the unobserved nodes. We also observed the necessity of the relevant latent nodes, by considering the set of sensors which are placed on the region of brain least related to the undertaken situation of 'both feet movement'. The same experiment is repeated to predict the activities of fixed nodes in consideration $1 \rightarrow 12$ and varying the total observed/latent nodes, but this time from a set of sensor IDs $\{56, \ldots, 64\}$. The prediction error values are reported in Table 2.2b. We notice that the error values are higher upon revealing nodes from the set $\{56, \ldots, 64\}$. This raises an important and intuitive point that, revealing/hiding time-series that have less relation to the data under consideration are very likely to increase inaccuracies in the model.

The experiments, both simulated and with real EEG data, provided evidence that the inclusion of latent model is helpful in the context of the accuracy of the retrieved model and prediction accuracies.

2.4 Brain Machine Interfaces

Brain interfaces aim to address the following problem.

Is it possible to classify a specific cognitive state, e.g., motor task or its imagination, by using measurements collected with a specific sensing technology that harvest information about brain activity?

In this Section, we revisit this problem in the context of brain-computer interfaces (BCI), when dealing with EEG-based noninvasive brain interfaces. Towards this goal, propose to use a systems' perspective that enables to enhance the BCI (see Figure 2.7

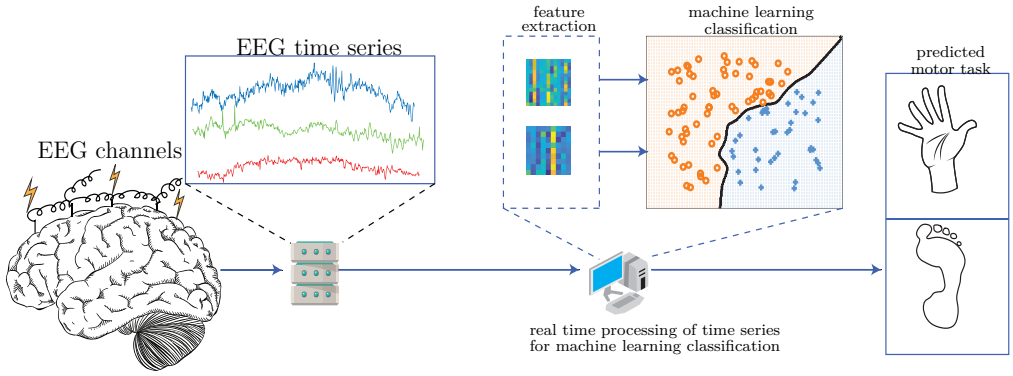


Figure 2.7: A systematic process flow of the Brain interface. The imagined motor movements of the subject are captured in the form of EEG time series which are then fed to the computational unit. A fractional-order dynamics based complex network model is estimated for the time series and the model parameters are used as features for machine learning classification. The output of classifier predicts various motor movements with certain confidence.

for an overview) reliability and resilience. We use the fractional spatio-temporal system model as we discussed in Section 2.1.

The unprocessed EEG signals coming from the sensors although carrying vital information may not be directly useful for making the predictions. However, by representing the signals in terms of parametric model (α, A) and the unknown signals as we did in the Section 2.1, we can gain better insights. The parameters of the model being representative of the original signal itself can be used to make a concise differentiation.

The A matrix represents how strong is the particular signal and how much it is affecting/being affected by the other signals that are considered together. While performing or imagining particular motor tasks, certain regions of the brain gets more activated than others. Simultaneously, the inter-region activity also changes. Therefore, the columns of A which represent the coefficients of the strength of a signal affecting other signals can be used as a feature for classification of motor tasks. In this work, we will be considering the machine learning based classification techniques like logistic regression and Support Vector Machines (SVM) (Murphy, 2012). The other classification techniques c The choice of kernels would vary from simple 'linear' to radial basis function (RBF), i.e., $k(x_i, x_j) = e^{-\gamma(x_i - x_j)^2}$. The value of parameters of the classifier and possibly of the kernels are determined using the cross-validation. The range of parameters in the cross-validation are from $2^{-5}, \ldots, 2^{15}$ for γ and $2^{-15}, \ldots, 2^{3}$ for $C = 1/\ \lambda$, both in the logarithmic scale, where λ is the regularization parameter which appears in optimization cost of the classifiers (Murphy, 2012).

2.4.1 Case Study

We will now illustrate the usefulness of the fractional-order dynamic model with unknown inputs in the context of classification for BCI. We have considered two datasets from the BCI competition (Blankertz et al., 2006). The datasets were selected on the priority of larger number of EEG channels and number of trials for training. The available data is split into the ratio of 60% and 40% for the purpose of training and testing, respectively.

2.4.1.1 Dataset-I

We consider for validation the dataset labeled 'dataset IVa' from BCI Competition-III (Dornhege et al., 2004). The recording was made using BrainAmp amplifiers and a 128 channel electrode cap and out of which 118 channels were used. The signals were band-pass filtered between 0.05 and 200 Hz and then digitized at 1000 Hz. For the purpose of this study we have used the downsampled version at 100 Hz. The dataset for subject ID 'al' is considered, and it contains 280 trials. The subject was provided a visual cue, and immediately after asked to imagine two motor tasks: (R) right hand, and (F) right foot.

Sensor Selection and Modeling: To avoid the curse-of-dimensionality, instead of considering 118 sensors available, which implies the use of 118×118 dynamics entries for classification, only a subset of 9 sensors is considered. Specifically, only the sensors indicated in Figure 2.8 are selected on the basis that only hand and feet movements need to be predicted, and only a 9×9 dynamics matrix and 9 fractional order coefficients are required for modeling the fractional order system. Besides, these sensors are selected because they are close to the region of the brain known to be associated with motor actions.

The model parameters (α, A) are jointly estimated with the unknown inputs using Algorithm 1, therefore the effect of the inputs is inherently taken care of in the parameters. The structure of matrix A for two different labels is shown in Figure 2.10. We have used the sensors C_3 and C_1 which are indexed as 3 and 4, respectively in Figure 2.10. It is apparent from Figure 2.10 that the columns corresponding to these sensors have different activity and hence deem to be fair candidates for the features to be used in classification. Therefore, the total number of features are $2 \times 9 = 18$.

2.4.1.2 Dataset-II

A 118 channel EEG data from BCI Competition-III, labeled as 'dataset IVb' is taken (Dornhege et al., 2004). The data acquisition technique and sampling frequencies are

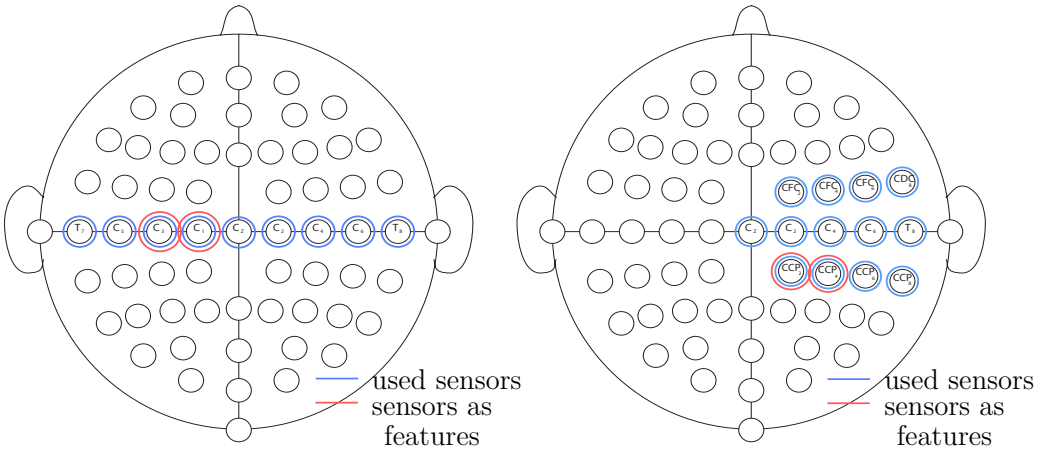


Figure 2.8: A description of the sensor distribution for the measurement of EEG. The channel labels for the selected sensors are shown for dataset-I.

Figure 2.9: A description of the sensor distribution for the measurement of EEG. The channel labels for the selected sensors are shown for dataset-II.

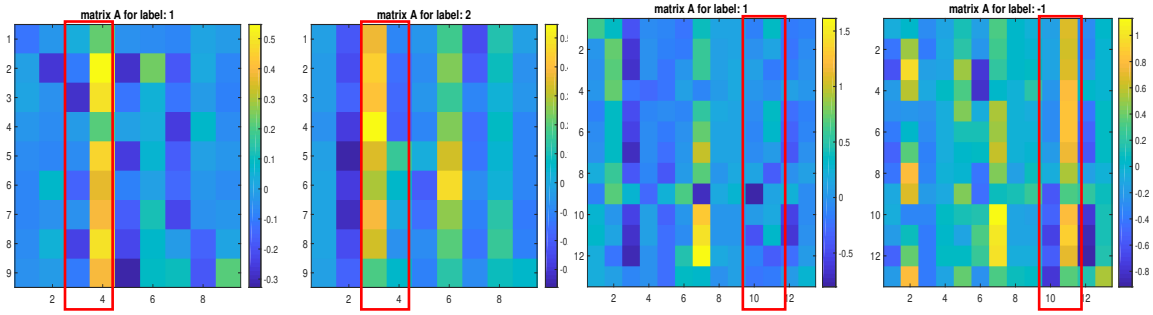
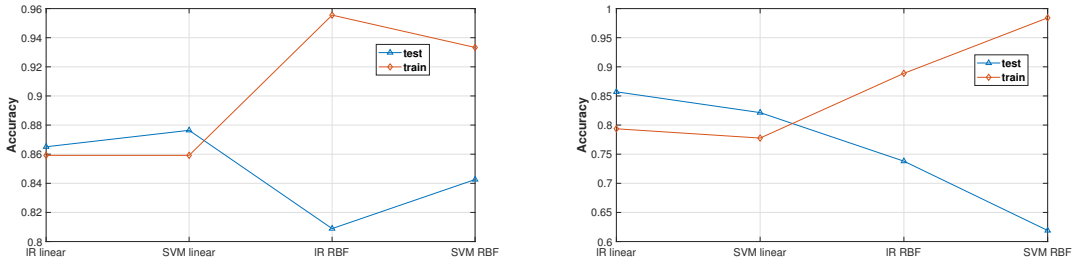


Figure 2.10: Estimated A matrix of size 9×9 for the dataset-I with marked columns corresponding to the sensor index 3 and 4 used for classification.

Figure 2.11: Estimated A matrix of size 13×13 for the dataset-II with marked columns corresponding to the sensor index 10 and 11 used for classification.

same as in dataset of the previous subsection. The total number of labeled trials are 210. The subjects upon provided visual cues were asked to imagine two motor tasks, namely (L) left hand and (F) right foot.

Sensor Selection and Modeling: Due to the small number of training examples, we have again resorted to select the subset of sensors for the model estimation as we did for the dataset-I in the previous section. Since the motor tasks were left hand and feet, therefore we have selected the sensors in the right half of the brain and close to the region which is known to be associated with hand and feet movements as shown in Figure 2.9. We will see in the final part of this section that selecting sensors based on such analogy helps not only in reducing the number of features, but also to gain better and meaningful results.



(a) (b)

Figure 2.12: Testing and training accuracies for various classifiers arranged in the order of classification model complexity from left to right. The estimated accuracies for dataset-I and dataset-II are shown in (a) and (b) respectively.

The estimated A matrix from Algorithm 1 is shown in Figure 2.11 for two different labels. Out of all 13 sensors, the sensors CCP_2 and CCP_4 which are indexed as 10 and 11 in the matrix have striking different activity. The columns corresponding to these two sensors seem good choice for being the features for classification. Therefore, the total number of features are $2 \times 13 = 26$ for this dataset. Next, we discuss the classification accuracy for both the datasets.

2.4.1.3 Classification Performance

Finally, the performance of the classifiers using the features explained for both the datasets are shown in Figure 2.12. The classifiers are arranged in the order of complexity from left to right with logistic regression (lR) and linear kernel being simplest and SVM with RBF kernel being most complex. The performance plot parallels the classic machine learning divergence curve for both the datasets. The accuracy for training data increases when increasing the classification model complexity while it reduces for the testing data. This is intuitive because a complex classification model would try to better classify the training data. But the performance of the test data would reduce due to overfitting upon using the complex models. We have very few training examples to build the classifier and hence such trend is expected. The performance of the classifiers for both the datasets are fairly high which reflects the strength of the estimated features. We can see a 87.6% test accuracy for dataset-I and 85.7% for dataset-II. While these accuracies depend a lot on the cross-validation numbers and other factors like choice of classifier which can be better tuned to get higher numbers.

For both the datasets we have seen that the proposed methodology efficiently extracts the features which serves as good candidate to differentiate the imagined motor movements. By implicitly removing the effects of the unwanted stimuli, the coefficients

of the coupling matrix A are shown to be sufficient for discriminating relation between various EEG signals which are indicative of the motor movements. The testing accuracies are high which indicate the good quality of the extracted features.

2.5 Discussion

We have revisited the EEG-based noninvasive brain interfaces feature extraction and translation from a cyber-physical systems' lens. Specifically, we leveraged spatiotemporal fractional-order models that cope with the unknown inputs. The fractional-order models provide us the dynamic coupling changes that rule the EEG data collected from the different EEG sensors, and the fractional-order exponents capture the long-term memory of the process. Subsequently, unknown stimuli is determined as the external input that least conforms with the fractional-order model. By doing so, we have filtered-out from the brain EEG signals the unknown inputs, that might be originated in the deeper brain structures. The presence of unknown stimuli is possibly the result of the structural connectivity of the brain that crisscrosses different regions, or due to artifacts originated in the muscles (e.g., eye blinking or head movement). As a consequence, the filtered signal does not need to annihilate an entire band in the frequency domain, thus keeping information about some frequency regions of the signal that would be otherwise lost.

We have shown how the different features obtained from the proposed model can be used towards rethinking the EEG-based noninvasive interfaces. In particular, two datasets used in BCI competitions were used to validate the performance of the methodology studied in this chapter, which is compatible with some of the state-of-the-art performances while requiring a relatively small number of training points. We believe that the proposed methodology can be used within the context of different neurophysiological processes and corresponding sensing technologies. Future research will focus on leveraging additional information from the unknown inputs retrieved to anticipate specific artifacts and enable the deployment of neuro-wearables in the context of real-life scenarios. Furthermore, the presented methodology can be used as an exploratory tool by neuroscientists and physicians, by testing input and output responses and tracking their impact in the unknown inputs retrieved by the algorithm proposed; in other words, one will be able to systematically identify the origin and dynamics of stimulus across space and time. Finally, it would be interesting to explore the proposed approach in the closed-loop context, where the present models would benefit from control-like strategies to enhance the brain towards certain tasks or attenuate side effects of certain neurodegenerative diseases or disorders.

Chapter 3

Neuron Spiking Models

In this chapter, we first describe our point process model of neuron system in discrete time with inclusion of unknown artifacts. The monitored neuron behavior is modeled as having the influence of (i) its own spiking history, (ii) other neurons activities, and (iii) the unknown unknowns. We formally describe the problem statement addressed in this work in the following sections.

3.1 Neuron Point Process

We consider a multi-neuron network with a total number of C neurons, and assume that their spiking activities are monitored simultaneously during an observation interval $[T_s, T_s + K\tau)$. The spiking interval length τ is usually small (in the orders of milliseconds), K is the total number of observations and T_s is the starting time. The neuron spiking activities are modeled as point process. Let N_k^c denote the spike counting process of the c-th neuron, $c \in \{1, \ldots, C\}$, during the time interval $[T_s, T_s + k\tau)$, $k = 1, \ldots, K$. Also, a more useful quantity called incremental spiking count ΔN_k^c is the number of spike count fired by the neuron c in time interval $[T_s + (k-1)\tau, T_s + k\tau), k = 1, 2, \ldots, K$, and $\Delta N_{1:K}^{1:C}$ is the incremental sample path of the entire monitored multi-neuron activity.

In the similar fashion, let ΔU_k^i represent the unknown artifact activity in time interval $[T_s+(k-1)\tau, T_s+k\tau)$, $k = 1, 2, \ldots, K$, where i is the index of the unknown, $1 \leq i \leq I$, and I is the total number of unknowns. In what follows, we always assume that $I < C$. Moreover, we can also define $\Delta U_{1:K}^{1:I}$ as the activity path of the unknowns during the observed time horizon $[T_s, T_s + K\tau)$.

The probability density function (PDF) of any c-th neuron with sample path $N^c_{1:K}$ can be expressed with respect to the conditional intensity function (CIF) $\lambda^c(k|\mathcal{H}_k;\Theta)$, where $\mathcal{H}_k$ is the spiking history till k time-interval and Θ are the associated parameters. The CIF $\lambda^c(k|\mathcal{H}_k;\Theta)$ fully characterizes the spike trains for some neuron c (Karr, 2017; Chornoboy, Schramm, and Karr, 1988; Brown, 2005). The CIF can be mathematically defined as the spiking probability on time interval $[k\tau, k\tau+\Delta)$

$$\lambda^c\left(k|\mathcal{H}_k,\Theta\right) = \lim_{\Delta\to 0}\frac{P(N(k\tau+\Delta)-N(k\tau))}{\Delta}, \tag{3.1}$$

where all other conditional covariates contribute to the neuron activity. In our model, the spiking history $\mathcal{H}_k = \left\{\Delta N^{1:C}_{1:k-1}, \Delta U^{1:I}_{1:k-1}\right\}$ with $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \ldots \subseteq \mathcal{H}_K$, and Θ is the parameter tuple to measure this process.

The joint conditional probability density function for the entire multi-neuron point process model can now be expressed with the CIF $\lambda^c(k|H(k);\Theta)$ (Brown, Kass, and Mitra, 2004). We assume that with the given spiking history $\mathcal{H}_k$, the activities of all neurons at k-th time-interval are independent, or they are conditionally independent with respect to the past. In other words, the correlation between neuronal activities appears only through the history. The joint probability of the spike train can be written as

$$P(N^{1:C}_{1:K}|\mathcal{H}_K;\Theta) = \exp\left\{\sum_{c=1}^{C}\sum_{k=1}^{K}\log\lambda^c(k|\mathcal{H}_k;\Theta)\Delta N^c_k - \tau\lambda^c(k|\mathcal{H}_k;\Theta)\right\}. \tag{3.2}$$

We use generalized linear model (GLM) framework to describe the CIF along with the exponential rectification. In this model, the CIF of any neuron c at time k is linear function by four covariates namely: (a) the spontaneous, (b) the intrinsic, (c) the extrinsic, and (d) the unknown covariate. The CIF is formally written as

$$\begin{aligned}\log\lambda^c(k|\mathcal{H}_k;\Theta) = \alpha(c) + &\sum_{q=1}^{\min(k-1,Q)}\varepsilon_q(c)\Delta N^c_{k-q} + \sum_{c'=1,c'\neq c}^{C}\sum_{r=1}^{\min(k-1,R)}\beta^{c'}_r(c)\Delta N^c_{k-r} \\ &+ \sum_{i=1}^{I}\sum_{m=1}^{\min(k-1,M)}\gamma^i_m(c)\Delta U^i_{k-m},\end{aligned} \tag{3.3}$$

where $\alpha(c)$ is the spontaneous spiking rate of the neuron c, $\varepsilon_q(c)$ are the intrinsic parameters corresponding to neuron's own spiking history with a memory length of Q. The extrinsic parameters $\beta^{c'}_r(c)$ relates the effect of neurons $c', c' \neq c$ in system towards neuron c with a memory length R. The unknown parameters $\gamma^i_m(c)$ describe the influence of unknown artifacts with the memory length of M. The complete parameter tuple Θ can now be formally expressed as $\Theta = (\alpha(c), \varepsilon_q(c), \beta^{c'}_r(c), \gamma^i_m(c))$ with appropriate indices of $\alpha, \varepsilon, \beta$ and γ. The CIF in equation (3.3) takes care of the interactions which are intrinsic, extrinsic as well as from unknown sources. Such

$$\nu_q^{i_0\,(n+1)} = \nu_q^{i_0\,(n)} \left[\frac{\sum_{c=1}^{C} \sum_{k=q+1}^{\min(q+M,K)} \Delta N_k^c \gamma_{k-q}^{i_0}(c) + \beta}{\sum_{c=1}^{C} \sum_{k=q+1}^{\min(q+M,K)} \gamma_{k-q}^{i_0}(c) \omega^c(k|\Theta) \tau \lambda_U^c(k|\nu^{(n)}) + \frac{\nu_q^{i_0\,(n)}}{\alpha}} \right]^{t_q^{i_0}}, \quad (3.4)$$

$$t_q^{i_0} = l \left[\frac{\sum_{c=1}^{C} \sum_{k=q+1}^{\min(q+M,K)} \Delta N_k^c \gamma_{k-q}^{i_0}(c) + \beta}{\sum_{c=1}^{C} \sum_{p=\max(q-M+1,1)}^{\min(q+M-1,K)} \sum_{k=\max(p+1,q+1)}^{\min(p+M,q+M,K)} \gamma_{k-q}^{i_0}(c) \gamma_{k-p}^{i_0}(c) \Delta N_k^c} \right], \quad (3.5)$$

interactions are often found among users' activities in the social networks, e.g., Twitter network. The modeling of such activities to decipher the user connectivity can exploit a similar CIF formulation.

3.2 Spiking Model Estimation

With the spiking data and some initial knowledge of the unknown parameters γ, the goal of the estimation procedure as described in this section is to perform two tasks, first (i) estimate the system model parameters $\Theta = (\alpha(c), \varepsilon_q(c), \beta_r^{c'}(c))$, and simultaneously (ii) estimate the unknown activities $\Delta U_{1:K}^{1:I}$. To perform these two tasks, we propose an Expectation-Maximization (EM) formulation (McLachlan and Krishnan, 1996; Dempster, Laird, and Rubin, 1977). The proposed algorithm like EM is split into two parts. First, it estimates the unknown activities having some previous knowledge of the system model parameters. In the next step, the algorithm uses this estimated unknown activity values to update the system model parameters Θ. These steps are repeated until convergence. The goal of the algorithm is to maximize the likelihood, and the proposed procedure being iterative will provide a maximal likelihood solution. The log likelihood associated with the current objective can be written as

$$l = \sum_{c=1}^{C} \sum_{k=1}^{K} \log \lambda^c(k|\mathcal{H}_k; \Theta) \Delta N_k^c - \tau \lambda^c(k|\mathcal{H}_k; \Theta). \quad (3.6)$$

The log-likelihood in (3.6) is a function of CIF, and at this point it is convenient to split the CIF in equation (3.3) into two parts as follows.

$$\lambda_U^c(k|\Delta U) = \exp\left\{ \sum_{i=1}^{I} \sum_{m=1}^{\min(k-1,M)} \gamma_m^i(c) \Delta U_{k-m}^i \right\}, \text{and} \quad (3.7)$$

$$\omega^c(k|\Theta) = \exp\left\{\alpha(c) + \sum_{q=1}^{\min(k-1,Q)} \varepsilon_q(c)\Delta N^c_{k-q} + \sum_{c'=1,c'\neq c}^{C} \sum_{r=1}^{\min(k-1,R)} \beta_r^{c'}(c)\Delta N^c_{k-r}\right\}, \tag{3.8}$$

where $\lambda^c(k|\mathcal{H}_k;\Theta) = \lambda^c_U(k|\Delta U)\,\omega^c(k|\Theta)$. It can be readily realized that the first part $\lambda^c_U(k|\Delta U)$ is a function of unknown activities ΔU, while the second part $\omega^c(k|\Theta)$ is function of system models parameters Θ. Hence, the 'E' and 'M' like step will alternatively update these two parts of the CIF, respectively, to maximize the log likelihood in equation (3.6) iteratively. The CIF partition $\lambda^c_U(k|\Delta U)$ will be used for the rest of the chapter in its most useful form as follows.

$$\lambda^c_U(k|\nu) = \prod_{i=1}^{I}\prod_{m=1}^{M} (\nu^i_{k-m})^{\gamma^i_m(c)}, \tag{3.9}$$

where $\nu^i_k = e^{\Delta U^i_k}$. The update of unknown activities, or also ν^i_k, is performed using the following result.

Theorem 3.1. *Given neuron spikes $\Delta N^{1:C}_{1:K}$, $\omega^c(k|\Theta)$ from (3.8), time interval τ, prior parameters α, β, and the unknown parameters γ, the unknown artifacts $\Delta U^{i_0}_q$ are estimated using fixed-point iterations at each iteration index n as in equation (3.4)-(3.5), where $\nu^{i_0}_q = e^{\Delta U^{i_0}_q}$ and $l \in (0,2)$. The maximum likelihood estimate of $\nu^{i_0}_k$ is denoted as $\widehat{\nu}^{i_0}_k$.*

The reason for restricting the values of unknown parameters γ^i_k to be non-negative in Section 3.2 can now be realized more concretely from equation (3.4) and (3.5). It can be seen that the denominators of terms in both (3.4) and (3.5) can go negative (depending on the data) and hence the fixed-point iterations would possibly become intractable. However, as already mentioned, this does not restrict our ability to model the inhibition and excitation effects, because now it can be decided through the sign of cumulative estimated unknown activities.

For the next step, we wish to update the other part of CIF written in (3.8) as $\omega^c(k|\Theta)$. Again, we express the $\omega^c(k|\Theta)$ in its most useful form for the rest of the chapter by defining the following vectors.

$$\mu^c = [e^{\alpha(c)}, \ldots, e^{\epsilon_n(c)}, \ldots, e^{\beta_r^{c'}(c)}, \ldots], \tag{3.10}$$

$$Y^c(k) = [1, \ldots, \Delta N^{c'}_{k-r}, \ldots, \Delta N^c_{k-n}, \ldots], \tag{3.11}$$

where $Y^c(k)$ and μ^c are $D \times 1$ vectors, $D = 1 + Q + (C-1)R$. The $\omega^c(k|\Theta)$ can now be written as

$$\omega^c(k|\Theta) = \prod_{l=1}^{D} (\mu^c_l)^{Y^c_l(k)}, \tag{3.12}$$

where μ_l^c and $Y_l^c(k)$ are l-th component of μ^c and $Y^c(k)$ in (3.10) and (3.11) respectively.

Theorem 3.2 ((Chornoboy, Schramm, and Karr, 1988)). *Given neuron spikes $\Delta N_{1:K}^{1:C}$, μ^c, $Y^c(k)$ from (3.10)-(3.11), and time interval τ, the exponential of system models parameters μ^c as defined in (3.10) are updated using fixed-point iterations at iteration index n as follows.*

$$\mu_j^{c(n+1)} = \mu_j^{c(n)} \left[\frac{\sum_{k=1}^{K} \Delta N_k^c Y_j^c(k)}{\sum_{k=1}^{K} \tau Y_j^c(k) \left[\prod_{l=1}^{D} (\mu_j^{c(n)})^{Y_l^c(k)} \right]} \right]^{\beta_j^c}, \tag{3.13}$$

$$\beta_j^c = \frac{\sum_{k=1}^{K} \Delta N_k^c Y_j^c(k)}{\sum_{k=1}^{K} \tau Y_j^c(k) \left[\prod_{l=1}^{D} (\widehat{\mu_l^c})^{Y_l^c(k)} \right] \left[\sum_{l=1}^{D} Y_l^c(k) \right]}, \tag{3.14}$$

where $\widehat{\mu_l^c}$ is the maximum likelihood estimate of μ_l^c.

The denominator of β_j^c is a variant of maximum likelihood (ML) estimate of μ_l^c which is problematic as it is not available at the time of iterations. However, an approximation with counting argument is provided in (Chornoboy, Schramm, and Karr, 1988; Chornoboy, 1986) which works well for the estimation problems.

$$\beta_j^c \approx \frac{\sum_{k=1}^{K} \Delta N_k^c Y_j^c(k)}{\sum_{k=1}^{K} Y_j^c(k) \Delta N_k^c \left[\sum_{l=1}^{D} Y_l^c(k) \right]}. \tag{3.15}$$

The estimation results in Theorem 3.1 and Theorem 3.2 are used to construct the following iterative algorithm.

It should be noted that in each fixed-point iteration, both in Theorem 3.1 and Theorem 3.2, the value of exponent $t_q^{i_0}$ and β_j^c is constant with respect to the iteration index-n. Similarly, the numerators of the fixed-point functions are constant. Hence, they need to be computed only once per EM update and lot of computations can be saved. It is also important to note that another big advantage offered by proposed fixed-point iterations is that they are independent across time index-q and unknown index-i_0, therefore they can be implemented in parallel using current *multi-threading/multi-processing* architectures. This make the computations very fast especially when we have large size of the data. On the other hand, the existing Kalman

Algorithm 2: EM algorithm

Input: $N_{1:K}^{1:C}$, γ_m^i $1 \le i \le I$, $1 \le m \le M$ and memory lengths Q, R

Output: $\Theta = (\alpha(c), \varepsilon_q(c), \beta_r^{c'}(c))$, $1 \le c \le C, 1 \le q \le Q, 1 \le r \le R$, and $\Delta U_{1:K}^{1:I}$

Initialize For $t = 0$, set $\Theta^{(0)}$ by using Theorem 3.2 with $\Delta U_{1:K}^{1:I} \sim U[-1, 1]$

repeat

'E-step':

(i) For each $i_0 \in [1, I]$ and $q \in [1, K]$, using $\omega^c(k|\Theta^{(j)})$ obtain $\nu_q^{i_0\,(j+1)}$ from Theorem 3.1 and $\lambda_U^c(k|\nu^{(j+1)})$;

'M-step':

(i) Using $\lambda_U^c(k|\nu^{(j+1)})$ obtain $\mu^{c\,(j+1)}$ from Theorem 3.2 and $\omega^c(k|\Theta^{(j+1)})$;

(ii) $\Theta^{(j+1)} \leftarrow \log \mu^{c\,(j+1)}$;

$j \leftarrow j + 1$;

until until converge;

smoothing techniques (Kulkarni and Paninski, 2007) have dependence across time and has to be computed serially. The fixed-point iterations and EM iterations are in the closed-form, and they are computationally efficient. The convergence is fast, as we will see in Section 3.3.1 and Section 3.3.3. The choice of scalar l in equation (3.5) can play an important role in convergence rate and hence can be taken as an input parameter as well.

The proposed techniques developed in this section are tested on simulated as well as real-world datasets in Section 3.3.1 and Section 3.3.3, respectively.

3.3 Results

The applicability of the proposed neuron spiking model estimation technique with unknown unknowns is studied hereafter. The estimation process, as detailed in the Algorithm 1, is applied to an artificially generated spiking data as well as real-world spiking dataset which is explained in the following parts.

3.3.1 Artificial Neuron Network

An artificial neuron network in Figure 3.1 is designed with a total of six neurons. Each neuron is assumed to be influenced by (a) its intrinsic activity, (b) extrinsic effects via other neurons in the network, and (c) unknown artifacts. The contribution of unknown sources is quantized by having two unknown nodes u_1 and u_2. The extrinsic

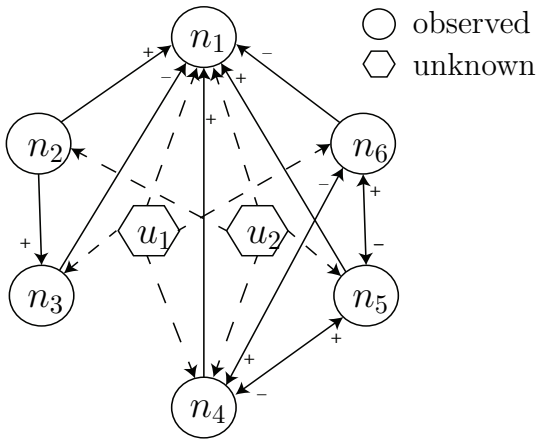


Figure 3.1: Neuron network assumed for the generating artificial neuronal spikes, with six observed neurons and two unknown artifacts. The excitation and inhibition effects are indicated by + and − respectively for each directed edge.

effect is modeled by having directed edges among neurons as shown in Figure 3.1. For example, neuron n_2 excites n_3, and the excitation effect is indicated using +, also, neuron n_6 inhibits n_1, and the inhibition effect is marked through − sign. The contribution of unknown nodes is indicated by the corresponding directed edges from u_i to n_j in the network.

The parameters required for defining the intensity function of the spiking point-process as in equation (3.3) are selected as follows: The values of spontaneous firing rate α for six neurons are selected as, $\alpha = [3.5, 2.4, 2.0, 3.0, 2.8, 2.5]$. The intrinsic effect memory length is selected as $Q = 50$ for all neurons. The previous spiking history of each neuron can have excitatory as well as inhibitory effects on the future spiking activity. Also, with the increasing length of the history, the effects get mitigated. The initial values of the intrinsic parameters are negative to model the refractory period of the neurons after firing a spike (Truccolo et al., 2005; Berry II and Meister, 1998). To collectively model these effects, we describe the intrinsic parameters using sinc function. The intrinsic parameter values are selected as described in (Yang, Gupta, and Bogdan, 2019).

Next, the extrinsic effect as indicated via directed edges in Figure 3.1 is quantized though parameters $\beta_r^{c'}(c)$ for each $n_{c'} \to n_c$. The extrinsic memory length is fixed as $R = 5$ for each directed edge. We have used exponential decay functions to model the parameters and the values are shown (Yang, Gupta, and Bogdan, 2019). Next, the unknown sources contribution memory length is fixed as $M = 5$ for both sources u_1 and u_2. The parameters associated with unknown excitations are always taken to be positive, as explained in Section 3.2, and the undertaken values are shown in (Yang, Gupta, and Bogdan, 2019). Finally, with these parameter values we can now proceed to the spike generation process.

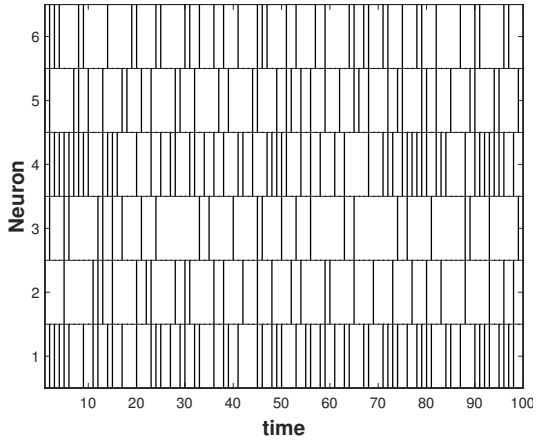


Figure 3.2: Simulated neuron spike trains for six neurons with system model parameters as described in Section 3.3.1.

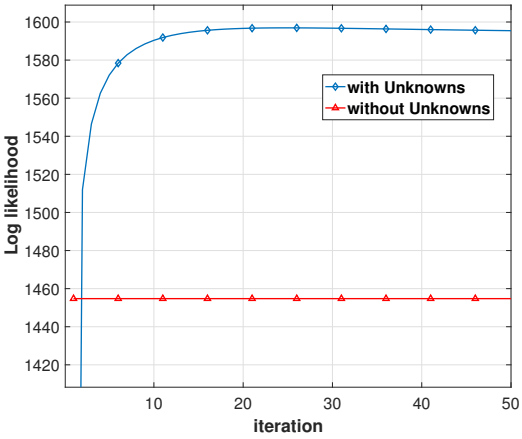


Figure 3.3: Log likelihood at each iteration using Algorithm 1 on the simulated spike trains.

3.3.2 Spike Generation with Unknown contributions

The multi-neuron spike generation can be performed by recursively compute the conditional intensity parameter, or firing rate, and adding the contributions of the unknown sources. As mentioned in Appendix B.1, we have used log-Gamma distribution for ΔU_k^i, and independent and identically distributed (iid) samples of size $[I \times K]$ are generated using the PDF in equation (B.2). The parameter values are taken to be $\alpha = 1$ and $\beta = 50$, and the samples are then mean centered to zero. The spike train generation procedure is similar to (Kim et al., 2011) and for the sake of completeness, we briefly write the steps as follows:

- At step k, compute the conditional intensity function from equation (3.3) using the system model parameters as stated in Section 3.3.1, and unknown sources contribution ΔU_k^i from (B.2);
- Generate a uniform random number $u \sim U[0, 1]$ and if $u < \tau\lambda^c(k|\mathcal{H}_k, \theta)$, then there is spike for c-th neuron at time interval k;and
- Repeat with recursively computing conditional intensity function from equation (3.3), until desired length of spike train.

where the value of τ is taken to be 0.05, and for the k-th step, the time-interval in consideration is $[T_s + (k-1)\tau, T_s + k\tau)$. For simulations, we have generated a total of $K = 500$ multi-neuron spikes. The spike train obtained using above procedure for all six neurons in the first 100 time-intervals is as shown in Figure 3.2. We now apply the proposed technique of model estimation with unknown unknowns.

The artificially generated spike train in Figure 3.2 with the parameters for unknown sources as in (Yang, Gupta, and Bogdan, 2019) is used as an input for the Algorithm 2. The algorithm recursively computes the unknown contribution ΔU_k^i and then update the system model parameters. The iterations are fixed-point based and the convergence is fast. The log likelihood plot (with constants ignored) is shown in Figure 3.3. The 'without unknowns' case is computed with setting unknowns to zero and, for consistency, we run the 'without unknowns' case with maximum 'M-step' iterations occurring in the proposed 'with unknowns' approach. As expected, there is sharp increase in the likelihood in the first few iterations, due to incorporation of unknowns, and it is observed that few EM iterations are sufficient for convergence. The proposed approach results in better log-likelihood as compared to the case of no unknowns.

3.3.3 Real-World Data

In this section, we explore various neuron spiking data recorded through real-world experiments. Each dataset poses some challenges as compared to the simulated dataset considered in the Section 3.3.1. We describe the datasets and discuss the results in the following sections.

3.3.3.1 Mouse Somatosensory Spiking Data

The mouse somatosensory spiking dataset is recorded using a 512-channel multi-electrode array system in an organotypic cultures of dorsal hippocampus and somatosensory cortex sections (Ito et al., 2014; Litke et al., 2004; Timme et al., 2014; Nigam et al., 2016; Shimono and Beggs, 2015; Timme et al., 2016). The cultures were not stimulated and the dataset represents spontaneous activity. The spikes were sorted using Principle Component Analysis (PCA). The dataset is downloaded from (Ito et al., 2016), where a total of 25 datasets are available, and hundreds of neurons were probed to have the spiking data. In this work, we have taken a total of 8 neurons to study the inter-neuronal interactions with unknown unknowns. The spiking activity of the considered neurons is as shown in Figure 3.4.

The inter-spiking interval is having a large value for spikes at some times, and it is very small for others. Therefore, while modeling such datasets it is possible that the assumed model for CIF in (3.3) may not apply in its original form. The proposed technique like any other data-driven estimators can work only if there are samples corresponding to the associated parameters. For example, when the data is too sparse

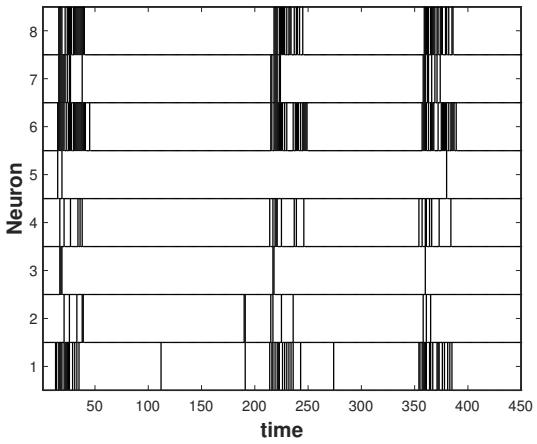


Figure 3.4: Multi-neuron spike train for Somatosensory data with a total of eight neurons.

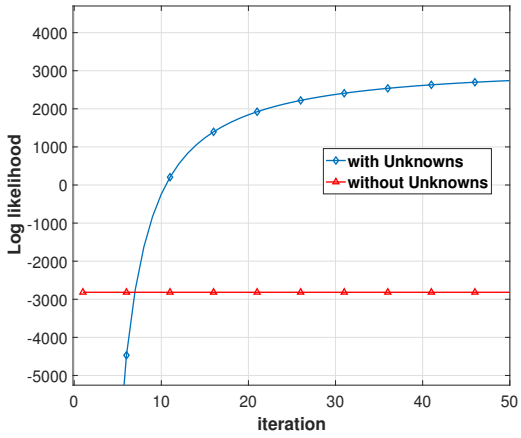


Figure 3.5: Log likelihood at each iteration using Algorithm 2 on the Somatosensory spike trains dataset.

in some time-intervals then the denominator term in equation (3.15)

$$\sum_{k=1}^{K} Y_j^c(k)\Delta N_k^c \left[\sum_{l=1}^{D} Y_l^c(k)\right],$$

will go to zero. It is an indication that there is no data corresponding to the j-th term in (3.10) and hence it cannot be recovered. A simple modification would be to increase the spiking interval length τ and take ΔN_k^c as accumulated counts in that enlarged interval. The proposed Algorithm 2 is then applied to the spike train and the log likelihood (with constants ignored) is shown in Figure 3.5. We observe that the convergence is fast and the likelihood increase sharply in the beginning. We also observe that the resulting log likelihood is better as compared to the case of not having unknowns in the model.

3.3.3.2 Mouse Retina Spiking Data

The mouse retina dataset contains neuronal responses of retinal ganglion cells to various visual stimuli recorded in the isolated retina from lab mice (Mus Musculus) using a 61-electrode array. The visual stimuli were displayed on a gamma-corrected cathode-ray tube monitor and projected on the photoreceptor layer of the retina (magnification $8.3\mu m$/pixel; luminance range 0.5-3.8 mW/m^2) from above through a custom-made lens system (Lefebvre et al., 2008). Extracellular action potentials were recorded (sampling rate 10 kHz) and single units were identified by a semi-automated spike-sorting algorithm. The dataset is downloaded from (Zhang, Asari, and Meister, 2014) which comprises 16 retinal preparations. The spike trains for a total of seven neurons is shown in Figure 3.6.

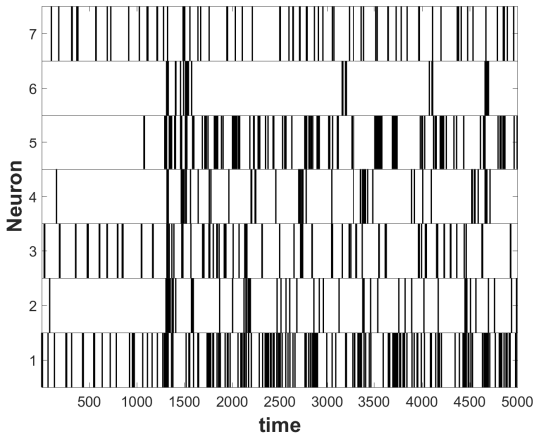


Figure 3.6: Multi-neuron spike train for Mice Retina data with a total of seven neurons.

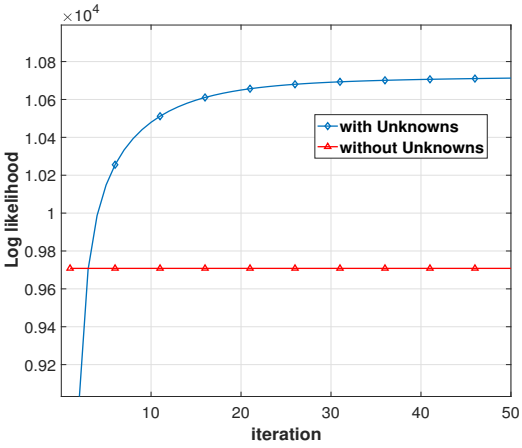


Figure 3.7: Log likelihood at each iteration using Algorithm 2 on the Mice Retina spike trains dataset.

The concept of having unknown sources is better in the sense that it provides flexibility to fill the gaps in the real-world data and the assumed model by exploiting the extra degrees-of-freedom that the data might have. However, we have observed in this dataset that this may not be true always, and sometimes unknown contributions are not necessary. In particular to our model, this can be realized when the denominator of equation (3.5)

$$\sum_{c=1}^{C} \sum_{p=\max(q-M+1,1)}^{\min(q+M-1,K)} \sum_{k=\max(p+1,q+1)}^{\min(p+M,q+M,K)} \gamma_{k-q}^{i_0}(c)\gamma_{k-p}^{i_0}(c)\Delta N_c^k,$$

goes to zero. This indicates that there is not enough degrees-of-freedom in the data (and given values of unknown parameters $\gamma_k^i(c)$) to estimate the $\Delta U_q^{i_0}$. In such cases, we can set the unknown activities to zero for the corresponding q, i_0 indices, or in other words assume that there is no requirement of unknowns at q-th time step for i_0-th unknown source. The log-likelihood (with constants ignored) after this adjustment on applying Algorithm 1 is shown in Figure 3.7. We observe that the likelihood increases quickly and convergence occurs in only 3-5 iterations. The resulting log-likelihood is also higher than the model estimation procedure without having unknowns.

3.4 Discussion

We have presented a multi-variate neuron system framework with the inclusion of unknown inputs. The single neuron activity is affected by its own spiking history, the influence from other neurons, and the unknown stimuli. The statistical framework is built on the non-stationary likelihood of the multi-channel point process, which

is characterized by the conditional intensity function with a generalized linear combination of the covariates. The proposed method aims at increasing the likelihood of the data, and the inclusion of unknown unknowns to minimize the discrepancies between the data and the model. The developed algorithm is based on fixed-point iterations and converges quickly. The proposed fixed-point method offers advantages of independence across time and can be computed in parallel using modern multi-core processors. Experiments on simulated spike trains and on the real-world datasets of mouse somatosensory, retinal and cat retina show promising results in terms of likelihood maximization. We have observed interesting insights into degrees-of-freedom offered by the data which sometimes suggest not to use unknown unknowns.

The proposed mathematical framework is general for non-stationary discrete time-series in the form of spike trains with missing data (or gaps) and the contribution of unknown sources. The developed techniques are computationally efficient especially when the data is recorded over long time-horizon. While we relied on a log-gamma distribution as the prior for unknown artifacts, we plan to investigate unknown unknowns phenomena that exhibit non-Gaussian statistical or multifractal behavior (Xue and Bogdan, 2017). The future research will also focus on exploring applications in the domain of pattern identification in social networks of opinion dynamics, smart cities monitoring for chemical and threat (explosive) detection and identification/localization, etc. A critical challenge with such datasets is the huge data size, and the computational advantages offered by the proposed techniques can make the statistical inference tractable.

Part II

Partial Differential Equations

Chapter 4

Fractional Diffusion equations

Finding the exact parameters of the corresponding governing PDE is not a trivial task. In this context, given a spatiotemporal dataset, we aim to develop an efficient algorithm for estimating the parameters of the generalized fractional-order PDE that models the dynamics of the process under investigation. Consequently, by identifying these parameters, one can also investigate the physical rules modeled by the PDE. In the results section, we analyze two types of PDEs and discuss algorithmic approaches to determine their parameters from the time-series data. We also provide a simulation study where we verify the correctness and effectiveness of the proposed algorithmic approaches on deriving the exact parameters using synthetic trajectories generated from the PDE model.

4.1 Data-Driven Approach for Analyzing Anomalous Diffusion

4.1.1 Space-Time Fractional Diffusion Equation

The space-time fractional diffusion equation has been proposed in previous works as a mathematical model to analyze anomalous diffusion (Metzler and Klafter, 2000; Gorenflo and Mainardi, 2012; Mainardi, Luchko, and Pagnini, 2007; Saichev and Zaslavsky, 1997; Gorenflo, Iskenderov, and Luchko, 2000; Scalas, Gorenflo, and Mainardi, 2000). In a nutshell, the space-time fractional diffusion equation in (4.1) consists of a fractional Riesz-Feller derivative of the order $\alpha > 0$ (space-derivative) that encodes the space variations and a fractional Caputo derivative of the order $\beta > 0$ (time-derivative) that measures the time variations. To better generalize, we

also consider the skewness factor in the space derivative of the diffusion equation. Hence, the space-time fractional diffusion equation is defined as

$$ {}_t\mathcal{D}_*^{\beta} u(x,t) = D \times {}_x\mathcal{D}_{\theta}^{\alpha} u(x,t), \qquad \forall x \in \mathbb{R}, \forall t \in \mathbb{R}^+, \tag{4.1} $$

where the operators ${}_x\mathcal{D}_{\theta}^{\alpha}$ and ${}_t\mathcal{D}_*^{\beta}$ designate the fractional Riesz-Feller derivative of order α and skewness θ (Feller, 1962), and the Caputo time-fractional derivative of order β (Caputo, 1967), respectively[1]. The parameter D denotes the generalized diffusion coefficient. The parameters α, β and θ satisfy the following constraints, $0 < \alpha \leq 2$, $0 < \beta \leq 1$ and $|\theta| \leq \min\{\alpha, 2-\alpha\}$.

Given a set of time-series trajectories that record the evolution of particles or agents that exhibits anomalous diffusion modeled by equation (4.1), without prior knowledge about the parameters of the space-time fractional diffusion equation, our goal is to use the dataset available to retrieve the exact fractional PDE that generates the given time-series. Towards this end, we develop a mathematical framework where the parameter and mathematical (operator) expression identification task is defined as a regression problem. Indeed, the regression is formulated as a least squares problem, where the minimization involves the theoretical and the empirical statistical (higher order) moments (i.e., specifically the absolute moments). The choice of the statistical moments for performing the regression is convenient because we could derive its closed form expressions just from the generalized fractional PDE given in equation (4.1). For the given time-series data, $X_n(t), 1 \leq n \leq N$, where N denotes the total number of trajectories, the time empirical moments are defined as follows

$$ M_t^{\delta} = \frac{1}{N} \sum_{n=1}^{N} |X_n(t)|^{\delta}, \qquad S_t^{\delta} = \frac{1}{N} \sum_{n=1}^{N} X_n(t)^{\langle \delta \rangle}, \tag{4.2} $$

where $x^{\langle \delta \rangle}$ denotes the signed absolute δ-th power of x and $x^{\langle \delta \rangle} = |x|^{\delta}\text{sign}(x)$. The time-dependent absolute moment of the data generated according to the fractional PDE in (4.1) is given by the following result.

Proposition 2. *The time-dependent absolute moment of the order δ with $0 < \delta < \alpha$ is written as follows*

$$ \mathbb{E}[|X(t)|^{\delta}] = t^{\delta \frac{\beta}{\alpha}} D^{\frac{\delta}{\alpha}} \times \frac{\Gamma(1-\frac{\delta}{\alpha})\Gamma(1+\frac{\delta}{\alpha})\cos(\frac{\delta\pi\theta}{2\alpha})}{\Gamma(1-\delta)\Gamma(1+\delta\frac{\beta}{\alpha})\cos(\frac{\delta\pi}{2})}, \tag{4.3} $$

where $\Gamma(\cdot)$ designates the gamma function.

To find the parameters of the fractional PDE in (4.1), we rely on analyzing the higher order moments and minimize the quadratic error between the theoretical and

[1]The two operators are clearly defined in the supplementary material document.

the empirical absolute higher order moments. However, due to the non-linear non-convex expression stated in the equation (4.3), such a regression problem is non-trivial and it is non-trivial to provide the theoretical guarantees about the convergence of a multi-dimensional optimization algorithm used to solve the problem in one-shot. To tackle the non-convexity, we aim to approach the parameters estimation via multi-step optimization. For obtaining additional information from the data, we also derive the signed absolute as in the following result.

Proposition 3. *The time-dependent signed absolute moment of the order δ with $0 < \delta < \alpha$ is written as follows*

$$\mathbb{E}[X(t)^{\langle\delta\rangle}] = -t^{\delta\frac{\beta}{\alpha}} D^{\frac{\delta}{\alpha}} \times \frac{\Gamma(1-\frac{\delta}{\alpha})\Gamma(1+\frac{\delta}{\alpha})\sin(\frac{\delta\pi\theta}{2\alpha})}{\Gamma(1+\delta\frac{\beta}{\alpha})\Gamma(1-\delta)\sin(\frac{\delta\pi}{2})}. \tag{4.4}$$

The detailed proofs of the above propositions are provided in Appendix C.2 and C.3. Using the above results, the estimation of parameters is detailed in the following section.

4.1.2 Parameter Estimation

4.1.2.1 Absolute moments approach

Starting from a dataset containing N independent trajectories (realizations of the equation (4.1) with unknown parameters) sampled uniformly at times $\{t_1, t_2, \cdots, t_L\}$, we aim at estimating the actual parameters of the equation (4.1) via a moments-based approach, i.e., determining the parameters using empirical moments and the theoretical expressions. Thus, the proposed scheme to find the parameters is mainly a two-step approach, regression over time on one hand and over space on the other hand. The method is summarized as follows.

For a given order δ, the log of absolute moments in equation (4.3) varies as

$$\log(\mathbb{E}[|X(t)|^{\delta}]) = \delta\frac{\beta}{\alpha}\log(t) + C_1, \tag{4.5}$$

where C_1 does not depend on t. Using the estimated empirical moments from equation (4.2), we replace the theoretical moments with the empirical values in the equation (4.5). The parameter ratio β/α can then be estimated by performing linear regression of $\log(t_l)$ versus $\log(M_{t_l}^{\delta})$ using a total of L points (i.e., $l = 1, 2, \cdots, L$). It is worthwhile to note that the precision in estimation of β/α can be improved upon using multiple values of moment exponent δ for increasing the total diverse points in linear

regression. For example, a set $\Delta = \{\delta_1, \delta_2, \ldots, \delta_K\}$ can be used for having $K \times L$ diverse points for linear regression by repeating the equation (4.5) with different δ_k. Hence, we can make a trade-off between space and time, i.e., enlarging the cardinality of Δ when the available time-series are short, in other words when L is small.

Next, we use the results of Proposition 2 and 3 to get the ratio of $\mathbb{E}[|X(t)|^\delta]$ and $\mathbb{E}[X(t)^{\langle\delta\rangle}]$ as the following ratio r

$$r = \frac{\mathbb{E}[X(t)^{\langle\delta\rangle}]}{\mathbb{E}[|X(t)|^\delta]} = -\frac{\tan(\frac{\pi\delta\theta}{2\alpha})}{\tan(\frac{\pi\delta}{2})}. \tag{4.6}$$

Upon replacing the theoretical moments with the ones derived in equation (4.2) we can invert the tangent function to have the ratio θ/α. In addition, although the ratio r is independent of time t, the empirical ratio of the data will possibly not be a constant across time. We replace r with time average of the ratio of $S^\delta_{t_l}$ and $M^\delta_{t_l}$ as $\overline{S^\delta_{t_l}/M^\delta_{t_l}}$. From Section 4.1.1, we know that the parameter θ is constrained as $|\theta| \leq \min(\alpha, 2-\alpha)$, so, we have $|\theta/\alpha| \leq 1$. Next, we define the following function

$$w_L(x) = \begin{cases} -1 & x < -1 \\ x & -1 \leq x \leq 1 \\ 1 & x > 1 \end{cases}.$$

Finally, the ratio of parameters θ and α can now be written as

$$\frac{\theta}{\alpha} = w_L\left(-\frac{2}{\pi\delta}\arctan\left(\tan\left(\frac{\pi\delta}{2}\right)\overline{\left(\frac{S^\delta_{t_l}}{M^\delta_{t_l}}\right)}\right)\right). \tag{4.7}$$

As argued before, to improve the precision of the estimation, we can add diversity by having a set of moment exponents $\Delta = \{\delta_1, \delta_2, \ldots, \delta_K\}$. For each $\delta_k \in \Delta$ (where $k = 1, 2, \cdots, K$), we use the equation (4.7) to obtain $\widehat{(\theta/\alpha)}_k$, and finally obtain the estimated ratio of parameters as $\widehat{(\theta/\alpha)} = \frac{1}{K}\sum_{k=1}^K \widehat{(\theta/\alpha)}_k$.

The absolute moments of order δ from Proposition 2 can be re-written as follows

$$\begin{aligned} \frac{\mathbb{E}[|X(t)|^\delta]}{t^{\delta\frac{\beta}{\alpha}}} &= D^{\frac{\delta}{\alpha}} \times \frac{\Gamma(1-\frac{\delta}{\alpha})\Gamma(1+\frac{\delta}{\alpha})\cos(\frac{\delta\pi\theta}{2\alpha})}{\Gamma(1-\delta)\Gamma(1+\delta\frac{\beta}{\alpha})\cos(\frac{\delta\pi}{2})} \\ &= D^{\frac{\delta}{\alpha}} \times \frac{\frac{\pi\delta}{\alpha}}{\sin(\frac{\pi\delta}{\alpha})} \times \frac{\cos(\frac{\pi\delta\theta}{2\alpha})}{\Gamma(1-\delta)\Gamma(1+\frac{\delta\beta}{\alpha})\cos(\frac{\pi\delta}{2})}. \end{aligned}$$

Therefore, for the given value of the order $\delta = \delta_k$, the estimation of α and diffusion coefficient D can be written as the following non-linear equation

$$C_k = \frac{\Gamma(1-\delta_k)\Gamma(1+\frac{\delta_k\beta}{\alpha})\cos(\frac{\pi\delta_k}{2})}{\cos(\frac{\pi\delta_k\theta}{2\alpha})} \times \frac{\mathbb{E}[|X(t)|^{\delta_k}]}{t^{\delta_k\frac{\beta}{\alpha}}} = D^{\frac{\delta_k}{\alpha}} \times \frac{\frac{\pi\delta_k}{\alpha}}{\sin(\frac{\pi\delta_k}{\alpha})}. \tag{4.8}$$

To solve the non-linear equation in (4.8) for α and D, we use non-linear least squares method (trust region reflective method). The input variables to the non-linear optimization are the values of order $\delta \in \Delta$, where $\Delta = \{\delta_1, \delta_2, \ldots, \delta_K\}$. Formally, the optimization problem is written as follows:

$$\{\hat{D}, \hat{\alpha}\} = \underset{\alpha, D}{\operatorname{argmin}} \sum_{k=1}^{K} \left| D^{\frac{\delta_k}{\alpha}} \times \frac{\frac{\delta_k\pi}{\alpha}}{\sin(\frac{\delta_k\pi}{\alpha})} - C_k \right|^2. \tag{4.9}$$

We note that the values of C_k can be efficiently estimated by first performing the linear regression of $t^{\delta\frac{\beta}{\alpha}}$ vs $\mathbb{E}[|X(t)|^{\delta}]$ with the condition of zero intercept, and then substituting the ratio (slope of the linear regression) in equation (4.8). The optimization problem in (4.9) is non-convex, therefore, the global optimum solution is not guaranteed by the solvers. Note that, in some scenarios, we may have a prior knowledge about boundaries of the parameter α (i.e., $\alpha_{\min} \leq \alpha \leq \alpha_{\max}$). Thus, we solve the constrained optimization problem.

Finally, the values of β and θ can be estimated upon estimating α, as we already have the ratios β/α and θ/α from equation (4.5) and equation (4.7), respectively. The approach described is summarized as Algorithm 3.

Remark: Note that for $\alpha \neq 2$, when the order δ is close to the boundary values, the theoretical absolute moment goes to $+\infty$. However, the empirical one is finite, so in order to have a small error associated to the estimated parameters, we choose the order δ to be far enough from the end points of allowed region. Notice that, the value of α is unknown while the condition $-\min\{\alpha, 1\} < \Re(\delta) < \alpha$ is dependent of it, so assuming an $\alpha_{\min}$ as lower bound for α is rational. We note that the Algorithm 3 is dependent upon the solution of non-linear non-convex optimization problem in (4.9), and, therefore, convergence to the global solution is not guaranteed. Also, we need to provide an input set Δ such that the absolute and signed moments are computable for unknown α. The choice of Δ has to be made by having some idea about lower bound of the α parameter. To take care of these two issues together, we next present an alternative approach in which we do not require non-linear optimization as well as do not require to have the knowledge of Δ set.

Algorithm 3: Space-Time Fractional Diffusion: Parameters Estimation

Input: Time-series data $\{X_n(t_l); 1 \leq n \leq N, 1 \leq l \leq L\}$, order δ, $\Delta = \{\delta_1, \delta_2, \cdots, \delta_K\}$

Output: Parameters: α, β, θ and D

1: **for** $l = 1, 2, \cdots, L$ **do**
2: Calculate the empirical absolute and signed moments $M_{t_l}^{\delta}$ and $S_{t_l}^{\delta}$ ▷ Eq.4.2
3: **end for**
4: $\widehat{\left(\frac{\beta}{\alpha}\right)} \leftarrow \frac{m_1}{\delta}$, m_1 being slope of linear regression $\log(t)$ vs $\log(M_t^{\delta})$
5: Get the estimate $\widehat{\left(\frac{\theta}{\alpha}\right)}$ ▷ Eq.4.7
6: **for** $k = 1, 2, \cdots, K$ **do**
7: Calculate the empirical absolute moments $M_{t_l}^{\delta_k}, \quad \forall l$ ▷ Eq.4.2
8: $m_2 \leftarrow$ slope of linear regression $t^{\delta_k \widehat{\left(\frac{\beta}{\alpha}\right)}}$ vs $M_t^{\delta_k}$ with zero intercept
9: $C_k \leftarrow m_2 . \frac{\Gamma\left(1+\delta_k \widehat{\left(\frac{\beta}{\alpha}\right)}\right)\Gamma(1-\delta_k)\cos\left(\frac{\pi\delta_k}{2}\right)}{\cos\left(\frac{\pi\delta_k}{2}\widehat{\left(\frac{\theta}{\alpha}\right)}\right)}$ ▷ Eq.4.3
10: **end for**
11: Find $\hat{\alpha}, \widehat{D} \leftarrow \underset{\alpha, D}{\operatorname{argmin}} \sum_{k=1}^{K} \left| D^{\frac{\delta_k}{\alpha}} \times \frac{\frac{\delta_k \pi}{\alpha}}{\sin(\frac{\delta_k \pi}{\alpha})} - C_k \right|^2$: non-linear regression over space (sinc inversion)
12: Calculate $\hat{\beta}, \hat{\theta}$.

4.1.2.2 Log absolute Moments Approach

In this subsection, we rely on the moments of log absolute values of the trajectories. Similar to absolute moments with order δ, the log absolute moments can be computed as we present in the following results.

Proposition 4. *The time-dependent expected log absolute value of $X(t)$ is written as follows*

$$\mathbb{E}[\log|X(t)|] = \frac{\beta}{\alpha}\log(t) + \frac{\log(D)}{\alpha} + \gamma\left(\frac{\beta}{\alpha} - 1\right), \tag{4.10}$$

where γ is the Euler-Mascheroni constant.

Next, the variance of the log absolute values of the trajectories can be written as the following results.

Proposition 5. *The variance of log absolute value of $X(t)$ is written as follows*

$$var(\log|X(t)|) = \frac{\pi^2}{6}\left(\frac{1}{\alpha^2} + \frac{1}{2}\right) - \left(\frac{\pi\theta}{2\alpha}\right)^2. \tag{4.11}$$

It is interesting to note that the variance is independent of time as well as a function of α and θ/α. We exploit this feature of the variance to obtain an estimate of the α with the ratio θ/α known. Hence, the need for performing non-linear optimization in Algorithm 1 is omitted. We also write the second moment of the log absolute values in the following result.

Proposition 6. *The time-dependent expected square of log absolute value of $X(t)$ is written as follows*

$$\mathbb{E}[(\log|X(t)|)^2] = \frac{\beta^2}{\alpha^2}\log^2(t) + 2\frac{\beta\gamma}{\alpha}\left(\frac{\beta}{\alpha} - 1\right)\log(t) + c, \tag{4.12}$$

where $c = \frac{\pi^2}{6}\left(\frac{1}{\alpha^2} + \frac{1}{2}\right) - \left(\frac{\pi\theta}{2\alpha}\right)^2 + \left(\frac{\log(D)}{\alpha} + \gamma\left(\frac{\beta}{\alpha} - 1\right)\right)^2 + \frac{\pi^2}{6\alpha^2}(1-\beta^2)$, and γ is the Euler-Mascheroni constant.

The proof of the Propositions 4, 5 and 6 are provided in the Appendix C.4, C.5, and C.6, respectively. Using the above results for log absolute values, we now present the second approach to estimate the parameters of the space-time fractional PDE in (4.1).

We proceed similarly to the first approach of the δ order absolute moments, however, now equating the theoretical and empirical expressions of the log absolute moments. The empirical log absolute moments are written as

$$\begin{aligned} L_t^{(1)} &= \frac{1}{N}\sum_{n=1}^{N}\log|X_n(t)|, \qquad L_t^{(2)} = \frac{1}{N}\sum_{n=1}^{N}\log|X_n(t)|^2, \\ \overline{\text{var}(\log|X(t)|)} &= \frac{1}{N-1}\sum_{n=1}^{N}\left(\log|X_n(t)| - L_t^{(1)}\right)^2. \end{aligned} \tag{4.13}$$

The parameter ratio β/α is estimated by performing the linear regression of $\log(t)$ vs $L_t^{(1)}$. The slope of the regression output is the estimated ratio $\widehat{(\beta/\alpha)}$. Next, the ratio of the parameters θ and α is estimated using the same approach as described previously in equation (4.7).

We note that upon having an estimate of the parameter ratio $\widehat{\theta/\alpha}$, the variance is one-to-one function of α since $\alpha \geq 0$. Therefore, on substituting the value of $\widehat{\theta/\alpha}$ in equation (4.11) we compute the value of $\widehat{\alpha}$. Finally, with $\widehat{\alpha}$ and $\widehat{\beta/\alpha}$ known, the value of diffusion coefficient is estimated from the intercept of the linear regression of $\log(t)$ vs $L_t^{(1)}$ as $\widehat{D}$. The above described approach is summarized as an algorithmic strategy in Algorithm 4.

Algorithm 4: Space-Time Fractional Diffusion: Parameters Estimation

Input: Time-series data $\{X_n(t_l); 1 \leq n \leq N, 1 \leq l \leq L\}$, order δ
Output: Parameters: α, β, θ and D

1: **for** $l = 1, 2, \cdots, L$ **do**
2: Calculate the empirical absolute and signed moment $M_{t_l}^{\delta}$ and $S_{t_l}^{\delta}$ ▷ Eq.4.2
3: Calculate log absolute moments $L_t^{(1)}$ ▷ Eq.4.13
4: **end for**
5: $\widehat{\left(\frac{\beta}{\alpha}\right)} \leftarrow m$, (m, c) being (slope, intercept) of linear regression $L_t^{(1)}$ vs $\log(t)$
6: Get the estimate $\widehat{\left(\frac{\theta}{\alpha}\right)}$ ▷ Eq.4.7
7: Calculate empirical variance of log absolute values $\overline{\text{var}(\log(|X(t)|))}$ as σ^2 ▷ Eq.4.13
8: $\hat{\alpha} \leftarrow \left(\sigma^2 \frac{6}{\pi^2} - \frac{1}{2}\right)^{-\frac{1}{2}}$
9: $\hat{D} \leftarrow \exp\left\{\hat{\alpha}\left(c - \gamma\left(\widehat{\left(\frac{\beta}{\alpha}\right)} - 1\right)\right)\right\}$ ▷ Eq.4.3
10: Calculate $\hat{\beta}, \hat{\theta}$.

It should be noted that the approach utilizing log absolute moments does not require a predefined set of order values Δ. In addition, this does not suffer from the convergence issues as there is no non-linear non-convex optimization involved.

Note: The estimated parameters in both algorithms are not guaranteed to be optimal. For example, it is not straightforward to guarantee maximum likelihood sense as solving maximum likelihood involves solution to non-convex problem. We evaluate the efficiency of the both algorithms in the following section.

4.2 Experimental Results

As we described previously, both algorithms depend mainly on the statistical absolute moments, statistical signed absolute moments and the expected log absolute value of the process $X(t)$. For this reason, we first start by validating the theoretical expressions derived in equation (4.3), (4.4) and (4.10). We consider different scenarios (normal diffusion equation, neutral diffusion equation (Gorenflo et al., 2002; Metzler and Nonnenmacher, 2002; Luchko, 2012; Tarasov, 2019), space diffusion equation (Tarasov, 2019), and time diffusion equation (Tarasov, 2019)). In these experiments, we generate synthetic data corresponding to $N = 100$ trajectories simulated according to the diffusion model under study with a generalized diffusion coefficient $D = 1$. Note that the data generation procedure is presented in details in the supplementary

information document. In Figure 4.1, we present a panel of 4×3 plots where we refer by a row the scenario considered (normal, time, space, neutral diffusion) and we plot the statistical absolute moments, statistical signed absolute moments and the expected log absolute value of the process $X(t)$ for an order $\delta = 0.001$ versus time in the columns. The signed moment deviates a little from the theoretical expression for some scenarios, due to lack of sufficient samples. For the particular case of Figure 4.1e, we observe that there is nearly perfect match with the theoretical expression. The reason being, in this case, the parameters $\alpha = \theta$, which implies that the trajectories are generated from a negatively skewed alpha-stable distribution, and, therefore $X(t) \leq 0$, $\forall t$. We note that a similar situation happens when we have the parameters $\alpha = -\theta$. In this case, we have a positive skewed alpha distribution, and therefore, $X(t) \geq 0, \forall t$. In all scenarios, we can observe that the empirical statistical moments match perfectly the theoretical ones in all scenarios. This result confirms our theoretical derivations and motivates us to move forward with this approach.

Parameters Estimation of Synthetic Data: In this experiment, we validate the proposed approach using artificially generated spatiotemporal data according to the PDE model presented in equation (1). More precisely, we use the above-mentioned schemes (Algorithm 3 and Algorithm 4) to retrieve the parameters (α, β, θ and D) used during the data generation step. Figure 4.2 summarize several experiments done for different diffusion models (classical, neutral, space, and time diffusion), where we assume a set of combination of α, β and θ parameters for a generalized diffusion coefficient $D = 1$. We refer the reader to (Znaidi et al., 2020) for complete results with diffusion coefficients $D = 2, 5$. In the figure, we present a panel of 4×4 different plots where a row represents the type of diffusion considered and columns $1, 2, 3$ and 4 designate the parameters α, β, θ and D, respectively. In each of the sub-figure, we plot the estimated parameter using both algorithms (blue line for Algorithm 3 and the black line for Algorithm 4) versus the number of trajectories considered during the estimation process. We also plot the true value as a red line, and a narrow interval around the true value using black dashed lines. The presented blue and grey shaded regions represent the standard deviation for the estimated parameters associated to Algorithm 3 and Algorithm 4, respectively. All sub-figures in Figure 4.2 show that the proposed schemes are doing well in all scenarios where we are able to retrieve the exact set of parameters α, β, θ and D with a small/negligible error [2]. Also, we can see how the standard deviation of the estimated parameters decreases as the the number of trajectories increases. Furthermore, we can remark that Algorithm 4 is performing slightly better than Algorithm 3 in terms of rate of convergence. Although the variance is quite high when fewer trajectories are considered, we remark that in some scenarios we can get good estimates of the parameters even with reduced number of trajectories.

[2] Additional simulations for other scenarios are presented in the supplementary information document.

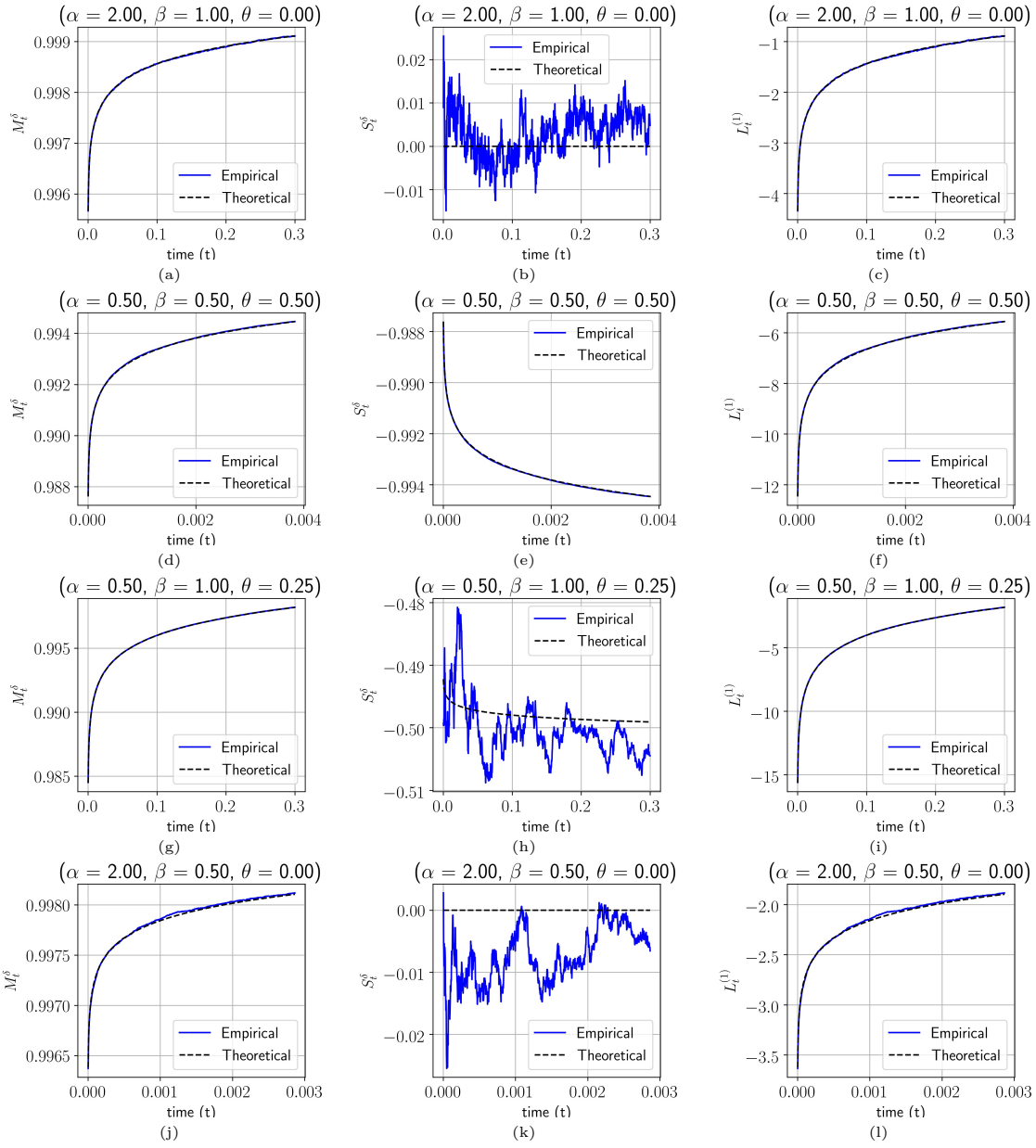


Figure 4.1: The time-dependent theoretical and empirical absolute moments and signed absolute moments of order $\delta = 0.001$, the time-dependent theoretical and empirical expected log absolute for the following four types of diffusion models: **(4.1a), (4.1b), (4.1c)** Normal diffusion equation, **(4.1d), (4.1e), (4.1f)** Neutral diffusion equation, **(4.1g), (4.1h), (4.1i)** Space fractional diffusion equation, **(4.1j), (4.1k), (4.1l)** Time fractional diffusion equation.

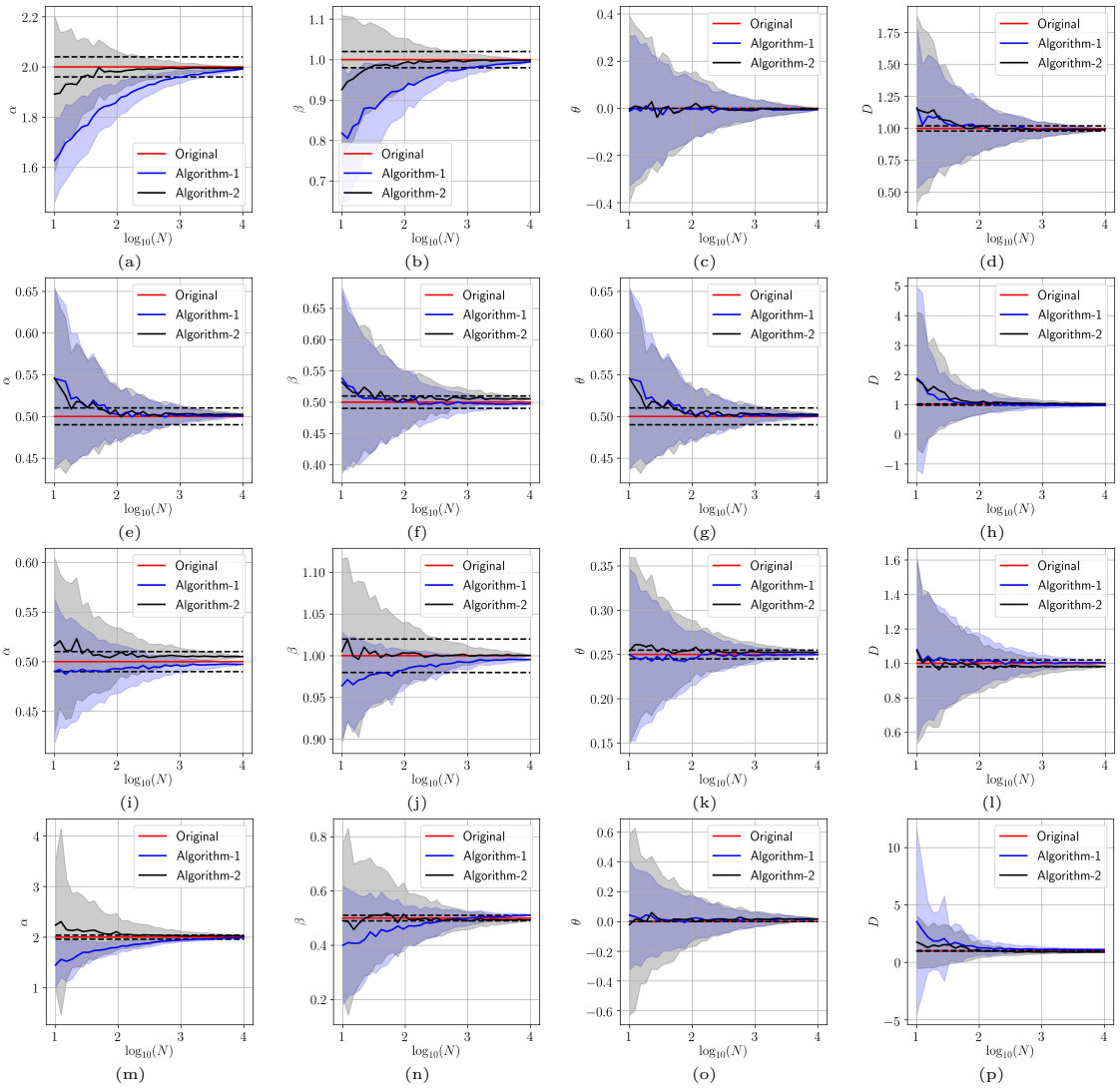


Figure 4.2: Determining the parameters of the space-time fractional diffusion via the two proposed algorithms while varying the number of trajectories and the generalized diffusion coefficient $D = 1$. The dotted line indicate 2% error tube around the original parameter value in the red: **(4.2a, 4.2b, 4.2c, 4.2d)** Normal diffusion equation ($\alpha = 2, \beta = 1, \theta = 0$), **(4.2e, 4.2f, 4.2g, 4.2h)** Neutral diffusion equation ($\alpha = 0.5, \beta = 0.5, \theta = 0.5$), **(4.2i, 4.2j, 4.2k, 4.2l)** Space fractional diffusion equation ($\alpha = 0.5, \beta = 1, \theta = 0.25$), **(4.2m, 4.2n, 4.2o, 4.2p)** Time fractional diffusion equation ($\alpha = 2, \beta = 0.5, \theta = 0$).

4.3 Discussion

Understanding complex dynamics remains a challenging task when their generative model is unknown. This task is more complicated when it comes to analyze spatiotemporal kinetics and infer the model that dictate their evolution. Although, the physics that drive the dynamics are unknowns, data-driven based approaches are prominent tools to discover the physical laws / rules governing complex observed dynamics (from heterogeneous, sparse, scarce or even noisy data). Indeed, such a discovery plays a crucial role in diverse fields ranging from system biology, neuroscience, econophysics to social studies. Towards addressing this goal, in this chapter, we have considered a generalized space-time fractional PDE and have developed an effective, rigorous and robust algorithmic strategies to estimate the parameters and so identify the main mathematical operators appearing in the PDE.

In contrast to prior work, we investigated the effectiveness and robustness of the proposed algorithmic approach for estimating the correct parameters as a function of the available number of trajectories. From our simulation results, we observe that for all considered types of diffusion models except the classical one (i.e., all combination of the parameters α, β, θ, and D), a few number of recorded time-series (less than 100 trajectories) is required to attain the correct estimation of the PDE parameters with less then 2% confidence interval. For the case of the normal diffusion (i.e., except for the case $\alpha = 2$ and $\beta = 1$), we may need more trajectories to achieve similar accuracy. Therefore, we hope that the proposed algorithms will help the community to better analyze complex spatiotemporal data, in order to unravel new physical laws in different applications (social networks, neuroscience, etc.) and decipher the causal interdependence between different processes.

This mathematical formalism can be further developed and generalized to include additional operators and take into account advection phenomena as well as combined with other advanced statistics and information theory inspired methods to discriminate among various mathematical expressions (operators) in order to either identify the dominant physical phenomenon (or rule) governing the measurements or to determine the degree to which multiple physical laws contribute to the observed dynamics. Also, analyzing noisy data originated from real world applications will be taken into account in order to cope with complex scenario. We plan to build on these grounds, enrich the mathematical formalism and contribute to a significant paradigm shift in the context of data-driven discovery architectures of physical phenomena as well as enabling accurate predictions concerning complex evolving systems without requiring to know the regimes of variation for parameters, the types of mathematical operators or the fact that the data should be sampled at a particular level.

Chapter 5

Operator Learning for Partial Differential Equations

We incorporate the multiwavelet filters derived using a variety of the orthogonal polynomial (OP) basis into our operator learning model, and show that the proposed architecture outperforms the existing neural operators. Our main contributions are as follows: **(i)** Based on some fundamental properties of the integral operator's kernel, we develop a multiwavelet-based model which learns the operator map efficiently. **(ii)** For the 1-D dataset of non-linear Korteweg-de Vries and Burgers equations, we observe an order of magnitude improvement in the relative $L2$ error (Section 5.2.1, 5.2.3). **(iii)** We demonstrate that the proposed model is in validation with the theoretical properties of the pseudo-differential operator (Section 5.2.2). **(iv)** We show how the proposed multiwavelet-based model is robust towards the fluctuation strength of the input signal (Section 5.2.1). **(v)** Next, we demonstrate the applicability on higher dimensions of 2-D Darcy flow equation (Section 5.2.4), and finally show that the proposed approach can learn at lower resolutions and generalize to higher resolutions.

5.1 Operator Learning using Multiwavelet Transform

We start by defining the problem of operator learning in Section 5.1.1. Section 5.1.2 defines the multiwavelet transform for the proposed operator learning problem and derives the necessary transformation operations across different scales. Section 5.1.3 outlines the proposed operator learning model. Finally, Section 5.1.4 lists some of the useful properties of the operators which leads to an efficient implementation of multiwavelet-based models.

5.1.1 Problem Setup

Given two functions $a(x)$ and $u(x)$ with $x \in D$, the operator is a map T such that $Ta = u$. Formally, let $\mathcal{A}$ and $\mathcal{U}$ be two Sobolev spaces $\mathcal{H}^{s,p}$ $(s > 0, p \geq 1)$, then the operator T is such that $T : \mathcal{A} \to \mathcal{U}$. The Sobolev spaces are particularly useful in the analysis of partial differential equations (PDEs), and we restrict our attention to $s > 0$ and $p = 2$. Note that, for $s = 0$, the $\mathcal{H}^{0,p}$ coincides with L^p, and, $f \in \mathcal{H}^{0,p}$ does not necessarily have derivatives in L^p. We choose $p = 2$ in order to be able to define projections with respect to (w.r.t.) measures μ in a Hilbert space structure.

We take the operator T as an integral operator with the kernel $K : D \times D \to L^2$ such that

$$Ta(x) = \int_D K(x, y)a(y)dy. \tag{5.1}$$

For the case of inhomogeneous linear PDEs, $\mathcal{L}u = f$, with f being the forcing function, $\mathcal{L}$ is the differential operator, and the associated kernel is commonly termed as Green function. In our case, we do not put the restriction of linearity on the operator. From eq. (5.1), it is apparent that learning the complete kernel $K(.,.)$ would essentially solve the operator map problem, but it is not necessarily a numerically feasible solution. Indeed, a better approach would be to exploit possible useful properties (see Section 5.1.4) such that a compact representation of the kernel can be made. For an efficient representation of the operator kernel, we need an appropriate subspace (or sequence of subspaces), and projection tools to map to such spaces.

Norm with respect to measures: Projecting a given function onto a fixed basis would require a measure dependent distance. For two functions f and g, we take the inner product w.r.t measure μ as $\langle f, g\rangle_\mu = \int f(x)g(x)d\mu(x)$, and the associated norm as $||f||_\mu = \langle f, f\rangle_\mu^{1/2}$. We now discuss the next ingredient, which refers to the subspaces required to project the kernel.

5.1.2 Multiwavelet Transform

In this section, we briefly overview the concept of multiwavelets (Alpert et al., 2002) and extend it to work with non-uniform measures at each scale. The multiwavelet transform synergizes the advantages of *orthogonal polynomials* (OPs) as well as the *wavelets* concepts, both of which have a rich history in the signal processing. The properties of wavelet bases like (i) vanishing moments, and (ii) orthogonality can effectively be used to create a system of coordinates in which a wide class of operators (see Section 5.1.4) have *nice* representation. Multiwavelets go few steps further, and provide a fine-grained representation using OPs, but also act as basis on a finite interval. For the rest of this section, we restrict our attention to the interval $[0, 1]$;

however, the transformation to any finite interval $[a, b]$ could be straightforwardly obtained by an appropriate shift and scale.

Multi Resolution Analysis: We begin by defining the space of piecewise polynomial functions, for $k \in \mathbb{N}$ and $n \in \mathbb{Z}^+ \cup \{0\}$ as, $\mathbf{V}_n^k = \bigcup_{l=0}^{2^n-1}\{f|\deg(f) < k \text{ for } x \in (2^{-n}l, 2^{-n}(l+1)) \wedge 0, \text{ elsewhere}\}$. Clearly, $\dim(\mathbf{V}_n^k) = 2^n k$, and for subsequent n, each subspace is contained in another as shown by the following relation:

$$\mathbf{V}_0^k \subset \mathbf{V}_1^k \ldots \subset \mathbf{V}_{n-1}^k \subset \mathbf{V}_n^k \subset \ldots. \tag{5.2}$$

Similarly, we define the sequence of measures $\mu_0, \mu_1, \ldots$ such that $f \in \mathbf{V}_n^k$ is measurable w.r.t. μ_n and the norm of f is taken as $||f|| = \langle f, f\rangle_{\mu_n}^{1/2}$. Next, since $\mathbf{V}_{n-1}^k \subset \mathbf{V}_n^k$, we define the multiwavelet subspace as $\mathbf{W}_n^k$ for $n \in \mathbb{Z}^+ \cup \{0\}$, such that

$$\mathbf{V}_{n+1}^k = \mathbf{V}_n^k \bigoplus \mathbf{W}_n^k, \quad \mathbf{V}_n^k \perp \mathbf{W}_n^k. \tag{5.3}$$

For a given OP basis for $\mathbf{V}_0^k$ as $\phi_0, \phi_1, \ldots, \phi_{k-1}$ w.r.t. measure μ_0, a basis of the subsequent spaces $\mathbf{V}_n^k, n > 1$ can be obtained by shift and scale (hence the name, multi-scale) operations of the original basis as follows:

$$\phi_{jl}^n(x) = 2^{n/2}\phi_j(2^n x - l), \quad j = 0, 1, \ldots, k-1, \quad l = 0, 1, \ldots, 2^n - 1, \text{w.r.t.} \quad \mu_n, \tag{5.4}$$

where, μ_n is obtained as the collections of shift and scale of μ_0, accordingly.

Multiwavelets: For the multiwavelet subspace $\mathbf{W}_0^k$, the orthonormal basis (of piecewise polynomials) are taken as $\psi_0, \psi_1, \ldots, \psi_{k-1}$ such that $\langle \psi_i, \psi_j\rangle_{\mu_0} = 0$ for $i \neq j$ and 1, otherwise. From eq. (5.3), $\mathbf{V}_n^k \perp \mathbf{W}_n^k$, and since $\mathbf{V}_n^k$ spans the polynomials of degree at most k, therefore, we conclude that

$$\int_0^1 x^i \psi_j(x) d\mu_0(x) = 0, \quad \forall\, 0 \leq j, i < k. \qquad \text{(vanishing moments)} \tag{5.5}$$

Similarly to eq. (5.4), a basis for multiwavelet subspace $\mathbf{W}_n^k$ are obtained by shift and scale of ψ_i as $\psi_{jl}^n(x) = 2^{n/2}\psi_j(2^n x - l)$ and ψ_{jl}^n are orthonormal w.r.t. measure μ_n, i.e. $\langle \psi_{jl}^n, \psi_{j'l'}^n\rangle_{\mu_n} = 1$ if $j = j', l = l'$, and 0 otherwise. Therefore, for a given OP basis for $\mathbf{V}_0^k$ (for example, Legendre, Chebyshev polynomials), we only require to compute ψ_i, and a complete basis set at all the scales can be obtained using scale/shift of ϕ_i, ψ_i.

Note: Since $\mathbf{V}_1^k = \mathbf{V}_0^k \bigoplus \mathbf{W}_0^k$ from eq. (5.3), therefore, for a given basis ϕ_i of $\mathbf{V}_0^k$ w.r.t. measure μ_0 and ϕ_{jl}^n as a basis for $\mathbf{V}_1^k$ w.r.t. μ_1, a set of basis ψ_i can be obtained by applying Gram-Schmidt orthogonalization using appropriate measures. We refer the reader to supplementary materials for the detailed procedure.

Note: Since $\mathbf{V}_0^k$ and $\mathbf{W}_0^k$ lives in $\mathbf{V}_1^k$, therefore, ϕ_i, ψ_i can be written as a linear combination of the basis of V_1^k. We term these linear coefficients as multiwavelet decomposition filters $(H^{(0)}, H^{(1)}, G^{(0)}, G^{(1)})$, since they are transforming a fine $n = 1$ to coarse scale $n = 0$. A uniform measure (μ_0) version is discussed in (Alpert et al., 2002), and we extend it to any arbitrary measure by including the correction terms $\Sigma^{(0)}$ and $\Sigma^{(1)}$. We refer to Appendix D.2 for the complete details. The capability of using the non-uniform measures enables us to apply the same approach to any OP basis with finite domain, for example, Chebyshev, Gegenbauer, etc.

For a given $f(x)$, the multiscale, multiwavelet coefficients at the scale n are defined as $\mathbf{s}_l^n = [\langle f, \phi_{il}^n \rangle_{\mu_n}]_{i=0}^{k-1}$, $\mathbf{d}_l^n = [\langle f, \psi_{il}^n \rangle_{\mu_n}]_{i=0}^{k-1}$, respectively, w.r.t. measure μ_n with $\mathbf{s}_l^n, \mathbf{d}_l^n \in \mathbb{R}^{k \times 2^n}$. The decomposition / reconstruction across scales is written as

$$\mathbf{s}_l^n = H^{(0)} \mathbf{s}_{2l}^{n+1} + H^{(1)} \mathbf{s}_{2l+1}^{n+1}, \tag{5.6}$$

$$\mathbf{d}_l^n = G^{(0)} \mathbf{s}_{2l}^{n+1} + H^{(1)} \mathbf{s}_{2l+1}^{n+1}, \tag{5.7}$$

$$\mathbf{s}_{2l}^{n+1} = \Sigma^{(0)} (H^{(0)\,T} \mathbf{s}_l^n + G^{(0)\,T} \mathbf{d}_l^n), \tag{5.8}$$

$$\mathbf{s}_{2l+1}^{n+1} = \Sigma^{(1)} (H^{(1)\,T} \mathbf{s}_l^n + G^{(1)\,T} \mathbf{d}_l^n). \tag{5.9}$$

The wavelet (and also multiwavelet) transformation can be straightforwardly extended to multiple dimensions using tensor product of the bases. For our purpose, a function $f \in \mathbb{R}^d$ has multiscale, multiwavelet coefficients $\mathbf{s}_l^n, \mathbf{d}_l^n \in \mathbb{R}^{k \times \ldots \times k \times 2^n}$ which are also recursively obtained by replacing the filters in eq. (5.6)-(5.7) with their Kronecker product, specifically, $H^{(0)}$ with $H^{(0)} \otimes H^{(0)} \otimes \ldots H^{(0)}$, where $\otimes$ is the Kronecker product repeated d times. For eq. (5.8)-(5.9) $H^{(0)}\Sigma^{(0)}$ (and similarly others) are replaced with their d-times Kronecker product.

Non-Standard Form: The multiwavelet representation of the operator kernel $K(x, y)$ can be obtained by an appropriate tensor product of the multiscale and multiwavelet basis. One issue, however, in this approach, is that the basis at various scales are *coupled* because of the tensor product. To untangle the basis at various scales, we use a trick as proposed in (Beylkin, Coifman, and Rokhlin, 1991) called the non-standard wavelet representation. The extra mathematical price paid for the non-standard representation, actually serves as a ground for reducing the proposed model complexity (see Section 5.1.3), thus, providing data efficiency. For the operator under consideration T with integral kernel $K(x, y)$, let us denote T_n as the projection of T on V_n^k, which essentially is obtained by projecting the kernel K onto basis ϕ_{jl}^n w.r.t. measure μ_n. If P_n is the projection operator such that $P_n f = \sum_{j,l} \langle f, \phi_{jl}^n \rangle_{\mu_n} \phi_{jl}^n$, then $T_n = P_n T P_n$. Using telescopic sum, T_n is expanded as

$$T_n = \sum_{i=L+1}^{n} (Q_i T Q_i + Q_i T P_{i-1} + P_{i-1} T Q_i) + P_L T P_L, \tag{5.10}$$

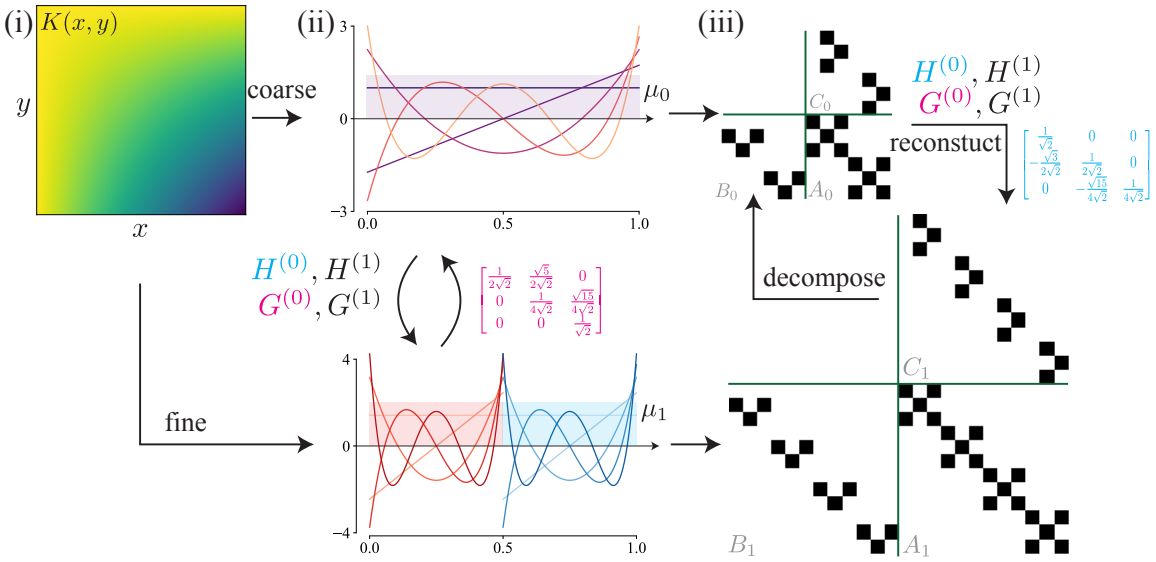


Figure 5.1: **Multiwavelet representation of the Kernel**. (i) Given kernel $K(x, y)$ of an integral operator T, (ii) the bases with different measures (μ_0, μ_1) at two different scales (coarse=0, fine=1) projects the kernel into 3 components A_i, B_i, C_i. (iii) The decomposition yields a sparse structure, and the entries with absolute magnitude values exceeding $1e^{-8}$ are shown in black. Given projections at any scale, the finer / coarser scale projections can be obtained by reconstruction / decomposition using a fixed multiwavelet filters $H^{(i)}$ and $G^{(i)}, i = 0, 1$.

where, $Q_i = P_i - P_{i-1}$ and L is the coarsest scale under consideration ($L \geq 0$). From eq. (5.3), it is apparent that Q_i is the multiwavelet operator. Next, we denote $A_i = Q_i T Q_i, B_i = Q_i T P_{i-1}, C_i = P_{i-1} T Q_i$, and $\bar{T} = P_L T P_L$. In Figure 5.1, we show the non-standard multiwavelet transform for a given kernel $K(x, y)$. The transformation has a sparse banded structure due to smoothness property of the kernel (see Section 5.1.4). For the operator T such that $Ta = u$, the map under multiwavelet domain is written as

$$\begin{aligned} U^n_{dl} &= A_n d^n_l + B_n s^n_l, \\ U^n_{\hat{s}l} &= C_n d^n_l, \\ U^L_{sl} &= \bar{T} s^L_l, \end{aligned} \tag{5.11}$$

where, $(U^n_{sl}, U^n_{dl})/(s^n_l, d^n_l)$ are the multiscale, multiwavelet coefficients of u/a, respectively, and L is the coarsest scale under consideration. With these mathematical concepts, we now proceed to define our multiwavelet-based operator learning model in the Section 5.1.3.

5.1.3 Multiwavelet-based Model

Based on the discussion in Section 5.1.2, we propose a multiwavelet-based model (MWT) as shown in Figure 5.2. For a given input/output as a/u, the goal of the

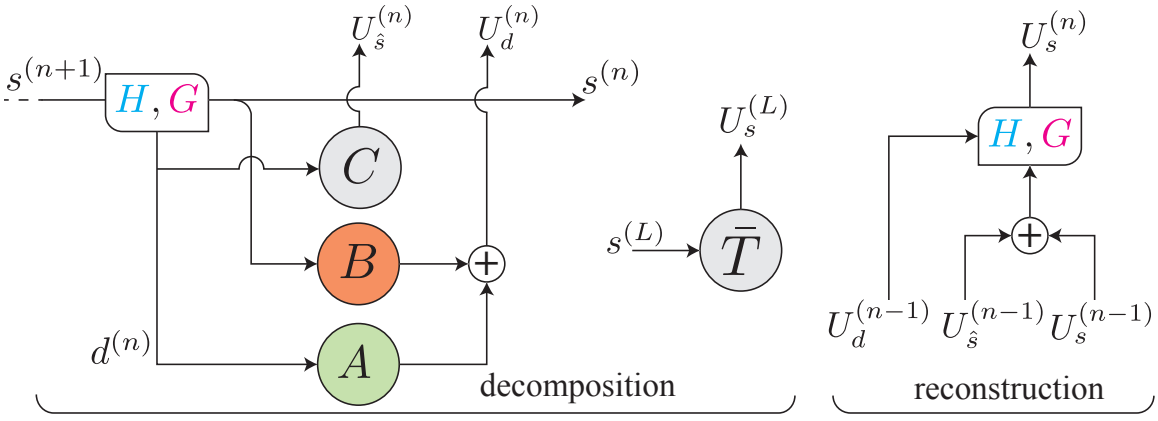


Figure 5.2: **MWT model architecture**. **(Left)** Decomposition cell using 4 neural networks (NNs) A, B and C, and T (for the coarsest scale L) performs multiwavelet decomposition from scale $n+1$ to n. **(Right)** Reconstruction module using pre-defined filters $H^{(i)}, G^{(i)}$ performs inverse multiwavelet transform from scale $n-1$ to n.

MWT model is to map the multiwavelet-transform of the input (s_l^N) to output (U_{sl}^N) at the finest scale N. The model consists of two parts: (i) Decomposition (dec), and (ii) Reconstruction (rec). The *dec* acts as a recurrent network, and at each iteration the input is s^{n+1}. Using (5.6)-(5.7), the input is used to obtain multiscale and multiwavelet coefficients at a coarser level s^n and d^n, respectively. Next, to compute the multiscale/multiwavelet coefficients of the output u, we approximate the non-standard kernel decomposition from (5.11) using four neural networks (NNs) A, B, C and $\bar{T}$ such that $U_{dl}^n \approx A_{\theta_A}(d_l^n) + B_{\theta_B}(s_l^n), U_{\hat{s}l}^n \approx C_{\theta_C}(d_l^n), \forall\, 0 \leq n < L$, and $U_{sl}^L \approx \bar{T}_{\theta_{\bar{T}}}(s_l^L)$. This is a ladder-down approach, and the *dec* part performs the decimation of signal (factor 1/2), running for a maximum of L cycles, $L < \log_2(M)$ for a given input sequence of size M. Finally, the *rec* module collects the constituent terms $U_{sl}^n, U_{\hat{s}l}^n, U_{dl}^n$ (obtained using the *dec* module) and performs a ladder-up operation to compute the multiscale coefficients of the output at a finer scale $n+1$ using (5.8)-(5.9). The iterations continue until the finest scale N is obtained for the output.

At each iteration, the filters in *dec* module downsamples the input, but compared to popular techniques (e.g., maxpool), the input is only transformed to a coarser multiscale/multiwavelet space. By virtue of its design, since the non-standard wavelet representation does not have inter-scale interactions, it basically allows us to reuse the same kernel NNs A, B, C at different scales. A follow-up advantage of this approach is that the model is resolution independent, since the recurrent structure of *dec* is input invariant, and for a different input size M, only the number of iterations would possibly change for a maximum of $\log_2 M$. The reuse of A, B, C by re-training at various scales also enable us to learn an expressive model with fewer parameters $(\theta_A, \theta_B, \theta_C, \theta_{\bar{T}})$. We see in Section 5.2, that even a single-layered CNN for A, B, C is sufficient for learning the operator.

The *dec* / *rec* module uses the filter matrices which are fixed beforehand, therefore, this part does not require any training. The model does not work for any arbitrary

choice of fixed matrices H, G. We show in Section 5.2.4 that for randomly selected matrices, the model does not learn, which validates that careful construction of filter matrices is necessary.

5.1.4 Operators Properties

This section outlines the definition of the integral kernels that are typically useful in an efficient compression of the operators through multiwavelets. We then discuss a fundamental property of the pseudo-differential operator.

Definition 5.1 ((Meyer, Coifman, and Salinger, 1997)). ***Calderón-Zygmund Operator***. *The integral operators that have kernel $K(x, y)$ which is smooth away from the diagonal, and satisfy the following.*

$$\begin{aligned} |K(x,y)| &\leq \frac{1}{|x-y|}, \\ |\partial_x^M K(x,y)| + |\partial_y^M K(x,y)| &\leq \frac{C_0}{|x-y|^{M+1}}. \end{aligned} \tag{5.12}$$

The smooth functions with decaying derivatives are *gold* to the multiwavelet transform. Note that, smoothness implies Taylor series expansion, and the multiwavelet transform with sufficiently large k zeroes out the initial k terms of the expansion due to vanishing moments property (5.5). This is how multiwavelet sparsifies the kernel (see Figure 5.1 where $K(x, y)$ is smooth). Although, the definition of Calderón-Zygmund is simple (singularities only at the diagonal), but the multiwavelets are capable to compresses the kernel as long as the *number of singularities are finite*.

The next property, from (Chou and Guthart, 2000), points out that with input/output being single-dimensional functions, for any pseudo-differential operator (with smooth coefficients), the singularity at the diagonal is also well-characterized.

Property 5.1. ***Smoothness of Pseudo-Differential Operator.*** *For the integral kernel K(x,y) of a pseudo-differential operator, $K(x, y) \in C^\infty \; \forall x \neq y$, and for $x = y$, $K(x, y) \in C^{T-1}$, where $T + 1$ is the highest derivative order in the given pseudo-differential equation.*

Property 5.1 implies that, for the class of pseudo-differential operator, and any set of basis with the initial J vanishing moments, the projection of kernel onto such bases will have the diagonal dominating the non-diagonal entries, exponentially, if $J > T - 1$ (Chou and Guthart, 2000). For the case of multiwavelet basis with k OPs,

$J = k$ (from eq. (5.5)). Therefore, $k > T - 1$ sparsifies the kernel projection onto multiwavelets, for a fixed number of bits precision ϵ. We see the implication of the Property 5.1 on our proposed model in the Section 5.2.2.

5.2 Empirical Evaluation

In this section, we evaluate the multiwavelet-based model (MWT) on several PDE datasets. We show that the proposed MWT model not only exhibits orders of magnitude higher accuracy when compared against the state-of-the-art (Sota) approaches but also works consistently well under different input conditions without parameter tuning. From a numerical perspective, we take the data as point-wise evaluations of the input and output functions. Specifically, we have the dataset (a_i, u_i) with $a_i = a(x_i), u_i = u(x_i)$ for $x_1, x_2, \ldots, x_N \in D$, where x_i are M-point discretization of the domain D. Unless stated otherwise, the training set is of size 1000 while test is of size 200.

Model architectures: Unless otherwise stated, the NNs A, B and C in the proposed model (Figure 5.2) are chosen as a single-layered CNNs following a linear layer, while $\bar{T}$ is taken as single $k \times k$ linear layer. We choose $k = 4$ in all our experiments, and the OP basis as Legendre (Leg), Chebyshev (Chb) with uniform, non-uniform measure μ_0, respectively. The model in Figure 5.2 is treated as single layer, and for 1D equations, we cascade 2 multiwavelet layers, while for 2D dataset, we use a total 4 layers with $ReLU$ non-linearity.

From a mathematical viewpoint, the *dec* and *rec* modules in Figure 5.2 transform only the multiscale and multiwavelet coefficients. However, the input and output to the model are point-wise function samples, i.e., (a_i, u_i). A remedy around this is to take the data sequence, and construct hypothetical functions $f_a = \sum_{i=1}^{N} a_i \phi_{ji}^n$ and $f_u = \sum_{i=1}^{N} u_i \phi_{ji}^n$. Clearly, f_a, f_u lives in V_n^k with $n = \log_2 N$. Now the model can be used with $s^{(n)} = a_i$ and $U_s^{(n)} = u_i$. Note that f_a, f_u are not explicitly used, but only a matter of convention.

Benchmark models: We compare our **MWT** model using two different OP basis (Leg, Chb) with the most recent successful neural operators. Specifically, we consider the graph neural operator (**GNO**) (Li et al., 2020b), the multipole graph neural operator (**MGNO**) (Li et al., 2020c), the **LNO** which makes a low-rank (r) representation of the operator kernel $K(x, y)$ (also similar to unstacked DeepONet (Lu, Jin, and Karniadakis, 2020)), and the Fourier neural operator (**FNO**) (Li et al., 2020a). We experiment on three competent datasets setup by the work of FNO (Burgers' equation (1-D), Darcy Flow (2-D), and Navier-Stokes equation (time-varying 2-D)). In addition, we also experiment with Korteweg-de Vries equation (1-D). For the 1-D cases, a

Networks	s = 64	s = 128	s = 256	s = 512	s = 1024
MWT Leg	**0.00338**	**0.00375**	**0.00418**	**0.00393**	**0.00389**
MWT Cheb	0.00715	0.00712	0.00604	0.00769	0.00675
FNO	0.0125	0.0124	0.0125	0.0122	0.0126
MGNO	0.1296	0.1515	0.1355	0.1345	0.1363
LNO	0.0429	0.0557	0.0414	0.0425	0.0447
GNO	0.0789	0.0760	0.0695	0.0699	0.0721

Table 5.1: Korteweg-de Vries (KdV) equation benchmarks for different input resolution s. Top: Our methods. Bottom: previous works of Neural operator.

modified FNO with careful parameter selection and removal of Batch-normalization layers results in a better performance compared with the original FNO, and we use it in our experiments. The MWT model demonstrates the highest accuracy in all the experiments. The MWT model also shows the ability to learn the function mapping through lower-resolution data, and able to generalize to higher resolutions.

All the models (including ours) are trained for a total of 500 epochs using Adam optimizer with an initial learning rate (LR) of 0.001. The LR decays after every 100 epochs with a factor of $\gamma = 0.5$. The loss function is taken as relative $L2$ error (Li et al., 2020a). All of the experiments are performed on a single Nvidia V100 32 GB GPU, and the results are averaged over a total of 3 seeds.

5.2.1 Korteweg-de Vries (KdV) Equation

The Korteweg-de Vries (KdV) equation was first proposed by Boussinesq (Boussinesq, 1877) and rediscovered by Korteweg and de Vries (Darrigol, 2005). KdV is a 1-D non-linear PDE commonly used to describe the non-linear shallow water waves. For a given field $u(x,t)$, the dynamics takes the following form:

$$\begin{aligned} \frac{\partial u}{\partial t} &= -0.5u\frac{\partial u}{\partial x} - \frac{\partial^3 u}{\partial x^3}, x \in (0,1), t \in (0,1] \\ u_0(x) &= u(x, t=0) \end{aligned} \tag{5.13}$$

The task for the neural operator is to learn the mapping of the initial condition $u_0(x)$ to the solutions $u(x, t = 1)$. We generate the initial condition in Gaussian random fields according to $u_0 \sim \mathcal{N}(0, 7^4(-\Delta + 7^2 I)^{-2.5})$ with periodic boundary conditions. The equation is numerically solved using *chebfun* package (Driscoll, Hale, and Trefethen, 2014) with a resolution 2^{10}, and datasets with lower resolutions are obtained by sub-sampling the highest resolution data set.

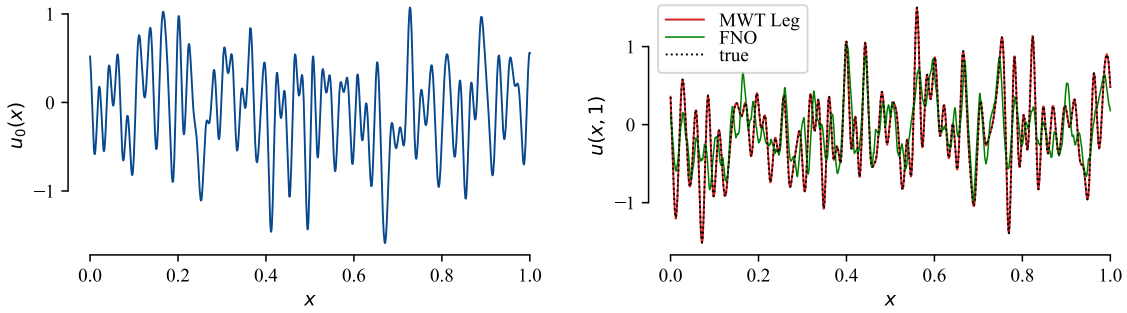


Figure 5.3: **The output of the KdV equation**. (Left) An input $u_0(x)$ with $\lambda = 0.02$. (Right) The predicted output of the MWT Leg model learning the high fluctuations.

Varying resolution: The experimental results of the KdV equation for different input resolutions s are shown in Table 5.1. We see that, compared to any of the benchmarks, our proposed MWT Leg exhibits the lowest relative error and is lowest nearly by an order of magnitude. Even in the case of the resolution of 64, the relative error is low, which means that a sparse data set with a coarse resolution of 64 is sufficient for the neural operator to learn the function mapping between infinite-dimensional spaces.

Varying fluctuations: We now vary the smoothness of the input function $u_0(x, 0)$ by controlling the parameter λ, where low values of λ imply more frequent fluctuations and $\lambda \to 0$ reaches the Brownian motion limit (Filip, Javeed, and Trefethen, 2019). To isolate the importance of incorporating the multiwavelet transformation, we use the same convolution operation as in FNO, i.e., Fourier transform-based convolution with different modes k_m (only single-layer) for A, B, C. We see in Figure 5.3 that MWT model consistently outperforms the recent baselines for all the values of λ considered. A sample input/output from test set is shown in the Figure 5.3. The FNO model with higher values of k_m has better performance due to more Fourier bases for representing the high-frequency signal, while MWT does better even with low modes in its A, B, C CNNs, highlighting the importance of using wavelet-based filters in the signal processing.

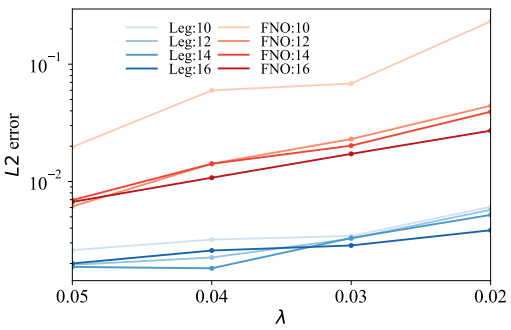


Figure 5.4: Comparing MWT by varying the degree of fluctuations λ in the input with resolution $s = 1024$. For each convolution, we fix the number of Fourier bases as k_m. For FNO, the width is 64.

5.2.2 Theoretical Properties Validation

We test the ability of the proposed MWT model to capture the theoretical properties of the pseudo-differential operator hereafter. Towards that, we consider the Euler-Bernoulli equation (Timoshenko, 1983) that models the vertical displacement of a finite length beam over time. A Fourier transform version of the beam equation with the constraint of both ends being clamped is as follows:

$$\begin{aligned} \frac{\partial^4 u}{\partial x} - \omega^2 u = f(x), \quad \frac{\partial u}{\partial x}\Big|_{\substack{x=1\\x=0}} = 0, \\ u(0) = u(1) = 0, \end{aligned} \tag{5.14}$$

where $u(x)$ is the Fourier transform of the time-varying beam displacement, ω is the frequency, $f(x)$ is the applied force. The Euler-Bernoulli is a pseudo-differential equation with the maximum derivative order $T+1=4$. We take the task of learning the map from f to u. In Figure 5.5, we see that for $k \geq 3$, the models relative error across epochs is similar, however, they are different for $k < 3$, which is in accordance with the Property 5.1. For $k < 3$, the multiwavelets will not be able to annihilate the diagonal of the kernel which is C^{T-1}, hence, sparsification cannot occur, and the model learns slow.

5.2.3 Burgers' Equation

The 1-D Burgers' equation is a non-linear PDE occurring in various areas of applied mathematics. For a given field $u(x,t)$ and diffusion coefficient v, the 1-D Burgers' equation reads as

$$\begin{aligned} \frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} + v\frac{\partial^2 u}{\partial x^2}, \quad x \in (0, 2\pi), \quad t \in (0, 1], \\ u_0(x) = u(x, t=0). \end{aligned} \tag{5.15}$$

The task for the neural operator is to learn the mapping of initial condition $u(x, t=0)$ to the solutions at $t = 1$ $u(x, t = 1)$. To compare with many advanced neural operators under the same conditions, we use the Burgers' data and the results that have been published in (Li et al., 2020a) and (Li et al., 2020c). The initial condition is sampled as Gaussian random fields where $u_0 \sim \mathcal{N}(0, 5^4(-\Delta + 5^2 I)^{-2})$ with periodic boundary conditions. Δ is the Laplacian, meaning the initial conditions are sampled by sampling its first several coefficients from a Gaussian distribution. In the Burgers' equation, v is set to 0.1. The equation is solved with resolution 2^{13}, and the data with lower resolutions are obtained by sub-sampling the highest resolution data set.

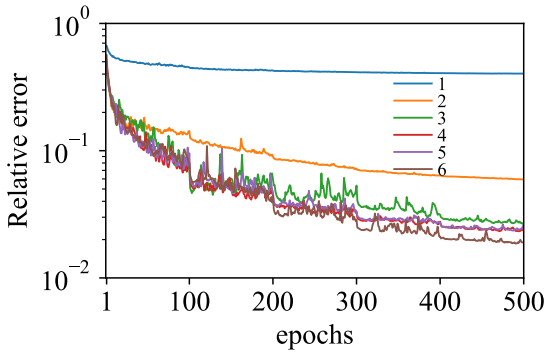


Figure 5.5: Relative $L2$ error vs epochs for MWT Leg with different number of OP basis $k = 1, \ldots, 6$.

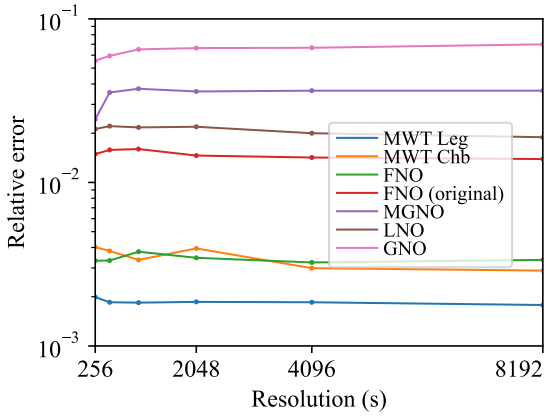


Figure 5.6: Burgers' Equation validation at various input resolution s. Our methods: MWT Leg, Chb.

The results of the experiments on Burgers' equation for different resolutions are shown in Figure 5.6. Compared to any of the benchmarks, our MWT Leg obtains the lowest relative error, which is an order of magnitude lower than the state-of-the-art. It's worth noting that even in the case of low resolution, MWT Leg still maintains a very low error rate, which shows its potential for learning the function mapping through low-resolution data, that is, the ability to map between infinite-dimensional spaces by learning a limited finite-dimensional spaces mapping.

5.2.4 Darcy Flow

Darcy flow formulated by Darcy (Darcy, 1856) is one of the basic relationships of hydrogeology, describing the flow of a fluid through a porous medium. We experiment on the steady-state of the 2-D Darcy flow equation on the unit box, where it takes the following form:

$$\begin{aligned} \nabla \cdot (a(x) \nabla u(x)) &= f(x), \quad && x \in (0,1)^2 \\ u(x) &= 0, && x \in \partial(0,1)^2 \end{aligned} \tag{5.16}$$

We set the experiments to learn the operator mapping the coefficient $a(x)$ to the solution $u(x)$. The coefficients are generated according to $a \sim \mathcal{N}(0, (-\Delta + 3^2 I)^{-2})$, where Δ is the Laplacian with zero Neumann boundary conditions. The threshold of $a(x)$ is set to achieve ellipticity. The solutions $u(x)$ are obtained by using a 2nd-order finite difference scheme on a 512×512 grid. Data sets of lower resolution are sub-sampled from the original data set.

The results of the experiments on Darcy Flow for different resolutions are shown in Table 5.2. MWT Leg again obtains the lowest relative error compared to other neural

Networks	s = 32	s = 64	s = 128	s = 256	s=512
MWT Leg	**0.0152**	**0.00899**	**0.00747**	**0.00722**	**0.00654**
MWT Chb	0.0174	0.0108	0.00872	0.00892	0.00891
MWT Rnd	0.2435	0.2434	0.2434	0.2431	0.2432
FNO	0.0177	0.0121	0.0111	0.0107	0.0106
MGNO	0.0501	0.0519	0.0547	0.0542	-
LNO	0.0524	0.0457	0.0453	0.0428	-

Table 5.2: Benchmarks on Darcy Flow equation at various input resolution s. Top: Our methods. MWT Rnd instantiate random entries of the filter matrices in (5.6)-(5.9). Bottom: prior works on Neural operator.

operators at various resolutions. We also perform an additional experiment, in which the multiwavelet filters $H^{(i)}, G^{(i)}, i = 0, 1$ are replaced with random values (properly normalized). We see in Table 5.2, that MWT Rnd does not learn the operator map, in fact, its performance is worse than all the models. This signifies the importance of the careful choice of the filter matrices.

5.2.5 Navier-Stokes Equation

Navier-Stokes equations (Acheson, 1991; Batchelor and Batchelor, 2000) describe the motion of viscous fluid substances, which can be used to model the ocean currents, the weather, and air flow. We experiment on the 2-D Navier-Stokes equation for a viscous, incompressible fluid in vorticity form on the unit torus, where it takes the following form:

$$\begin{aligned} \frac{\partial w(x,t)}{\partial t} + u(x,t)\cdot\nabla w(x,t) - \nu\Delta w(x,t) &= f(x), && x\in(0,1)^2, t\in(0,T], \\ \nabla\cdot u(x,t) &= 0, && x\in(0,1)^2, t\in[0,T], \\ w_0(x) &= w(x,t=0), && x\in(0,1)^2 \end{aligned} \tag{5.17}$$

We set the experiments to learn the operator mapping the vorticity w up to time 10 to w at a later time $T > 10$. More specifically, task for the neural operator is to map the first T time units to last $T - 10$ time units of vorticity w. To compare with the state-of-the-art model FNO (Li et al., 2020a) and other configurations under the same conditions, we use the same Navier-Stokes' data and the results that have been published in (Li et al., 2020a). The initial condition is sampled as Gaussian random fields where $w_0 \sim \mathcal{N}(0, 7^{\frac{3}{2}}(-\Delta + 7^2 I)^{-2.5})$ with periodic boundary conditions. The forcing function $f(x) = 0.1(\sin(2\pi(x_1 + x_2)) + \cos(2\pi(x_1 + x_2)))$. The experiments are conducted with ① the viscosities $\nu = 1e-3$, the final time $T = 50$, the number of training pairs $N = 1000$; ② $\nu = 1e-4, T = 30, N = 1000$; ③ $\nu = 1e-4, T =$

Networks	$\nu = 1e-3$ $T = 50$ $N = 1000$	$\nu = 1e-4$ $T = 30$ $N = 1000$	$\nu = 1e{-}4$ $T = 30$ $N = 10000$	$\nu = 1e-5$ $T = 20$ $N = 1000$
MWT Leg	**0.00625**	**0.1518**	**0.0667**	**0.1541**
MWT Chb	0.00720	0.1574	0.0720	0.1667
FNO-3D	0.0086	0.1918	0.0820	0.1893
FNO-2D	0.0128	0.1559	0.0973	0.1556
U-Net	0.0245	0.2051	0.1190	0.1982
TF-Net	0.0225	0.2253	0.1168	0.2268
Res-Net	0.0701	0.2871	0.2311	0.2753

Table 5.3: Navier-Stokes Equation validation at various viscosities ν. Top: Our methods. Bottom: previous works of Neural operators and other deep learning models.

$30, N = 10000$; ④ $\nu = 1e-5, T = 20, N = 1000$. The data sets are generated on a 256×256 grid and are subsampled to 64×64.

We see in Table 5.3 that the proposed MWT Leg outperforms the existing Neural operators as well as other deep NN benchmarks. The MWT models have used a 2-D multiwavelet transform with $k = 3$ for the vorticity w, and 3-D convolutions in the A, B, C NNs for estimating the time-correlated kernels. The MWT models (both Leg and Chb) are trained for 500 epochs for all the experiments except for $N = 10000, T = 30, \nu = 1e-4$ case where the models are trained for 200 epochs. Note that similar to FNO-2D, a time-recurrent version of the MWT models could also be trained and most likely will improve the resulting $L2$ error for the less data setups like $N = 1000, \nu = 1e-4$ and $N = 1000, \nu = 1e-5$. However, in this work we have only experimented with the 3d convolutions (for A, B, C) version.

5.2.6 Prediction at Higher resolutions

The proposed multiwavelets-based operator learning model is resolution-invariant by design. Upon learning an operator map between the function spaces, the proposed models have the ability to generalize beyond the training resolution. Next, we evaluate the resolution extension property of the MWT models using the Burgers' equation dataset as described in the Section 5.2.3. A pipeline for the experiment is shown in Figure 5.7. The numerical results for the experiments are shown in Table 5.4. We see that on training with a lower resolution, for example, $s = 256$, the prediction error at 10X higher resolution $s = 2048$ is 0.0226, or 2.26%. A sample input/output for learning at $s = 256$ while predicting a $s = 8192$ resolution is shown in Figure 5.7. Also, learning at an even coarser resolution of $s = 128$, the proposed model can

Train \ Test	s = 2048	s = 4096	s = 8192
s=128	0.0368	0.0389	0.0456
s=256	0.0226	0.0281	0.0321
s=512	0.0140	0.0191	0.0241

Table 5.4: MWT Leg model trained at lower resolutions can predict the output at higher resolutions.

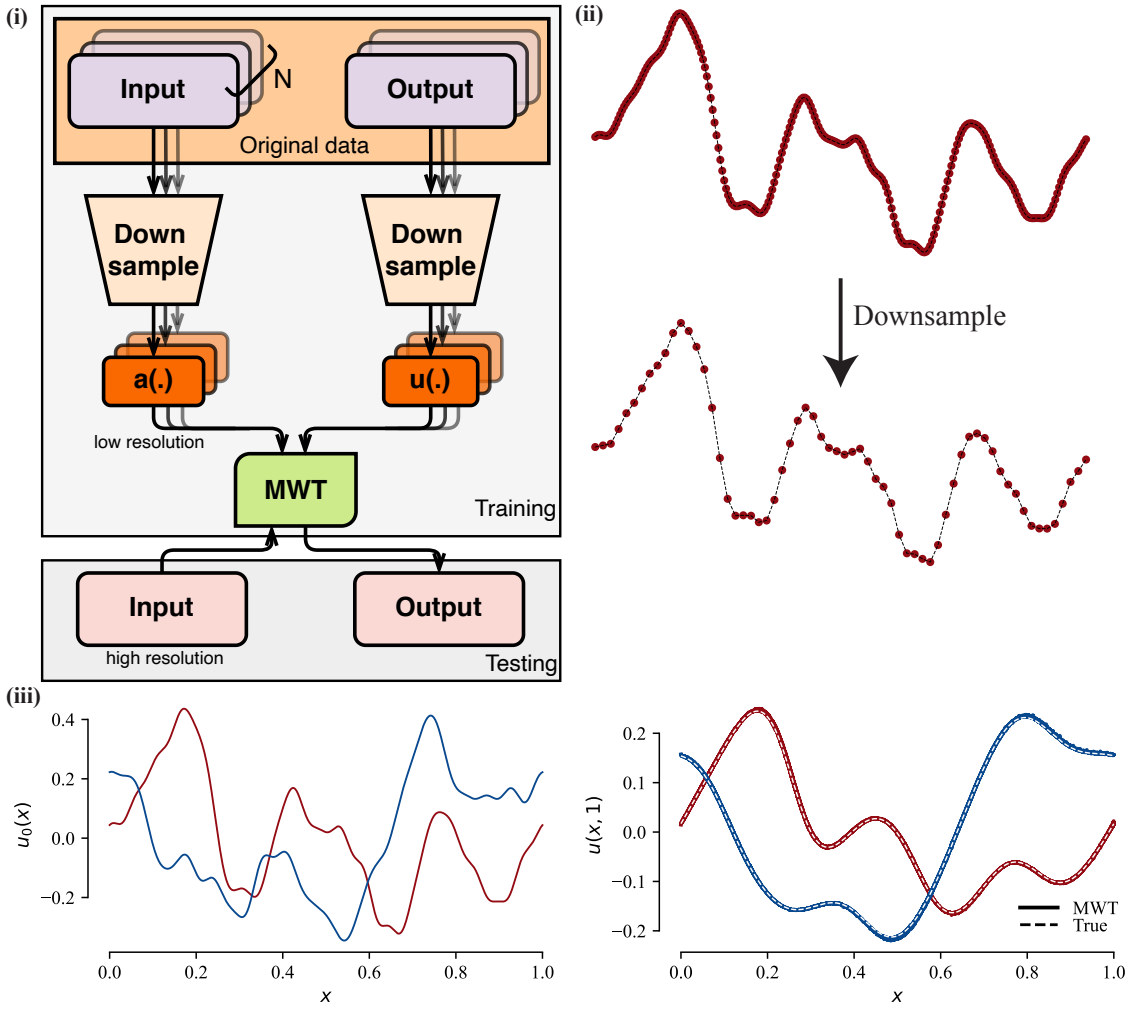


Figure 5.7: **Prediction at higher resolution:** The proposed model (MWT) learns the function mapping using the data with a coarse resolution, and can predict the output at a higher resolution. **(i)** The resolution-extension experiment pipeline. **(ii)** An example of down-sampling of the associated functions used in the training. **(iii)** We show two test samples with example-1 marked as blue while example-2 is marked as red. **Left:** input functions (u_0) of the examples. **Right:** corresponding outputs $u(x, 1)$ at $s = 8192$ from MWT Leg (trained on $s = 256$) of the 2 examples, and their higher-resolution ($s = 8192$) ground truth (dotted line).

predict the output of 2^6 times the resolution (i.e., $s = 8192$) data with an relative $L2$ error of 4.56%.

Networks	s = 64	s = 128	s = 256	s = 512	s = 1024
MWT Leg	**0.00190**	**0.00214**	**0.00204**	**0.00201**	**0.00211**
MWT Cheb	0.00239	0.00241	0.00221	0.00289	0.00348
FNO	0.00335	0.00330	0.00375	0.00402	0.00456
MGNO	0.0761	0.0593	0.0724	0.0940	0.0660
LNO	0.0759	0.0756	0.0673	0.0807	0.0792
GNO	0.0871	0.0801	0.0722	0.0798	0.0777

Table 5.5: Korteweg-de Vries (KdV) equation benchmarks for different input resolution s with input $u_0(x)$ sampled from a squared exponential kernel. Top: Our methods. Bottom: previous works of Neural operator.

5.2.7 Additional Experiments

We present additional results for the KdV equation (see Section 5.2.1). First, we demonstrate the operator learning when the input is sampled from a squared exponential kernel. Second, we experiment on the learning behavior of the Neural operators when the train and test samples are generated from different random sampling schemes.

5.2.7.1 Squared Exponential Kernel

We sample the input $u_0(x)$ from a squared exponential kernel, and solve the KdV equation in a similar setting as mentioned in Section 5.2.1. Due to the periodic boundary conditions, a periodic version of the squared exponential kernel (Rasmussen and Williams, 2006) is used as follows:

$$k(x, x') = \exp\left(-2\frac{\sin^2(\pi(x - x')/P)}{L^2}\right),$$

where, P is the domain length and L is the smoothing parameter of the kernel. The random input function is sampled from $\mathcal{N}(0, K_m)$ with K_m being the kernel matrix by taking $P = 1$ (domain length) and $L = 0.5$ to avoid the sharp peaks in the sampled function. The results for the Neural operators (similar to Table 5.1) is shown in Table 5.5. We see that MWT models perform better than the existing neural operators at all resolutions.

5.2.7.2 Training/Evaluation with Different Sampling Rules

The experiments in the current work and also in all of the recent neural operators work (Li et al., 2020a; Li et al., 2020c) have used the datasets such that the train and test samples are generated by sampling the input function using the same rule.

Networks	$\lambda = 0.25$	$\lambda = 0.5$	$\lambda = 0.75$
$s = 64$			
MWT Leg	0.00819	0.00413	**0.00202**
MWT Cheb	**0.00751**	**0.00347**	0.00210
FNO	0.0141	0.00822	0.00404
MGNO	0.3701	0.2030	0.0862
LNO	0.1012	0.0783	0.0141
$s = 256$			
MWT Leg	0.00690	**0.00322**	**0.00145**
MWT Cheb	**0.00616**	0.00344	0.00182
FNO	0.0134	0.00901	0.00376
MGNO	0.4492	0.2114	0.1221
LNO	0.1306	0.0821	0.0161
$s = 1024$			
MWT Leg	**0.00641**	0.00408	**0.00127**
MWT Cheb	0.00687	**0.00333**	0.00176
FNO	0.0141	0.00718	0.00359
MGNO	0.4774	0.2805	0.1309
LNO	0.1140	0.0752	0.0139

Table 5.6: Neural operators performance when training on random inputs sampled from Squared exponential kernel and testing on samples generated from smooth random functions (Filip, Javeed, and Trefethen, 2019) with controllable parameter λ. The random functions are used as the input $u_0(x)$ for Korteweg-de Vries (KdV) equation as mentioned in Section 5.2.1. In the test data, λ is inversely proportional to sharpness of the fluctuations.

For example, in KdV, a complete dataset is first generated by randomly sampling the inputs $u_0(x)$ from $\mathcal{N}(0, 7^4(\Delta + 7^2 I)^{-2.5})$ and then splitting the dataset into train/test. This setting is useful when dealing with the systems such that the future evaluation function samples have similar patterns like smoothness, periodicity, presence of peaks. However, from the viewpoint of learning the operator between the function spaces, this is not a general setting. We have seen in Figure 5.4 that upon varying the fluctuation strength in the inputs (both train and test), the performance of the neural operators differ. We now perform an addition experiment in which the neural operator is trained using the samples from a periodic squared exponential kernel (Section 5.2.7.1) and evaluated on the samples generated from random fields (Filip, Javeed, and Trefethen, 2019) with fluctuation parameter λ. We see in Table 5.6 that instead of different generating rules, the properties like fluctuation strength matters more when it comes to learning the operator map. Evaluation on samples that are generated from a different rule can still work well provided that the fluctuations are of similar nature. It is intuitive that by learning only from the low-frequency signals, the generalization to higher-frequency signals is difficult.

5.3 Conclusion

We address the problem of data-driven learning of the operator that maps between two function spaces. Motivated from the fundamental properties of the integral kernel, we found that multiwavelets constitute a natural basis to represent the kernel sparsely. After generalizing the multiwavelets to work with arbitrary measures, we proposed a series of models to learn the integral operator. This work opens up new research directions and possibilities toward designing efficient Neural operators utilizing properties of the kernels, and the suitable basis. We anticipate that the study of this problem will solve many engineering and biological problems such as aircraft wing design, complex fluids dynamics, metamaterials design, cyber-physical systems, neuron-neuron interactions that are modeled by complex PDEs.

Part III

Extensions and Applications

Chapter 6

Approximate Submodularity and Sensor Selection

The combinatorial nature of set optimization is usually dealt with by exploiting some structures in the optimization function. The popular one is submodularity property. The greedy selection in the presence of submodularity results in performance guarantees. However, greedy has also been proposed for non-submodular function, and it results in good empirical performance. In this Section, specifically, we seek answers to the following questions:

- Is it possible to approximate an arbitrary set function by a submodular one within a given error everywhere (i.e., for all possible sets)?
- Can we quantify the error incurred by such approximation?
- Can we trade-off the approximation error with the sub-optimal guarantees ascertain by greedy algorithms in submodular functions to determine which submodular approximation function leads to tighter guarantees on the optimality gap?

6.1 Preliminaries

A function $f : 2^N \to \mathbb{R}^+$ defined over the ground set $\Omega = \{1, 2, \ldots, N\}$ is monotone non-decreasing if for all $S \subseteq T \subseteq \Omega$, $f(S) \leq f(T)$. The marginal gain of an element $a \in \Omega$ with respect to a set $S \subseteq \Omega$ is defined as $f_S(a) = f(S \cup \{a\}) - f(S)$. A set function $f : 2^\Omega \to \mathbb{R}$ over a ground set $\Omega = \{1, 2, \ldots, N\}$ is referred as submodular if and only if for all sets $S, T \subseteq \Omega$, $f(S \cup T) + f(S \cap T) \leq f(S) + f(T)$ (Bach, 2013).

Throughout this work, we focus on the *cardinality constrained maximization problem*, i.e.,

$$\max_{|S|\leq k,\, S\subseteq\Omega} f(S). \tag{6.1}$$

The objective function f in the above discrete optimization problem will possibly be a non-submodular function. The optimal solution of this problem will be referred as OPT such that $f(\Omega^*) = OPT$ and $|\Omega^*| = k$. Before stating the definition of δ-approximate submodularity, we define the divergence between two discrete set functions.

Definition 6.1. *The divergence d between two non-decreasing set functions $f : 2^N \to \mathbb{R}^+$ and $g : 2^N \to \mathbb{R}^+$ defined over the same set Ω is denoted as*

$$d(f, g) = \max_{S\subseteq\Omega,\, a\in\Omega\setminus S} \left| \frac{f_S(a)}{g_S(a)} - 1 \right|. \tag{6.2}$$

The definition of divergence can be used to introduce the notion of δ-approximate submodular function as presented next.

Definition 6.2. *A function f, possibly non-submodular, is referred as δ-approximate submodular if there exists a non-decreasing submodular function $g : 2^N \to \mathbb{R}^+$ such that $d(f, g) \leq \delta$.*

In other words, Definition-6.2 can be re-written in its most useful form as

$$(1 - \delta)\, g_S(a) \leq f_S(a) \leq (1 + \delta)\, g_S(a),\ \forall S \subseteq \Omega, a \in \Omega \setminus S. \tag{6.3}$$

Intuitively, any finite-valued set function can be looked as $\infty-$approximate submodular function, but we will restrict to the scenario where $\delta \in [0, 1]$ in Definition 6.2. Observe that f is exactly submodular if and only if $\delta = 0$.

There are other measures to assess closeness to submodularity. For instance, for any non-negative set function f, we can use the submodularity ratio (Das and Kempe, 2011) that can also be generalized as follows.

Definition 6.3 (generalized submodularity ratio (Bian et al., 2017))**.** *The submodularity ratio of a non-negative set function f is given by*

$$\gamma_f = \min_{S,T\subseteq\Omega} \frac{\sum_{t\in T\setminus S} f_S(t)}{f_S(T)}. \tag{6.4}$$

It is worth noticing that $0 \leq \gamma_f \leq 1$, and is equal to 1 if and only if f is submodular. Another characterization is so-called curvature of submodular functions, which is a

measure of deviation from modularity (Conforti and Cornuejols, 1984), and enables us to write better performance bounds. The total curvature of a submodular function is given by

$$\alpha_T = 1 - \min_{a \in \Omega} \frac{f_{\Omega \setminus a}(a)}{f(a)}, \tag{6.5}$$

with $0 \leq \alpha_T \leq 1$ and is equal to 0 for modular/additive functions. This can be further generalized for any non-negative set function as follows.

Definition 6.4 (Generalized curvature (Bian et al., 2017))**.** *The curvature of a non-negative set function $f(.)$ can be written as*

$$\alpha = 1 - \min_{S,T \subseteq \Omega, s \in S \setminus T} \frac{f_{S \setminus \{s\} \cup T}(s)}{f_{S \setminus \{s\}}(s)}. \tag{6.6}$$

Similarly, $0 \leq \alpha \leq 1$ for the case of non-decreasing functions, and $\alpha = \alpha_T$ when f is submodular.

Now suppose that $S_G = \{s_1, s_2, \ldots, s_k\}$ denotes the ordered set solution of cardinality constrained maximization problem obtained using the greedy algorithm, and Ω^* be the optimal solution such that $|S_G| = |\Omega^*| = k$. Then, we can define the greedy curvature – in contrast to that in (Conforti and Cornuejols, 1984) – as follows.

Definition 6.5 (Greedy curvature)**.** *For a non-decreasing set function f, and Ω^* such that $|\Omega^*| = k$, the greedy curvature is defined as*

$$\alpha_G \;=\; 1 - \min_{1 \leq i \leq k} \left\{ \min_{a \in S_G \setminus (S_G^{i-1} \cup \Omega^*)} \frac{f_{S_G^{i-1} \cup \Omega^*}(a)}{f_{S_G^{i-1}}(a)}, \min_{\substack{a \in (S_G \cap \Omega^*) \setminus S_G^{i-1} \\ i \leq j \leq k}} \frac{f_{S_G^{j-1}}(s_j)}{f_{S_G^{i-1}}(a)} \right\}, \tag{6.7}$$

where $S_G^i = \{s_1, s_2, \ldots, s_i\}$ for $1 \leq i \leq k$, $S_G^0 = \phi$.

Remark that the greedy curvature defined above is always less than or equal to the greedy curvature defined in (Conforti and Cornuejols, 1984). The second term introduced in the above expression bounds the consecutive marginals of the greedy selection. This kind of technique will play a key role in proving performance bounds of the greedy algorithm and will simplify the proofs to a great extent. Furthermore, it can be easily verified that $\alpha_G \leq \alpha$. The main results and interpretations are presented in the following section.

Next, we start by addressing the necessary conditions that a set function must satisfy to be δ-approximate submodular. First, notice that if the function is submodular,

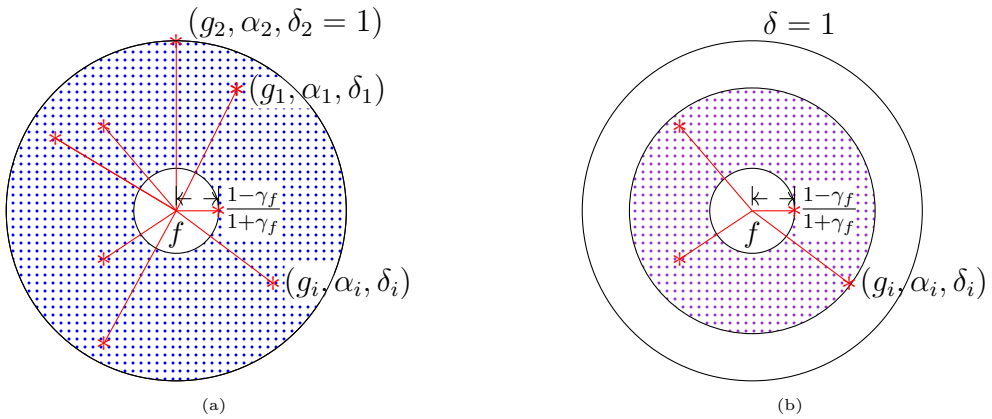


Figure 6.1: Region of submodularity ROS(f, 1) of a δ-approximate function f is depicted by the shadowed area in (a), and ROS(f, δ_2) in (b). The minimum possible value of δ is $\frac{1-\gamma_f}{1+\gamma_f}$, with γ_f denoting the submodularity ratio of f, and its maximum value is 1. A tuple $(g_i, \alpha_i, \delta_i)$ represent the δ_i-approximation of f and α_i its total curvature. For $\delta_i < 1$, the ROS(f, δ_i) in (b) may not include all the tuples as we see in (a).

then δ can be set to zero, and vice versa. On the other hand, if δ is different from zero, then we show that it cannot be arbitrary close. In other words, we obtain a necessary condition that establishes a *'submodularity gap'*. Subsequently, we aim to dig in further detail the properties of these functions, which lead us to introduce the notion of the region of submodularity (ROS). Within this region, there may exist different submodular functions, whose curvature differs and directly impact the optimality guarantees. Specifically, the lower the curvature is, the better the performance. Therefore, we aim to leverage these properties and the notion of generalized greedy curvature to show the performance of the greedy algorithm and obtain the improved constant optimality guarantees.

6.1.1 Approximate Submodularity

Lemma 6.1. *A non-submodular function f with submodularity ratio γ_f has a δ-approximation submodular function g only if $\delta \geq \frac{1-\gamma_f}{1+\gamma f}$.*

Proof. We will prove this by contradiction, let us assume that g is a submodular function and δ-approximation of f with $\delta < \frac{1-\gamma_f}{1+\gamma_f}$ (or, equivalently, $\frac{1+\delta}{1-\delta} < \frac{1}{\gamma_f}$). The submodularity ratio of any g satisfying (6.3) can be written as

$$\gamma_g = \min_{S,T\subseteq\Omega} \frac{\sum_{t\in T\setminus S} g_S(t)}{g_S(T)} \leq \frac{1+\delta}{1-\delta} \min_{S,T\subseteq\Omega} \frac{\sum_{t\in T\setminus S} f_S(t)}{f_S(T)} < \frac{1}{\gamma_f}\gamma_f = 1,$$

where the first inequality is obtained using (6.3). Hence, g cannot be submodular. □

The above result gives an interesting insight into what do we mean by closeness of a function to being submodular using the submodularity ratio. The value of γ_f limits the smallest possible value of δ and determines how close can be the function marginals to that of some submodular function. These insights can be better captured by the notion of **region of submodularity**, defined as

$$\text{ROS}(f, \delta) = \left\{ g \mid \frac{1 - \gamma_f}{1 + \gamma_f} \leq d(f, g) \leq \delta \right\}, \tag{6.8}$$

which is the collection of *delta*-approximations of a given function f. As shown in Figure 6.1, for a given f, the submodular functions in the shaded region can be used to describe it as δ-approximation, but the closest value of δ is restricted by γ_f. It should be noted that multiple g_i can be used as δ-approximation of a given f; hence, from the viewpoint of performance bound, we are interested in the g with minimum total curvature for the given value of δ. Let us denote α_δ as the curvature of the selected δ-approximation g, i.e.,

$$\alpha_\delta = \min_{g \in \text{ROS}(f,\delta)} \alpha_T(g). \tag{6.9}$$

With this in mind, in the next section, we state one of the main result regarding performance of naive greedy selection for δ-approximate submodular functions.

6.1.2 Constant Performance Bound

It is worth to mention the result for the performance bound of a submodular function with total curvature α.

Theorem 6.1 (from (Conforti and Cornuejols, 1984)). *For a non-negative non-decreasing submodular function g with total curvature α, if OPT denotes the optimal value of $\max g(\mathcal{S})$ subject to $|\mathcal{S}| \leq k$, then the output of greedy algorithm, $\mathcal{S}_G$ satisfies $g(\mathcal{S}_G) \geq \frac{1}{\alpha}(1 - (1 - \frac{\alpha}{k})^k)OPT$.*

Also, for any set function (may or may not be δ-approximate), the performance of greedy can be lower bounded as explained in the next result.

Theorem 6.2 (from (Bian et al., 2017)). *Let f be a non-negative non-decreasing set function with submodularity ratio $\gamma \in [0, 1]$ and curvature $\alpha \in [0, 1]$. The output of greedy algorithm satisfies $f(\mathcal{S}_G) \geq \frac{1}{\alpha}(1 - (1 - \frac{\alpha\gamma}{k})^k)OPT$.*

For δ-approximate submodular functions it can be shown that the greedy selection algorithm, described in Algorithm 5, offers constant performance bound as captured in the next result.

Algorithm 5: Greedy Selection Algorithm

Output: S_G

Initialize $S_G = \phi$;

repeat

$s^* = \arg\max_{s \in \Omega \setminus S_G} f_{S_G}(s)$;

$S_G \leftarrow S_G \cup s^*$;

until $|S_G| = k$;

Approximation	Feasibility	Performance bound
$(1-\delta)g_S(a) \leq f_S(a) \leq (1+\delta)g_S(a)$	$\delta \geq \frac{1-\gamma_f}{1+\gamma_f}$	$\frac{1}{\alpha_\delta \frac{1-\delta}{1+\delta} + \frac{2\delta}{1+\delta}} \left(1 - \exp\{-(\alpha_\delta \frac{1-\delta}{1+\delta} + \frac{2\delta}{1+\delta}) \frac{1-\delta}{1+\delta}\}\right)$
$\delta_l g_S(a) \leq f_S(a) \leq \delta_u g_S(a)$	$\delta_l \leq \delta_u \gamma_f$	$\frac{1}{1-\frac{\delta_l}{\delta_u}(1-\alpha_\delta)} \left(1 - \exp\{-(1 - \frac{\delta_l}{\delta_u}(1-\alpha_\delta)) \frac{\delta_l}{\delta_u}\}\right)$
$g_S(a) \leq f_S(a) \leq \delta_u g_S(a)$	$\delta_u \geq \frac{1}{\gamma_f}$	$\frac{1}{1-\frac{1-\alpha_\delta}{\delta_u}} \left(1 - \exp\{-(1 - \frac{1-\alpha_\delta}{\delta_u}) \frac{1}{\delta_u}\}\right)$

Table 6.1: Feasibility and performance bound for different approximation structures.

Theorem 6.3. *For a given δ-approximate submodular function with submodularity ratio γ_f and α_δ such that $\delta \geq \frac{1-\gamma_f}{1+\gamma_f}$, the greedy algorithm has guaranteed constant performance.*

$$f(S_G) \geq \frac{1}{\alpha_\delta \frac{1-\delta}{1+\delta} + \frac{2\delta}{1+\delta}} \left(1 - \left(1 - \frac{1}{k}\left(\alpha_\delta \frac{1-\delta}{1+\delta} + \frac{2\delta}{1+\delta}\right)\left(\frac{1-\delta}{1+\delta}\right)\right)^k\right) OPT. \quad (6.10)$$

The detailed proof is provided in the Appendix. It is worth noticing that for $\delta = 0$, the above result is the same as Theorem 6.1 because f would be submodular. For $\delta \geq \frac{1-\gamma_f}{1+\gamma_f}$, the performance guarantee decreases due to the divergence from submodularity property. While α_δ corresponds to the total curvature of a δ-approximation submodular function of the given function f, the term $\frac{2\delta}{1+\delta}$ can be looked upon as penalty paid for deviating from this submodular function.

6.1.3 Extensions on Approximation and Performance Bounds

The definition of approximate submodularity in (6.3) is sufficient for identifying such functions but it often happens that the upper and lower bounds may not be symmetric as we will see in the next section. However, instead of making the bounds loose to bring regularity, such asymmetric behavior can be leveraged to get tighter bounds for the performance of greedy algorithm from Theorem 6.3. Specifically,

(i) If the δ-approximation in equation (6.3) is available in the following form

$$\delta_l \, g_S(a) \leq f_S(a) \leq \delta_u \, g_S(a), \ \forall S \subseteq \Omega, a \in \Omega \setminus S. \tag{6.11}$$

Then we have the following results on feasibility and performance bounds.

Corollary 6.1. *A non-submodular function f with submodularity ratio γ_f has a δ-approximation submodular function g, according to (6.11), only if $\delta_l \leq \delta_u \gamma_f$.*

Corollary 6.2. *For a given δ-approximate submodular function, according to (6.11), with submodularity ratio γ_f and α_δ such that $\delta_l \leq \delta_u \gamma_f$, the greedy algorithm has guaranteed constant performance.*

$$f(S_G) \geq \frac{1}{1 - \frac{\delta_l}{\delta_u}(1-\alpha_\delta)} \left(1 - \left(1 - \frac{1}{k}\left(1 - \frac{\delta_l}{\delta_u}(1-\alpha_\delta)\right)\left(\frac{\delta_l}{\delta_u}\right)\right)^k\right) OPT. \tag{6.12}$$

(ii) If the δ-approximation in equation (6.3) is available in the following form

$$g_S(a) \leq f_S(a) \leq \delta_u \, g_S(a), \ \forall S \subseteq \Omega, a \in \Omega \setminus S. \tag{6.13}$$

Then, we have the following results on feasibility and performance bounds.

Corollary 6.3. *A non-submodular function f with submodularity ratio γ_f has a δ-approximation submodular function g, according to (6.13), only if $\delta_u \geq \frac{1}{\gamma_f}$.*

Corollary 6.4. *For a given δ-approximate submodular function, according to (6.13), with submodularity ratio γ_f and α_δ such that $\delta_u \geq \frac{1}{\gamma_f}$, the greedy algorithm has guaranteed constant performance.*

$$f(S_G) \geq \frac{1}{1 - \frac{1-\alpha_\delta}{\delta_u}} \left(1 - \left(1 - \frac{1}{k}\left(1 - \frac{1-\alpha_\delta}{\delta_u}\right)\left(\frac{1}{\delta_u}\right)\right)^k\right) OPT. \tag{6.14}$$

The variations of such asymmetric behavior is summarized in Table 6.1. Next, we identify some important examples that can be recognized as approximate submodular functions.

6.2 Applications

In this section, we identify some objective functions which are not submodular and often appear in the applications (e.g., sensor selection in cyber-physical systems and

sparse learning). Next, we provide the foundation why greedy algorithm has empirically performed well in such non-submodular functions. Towards this goal, we exploit the δ-approximations of these functions, that help us to derive improved performance bounds, as well as develop some closeness guarantee to being submodular.

6.2.1 Trace of Inverse Gramian

Let us consider a Gramian matrix $W_S \in \mathbb{R}^{n\times n}$, obtained as $W_S = \Lambda_0 + \sum_{s\in S} x_s x_s^T$, where Λ_0 is a symmetric matrix and $x_i \in \mathbb{R}^n$ is taken from the data matrix $X \in \mathbb{R}^{n\times N}, X = [x_1, x_2, \ldots, x_N]$. As such, let the ordered eigenvalues of W_S be denoted as $\lambda_n \geq \ldots \geq \lambda_1$.

There are several practical settings where the Gramian is used . For instance, the negative trace of the covariance matrix inverse is used as criteria for Bayesian-A optimality (Krause, Singh, and Guestrin, 2008). Another example consists in determining the minimum sensor selection (constrained to a given sensor budget) that seeks to reduce the variance of the Bayesian estimate of parameters (Bian et al., 2017). Briefly, for the sake of completeness, this problem can be described as follows. Consider some set of observations $\mathbf{y} \in \mathbb{R}^N$ and the linear model $y = X^T\theta + w$ which relates the parameter θ with observations in the presence of Gaussian noise $w \sim \mathcal{N}(0, I_N)$. The problem of sensor selection can be stated as minimizing the conditional variance of θ with a given sensor budget. If $S \subseteq \{1, 2, \ldots, N\}$ denotes the set of selected sensors, then we have $y_S = X_S^T\theta + w$, where y_S are the set of observations indexed according to the elements in S and similarly X_S has columns taken from X. With the Gaussian prior assumption for $\theta \sim \mathcal{N}(0, \Lambda_0^{-1})$ with $\Lambda_0 = \beta^2 I_N$, the conditional covariance of θ given y_S can be written as $\Sigma_{\theta|y_S} = (\Lambda_0 + X_S X_S^T)^{-1}$. Consequently, the objective function is defined as

$$f(S) = \text{tr}(\Lambda_0) - \text{tr}(\Lambda_0 + \sum\nolimits_{s\in S} x_s x_s^T)^{-1}. \tag{6.15}$$

Notice that the negative trace of the inverse of the Gramian matrix, W_S, is a non-submodular function (Krause, Singh, and Guestrin, 2008). Notwithstanding, it can be labeled as δ-approximate to a known submodular function, log determinant of W_S, which is formalized in the next result.

Proposition 7. *The negative trace of matrix inverse, $f(S) = -tr(W_S^{-1})$ is a δ-approximate, where the δ-approximation is given by the submodular function, $g(S) = \log\det(W_S)$ with the upper and lower bounds given by $\delta_u = \frac{1}{\lambda_1(W_\phi)}$ and $\delta_l = \frac{1}{\lambda_n(W_\Omega)}$, respectively.*

To assess the tightness of the associated performance bounds (summarized in the Table 6.1), we conduct different numerical experiments in the Experimental Section.

6.2.2 Minimum Eigenvalue

It has been stated in (Pasqualetti, Zampieri, and Bullo, 2014) that minimum eigenvalue of the so-called observability Gramian can be used to retrieve the networked dynamical system's state. In particular, the lower the value, the harder it is to retrieve the state; hence, we seek to maximize the value of minimum eigenvalue of the observability Gramian. The maximization of minimum eigenvalue also serves as criteria for matrix inversion in the presence of numerical errors for sparse matrices.

Next, the minimum eigenvalue of the Gramian given by

$$f(S) = \lambda_1(W_S) \tag{6.16}$$

is a non-submodular function (Summers, Cortesi, and Lygeros, 2016), but in the context of our framework, it can be shown as a δ-approximate submodular function, as described next.

Proposition 8. *The minimum eigenvalue of Gramian, $f(S) = \lambda_1(W_S)$ is a δ-approximate, where the δ-approximation is given by the submodular function, $g_1(S) = \lambda_n(W_S)$, with the upper and lower bounds as $\delta_u = \max_{\omega\in\Omega} \frac{\lambda_n(W_\omega)}{\lambda_1(W_\omega)}$ and $\delta_l = \min_{\omega\in\Omega} \frac{\lambda_1(W_\omega)}{\lambda_n(W_\omega)}$, respectively.*

Next, we show that there exists another function in the ROS of f through the following result.

Proposition 9. *The minimum eigenvalue of Gramian, $f(S) = \lambda_1(W_S)$ is a δ-approximate, where the δ-approximation is given by the submodular function, $g_2(S) = tr(W_S)$, with the upper and lower bounds as δ_u and δ_l, respectively, where*

$$\delta_u = 1 - \frac{n-1}{n} \min_{\omega\in\Omega} \frac{\lambda_1(W_\omega)}{\lambda_n(W_\omega)}, \text{ and } \quad \delta_l = \frac{1}{n} \min_{\omega\in\Omega} \frac{\lambda_1(W_\omega)}{\lambda_n(W_\omega)}.$$

The above result represents the non-submodular function f as an approximation to a modular function. Since g_2 is modular, therefore its curvature is equal to 0. This property can be leveraged in Table 6.1 to get tight bounds when n is not large.

6.2.3 Tightness Analysis of the Performance Bounds

The performance bound presented in (Bian et al., 2017) requires a combinatorial exhaustive search for the computation of parameters, making it extremely difficult to

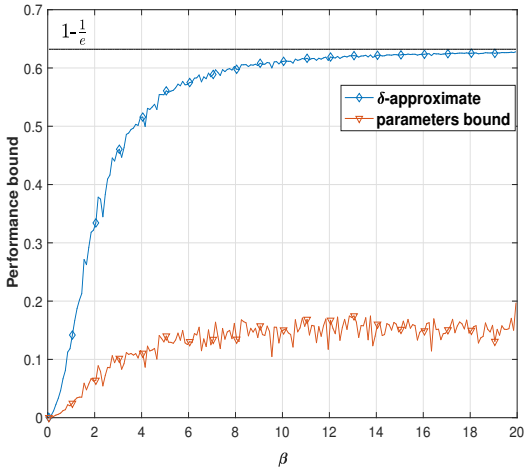


Figure 6.2: Comparison of performance bound for non-submodular function $f(S) = -\text{tr}(W_S^{-1})$ using δ-approximate method and using bound of parameters (α, γ) from (Bian et al., 2017).

obtain in practical scenarios. To remedy this, the authors have presented approximate bounds given in terms of the parameters, curvature and submodularity ratio. The performance bound for greedy algorithm presented in this work (in Theorem 6.3) requires only linear search to compute the curvature of the submodular function, α_δ from (6.5).

We make a comparison between our presented bounds and the ones in (Bian et al., 2017) through simulation of negative trace of W_S inverse described in (6.15). To simulate the matrix W_S, the entries of the data matrix $\mathbf{X} \in \mathbb{R}^{10\times 30}$ are generated from Gaussian distribution $\mathcal{N}(0, 1)$ and each column is normalized (i.e., $||\mathbf{x}_i||_2 = 1$). The matrix Λ_0 in W_S is taken as $\beta^2 I$. Figure 6.2 shows the performance bound attained for different values of β. We observe that there is a considerable gap between the performance bounds using δ_u and δ_l from Proposition 7 and Corollary 6.2, and the ones computed as proposed in (Bian et al., 2017).

Remarkably, an interesting observation can be made when β grows large, which can be explained using the current theory of δ-approximation. Specifically, the ratio of bounds $\delta_u/\delta_l \to 1$, the identified submodular function (i.e., log determinant of Gramian) becomes constant, and consequently, $\alpha_\delta \to 1$. Substituting these values in Table 6.1, it follows that the performance bound goes to $1 - 1/e$ which matches that of a submodular function.

Chapter 7

Non-Markovian Reinforcement Learning

Reinforcement learning (RL) (Bertsekas, 2019) is a technique to synthesize control policies for autonomous agents that interact with a stochastic environment. The RL paradigm now contains a number of different kinds of algorithms, and has been successfully used across a diverse set of applications including autonomous vehicles, resource management in computer clusters (Mao et al., 2016), traffic light control (Arel et al., 2010), web system configuration (Bu, Rao, and Xu, 2009), and personalized recommendations (Zheng et al., 2018). In RL, we assume that in each state, the agent performs some action and the environment picks a probability distribution over the next state and assigns a reward (or negative cost). The reward is typically defined by the user with the help of a state-based (or state-action-based) reward function. The expected payoff that the agent may receive in any state can be defined in a number of different ways; in this chapter, we assume that the payoff is an discounted sum of the local rewards (with some discount factor $\gamma \in [0, 1]$) over some time horizon H. The purpose of RL is to find the stochastic policy (i.e. a distribution over actions conditioned on the current state), that optimizes the expected payoff for the agent. Most RL algorithms assume that the environment satisfies Markov assumptions, i.e. the probability distribution over the next state is dependent only on the current state (and not the history). In contrast, here, we investigate an RL procedure for a non-Markovian environment.

Broadly speaking, there are two classes of RL algorithms (Chua et al., 2018): model-based and model-free algorithms. Most classical RL algorithms are *model-based*; they assume that the environment is explicitly specified as a Markov Decision Process (MDP), and use dynamic programming to compute the expected payoff for each state of the MDP (called its value), as well as the optimal policy (Sutton, Precup, and Singh,

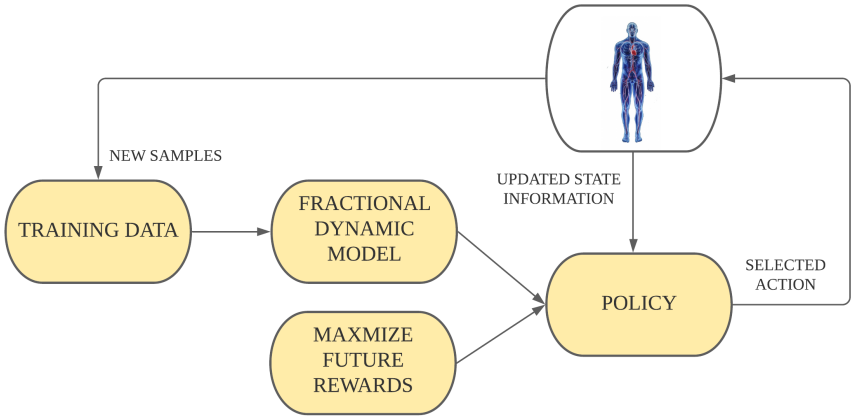


Figure 7.1: Non-Markovian Model Based Reinforcement Learning setup. The model based predictions are used to select actions, and then iteratively update the model dynamics.

1999; Strehl, Li, and Littman, 2009). Classical RL algorithms have strong convergence guarantees stemming from the fact that the value of a state can be recursively expressed in terms of the value of the next state (called the Bellman equation), which allows us to define an operator to update the value (or the policy) for a given state across iterations. This operator (also known as the Bellman operator) can be shown to be a contraction mapping (Bertsekas, 2019). However, obtaining exact symbolic descriptions of models is often infeasible. This led to the development of model-free reinforcement learning (MFRL) approaches that rely on sampling many model behaviors through simulations and eschew building a model altogether.

MFRL algorithms can converge to an optimal policy under the right set of assumptions; however, can suffer from high sample complexity (i.e. the number of simulations required to learn an optimal policy). This has led to investigation of a new class of model-based RL (MBRL) algorithms where the purpose is to simultaneously learn the system model as well as the optimal policy (Mao et al., 2016). Such algorithms are called *on policy*, as the policy learned during any iteration is used for improving the learned model as well as optimizing the policy further. Most MBRL approaches use function approximators or Bayesian models to efficiently learn from scarce sample sets of system trajectories. MBRL approaches tend to have lower sample complexity than MFRL as the learned model can accelerate the convergence by focusing on actions that are likely to be close to the optimal action. However, MBRL approaches can suffer severely from modeling errors (Todorov, Erez, and Tassa, 2012), and may converge to less optimal solutions.

In both MFRL and MBRL algorithms, a fundamental assumption is that the environment satisfies Markovian properties, partly to avoid the complexity of dealing with the historical dependence in transitions. To overcome this challenge, we propose a non-Markovian MBRL framework that captures non-Markovian characteristics

through a fractional dynamical systems formulation. Fractional dynamical systems can model non-Markovian processes characterized by a single fractal exponent and commonly arise in mathematical models of human physiological processes (West, 2010; Xue et al., 2016), biological systems, condensed matter and material sciences, and population dynamics (Gupta, Pequito, and Bogdan, 2018b; Yin, Gupta, and Bogdan, 2020; Gupta, Pequito, and Bogdan, 2018c; Gupta, Pequito, and Bogdan, 2018d). Such systems can effectively model spatio-temporal properties of physiological signals such as blood oxygenation level dependent (BOLD), electromyogram (EMG), and electrocardiogram (ECG). (Gupta, Pequito, and Bogdan, 2018b; Baleanu, Machado, and Luo, 2011; Magin, 2006).

The advantage of using fractional dynamical models is that they can accurately represent long-range (historical) correlations (memory) through a minimum number of parameters (e.g., using a single fractal exponent to encode a long-range historical dependence rather than memorizing the trajectory itself or modeling it through a large set of autoregressive parameters). Though fractional models can be used to perform predictive control (Ghorbani and Bogdan, 2014), problems such as learning these models effectively or obtaining optimal policies for such models in an RL setting have not been explored.

In this chapter, we present a novel non-Markovian MBRL technique in which our algorithm alternates between incrementally learning the fractional exponent from data and learning the optimal policy on the updated model. We show that the optimal action in a given state can be efficiently computed by solving a quadratic program over a bounded horizon rollout from the state. The overview of our model-based reinforcement learning algorithm is shown in Fig. 7.1. In this algorithm, we use on-policy simulations to gather additional RL data that is then used to update the model. Our model learning algorithm is based on minimizing the distance between the data's state-action distribution and the next state distribution induced by the controller. The fractional dynamic model is then retrained using the cumulative dataset. The MBRL procedure is run for a finite number of user-specified iterations.

7.1 Problem Formulation

The reinforcement learning deals with the design of the controller (or policy) which minimizes the expected total cost. In the setting of a memoryless assumption, the Markov Decision Process (MDP) (Bellman, 1957) is used to model the system dynamics such that the future state depends only on the current state and action. For a state $\mathbf{s}_t \in \mathbb{R}^n$ and action $\mathbf{a}_t \in \mathbb{R}^p$, the future state evolve as $\mathbf{s}_{t+1} \sim P(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$, and a cost function $r_t = c(\mathbf{s}_t, \mathbf{a}_t)$. However, the Markov assumption does not work well with the long-range memory processes (Micciche, 2009). In this work, we take

the non-Markovian setting, or history dependent process (HDP), and hence, the future state depends not only on the current action but also the history of states. The history at time t is the set $\mathcal{H}_t = \{(s_k)_{k\leq t}\}$, and for a trajectory $h \in \mathcal{H}_t$, we have $P(\mathbf{s}_{t+1}|h, \mathbf{a}_t)$, or alternatively, $P_h(\mathbf{s}_{t+1}|s_t, \mathbf{a}_t)$, where the terminal state of the trajectory h is written as $h(t) = s_t$. We consider a model-based approach for reinforcement learning in a finite-horizon setting. A non-Markovian policy $\pi(.|h)$ provides a distribution over actions given the history of states until time t as $h \in \mathcal{H}_t$. For a given policy, the value function is defined as $V_h^\pi = \mathbb{E}_{\pi(.|h)} \sum_{t=0}^{T-1} c(s_t, a_t)$, where the expectation is taken over state trajectories using policy π and the HDP, and T is the horizon under consideration. We formally define the non-Markovian MBRL problem in the Section 7.1.2.

7.1.1 Fractional Dynamical Model

We revisit the fractional dynamical model as discussed in Section 2.1. A linear discrete time fractional-order dynamical model is described as follows:

$$\Delta^\alpha \mathbf{s}[k+1] = \mathbf{A}\mathbf{s}[k] + \mathbf{B}\mathbf{a}[k], \tag{7.1}$$

where $\mathbf{s} \in \mathbb{R}^n$ is the state, $\mathbf{a} \in \mathbb{R}^p$ is the input action. The difference between a classic linear time-invariant (or Markovian) and the above model is the inclusion of fractional-order derivative whose expansion and discretization for any ith state $(1 \leq i \leq n)$ can be written as

$$\Delta^{\alpha_i} s_i[k] = \sum_{j=0}^{k} \psi(\alpha_i, j) s_i[k-j], \tag{7.2}$$

where α_i is the fractional order corresponding to the ith state dimension and $\psi(\alpha_i, j) = \frac{\Gamma(j-\alpha_i)}{\Gamma(-\alpha_i)\Gamma(j+1)}$ with $\Gamma(.)$ denoting the gamma function. The system dynamics can also be written in the probabilistic manner as follows:

$$\begin{aligned} &P_{\boldsymbol{\theta}}(\mathbf{s}[k+1]|\mathbf{s}[0], \ldots, \mathbf{s}[k], \mathbf{a}[k]) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}), \\ &\boldsymbol{\mu}_{\boldsymbol{\theta}} = \begin{bmatrix} \sum_{j=1}^{k} \psi(\alpha_i, j) s_0[k-j] + \mathbf{a}_0^T \mathbf{s}[k] + \mathbf{b}_0^T \mathbf{a}[k] + \mu_0 \\ \sum_{j=1}^{k} \psi(\alpha_i, j) s_1[k-j] + \mathbf{a}_1^T \mathbf{s}[k] + \mathbf{b}_1^T \mathbf{a}[k] + \mu_1 \\ \vdots \\ \sum_{j=1}^{k} \psi(\alpha_i, j) s_{n-1}[k-j] + \mathbf{a}_{n-1}^T \mathbf{s}[k] + \mathbf{b}_{n-1}^T \mathbf{a}[k] + \mu_{n-1} \end{bmatrix}, \end{aligned} \tag{7.3}$$

where $\boldsymbol{\theta} = \{\alpha, \mathbf{A}, \mathbf{B}, \mu, \boldsymbol{\Sigma}\}$, and $\mathbf{A} = [\mathbf{a}_0, \ldots, \mathbf{a}_{n-1}]$, $\mathbf{B} = [\mathbf{b}_0, \ldots, \mathbf{b}_{n-1}]$. The fractional differencing operator in (7.3) introduce the non-Markovianity by having long-range filtering operation on the state vectors.

7.1.2 Non-Markovian Model Based Reinforcement Learning

The actions in MBRL are preferred on the basis of predictions made by the undertaken model of the system dynamics. For many real-world systems, example blood glucose (Ghorbani and Bogdan, 2014; Otoom et al., 2013), ECG activities (Xue et al., 2016), the assumption of Markovian dynamics does not hold and the predictions are not accurate, leading to less rewarding actions selected for the system. As we note in the previous section 7.1.1 that non-Markovian dynamics can be effectively and compactly modeled as fractional dynamical system, we aim to use this system model for making predictions. The non-Markovian MBRL problem is formally defined as follows.

Problem Statement: Given non-Markovian state transitions, and actions dataset in the time horizon $k \in [0, T-1]$ as $\mathcal{D} = \Big\{(\mathbf{s}[0], \ldots, \mathbf{s}[k], \mathbf{a}[k]), \mathbf{s}[k+1]\Big\}$. Let $P_{\boldsymbol{\theta}}(\mathbf{s}[k+1]|\mathbf{s}[0], \ldots, \mathbf{s}[k], \mathbf{a}[k])$ be the non-Markovian system dynamics parameterized by the model parameters $\boldsymbol{\theta}$. Estimate the optimal policy which minimizes the expected future discounted cost

$$\pi^* = \arg\min_{\pi} \ \mathbb{E} \sum_{k=0}^{T-1} \gamma^k c(\mathbf{s}[k], \mathbf{a}[k]), \tag{7.4}$$

where γ is the discount factor satisfying $\gamma \in [0, 1]$, and T is the horizon under consideration.

7.2 Non-Markovian Reinforcement Learning

The MBRL comprises of two key steps, namely (i) the estimation of the model dynamics from the given data $\mathcal{D}$, and (ii) the design of a policy for optimal action selection which minimizes the total expected cost using estimated dynamics. We discuss the solution to the non-Markovian MBRL as follows.

7.2.1 Non-Markovian Model Predictive Control

The Model Predictive Control (MPC) aims at estimating the closed-loop policy by optimizing the future discounted cost under a limited-horizon H using some approximation of the environment dynamics and the cost. In this work, we are concerned with HDP using non-Markovian state dynamics. In MPC, the policy could be a deterministic action, or a distribution over actions, and we sample the action at each time-step in the latter. The MPC problem to estimate the policy at time-step k for

a given $h \in \mathcal{H}_k$ can be formally defined as

$$\begin{aligned} \min_{\pi(.|h)} \quad & \sum_{l=k}^{k+H-1} \gamma^{l-k} \hat{c}(s[l], a[l]) \\ \text{subject to} \quad & \\ & s[l+1] = f(h, a[l], e[l]), \forall l \geq k \end{aligned} \tag{7.5}$$

The approximation of the environment dynamics f could be non-linear in general, and $e[l]$ is the system perturbation noise following some distribution $e \sim g_e$. The presence of e provides randomness in the action sampling through policy, and the sampled action at each step is $a[k]$. The performance of the non-Markovian MPC based policy is bounded within the optimal policy using the following result.

Theorem 7.1. *Given an approximate HDP with* $||\hat{P}_{h'}(s'|s,a) - P_h(s'|s,a)||_1 \leq \mathcal{O}(t^q)$, $\forall h, h' \in \mathcal{H}_t$ *with* $h(t) = h'(t) = s$, *and* $||c(s,a) - \hat{c}(s,a)||_\infty \leq \varepsilon$. *The performance of the non-Markovian MPC based policy* $\hat{\pi}$ *is related to the optimal policy* π^* *as*

$$||V_{h_0}^{\hat{\pi}} - V_{h_0}^{\pi^*}||_\infty \leq 2\frac{1-\gamma^H}{1-\gamma}\left(\frac{c_{max} - c_{min}}{2}\right) H\mathcal{O}(T^q) + 2\varepsilon\frac{1-\gamma^H}{1-\gamma}\frac{1-\gamma^T}{1-\gamma}, \tag{7.6}$$

where, $h_0 \in \mathcal{H}_0$ *is the initial history given to the system.*

The assumption of model approximation is critical here, and the error increases if the exponent q increases. For the MDP setting, the approximation is taken as $q = 0$. However, for a HDP with the history of length t, we scale the approximation gap with t. The MPC horizon also plays a role in the error bound, and the error increases for larger H.

The non-Markovian MPC could be computationally prohibitive (expensive) in the general setting. Consequently, we now discuss the fractional dynamical MPC approach which is non-Markovian but computationally tractable.

7.2.2 Fractional Model Predictive Control

The linear discrete fractional dynamical model as discussed in (7.1) is used as an approximation to the non-Markovian environment dynamics. Formally, for our purpose,

the fractional MPC problem using (7.5) is defined as

$$\begin{aligned}
&\min_{\mathbf{a}[k]} \sum_{l=k}^{k+H-1} \gamma^{l-k} \hat{c}(\mathbf{s}[l], \mathbf{a}[l]) \\
&\text{s.t.} \\
&\qquad \Delta^{\alpha} \bar{\mathbf{s}}[l+1] = \mathbf{A}\bar{\mathbf{s}}[l] + \mathbf{B}\mathbf{a}[l] + e[l], \\
&\qquad \bar{\mathbf{s}}[k'] = \mathbf{s}[k], \forall k' \leq k, \\
&\qquad \mathbf{s}_{min} \leq \bar{\mathbf{s}}[l] \leq \mathbf{s}_{max}, \forall l,
\end{aligned} \tag{7.7}$$

where $\mathbf{s}_{min}, \mathbf{s}_{max}$ are feasibility bounds on the problem according to the application, and the model noise $e \sim \mathcal{N}(0, \Sigma)$. Note that (7.7) provides a policy using fractional MPC. The action $a[k]$ is sampled from this policy by first sampling $e \sim \mathcal{N}(0, \Sigma)$, and then solving (7.7). The non-Markovian fractional dynamics would introduce the computation complexities in optimally solving the problem in (7.7). However, since the constraints in (7.7) are linear, for cost approximations $\hat{c}$ that are quadratic, a quadratic programming (QP) solution can be developed to solve the fractional MPC efficiently. We refer the reader to Appendix F.2 for the QP version of the fractional MPC. Further, a convex formulation of the costs $\hat{c}$ also enables efficient solution of the fractional MPC using convex programming solvers, for example, CPLEX and Gurobi (Kronqvist et al., 2019; Hutter, Hoos, and Leyton-Brown, 2010).

Next, we discuss the methodologies required to make an approximation of the non-Markovian environment using fractional dynamics.

7.2.3 Model Estimation

The fractional dynamical model as described in the Section 7.1.1 is estimated using the approach proposed in (Gupta, Pequito, and Bogdan, 2018b) by replacing the unknown inputs with known actions at any time-step. For the sake of completeness, we present estimation algorithm as Algorithm 6. We note that in (Gupta, Pequito, and Bogdan, 2018b) the input data is obtained only once, and hence in this work appropriate modification in Algorithm 6 is performed to work with recursively updated dataset as we see in Section 7.2.4.

The Markovian model assume memoryless property and hence lacks long-range correlations for further accurate modeling. The existence of long-range correlations can be estimated by computing the Hurst exponent $\bar{H}$. For long-range correlations, the $\bar{H}$ lies in the range of $(0.5, 1]$. The fractional coefficient α in our model is related with $\bar{H}$ as $\alpha = \bar{H} - 0.5$. The Hurst exponent can be estimated from the slope of log-log variations of the variance of wavelets coefficients vs scale as noted in (Flandrin,

Algorithm 6: Fractional_Dynamics_Estimation

Input: $\mathcal{D} = \Big\{(\mathbf{s}[0], \ldots, \mathbf{s}[k], \mathbf{a}[k]), \mathbf{s}[k+1]\Big\}$ in the time-horizon $k \in [0, T-1]$
Output: $\boldsymbol{\theta} = \{\alpha, \mathbf{A}, \mathbf{B}, \mu, \boldsymbol{\Sigma}\}$

1: Estimate α using wavelets fitting for each state dimension
2: **for** $i = 1, 2, \ldots, n$ **do**
3: Compute $z_i[k] = \Delta^{\alpha_i} s[k+1]$ using α_i ▷ Eq.(7.2)
4: Aggregate $z_i[k], s[k], a[k]$ as Z_i, S, U
5: $[a_i^T, b_i^T, \mu] = \arg\min_{a,b,\mu} ||Z_i - Sa - Ub - \mu||_2^2$ with $\boldsymbol{\Sigma}$ as squared error
6: **end for**

1992). In the experiments Section 7.3.2, we show log-log plot to observe the presence of long-range correlations in the real-world data.

7.2.4 Model-based Reinforcement Learning

The non-Markovian MPC exploiting the fractional dynamical model formulation in Section 7.2.2 utilizes a dataset of the form $\mathcal{D} = \Big\{(\mathbf{s}[0], \ldots, \mathbf{s}[k], \mathbf{a}[k]), \mathbf{s}[k+1]\Big\}$ in the time-horizon $k \in [0, T-1]$. We note that the performance of such MPC can be further improved by using reinforcement learning. The selected actions by the MPC $\mathbf{a}[k]$ can be used to gather new transitions $\mathbf{s}[k+1]|\mathbf{s}[0], \ldots, \mathbf{s}[k], \mathbf{a}[k]$, or acquiring data using on-policy. The aggregated data is now used to re-estimate the model dynamics, and then perform MPC. Specifically, the MBRL proceeds as follows. Using the seed dataset, a parameterized fractional model dynamics is estimated as $P_{\boldsymbol{\theta}}(\mathbf{s}[k+1]|\mathbf{s}[0], \ldots, \mathbf{s}[k], \mathbf{a}[k])$. The model dynamics is used to minimize the discounted future cost as MPC in equation (7.7). The selected action along with the history of states $\mathbf{s}[0], \ldots, \mathbf{s}[k]$ is used to gather the next transition using on-policy as $\mathbf{s}[k+1]|\mathbf{s}[0], \ldots, \mathbf{s}[k], \mathbf{a}[k]$. The seed dataset is updated with the gathered on-policy data $\mathcal{D}_{RL}$ to get aggregated dataset. The fractional dynamics are updated using the new dataset, and the aforementioned steps are repeated for a given number of iterations. The above steps are summarized as Algorithm 7. The Algorithm 7 utilizes Algorithm 6 iteratively for the fractional model estimation. We now proceed to Section 7.3 for numerical demonstration of the proposed schemes.

7.3 Experiments

We show example of the fractional MBRL on the blood glucose (BG) control. The motive of blood glucose control is to make the BG in the range of $70-180 mg/dL$. The

Algorithm 7: Fractional_Reinforcement_Learning

Input: Seed dataset $\mathcal{D}_s = \Big\{(\mathbf{s}[0], \ldots, \mathbf{s}[k], \mathbf{a}[k]), \mathbf{s}[k+1]\Big\}$ in the time-horizon $k \in [0, T-1]$

Output: $\boldsymbol{\theta}$

Initialize: $\mathcal{D}_{RL} \leftarrow \phi$

for $iter = 1, 2, \ldots, iter\_max$ **do**
  $\theta \leftarrow$ Fractional_Dynamics_Estimation$(\mathcal{D}_s \cup \mathcal{D}_{RL})$
  Set initial state $\bar{\mathbf{s}}[0] \leftarrow \mathbf{s}[0]$
  for $k = 0, 1, \ldots, T-1$ **do**
    Sample action $a[k]$ from the fractional MPC based policy using $\bar{\mathbf{s}}[l], \forall l \leq k$ ▷ Eq.(7.7)
    Get $\bar{\mathbf{s}}[k+1]$ by executing $\mathbf{a}[k]$
    $\mathcal{D}_{RL} \leftarrow \mathcal{D}_{RL} \cup \Big\{(\bar{\mathbf{s}}[0], \ldots, \bar{\mathbf{s}}[k], \mathbf{a}[k]), \bar{\mathbf{s}}[k+1]\Big\}$
  end for
end for

BG control is crucial in the treatment of T1 diabetes patients which have inability to produce the required insulin amounts. The low levels of glucose in the blood plasma is termed as hypoglycemia, while the high levels is termed as hyperglycemia. For the application of reinforcement learning, the cost function is taken as risk associated with different levels of BG in the system. In (Clarke and Kovatchev, 2009) a quantified version of risk is proposed as function of BG levels which is written as follows.

$$\begin{aligned} f(b) &= 1.509 \times (\log(b)^{1.084} - 5.381), \\ R(b) &= 10 \times (f(b))^2. \end{aligned} \tag{7.8}$$

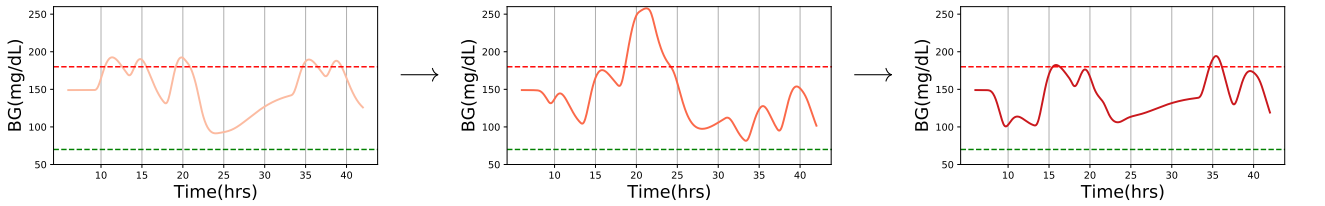
Next, the cost for the transition instance $s[k+1]|s[0], \ldots, s[k], a[k])$ is written as

$$\hat{c}(s[k], a[k]) = R(s[k+1]) - R(s[k]), \tag{7.9}$$

where the state $s[k] \in \mathbb{R}$ represents the BG level at time instant k, and $a[k]$ represents the insulin dose and $R(.)$ is from (7.8). In rest of the section, we experiment with simulated and real-world dataset, respectively.

7.3.1 UVa T1DM Simulator

The UVa/Padova T1DM (Kovatchev et al., 2009) is a FDA approved T1 Diabetes simulator which supports multiple virtual subjects. An open-source implementation of the simulator (Xie, 2018) is used in this work. We take similar simulation setup as

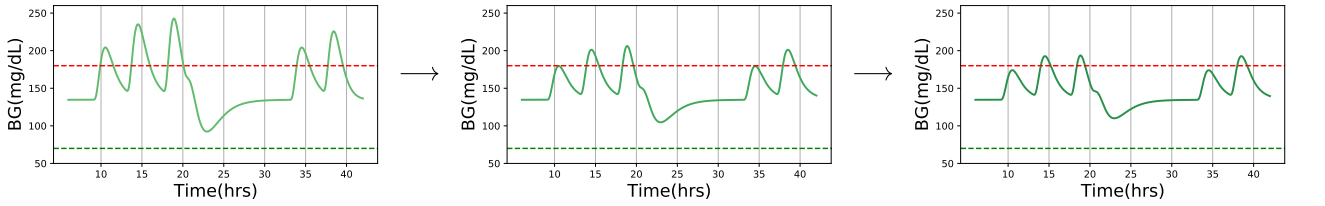


< 70 mg/dL	$70 - 180$ mg/dL	> 180 mg/dL
0%	76.67%	23.32%

< 70 mg/dL	$70 - 180$ mg/dL	> 180 mg/dL
0%	84.52%	15.47%

< 70 mg/dL	$70 - 180$ mg/dL	> 180 mg/dL
0%	93.53%	6.46%

(a)



< 70 mg/dL	$70 - 180$ mg/dL	> 180 mg/dL
0%	74.13%	25.86%

< 70 mg/dL	$70 - 180$ mg/dL	> 180 mg/dL
0%	87.52%	12.47%

< 70 mg/dL	$70 - 180$ mg/dL	> 180 mg/dL
0%	90.30%	9.69%

(b)

Figure 7.2: Blood Glucose (BG) level with time, by implementation of fractional Reinforcement Learning Scheme as Controller, of two Adults in (a) and (b). For each subject, the BG level trajectories are shown from left-to-right in the increasing number of RL iterations with leftmost, middle, and rightmost are outputs at 5, 10, and 15 iterations. As RL iterations increase the MBRL scheme learns better policy and the BG level stays more in the desired level of $70 - 180 mg/dL$. The percentage of time spend in different BG level zone is shown in tables below each plot.

in (Wang et al., 2015). Each subject is simulated for a total of 36 hours starting from 6 a.m. in the morning. The meal timings/quantity are fixed as $50g$ CHO at 9 a.m., $70g$ at 1 p.m, $90g$ at 5:30 p.m, and $25g$ at 8 p.m. On day 2, $50g$ at 9 a.m., and $70g$ at 1 p.m. The continuous glucose monitor (CGM) sensor measures the BG at every 5 mins.

For applying Algorithm 7, we set the horizon length H in MPC be 100 samples, discount factor $\gamma = 0.99$. The s_{min}, s_{max} in MPC problem (7.7) are set as $70, 180$ respectively. The maximum number of RL iterations $iter\_max$ are set as 30. We show the BG output of the simulator using Algorithm 7 as controller in Fig. 7.2. We observe that the fraction of time BG stays in the desired zone $70-180mg/dL$ increase with increasing the learning iterations in the Algorithm 7. The data gathered using on-policy helps the model making better prediction, and with as few iterations as 15 we have more than 90% of time BG stays in the desired levels.

7.3.2 Real-World Data

Testing the controllers on real-world systems is difficult because of the health risks associated with the patients. We take the Diabetes dataset from UCI repository (Dua and Graff, 2017) which records the BG level and insulin dosage for 70 patients. While testing controller is not possible here, hence we present the analysis regarding the modeling part. The long-range memory in the signals exist if the associated fractional exponent lies in the range of $(0, 0.5]$ as noted in Section 7.2.3. In Fig. 7.3, we show the log-log plots of the variance of wavelets coefficients at various scales, for two subjects. We observe that the estimated value of α lies in $(0, 0.5]$ which indicates presence of long-range memory, and hence fractional models can be used to make better predictions.

7.3.3 Discussion

Insulin dependent diabetes mellitus (IDDM) is a kind of chronic disease characterized by abnormal BG level. Topically, high level of BG, which is caused by either the pancreas does not compound enough insulin (a hormone that signals cells to uptake glucose in the bloodstream) or the produced insulin cannot be effectively used by the human body, can result in a disorder of metabolic that give rise to irreversible damage (such as lesion of patients' organs, retinopathy, nephropathy, peripheral neuropathy and blindness) (Tejedor, Woldaregay, and Godtliebsen, 2020; Derouich and Boutayeb, 2002). According to recent research, nowadays, IDDM is influencing 20-40 million people around the world and this amount is increasing over time (You and Henneberg, 2016). Related work (Diabetes Control and Complications Trial Research Group and

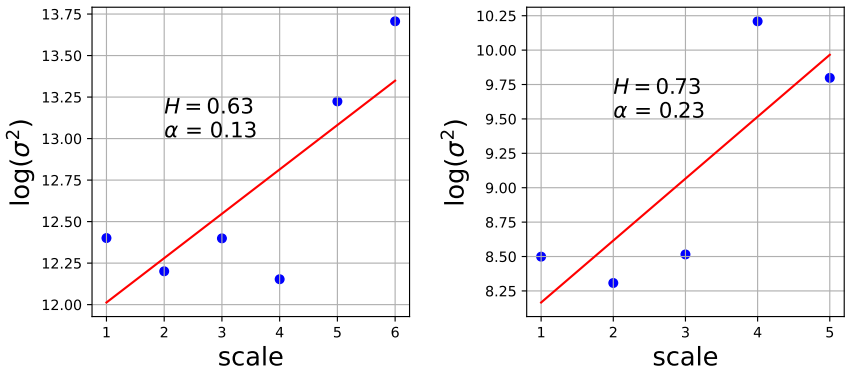


Figure 7.3: The log-log plot of variance of wavelet coefficients vs scale of two subjects in (a) and (b). The values of α lies in $(0, 0.5]$ which indicates long-range correlations.

others, 1995) presents that tight blood glucose control along with insulin injections can help to control the disease, however intensive control can result in the risk of low blood sugar. This symptom can increase the risk of heart disease, or even sudden death.

To effectively and safely against with IDDM, in the work of (Coffen and Dahlquist, 2009), the authors constructed a CGM to detect the insulin among in the individuals in real time. This monitor can read the blood glucose of the patients for every 5 mins. Combining with an insulin pump (a small device that automatically inject insulin), the CGM can constructed a system called "artificial pancreas" (AP). The AP system is designed to control the symptom of patients which can dynamically predict the among of insulin the individual should be delivered. For many years, researchers have worked on designing efficient algorithms/models to correctly predict the required insulin of individuals in AP systems. In this chapter, we present an innovative MBRL algorithm which is explored in the non-Markovian model to dynamically make the prediction of the among of insulin patients demand with high-accuracy and high-efficiency.

7.4 Conclusion

There are many important learning control problems that are not naturally formulated as Markov decision processes. For example, if the agent cannot directly observe the environment state, then the use of a partially observable Markov decision process (POMDP) (Smallwood and Sondik, 1973) model is more appropriate. Even in presence of full observability, the probability distribution over next states may not depend only on the current state. A more general class can be termed as history dependent process (HDP), which can be looked as infinite-state POMDP (Leike, 2016). Another non-MDP class for model-free is Q-value uniform decision process (QDP) (Majeed

and Hutter, 2018). The non-Markovianity in the rewards structure is explored in (Gaon and Brafman, 2019; Agarwal and Aggarwal, 2021) which utilize model-free learning, and RL for POMDP is explored in (Perez and Silander, 2017) which is also model-free. MBRL is used for various robotics application (Nagabandi et al., 2018) in the MDP setting. The deep probabilistic networks using MDP is used in (Chua et al., 2018).

In this work, we constructed a non-Markovian Model Based Reinforcement Learning (MBRL) algorithm consisted with fractional dynamics model and the model predictive control. The current Reinforcement learning (RL) approaches have two kinds of limitations: (i) model-free RL models can achieve a high predict accuracy, but these approaches need a large number of data-points to train the model; (ii) current models don't make latent behavioral patterns into considerations which can affect the prediction accuracy in MBRL. We show that our non-Markovian MBRL model can validly avoid these limitations. Firstly, in our algorithm, we gather additional on-policy data to alternate between gathering the initial data, hence it needs less sample points than the general model-free RL approaches. Secondly, fractional dynamical model is the key element in our algorithm to improve/guarantee the prediction accuracy. The experiments on the blood glucose (BG) control to dynamically predict the desired insulin amount show that the proposed non-Markovian framework helps in achieving desired levels of BG for longer times with consistency.

The richness of complex systems cannot be always modeled as Markovian dynamics. Previous works have shown that the long-range memory property of fractional differentiation operators can model biological signals efficaciously and accurately. Thus, we have modeled the blood glucose as non-Markovian fractional dynamical system and developed solutions using reinforcement learning approach. Finally, while the application of non-Markovian MBRL open venues for real-world implementation but proper care has to be taken especially when we have to deal with the healthcare systems. The future investigations would involve more personalized modeling capabilities for such systems with utilization of the domain knowledge. Nonetheless, we show that the use of long-range dependence in the biological models is worth exploring and simple models yield benefits of compactness as well as better accuracy of the predictions.

Chapter 8

Neuron Particles

Unlike existing work that assumes that the neuronal spike trains are Markovian and models their activity through the Poisson point processes formalism, we show that the inter-spike intervals are more regular, possess long-range memory and fractal characteristics and their statistics are better described by fractional order partial differential equations. Moreover, we show that the microscopic statistics encode emergent macroscopic topological properties about the underlying network, and display superior predictions of animal behavior during multiple cognitive tasks. Through simulations, we further discover how the macroscopic neural network activity influences the microscopic statistics of neuronal spiking dynamics. This framework formalizes, generalizes and unifies the mathematical modeling of microscopic neuronal dynamics while opening new analytical avenues for cognitive and sensory computational neuroscience including detecting and studying cognitive and sensory abnormalities or psychiatric disorders.

8.1 Neuron Particles Model

We introduce a novel concept of neuron particles to explain the causal fractal memory and scale component in the spike trains. We define the inter-spiking intervals (ISI) in the spike trains as neuron particles of the size corresponding to the length of ISI, Figure 8.1 (left). As shown in Figure 8.1 (right), the neuron particles start from the origin of space-time and make a jump when the associated ISI recur in the spike train at time t. The jump made by a particle in space is a fixed proportion of its size. Therefore, smaller particles make smaller jumps and bigger particles bigger jumps. The neuron particles traverse trajectories $X(t)$, which we model as realization of a multi-parameter fractional partial differential equation (FPDE) for each spike train

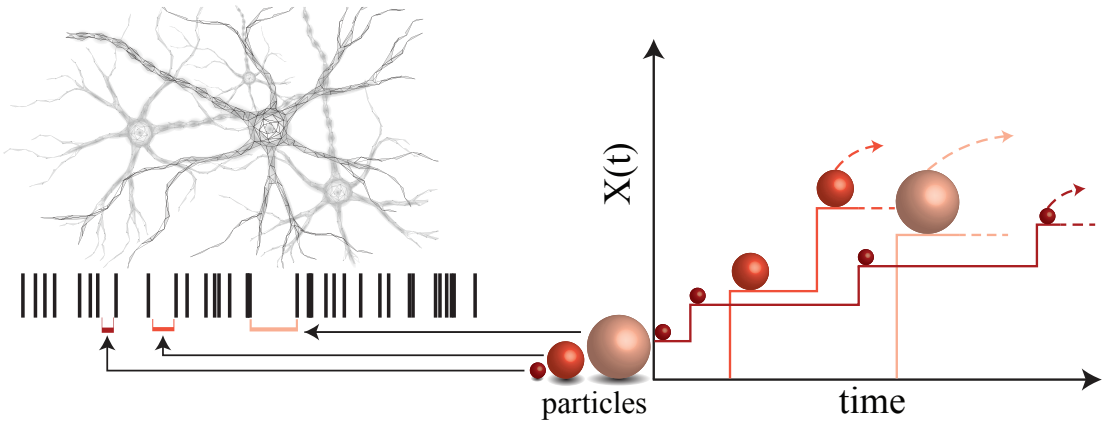


Figure 8.1: Neuron particles extracted from discrete events data. (Left) Particle size is proportional to the inter-event times, (Right) and the corresponding trajectories $X(t)$ of the particles across time t.

as follows

$$ {}_{t}\mathcal{D}_{*}^{\beta} u(x,t) = D \times {}_{x}\mathcal{D}_{\theta}^{\alpha} u(x,t), \qquad \forall x \in \mathbb{R}, \forall t \in \mathbb{R}^{+}, \tag{8.1} $$

The space-time fractional diffusion equation captures the scale and fractal memory using fractional Riesz-Feller (Feller, 1962) (order α, skewness θ) and Caputo time-fractional (Caputo, 1967) (order β) derivative, respectively. For more details, we refer to Chapter 4. The trajectories are used to estimate the parameters of the FPDE using the fractional moments algorithm, Algorithm 3. For constructing the trajectories, we first need to identify the eligible particles from the ISI.

The procedure to obtain the neuron particles, Neuron Particles Process (NPP), is formally outlined next.

8.1.1 Neuron Particles Process

A neuron particle trajectory is computed by identifying the repeated patterns in the spike train data at various scales. The spike train data recorded from a single unit is represented as a sequence of inter-arriving intervals $T = \tau_1, \tau_2, \ldots, \tau_M$ which is resulted from $M + 1$ spike times. A particle is termed as unique inter-spiking interval, and for the computational purpose is used as a continuous range of inter-spiking intervals. For complete details of neuron particles process is discussed in the next subsection. For a particle τ_p with continuous range $[\tau_p^l, \tau_p^u)$, we first define the particle arrival times as

$$ \begin{aligned} i^* &= \inf\{i | \tau_p^l \leq \tau_i < \tau_p^u \wedge (i-1)^* < i\}, \\ t_i &= \sum_{j=1}^{i^*} \tau_j, \end{aligned} \tag{8.2} $$

where, t_i is the ith arrival time of the particle τ_p in the inter-arrival time sequence T and $0^* = 0$. Next, using the arrival times, we now define the neuron particle trajectory as follows:

$$X(t) = l\bar{\tau}_p, \quad t_l \leq t < t_{l+1}, \tag{8.3}$$

where, $\bar{\tau}_p$ can be defined as average particle size with $\bar{\tau}_p = (\tau_l + \tau_u)/2$ and the start time is taken as $t_0 = 0$. The multiple trajectories are used as input for the FPDE estimation Algorithm (Znaidi et al., 2020) using moments approach.

8.1.2 Neuron Particles Algorithm

The neuron particles process (NPP) aims at obtaining a computational version of the neuron particles from the real-world spike trains. For a scenario where the inter-spike intervals (ISIs) belong to a set of finite cardinalities, for example, Cantor spikes (in Section 8.1.3), the neuron particles can be uniquely identified. However, for a general setup, such assumptions do not hold and ISIs are some real numbers, identifying unique ISIs (or neuron particles) is not computationally useful. Therefore, we resort to 'range of ISI' as 'neuron particles', for the sake of computation, as we discussed in the previous section. We now present a novel algorithm NPP in Algorithm 8, that computes neuron particles from the spike train data which is also pictorially shown in the Figure 8.2.

8.1.3 Cantor Spikes

The multiscale phenomena and causal fractal memory captured by neuron particles in a spike train can be understood and validated by the hypothetical concept of Cantor spikes. Cantor sets (Vallin, 2013) are formed by repeated deletion of a fixed ratio r from a unit interval $[0, 1]$ as shown in Figure 8.3 (Top). The Cantor sets have numerous remarkable properties, and particularly useful in our case, is its fractal nature (statistical self-similarity). We generate Cantor spikes such that the successive ISIs are deleted segments of a Cantor set from left to right for given depth N and ratio r. The estimated values of scale and fractal memory parameters using neuron particles concept for Cantor spikes are shown in Figure 8.3 (bottom). The box dimension captures the fractality of Cantor spikes. We note that different depth N generate different lengths of Cantor spikes $(2N)$ as well as different neuron particles $(r, r\rho, \ldots, r\rho^{(N-1)})$, where $\rho = (1 - r)/2$. From Figure 8.3 (bottom), we see that the fractal memory parameter β primarily is a variant of the fractality measure (box dimension) while estimations for various depths N cluster together. Therefore, the memory β captures fractal repeating patterns at various scales and does not depend on the depth. The overlap in scale parameter α of Cantor spikes across different box

Algorithm 8: Neuron Particles Process (NPP) Algorithm

Initialization: Given a spike train series with the sequence of inter-spike intervals (ISI) as $T = \tau_1, \tau_2, \ldots, \tau_M$, a binning algorithm $\mathcal{A}$, an initial number of neuron particles n_i, lower/upper acceptable values of ISI frequency as *lb/ub*.

1: Using algorithm $\mathcal{A}$ with number of bins n_i get the histogram of ISI sequence T with bin boundaries $(\tau_1^l, \tau_1^u), (\tau_2^l, \tau_2^u), \ldots, (\tau_{n_i}^l, \tau_{n_i}^u)$ and bin counts $b_1, b_2, \ldots, b_{n_i}$. The constraint on data sequence T is such that $b_1 \geq lb$, otherwise terminate the NPP algorithm.

2: **for** $i = 1, 2, \ldots, n_i$ **do**
3: **if** $b_i > ub$ **then**
4: For the data subsequence $T \cap \{\tau | \tau \in [\tau_i^l, \tau_i^u)$ get the histogram with number of bins as $\hat{b}_i = \lfloor b_i/ub \rfloor$. Update the histogram of the sequence T with the new bin edges and bin counts. This is termed as 'split' step. The updated histogram of T has $m_u \geq n_i$ bins with bin counts $b_1, b_2, \ldots, b_{m_u}$.
5: **end if**
6: **end for**

7: With $i = m_u$,
8: **while** $i > 1$ **do**
9: **if** $b_i < lb$ **then**
10: Start with ith bin and 'merge' consecutive bins such that merged bins have bin counts exceeding lb or with $\hat{b}_i = b_i$, iterate over $j = i-1, i-2, \ldots, 1$ as $\hat{b}_i = \hat{b}_i + b_j$ till $\hat{b}_i \geq lb$. Update the new bin boundaries as (τ_j^l, τ_i^u) and ith bin count $\hat{b}_i$ and set $i = j - 1$. This is termed as 'merge' step. The resulting histogram after merging has total bins m with bin counts $b_1, b_2, \ldots, b_m$.
11: **end if**
12: **end while**

13: Output the final iterated bin edges $(\tau_1^l, \tau_1^u), (\tau_2^l, \tau_2^u), \ldots, (\tau_m^l, \tau_m^u)$. The bin edges are referred as neuron meta-particles which consists of neuron particles such that ith meta-particle represents a group of particles with size ranging between $[\tau_i^l, \tau_i^u)$.

dimensions by varying N is due to common neuron particles among different (r, N) pair. Finally, a special case of uniform spikes is shown at the bottom (in pink) where the ISIs are equal. Since the increments, or ISIs, are constant across time, the uniform spike train is straightforwardly modeled as a first order linear differential equation

$$\frac{\partial u(x,t)}{\partial x} = D \times \frac{\partial u(x,t)}{\partial t}.$$

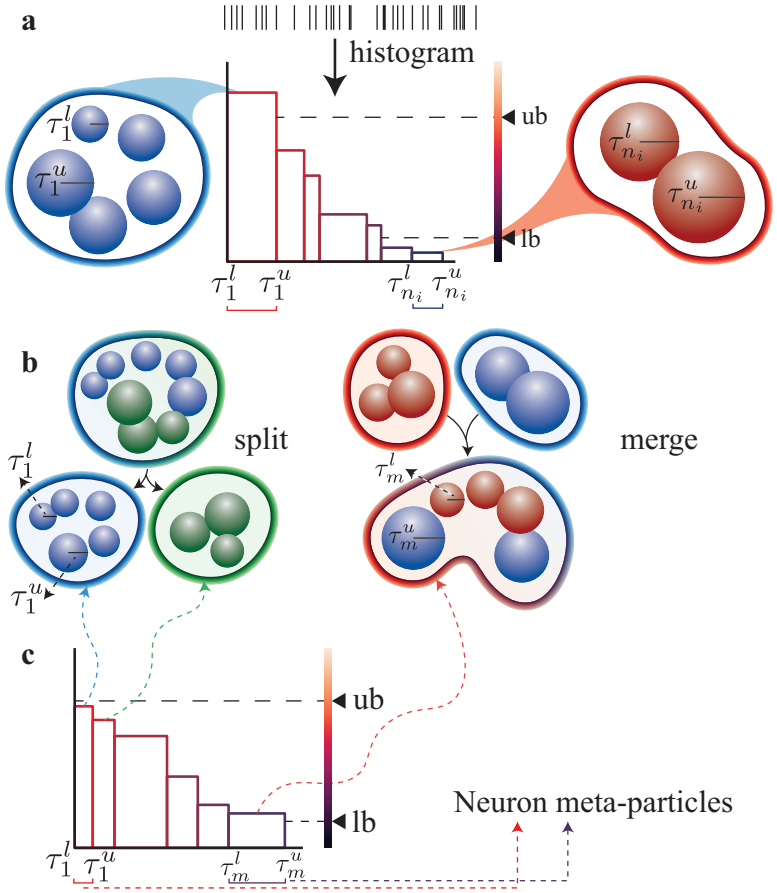


Figure 8.2: **Neuron particles process**. (a) A histogram of the inter-spike intervals (ISIs) from a single unit spiking data (see the spike train on top) is generated. Each interval is treated as a group of particles with the minimum radius τ_i^l and the maximum radius τ_i^u. Two specified parameters 'ub' and 'lb' determine the upper- and lower-bound on the number of particles in each group. (b) Groups with a larger number of particles than the upper bound are broken into smaller groups, while smaller groups are combined using 'split' and 'merge' steps, respectively. The iteration continues until each particles' group has a number of particles in the range $[lb, ub]$. (c) In the resulting histogram of ISIs, the ISI intervals are termed 'neuron meta-particles' and are such that the count of particles within them are in the range $[lb, ub]$. In the current work, we refer to the "neuron meta-particles" simply as "neuron particles".

Also, by using the neuron particles concept, we see that for the uniform spikes, the estimated value of $\alpha = \beta = 1$ in Figure 8.3 (pink dot) represents first order differentials for both space and time.

8.2 Topological Features of Neuronal Networks influence the Microscopic Neuronal Dynamics

The complex patterns in the neuron spikes are the result of the individual neuron transfer function and the network feeding into each neuronal unit (Holtmaat and

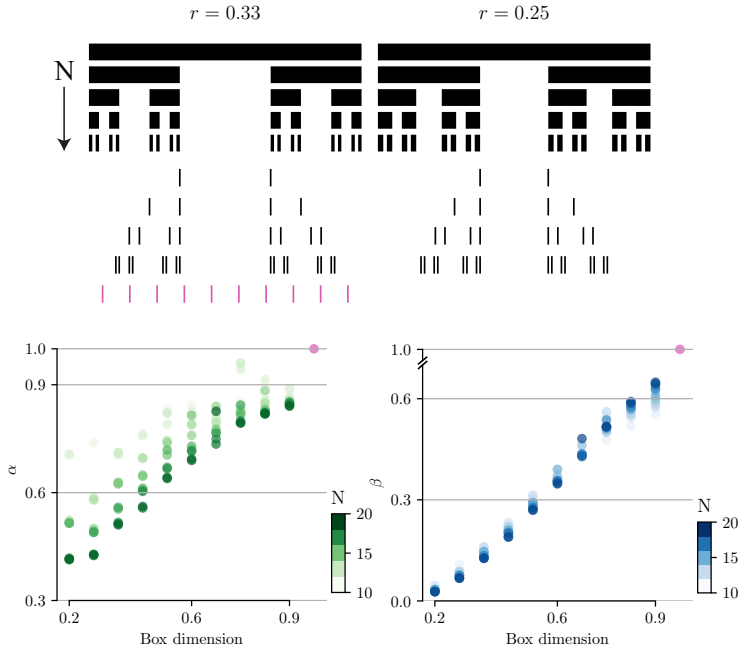


Figure 8.3: (Left) Cantor set example to generate scale-free spike train patterns. The deletion ratio (r) can vary from 0 to 1 ($0 < r < 1$) and division factor $N > 1$. A special case of uniform spike trains with equal ISI across time in pink color. (Right) Estimated fractional PDE parameters (α, β) for Cantor spike trains with different box dimensions. The box dimension is the slope of the logarithm of the number of ISI of size ϵ, i.e., $\log(\epsilon)$ and the logarithm of the inverse ISI size, i.e., $\log[1/\epsilon]$. For a Cantor-like spike train, resulting from the corresponding Cantor set, the box dimension is $-\log(2)/\log((1-r)/2)$. The special case of uniform spikes in pink can be represented as a differential equation in pink in c which has $\alpha = \beta = 1$. The estimated values of (α, β) for uniform spikes is pink dot in e.

Caroni, 2016; Parr et al., 2019). An individual neuron is part of multiple closed-loops of different lengths that induce a complex pattern of recurring ISIs. By capturing the multi-scale repeating patterns in the ISIs, from the network perspective, β is related to the cumulative effect of multiple closed-loops around a neuron in the network. The network has influence on the statistics of the spike train (Trousdale et al., 2012; Pernice et al., 2012). The scaling of path lengths from a neuron to its neighbors at different hops can be captured using node-based multifractal spectrum. The scale of ISI which is represented using α is intuitively related to the node-based multifractal dimension. We studied the effects of a wide range of complex network topologies on the fractal memory and scaling properties of a single neuronal spike train.

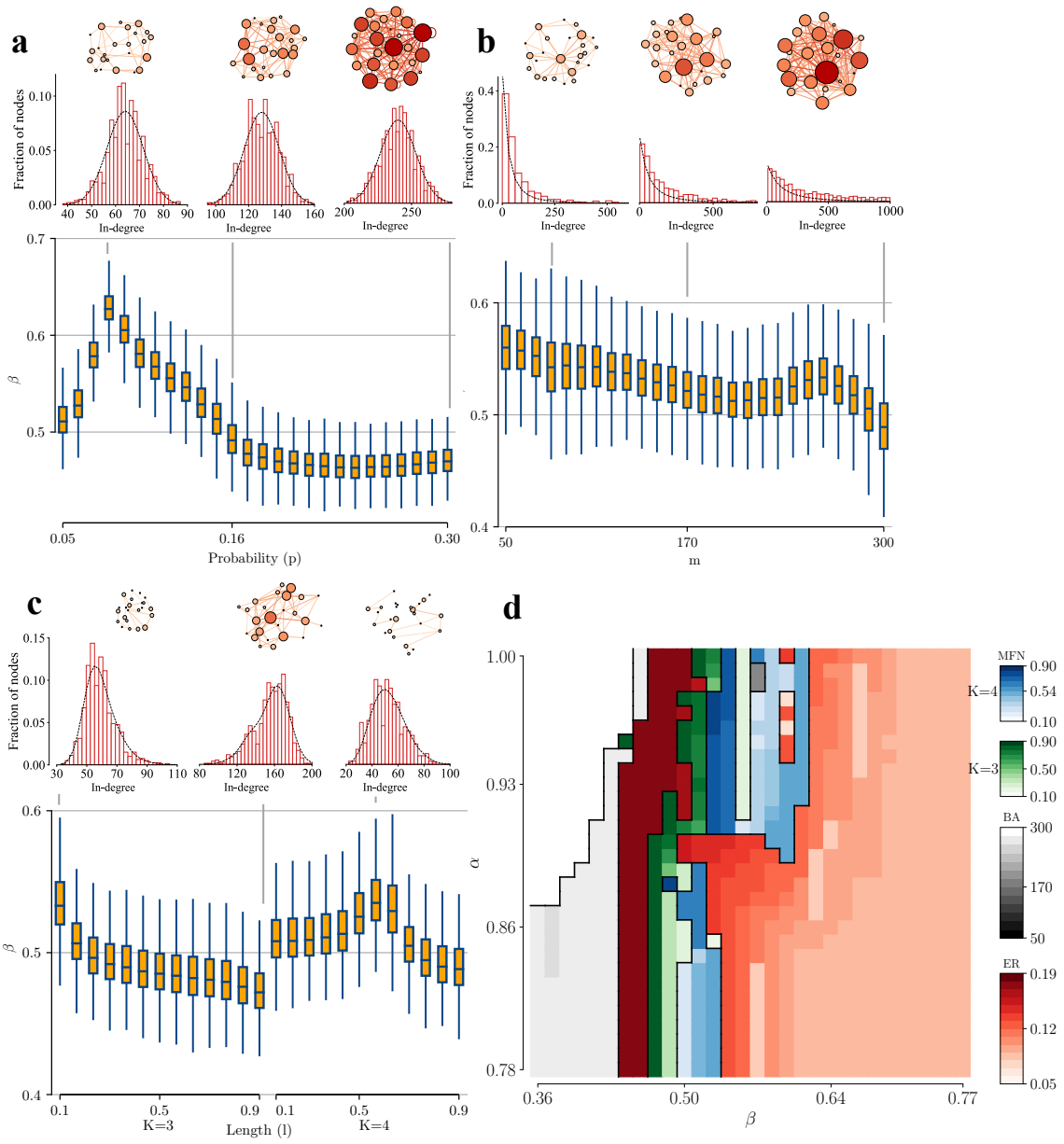


Figure 8.4: **Network topologies influence the memory of neuronal dynamics**. (a), An Erdös-Rényi random network with probability of existence of edge being p. The probability varies linearly from 0.05 to 0.30 and consequently increases the average node-degree. The excitatory neuron node-degree scaled histogram is in red bars and the black curve fits the histogram with Gaussian distribution. The variation of the excitatory neuron β with respect to p is represented as box plots. The line across the box plot indicates the median, and upper and lower hinges indicate 75th and 25th percentiles (making IQR), respectively. The whiskers extend to the lowest and the largest values but within 1.5xIQR. (b) Dorogovtsev (Dorogovtsev, Mendes, and Samukhin, 2000) extension of Bárabasi-Albert (Barabasi and Albert, 1999) scale-free network with outgoing links m and attraction parameter $a = m$. The excitatory neuron node-degree scaled histogram is shown in red bars and the black curve fits the histogram using analytical expression (Dorogovtsev, Mendes, and Samukhin, 2000). The outgoing links m varies from 50 to 300, and the variation of excitatory population fractal memory parameter β is shown as box plots. (c) Multi-fractal network (Palla, Lovász, and Vicsek, 2010; Yang and Bogdan, 2020) with base measure $P = [0.6, 0.5; 0.5, 0.4] (m = 2)$ and length measure $L = [l, 1 - l]$ with l variation linearly from 0.1 to 0.9 for two values of depth $K = 3, 4$. The excitatory neuron node-degree scaled histogram is in red bars and the black curve fits the histogram with the analytical expression of degree distribution (Palla, Lovász, and Vicsek, 2010). The excitatory neuron population fractal memory parameter

β is represented as box plots. (d) Prediction spectrum of the networks using α and β computed from a single spike train. For each tuple (α, β) the most likely network is indicated by its corresponding color. For each network type (topology as well as parameters), 50 random realizations of networks were generated, and the estimated parameters for the neuron populations were concatenated for box plots.

The variation of fractal memory parameter β with different network topologies is shown in Figure 8.4. For Erdös-Rényi (Erdos and Renyi, 1960) (ER) model of a random network, we see in Figure 8.4a that as the network becomes denser from the sparse state ($p = 0.05$) the memory parameter β of the ensemble increases and then decreases. A saturation occurs for dense networks with $p > 0.2$. The large number of connections provides a large input current to the neurons, thus, saturating the spiking patterns. The existence of a maximum point in the memory parameter shows that for some topologies (near $p = 0.08$), the network spikes are in a structured fashion. Next, in Figure 8.4b, we see that for scale-free Bárabasi-Albert (BA) networks, the fractal memory decreases as outward links (m), independent of total number of neurons in the network, increases. Finally, for multi-fractal network (MFN) in Figure 8.4c, the trend of fractal memory is inversely proportional to length measure l for depth $K = 2$. However, as depth K increases the network gets sparser, and for $K = 4$ the fractal memory attains a peak for l near 0.5 and then decreases. Taken together these results suggest that critical information about the type of network feeding into the single unit spike trains can be deduced from analyzing the spike trains using the neuron particle method. The information is not complete, but over certain values of α, β, and assuming a finite range of networks, we can describe the probability of features of the underlying network given a single spike train. To the best of our knowledge this has never been accomplished before. Such a probabilistic representation of the topology of the underlying network enables predictions of neural dynamics and behavior.

Using the observed variation in the fractal memory parameter β (Figure 8.4a-c) and scale parameter α (as presented in (Gupta et al., 2021b)), a prediction spectrum is constructed in Figure 8.4d (full version in (Gupta et al., 2021b)). For each constituent networks ER, BA, and MFN with various parameter ranges, we show the most likely network that an estimated tuple (α, β) belongs. In the low values of α and β, the most probable network is BA with $m \sim 300$, and ER with high connection probability (~ 0.19). On the contrary, high values for both parameters indicate ER with low connection probability. For some tuple values, almost all networks are equiprobable, and hence, it is difficult to predict a single one.

8.3 Neuron Particles Predict Behavior

To determine the extent to which the neuron particle analysis of single unit data is capable of predicting the behavior of an animal during a cognitive task we analyzed a

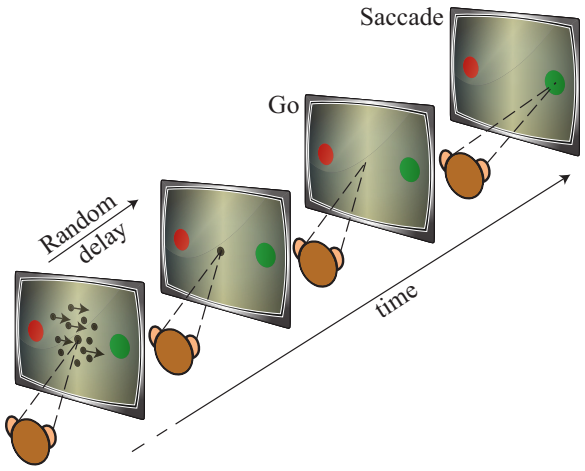


Figure 8.5: Direction discrimination task by the animal.

few different publicly available data sets. In the first data set, hundreds of single units were recorded simultaneously in the pre-arcuate gyrus of Macaque monkeys while they were performing a direction discrimination task (Kiani et al., 2014). Briefly, on each trial, the monkey views moving dots on a screen for 800 ms. A fraction of dots moves coherently in the same direction towards one target or another (T1 or T2) while the rest of the dots are displaced randomly (Figure 8.5). After the lapse of a random delay period, upon receiving a Go cue, the monkey makes a saccadic eye movement towards the target T1 or T2, where it thinks the dots were moving. We compared the prediction accuracy of the monkey's choices based on a commonly used mean firing rate approach versus the neuron particles approach (Figure 8.6). We found a performance gain, with an averaged difference of 5.75% around the Go cue. Around the saccade initiation time, the spikes are distinct enough across the units such that both the firing rate and neuron particles trajectories perfectly predict the choice. However, neuron particles consistently outperform during motion viewing and delay periods.

In an additional experiment, we randomly (uniform) sampled only 50% of the units to mimic a partial knowledge setup and predicted the monkey's choice from the reduced data. The difference between firing rate and particles approach around the Go cue is 6.74%. With half the population, the prediction accuracy gap further increases. Another partial data setup is presented in the Fig. S7, where for each single unit we randomly removed some of the spiking event data. We see that when using only 50% of the data from each unit, the neuron particles approach suffers 1.76% performance loss, while mean firing rate suffers 4.70% loss at Go cue. Next, on using only 25% of the spiking data from each unit, the performance loss of neuron particles/mean firing rate is 4.52%/11.23% at Go cue. The fractality in the spike trains is less affected if some spikes are removed. The particles approach extracts more information from the

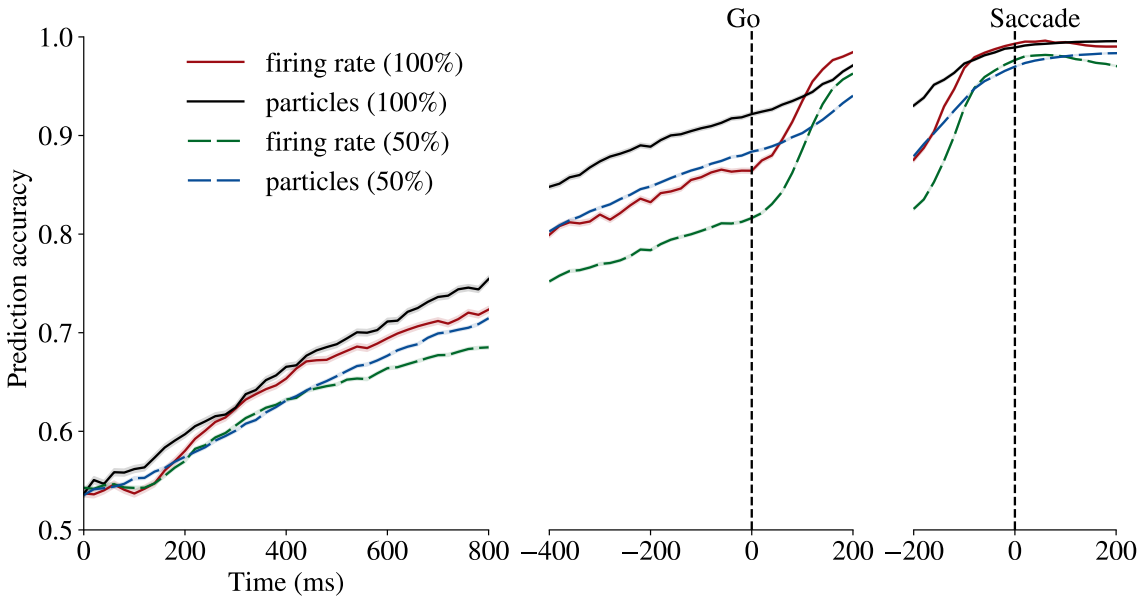


Figure 8.6: A comparison of choice prediction using recordings of neuron ensembles for neuron particles and mean firing rate approach. Both approaches use logistic regression with linear kernel and train/test split as 90/10. The particles trajectory is sampled every 20ms, and the firing rates are computed in a sliding 100ms bin with 20ms slide. The mean accuracy ± s.e.m. (across sessions and resampling) is indicated in dark and shaded colors. An additional case of 50% is considered, where to mimic the reduced data scenario, out of total, 50% uniformly random sampled units are taken, respectively. The sampling is done 5 times independently.

same data which in turn can be utilized to better discriminate the spike patterns over time to predict the monkey's choice.

8.4 Scale and Memory Parameters are Invariant to Network Size

The features used to infer the network topological structure from a single unit should ideally be independent of the network size. Additional experiments in Figure 8.7 suggest that the scale (α) and fractal memory (β) parameters are indeed invariant to the network size under consideration. In Figure 8.4, the network size is $N = 1000$ and we take this as baseline to observe the network size effects upon changing N. By varying the network size from $N = 1000$ to $N = 10000$, the patterns in α and β are relatively stable for each network topology. This shows that the topology shapes the fractality of each network unit, independently of its size. The neuron particles, by capturing such fractality, can differentiate among different topological structures to a certain extent. We note that, for large BA networks with higher values of m, many connections occur for some nodes, and thus parameter estimates are variable due to massive incoming currents in the hub nodes. For the smaller

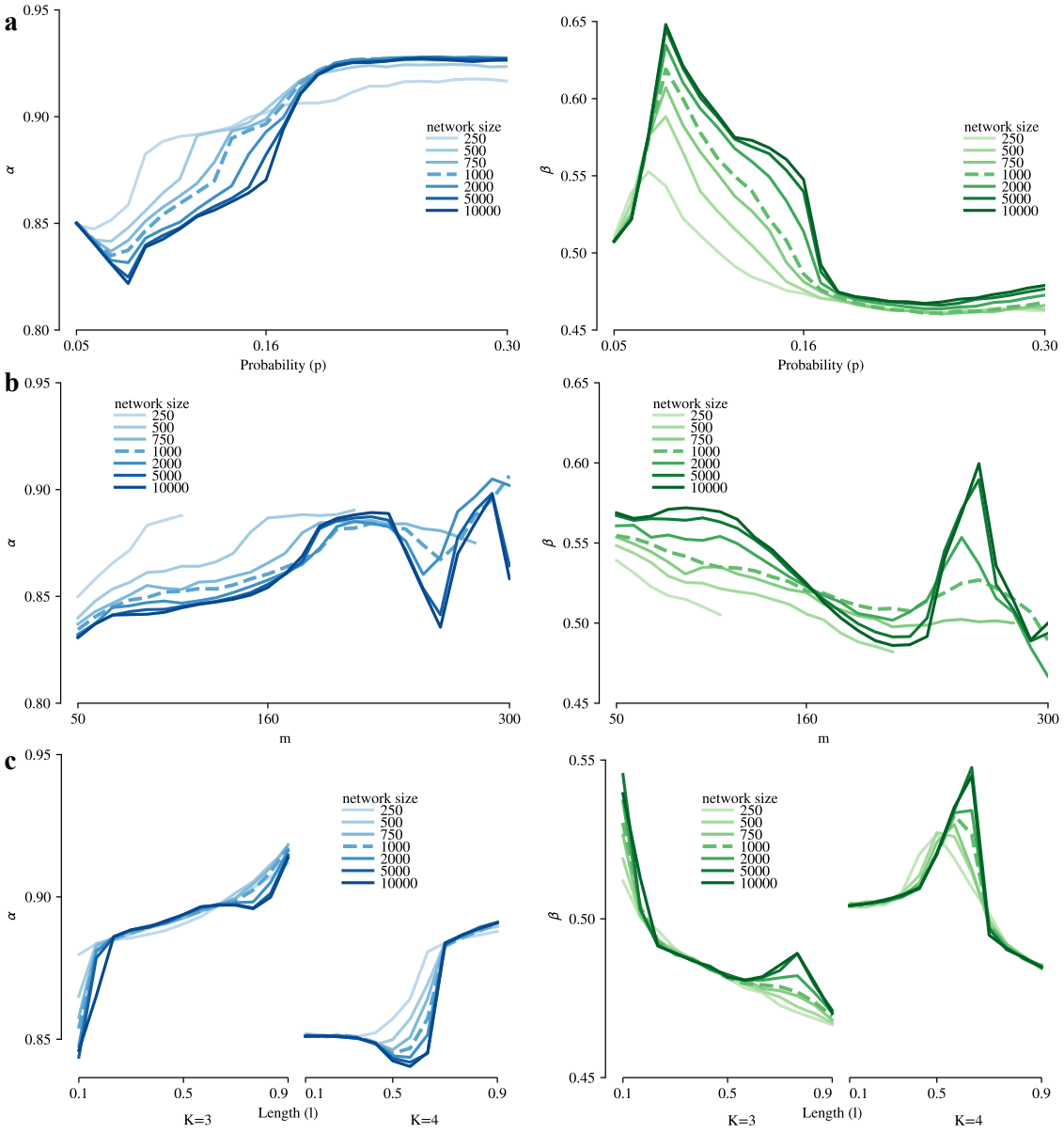


Figure 8.7: **Memory and scale invariance to the network size**. Network configurations for Erdös-Renyi (ER), Barabasi-Albert (BA), and Multi-fractal network (MFN) is same as in Figure 8.4. For the BA networks, the smaller networks $N = 250, 500$ and 750, the larger values of m are not feasible to simulate a power-law degree distribution. The highest value of m is limited as $m = 110, 210, 290$ for $N = 250, 500, 750$, respectively. The median of node parameters (scale α, memory β) is shown for different network size for each network topology in a, b, c.

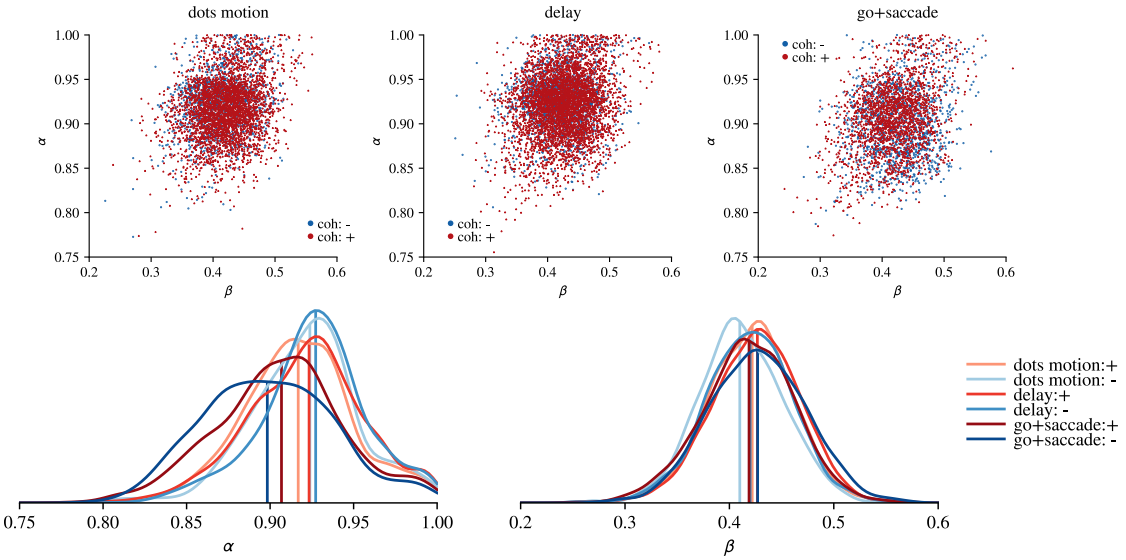


Figure 8.8: The estimated parameters for monkey 1 across three epochs: motion viewing (200-800 ms after motion onset), delay (200 ms after dots offset to -50 ms from go cue), and go+saccade (go cue to 200 ms after saccade initiation). The parameter tuple was separately calculated for motion in favor and against the neuron's preferred saccade (+ and – motion coherence, respectively). (d, e) Gaussian-kernel smoothed densities for α, β across three epochs and 2 coherence levels with straight line being the median.

networks, to maintain the same topological setup, the paucity of stimulating currents leads to a reduction in the spiking activities of the neurons. Therefore, for smaller networks the extracted network fractality patterns saturate and diverge from the bigger networks. The patterns quickly change for all three network types between $N = 250$ to $N = 750$. For ER and MFN topologies with dense neuron connections, the reduction in N doesn't impact the patterns as expected due to sufficient current flow. Finally, the maximum network size used in the current experiment was $N = 10000$, because of computational restrictions. However, the current observations of similar patterns in (α, β) should hold for larger neuron networks as well. From these comprehensive experiments, we observe that by making the networks larger and with proper upscaling of the stimulating currents, the spiking activity of each neuron does not diminish. Therefore, the fractality induced by network topology in the spiking data can be captured.

8.5 Stability of Memory and Scale Parameters

The neuron particles model also enables tracking functional changes in the network topology across species, tasks, and task epochs. Figure 8.8 shows the inferred model parameters (α, β) for different motion directions and epochs of the direction discrimination task in the monkey pre-arcuate gyrus. The memory parameter β is highly

stable, compatible with the stability of intrinsic connectivity and the underlying dynamics of neural responses across tasks and task epochs in this region of the monkey cortex (Kiani et al., 2014). Since β reflects the collective effect of the multiple closed-loops in the topology, we predicted little change across task epochs. The scale parameter α which reflects path lengths and in-degree is also stable (Figure 8.8). The stability of α and β despite obvious changes in the sensory inputs and motor outputs involved in the three distinct epochs suggests the neuron population-level response statistics and network topology are stable. This stability is reminiscent of the random network topology which has been hypothesized to be present in the pre-frontal cortex (Rigotti et al., 2013) to support adaptive learning and behavior. In contrast, if changes in input and output across epochs were associated with emergence of new loops or changes in path lengths of network (e.g., activation of a hub neuron in a BA network), such stability would be unexpected.

8.6 Discussion

In this chapter, we discussed a novel statistical physics inspired model, which captures the causal memory and fractal patterns of inter-spike intervals recorded in vivo from single neurons. More precisely, we define the "neuron particle" as the time interval between consecutive spikes, that through its path-based history captures the degree of long-range memory emerging in neuronal activity due to intrinsic local computation and interaction with other networked neurons. We first show how the hierarchical information encoded in the spike trains is captured using the proposed neuron particles model (Figure 8.1). Next, we show how the neuron particles model can encode information about the underlying neural networks that were not directly and completely observed (Figure 8.4).

Although it is widely recognized that the pattern of spike trains in one specific neuron is a reflection of a network of interacting neurons and glia connected to it, to the best of our knowledge, no previous study has shown that it is possible to predict network topology based on single unit activity. Previous studies largely focused on pairwise correlations or higher order statistical features based on neural population responses (Trousdale et al., 2012; Pernice et al., 2012). Although knowledge of network connectivity enables predictions of neural responses features, the converse has been elusive until now (Kispersky, Gutierrez, and Marder, 2011; Sporns, 2012).

The proposed method of neuron particles generalizes to any complex network with interactions between nodes with discrete time intervals between activities. As an example, first, the online hate events which have been linked with several extremist activities (Johnson et al., 2019) could be used to infer the underlying terrorist-group network structure. Monitoring the complete network is difficult as such individuals

are very discreet. Using the 'social particles' analogy, using very few users' data, the network topology could be inferred using computed fractality of events. Armed with the topology, it may be possible to predict the timing and implicated parties involved in a terrorist attack by detecting activity in only a few of the nodes, and not necessarily the actual perpetrators. As another example, climate change events impact rainforests across the globe. The detailed forest maps (Réjou-Méchain et al., 2021) and tree species-based information could be used to make a time-varying complex network. Extracting the fractality in the climate change events by 'climate particle' jumps and observing its variations by changing tree population network, the next damaging event, especially for vulnerable tree species could be foretold. Third, the impact of the introduction of species by humans to the network of interactions among other species brings ecological and evolutionary consequences (Fricke and Svenning, 2020). Capturing the change in the network over a global scale is challenging, but using the particles approach, the change in the change in α and β obtained using a 'species particles' jumps from only a handful of nodes can provide a change in the underlying network topology. The time fractality computed over multiscale interactions provides a way to capture human interventions over several years.

Our advance has several important implications for neuroscience applications. The first is that the neuron particles model can be used to analyze single unit data to capture the causal information and long-range memory contained in the statistically self-similar (repeating) patterns of time intervals between spikes for superior inference. The underlying networks of the brain could be inevitably more complex and may not be a direct match with any of the considered network topologies here. But, since the considered networks in Figure 8.4d span a wide spectrum of complex networks, they can be treated as canonical networks. Future studies are needed to consider scenarios where the network is a blend of various canonical structures, and whether the neuron particles can extract proportions of these constituent networks.

Second, if features of the underlying neuronal networks can be decoded then theoretically better predictions of behavior should be possible from single unit recordings. Indeed, the real-world examples show how using the neuron particles produces superior predictions of behavior when compared to conventional approaches which use the mean firing rates (Figure 8.6). By the same logic, better predictions of neuronal activities should also be possible. If the network can be inferred, then that should constrain possible paths of information in response to new stimuli, or specific neuronal activation patterns. To enable predictions of neuronal responses to new stimuli, future work should develop causal inference techniques to identify composable sets of multi-fractional PDEs for multi-dimensional trajectories. Doing so would enable to study the interactions of neurons explicitly using population spike trains. The working hypothesis of the current work is that the reflection of interaction with the complete network is present in the spike train of each neuron, which we capture

through α, β. However, with multi-dimensional modeling, a better prediction could be made of joint neuronal activities, for example, inter-neuron spike correlations.

Third, the neuron particles model can allow insights into network structures that enable how the brain processes information. In the brain, interactions of large neural populations implement computations, decisions, and perceptions. The brain, therefore, does not need to rely on the neuron particle concept for its computations. In fact, it is unlikely that the brain extracts information in an analogous way as our statistical model. However, the neuron particles model provides a way to obtain information about the underlying network structure and dynamics, and in this manner, has the potential to provide new insights into mechanisms of brain function that were not possible before. For example, it may be possible to infer that a group of neurons or glia are disrupted by detecting the change in the network dynamics through a single unit, even if the single unit is intact. Although this speculation would need to be confirmed using experimental data from single units where specific known defects are introduced in a circuit to see how well the neuron particles perform in identifying the defect (i.e., needle in the haystack), the implications are tantalizing. For example, it may be possible to measure a reasonable number of neurons and be able to make predictions about the early onset of neurodegenerative disease, epilepsy, brain cancer or other neurological disorders by detecting a change in the network dynamics without directly measuring the implicated tissues.

Chapter 9

Conclusion and Future Directions

In this thesis work, we were primarily concerned with the role of unknown sources towards modeling techniques. Apart from some very controlled toy settings, whether we acknowledge or not, the unknown unknowns (UUs) contribute towards the observed phenomena. We have developed several mathematical tools to take care of the unknown sources for different kind of data streams.

In Chapter 2, we presented a fractional-order dynamical system with unknown unknowns (i.e., inputs). By relaxing various assumptions, we further developed other latent variable models. The applications were demonstrated for real-valued time-series data format. We show better modeling for electroencephalogram (EEG), spatio-temporal signals, for example, blood pressure, heart rate, BOLD signals. Next, we have also shown that challenging and emerging applications like brain imagined task predictions can be solved effectively using the proposed mathematical tools. We have also extended the fractional modeling to model-based reinforcement learning (RL) paradigm. After showing performance bound for non-Markovian RL, we have shown applications for some real-world problems of blood glucose control through a diabetes simulator. A different modeling technique but following the same theme of UUs is presented in Chapter 3, where the mathematical models were developed for event-type data streams, for example, neural spiking trains.

Next, we have developed series of mathematical techniques building on the concepts of partial differential equations (PDEs). First, we have demonstrated two algorithms to estimate the arguments of fractional diffusion equations from a limited set of data streams in Chapter 4. Using the fractional moments approach, we have shown that the proposed algorithms were accurate and data-efficient. Second, we have taken a 'neural operators' approach to solve PDEs and achieve time-efficiency. After projecting the operator kernel to the multiwavelet domain and then utilizing the smoothness

property, we have achieved significantly better results as well as data efficiency for deep learning based PDE solvers in Chapter 5.

Through our work, we have elaborated that complex networks and partial differential equations go hand in hand. We presented a novel technique of neuron particles to estimate the fractal causal memory in the spiking events-type data. By extracting particle trajectories from spiking data, and modeling them as fractional PDEs, we have observed that scale and memory parameter varies uniquely with the underlying network topology.

Future directions: The current thesis has worked along the way to develop mathematical models with unknown unknowns and make some interesting findings about the causal fractality. In this pursuit, we have solved some finite number of problems and also identified numerous problems that are interesting for future research.

An important assumption of the fractional dynamical models in the Chapter 2 is constant fractional-order systems. A more general approach would be to have time-varying fractional-order dynamical model. Wavelets are useful in computing such coefficients and the work of Gu et al. (2020) could be helpful in writing time-varying ordinary differential equations (ODEs) for fractional coefficients. Some immediate future directions to pursue are applications of the proposed mathematical models on spatio-temporal signals. In this work, we are primarily concerned with the EEG signals, however, the current formulation could be successfully applied to other applications, including, physiological signals from portable devices to monitor critical health events. Preliminary work has demonstrated promising results on predicting viral infections and respiratory diseases using readily available data from wearable devices.

Modeling of the data through PDEs is a futuristic approach for machine learning (ML) methods. Through our work on fractional diffusion we have observed critical phenomena can be modeled with less data using fractional PDEs. Implicitly building physics-inspired fractional PDEs inside the ML models could be next generation of artificial intelligence (AI) models. The recent success of neural ODEs (Chen et al., 2019) is just one example long this long path. In the context of the current work in Chapter 4, the fractional PDE for diffusion process is assumed to be carrying single fractional exponents, both in space and time. Relaxing such assumption and having generalized equation with multiple scale and time fractional exponents would offer flexibility in compact modeling of challenging fast-varying, non-stationary, memory dependent, and multi-scale data streams. A step towards this direction is made by us in Yin, Gupta, and Bogdan (2020) but the generalization is again an open problem for now.

We saw an application of fractional PDEs to model the neural spiking data in Chapter 8. The multi-dimensional information from a single spike train is used to infer

the underlying network topology. However, a promising future direction is to study coupled fractional PDEs such that multiple spike trains could be modeled together using neuron particles. Doing so would address the long-standing problem of how network topology shapes the neural spike correlations? Estimating spike correlations and showing their variations network topology would unify the existing but two separated branch of results in experimental neuroscience. The network topologies knitting the neurons together in the brain are not restricted to the simple canonical topologies considered in Chapter 8. Another interesting direction to pursue is showing how a mixture of network topology influence the spiking activities. A follow-up question to address in the reverse direction would be: how to isolate the contribution of canonical network topologies on the observed outcomes? Such inference would also help in the long-run to design artificial neural networks which can mimic biological neural network functionalities.

The neural operators delegate the time-consuming part of the PDE solvers to the training phase, which then has to be done only once. However, one issue with the current line of work of neural operator, as in Chapter 5, is that they are not regulated to satisfy boundary conditions. An immediate promising direction is to enable black-box neural operators to be constrained to satisfy boundary conditions. This would be a merger of physics-aware neural operators and the proposed data efficient neural operators. Availability of the data is itself a restriction on learning neural operators. In certain setups, the values of force, velocity fields are only available at the boundaries. Developing neural operators that could learn the solution with only a partial knowledge of boundary values is an interesting extension to Chapter 5 work. The operator learning tool could be extended to model coupled PDEs and jointly learn the cross-kernel with being data-efficient. Some of the problems of inferring causality could also be formulated as PDE with causal component modeled directly inside the operator kernel. Such approaches could enable detecting causality in the non-linear systems in a model-agnostic fashion and generalize over linear models of Granger causality (Granger, 1969).

Part IV

Appendices

Appendix A

Time-varying Complex Networks

A.1 EM Formulation

We present the detailed construction of EM like algorithm in this section. In our formulation, the observed (incomplete) data is x and z while u is the hidden data, therefore the complete data would be (z, x, u). Let us consider, $\Sigma = \sigma^2 I$, and denote $\theta = \{A, B\}$. At the lth iteration we denote

$$u[k]^* = \arg\max_u \mathbb{P}\left(u\Big|z[k], x[k]; \theta^{(l)}\right).$$

We can enforce Laplacian prior for $u[k]$ for sparsity (any other prior could also be used) such that $\mathbb{P}(u[k]) \propto \exp(-\lambda||u[k]||_1)$. Therefore, $u[k]^*$ is then derived as

$$\begin{aligned} u[k]^* &= \arg\max_u \log\mathbb{P}\left(u\Big|z[k], x[k]; \theta^{(l)}\right) \\ &= \arg\max \log\mathbb{P}(u) + \log\mathbb{P}\left(z[k], x[k]\Big|u; \theta^{(l)}\right) \\ &= \arg\max -\frac{1}{2\sigma^2}||z[k] - A^{(l)}x[k] - Bu||_2^2 - \lambda||u||_1. \end{aligned}$$

We have approximated the conditional distribution as $\mathbb{P}(u[k]\Big|z[k], x[k]; \theta^{(l)}) \approx \mathbb{1}_{\left\{u[k]=\overset{*}{u}[k]\right\}}$.
In the final step of expectation, we can write

$$\begin{aligned}
Q(\theta; \theta^{(l)}) &= \mathbb{E}_{\theta^{(l)}}\left[\log\mathbb{P}_c(z[k], x[k], u[k])\Big|x[k], z[k]\right] \\
&= \mathbb{E}_{u[k]|z[k],x[k];\theta^{(l)}}\left[\log\mathbb{P}\left(z[k], x[k], u[k]; \theta^{(l)}\right)\right] \\
&= \mathbb{E}_{u[k]|z[k],x[k];\theta^{(l)}}\left[\log\mathbb{P}\Big(z[k], x[k]; \theta^{(l)}\Big)\right] + \log\mathbb{P}\left(u[k]\Big|z[k], x[k]; \theta^{(l)}\right) \\
&= \log\mathbb{P}\left(z[k], x[k]; \theta^{(l)}\right) \mathbb{1}_{\left\{u[k]=\overset{*}{u}[k]\right\}},
\end{aligned}$$

where $\mathbb{P}_c$ is used to signify the likelihood of the complete data. For the Maximization step, we can simply write

$$\begin{aligned}
\theta^{(l+1)} &= \arg\max_{\theta} Q(\theta; \theta^{(l)}) \\
&= \arg\max_{\theta} \log\mathbb{P}\left(z[k], x[k]; \theta\right) \mathbb{1}_{\left\{u[k]=\overset{*}{u}[k]\right\}},
\end{aligned}$$

or in other words,

$$[a_i^{(l+1)\,T}, b_i^{(l+1)\,T}]^T = \arg\min_{a,b} ||Z_i - Xa - Ub||_2^2,$$

A.2 Proof of Proposition 1

Proof. We show that the likelihood for incomplete (observed) data is bounded at each EM update step. Let us denote the likelihood of the observed data in relation to the parameter $A^{(l)}, B^{(l)}$ as

$$\mathbb{P}(A^{(l)}, B^{(l)}) = \mathbb{P}(z, x; A^{(l)}, B^{(l)}), \tag{A.1}$$

which is further written as

$$\begin{aligned}
\mathbb{P}(A^{(l)}, B^{(l)}) &\propto \int \mathbb{P}(z, x\Big|u; A^{(l)}, B^{(l)})\mathbb{P}(u)du \\
&= C\int \exp\left(-\frac{1}{2\sigma^2}||z - A^{(l)}x - B^{(l)}u||_2^2\right)\exp\left(-\lambda||u||_1\right)du \\
&\leq C\int \exp\left(-\lambda||u||_1\right)du \leq \mathcal{O}(1),
\end{aligned}$$

where C is the normality constant. Therefore $\mathbb{P}(A^{(l)}, B^{(l)})$ is bounded for every iteration index $l \geq 0$. □

Appendix B

Spiking Event Complex Network Models

B.1 Proof of Theorem 3.1

Proof. The Expectation step of the algorithm is concerned with computation of $P(\Delta U_k^i | N_{1:K}^{1:C}; \theta)$ which is not computationally tractable in various cases. In this work, we will approximate it as $P(\Delta U_k^i | N_{1:K}^{1:C}; \theta) = 1_{\Delta U_k^i = \Delta \widehat{U}_k^i}$ (this is sometimes referred as Hard EM (Murphy, 2012)), where

$$\Delta \widehat{U}_k^i = \arg\max_{\Delta U_k^i} \log P(\Delta U_k^i | N_{1:K}^{1:C}; \theta). \tag{B.1}$$

The equation (B.1) can be expanded using Bayesian formulation, and the choice of the prior distribution of ΔU_k^i is critical. Intuitively, ΔU_k^i represents all the hidden sources, including undetected neuron activity and environmental stimuli. Unlike binary spike count ΔN_k^c, the unknown artifacts ΔU_k^i don't necessarily have to be discrete and non-negative, and are real values without further constraint. While there can be wide selections of priors, in this work we are motivated by using conjugate priors approach (Raiffa, 1974) to have computable expressions. Therefore, we assume the log-Gamma prior for ΔU_k^i as it collaborates well with the existing CIF model. The log-Gamma distribution (Demirhan and Hamurkaroglu, 2011) can be mathematically written as

$$P(\Delta U_{1:K}^{1:I}; \theta) = \prod_{i=1}^{I} \prod_{k=1}^{K} \frac{e^{\beta \Delta U_k^i} e^{-e^{\Delta U_k^i / \alpha}}}{\alpha^\beta \Gamma(\beta)}, \Delta U_k^i \in \mathbb{R}, \tag{B.2}$$

where α is the shape parameter and β is the scale parameter. We also assume that the unknown artifacts behavior are independent and identical distributed across time and for each unknown sources. Therefore, the equation (B.1) can be expanded as

$$\begin{aligned}\Delta\widehat{U}_{1:K}^{1:I} &= \arg\max_{\Delta U_{1:K}^{1:I}} \log P(\Delta U_{1:K}^{1:I}|N_{1:K}^{1:C};\theta) \\ &= \arg\max_{\Delta U_{1:K}^{1:I}} \left\{\log P(N_{1:K}^{1:C}|\Delta U_{1:K}^{1:I};\theta) + \log P(\Delta U_{1:K}^{1:I};\theta)\right\} \\ &\overset{(a)}{=} \arg\max_{\Delta U_{1:K}^{1:I}} \sum_{c=1}^{C}\sum_{k=1}^{K}\{\Delta N_k^c \log\lambda_U^c(k) - \omega^c(k)\tau\lambda_U^c(k)\} + \sum_{i=1}^{I}\sum_{k=1}^{K}\left\{\beta\Delta U_k^i - \frac{1}{\alpha}e^{\Delta U_k^i}\right\},\end{aligned} \tag{B.3}$$

where in (a) we have used the definition of $\lambda_U^c(k)$ and $\omega^c(k)$ from equation (3.7) and (3.8), respectively. Now for the maximization, the equation (B.3) is an unconstrained optimization problem and we solve it by setting the partial derivatives with respect to $\nu_q^{i_0} = e^{\Delta U_q^{i_0}}$ to be zero for each $q = 1, 2, ..., K$, and $i_0 = 1, 2, ..., I$. Therefore, we obtain

$$\beta - \frac{\nu_q^{i_0}}{\alpha} + \sum_{c=1}^{C}\sum_{k=q+1}^{\min(q+M,K)}\Delta N_k^c\gamma_{k-q}^{i_0}(c) - \sum_{c=1}^{C}\sum_{k=q+1}^{\min(q+M,K)}\omega^c(k)\tau\gamma_{k-q}^{i_0}(c)\lambda_U^c(k|\nu) = 0. \tag{B.4}$$

The maximum likelihood (ML) estimate of $\nu_q^{i_0}$ is computed with a fixed-point iterative method. We rearrange the terms in equation (B.4) with additional exponent $t_q^{i_0}$ and build the fixed-point function as following.

$$G_q^{i_0}(\nu) = \nu_q^{i_0} \times \left[\frac{\sum_{c=1}^{C}\sum_{k=\max(q+1,1)}^{\min(q+M,K)}\Delta N_k^c\gamma_{k-q}^{i_0}(c) + \beta}{\sum_{c=1}^{C}\sum_{k=\max(q+1,1)}^{\min(q+M,K)}\gamma_{k-q}^{i_0}(c)\omega^c(k)\tau\lambda_U^c(k|\nu) + \frac{\nu_q^{i_0}}{\alpha}}\right]^{t_q^{i_0}}, \tag{B.5}$$

where the fixed-point iterations would be $\nu_q^{i_0\,(n+1)} = G_q^{i_0}(\nu^{(n)})$ at iteration n. The exponent $t_q^{i_0}$ need to be chosen carefully such that the fixed-point iterations would converge to the ML solution. We follow the procedure as mentioned in (Peters and Coberly, 1976) to prove that $G_q^{i_0}(\nu)$ is a local contraction and hence find the value of $t_q^{i_0}$ such that the fixed-point iterations are convergent.

Let us denote the ML estimate $\widehat{\nu}_q^{i_0}$ as the solution of fixed point equation $\nu_q^{i_0} = G_q^{i_0}(\nu)$. Now $G^i(\cdot)$ is a local contraction if $\|\nabla G^i(\widehat{\nu})\| < 1$, where $\nabla G^i(\widehat{\nu})$ is a matrix such that

$$\left[\nabla G^i(\widehat{\nu})\right]_{q,p} = \frac{\partial G_q^i(\nu)}{\partial\nu_p^i}|_{\nu=\widehat{\nu}}. \tag{B.6}$$

The norm $\|\nabla G^i(\widehat{\nu})\|$ is less than 1 if and only if the spectral radius $\rho(\nabla G^i(\widehat{\nu})) < 1$ (Pattern, 1973). After writing the $\nabla G^i(\widehat{\nu})$ in (B.6) using (B.5) and upon setting

$$t_q^{i_0} = l\frac{n_q^{i_0}}{d_q^{i_0}}, \tag{B.7}$$

where, $0 < l < 2$ and

$$n_q^{i_0} = \sum_{c=1}^{C} \sum_{k=q+1}^{\min(q+M,K)} \Delta N_k^c \gamma_{k-q}^{i_0}(c) + \beta, \tag{B.8}$$

$$d_q^{i_0} = \sum_{c=1}^{C} \sum_{p=\max(q-M+1,1)}^{\min(q+M-1,K)} \sum_{k=\max(p+1,q+1)}^{\min(p+M,q+M,K)} \gamma_{k-q}^{i_0}(c) \times \gamma_{k-p}^{i_0}(c)\omega^c(k)\tau\lambda_U^c(k|\widehat{\nu}), \tag{B.9}$$

we obtain

$$\nabla G^i(\widehat{\nu}) = I - B^i, \tag{B.10}$$

where the matrix B^i is non-negative and it has a positive eigenvector $\widehat{\nu}^i$ with positive eigenvalue l. Now, using the Perron-Frobenius theorem (Householder, 1964), given the condition that l is the positive eigenvalue of B^i with positive eigenvector, the absolute value of all other eigenvalues are less or equal than l. Thus, the spectral radius $\rho(\nabla G^i(\widehat{\nu})) = |1 - l| < 1$. Hence, the fixed-point equations are convergent. It should be noted that the denominator expression $d_q^{i_0}$ is depending on the ML estimate $\widehat{\nu}$ which is not available during iterations. We approximate the $d_q^{i_0}$ using the similar counting arguments of (Chornoboy, Schramm, and Karr, 1988) to write the final expression as in equation (3.5). □

Appendix C

Fractional Diffusion Equations

C.1 Preliminaries

C.1.1 Fractional Derivatives

C.1.1.1 The Fourier transform and the Riesz-Feller space-fractional derivative

Let (C.1) be the Fourier transform of a general function $f(x)$,

$$\widehat{f}(\kappa) = \mathcal{F}\{f(x)\} = \int_{-\infty}^{+\infty} \mathrm{e}^{i\kappa x} f(x) dx, \quad \kappa \in \mathbb{R} \tag{C.1}$$

and let (C.2),

$$f(x) = \mathcal{F}^{-1}\{\widehat{f}(\kappa)\} = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \mathrm{e}^{-i\kappa x} \widehat{f}(\kappa) d\kappa, \quad x \in \mathbb{R} \tag{C.2}$$

be the inverse Fourier transform. For a sufficiently well-behaved function $f(x)$ we define the Riesz-Feller space-fractional derivative of order α and skewness θ as

$$\begin{cases} \mathcal{F}\{{}_x\mathcal{D}_\theta^\alpha f(x);\kappa\} = \psi_\alpha^\theta(\kappa)\widehat{f}(\kappa), \quad \psi_\alpha^\theta(\kappa) = -|\kappa|^\alpha \mathrm{e}^{i(\mathrm{sign}(\kappa))\theta\pi/2} \\ 0 < \alpha \leq 2, \quad |\theta| \leq \min\{\alpha, 2-\alpha\} \end{cases} \tag{C.3}$$

$$\begin{aligned} {}_x\mathcal{D}_\theta^\alpha f(x) = \frac{\Gamma(1+\alpha)}{\pi} \{ & \sin[(\alpha+\theta)\pi/2] \int_0^\infty \frac{f(x+\xi)-f(x)}{\xi^{1+\alpha}} d\xi \\ & + \sin[(\alpha-\theta)\pi/2] \int_0^\infty \frac{f(x-\xi)-f(x)}{\xi^{1+\alpha}} d\xi \}. \end{aligned} \tag{C.4}$$

The symbol $\psi_\alpha^\theta(\kappa)$ is the logarithm of the characteristic function of a general Levy strictly stable probability density with index of stability α and asymmetry parameter θ (improperly called skewness) according to Feller's parameterization.

C.1.1.2 The Laplace transform and the Caputo fractional derivative

Let

$$\widetilde{f}(s) = \mathcal{L}\{f(t)\} = \int_0^\infty \mathrm{e}^{-st} f(t)dt, \quad \Re(s) > a_f, \tag{C.5}$$

be the Laplace transform of a function $f(t)$, and let

$$f(t) = \mathcal{L}^{-1}\{\tilde{f}(s)\} = \frac{1}{2\pi i}\int_{\gamma-i\infty}^{\gamma+i\infty} \mathrm{e}^{st}\tilde{f}(s)ds, \quad \Re(s) = \gamma > a_f \tag{C.6}$$

where $t > 0$ and a_f is a constant defined such that the product $e^{-a_f t}|f(t)|$ is bounded for all t greater than some T (i.e., the constant a_f exists provided the existence of the Laplace transform). For a sufficiently well-behaved function $f(t)$ we define the Caputo time-fractional derivative of order $\beta, (0 < \beta \leq 1)$ through

$$\mathcal{L}\{{}_t\mathcal{D}_*^\beta f(t)\} = s^\beta \widetilde{f}(s) - s^{\beta-1}f(0^+), \quad 0 < \beta \leq 1 \tag{C.7}$$

Hence, we can write

$${}_t\mathcal{D}_*^\beta f(t) = \begin{cases} \frac{1}{\Gamma(1-\beta)}\int_0^t \frac{f^{(1)}(\tau)}{(t-\tau)^\beta}d\tau, & 0 < \beta < 1 \\ \frac{d}{dt}f(t), & \beta = 1 \end{cases}. \tag{C.8}$$

C.1.2 Stable distribution

A non-degenerate random variable X is called stable if for all $n > 1$, there exist constants $c_n > 0$ and $d_n \in \mathbb{R}$ such that $X_1 + \cdots + X_n \stackrel{d}{=} c_n X + d_n$[1], where $X_1, X_2, \cdots, X_n$ are i.i.d realizations of X. The random variable X is strictly stable if and only if $d_n = 0, \forall n$. It can be shown that the only possible choice for the scaling constants is $c_n = n^{1/\alpha}$ for a certain value $\alpha \in (0, 2]$.

[1]The symbol $\stackrel{d}{=}$ designates the equality in distribution.

C.1.2.1 Parameterizations of stable laws:

There are different parameterizations for stable distribution. The variety of parameterizations is caused by a combination of historical evolution, plus the numerous problems that have been analyzed applying specialized forms of stable distributions.

1. A random variable X is stable if and only if $X \stackrel{d}{=} aZ + b$ where $a \neq 0, b \in \mathbb{R}$ and Z is a random variable with characteristic function ((Nolan, 2003))

$$\mathbb{E}[\exp(i\kappa Z)] = \begin{cases} \exp(-|\kappa|^{\alpha}[1 - i\zeta \tan\frac{\pi\alpha}{2}(\operatorname{sign}\kappa)]) & \alpha \neq 1 \\ \exp(-|\kappa|[1 + i\zeta\frac{2}{\pi}(\operatorname{sign}\kappa)\log|\kappa|]) & \alpha = 1 \end{cases}, \tag{C.9}$$

   where $0 < \alpha \leq 2$, $-1 \leq \zeta \leq 1$ [2].

2. A random variable X is parameterized as $S(\alpha, \zeta, c, \mu; 1)$ if

$$\mathbb{E}(\exp(i\kappa X)) = \exp(i\kappa\mu - |c\kappa|^{\alpha}(1 - i\zeta\operatorname{sign}(\kappa)\Phi_1)), \tag{C.10}$$

   where,

$$\Phi_1 = \begin{cases} \tan(\frac{\pi\alpha}{2}) & \alpha \neq 1 \\ -\frac{2}{\pi}\log|\kappa| & \alpha = 1 \end{cases}. \tag{C.11}$$

   The distribution is assumed to be standard when the scale $c = 1$ and the location $\mu = 0$ ((Nolan, 2003)).

3. A random variable X is $S(\alpha, \zeta, \gamma, \delta; 0)$ ((Nolan, 2003)) if

$$X \stackrel{d}{=} \begin{cases} \gamma\left(Z - \beta\tan\frac{\pi\alpha}{2}\right) + \delta & \alpha \neq 1 \\ \gamma Z + \delta & \alpha = 1 \end{cases}, \tag{C.12}$$

   where the Z is defined at (C.9). This can also be rewritten as:

$$\mathbb{E}(\exp(i\kappa X)) = \exp(i\kappa\delta - |\gamma\kappa|^{\alpha}(1 - i\zeta\operatorname{sign}(\kappa)\Phi_0)), \tag{C.13}$$

   where

$$\Phi_0 = \begin{cases} (1 - |\gamma\kappa|^{1-\alpha})\tan(\frac{\pi\alpha}{2}) & \alpha \neq 1 \\ -\frac{2}{\pi}\log|\gamma\kappa| & \alpha = 1 \end{cases}. \tag{C.14}$$

   This form $(S(\alpha, \zeta, \gamma, \delta; 0))$ is continuous at $\alpha = 0$. Note that this form is the one used in MATLAB.

4. Feller's parameterization: ((Sato et al., 1999; Feller, 1962; Gorenflo and Mainardi, 1999; Takayasu, 1990; Mainardi, Luchko, and Pagnini, 2007)) A random variable X is stable if and only if $X \stackrel{d}{=} aY + b$ where $0 < \alpha \leq 2$, $\theta \leq \min(\alpha, 2-\alpha)$, $a \neq$

[2] The parameter ζ is usually called β but β is used to describe another parameter in this work.

$0, b \in \mathbb{R}$ and Y is a random variable with characteristic function

$$\mathbb{E}(\exp(i\kappa Y)) = \exp(i\psi_\alpha^\theta(\kappa)), \tag{C.15}$$

where $\psi_\alpha^\theta(\kappa)$ is given by (C.3). It is also worth to mention that for $b = 0$ the characteristic function of X (which is strictly stable) is given as

$$\mathbb{E}(\exp(i\kappa X)) = \exp\left(i\psi_\alpha^\theta\left(\frac{\kappa}{a}\right)\right) = \exp\left(i|a|^\alpha \psi_\alpha^\theta(\kappa)\right). \tag{C.16}$$

For the sake of performing simulation using existing methods on MATLAB, we have to express the Feller's parameterization in the $S(\alpha, \zeta, \gamma, \delta; 0)$ form. First, we are interested in the strictly stable case ($\delta = \zeta\gamma\tan(\frac{\pi\alpha}{2})$) so we have

$$\exp(i|a|^\alpha \psi_\alpha^\theta(\kappa)) = \exp\left(i\kappa\zeta\gamma\tan\left(\frac{\pi\alpha}{2}\right) - |\gamma\kappa|^\alpha(1 - i\zeta\,\mathrm{sign}(\kappa)\Phi)\right). \tag{C.17}$$

The above equation should be correct for any $\kappa \in \mathbb{R}$. Solving them (considering separate equation for imaginary and real parts) gives

$$\begin{aligned}
\gamma &= a\left(\cos\left(\frac{\pi\theta}{2}\right)\right)^{1/\alpha} \\
\zeta &= -\tan\left(\frac{\pi\theta}{2}\right)\cot\left(\frac{\pi\alpha}{2}\right) \\
\delta &= \zeta\gamma\tan\left(\frac{\pi\alpha}{2}\right) = -a\tan\left(\frac{\pi\theta}{2}\right)\left(\cos\left(\frac{\pi\theta}{2}\right)\right)^{1/\alpha}.
\end{aligned} \tag{C.18}$$

C.1.2.2 Fractional order absolute moment

Suppose the characteristic function of random variable X is denoted as $\varphi_X(\kappa) = \mathbb{E}[\exp(i\kappa X)]$. Applying the method described in ((Harvill, 2009)) and using its general result (C.20), the fractional order absolute moment of stable distributions are computed.
Define an auxiliary function $\rho(.)$:

$$\rho(\delta) = \int_0^\infty u^{-(\delta+1)}\sin^2(u)\mathrm{d}u = \begin{cases} \delta^{-1}2^{\delta-1}\Gamma(1-\delta)\cos(\pi\delta/2), & \text{if } 0 < \delta < 2, \delta \neq 1 \\ \pi/2, & \text{if } \delta = 1 \end{cases}. \tag{C.19}$$

The general result of ((Harvill, 2009)) is:

$$\rho(\delta)\mathbb{E}[|X|^\delta] = -\frac{1}{4}\int_0^\infty \kappa^{-(\delta+1)}[\varphi_X(2\kappa) + \varphi_X(-2\kappa) - 2]\mathrm{d}\kappa. \tag{C.20}$$

When X is a strictly stable random variable with decomposing the characteristic function as

$$\varphi_X(\kappa) = \exp\{-z_p\kappa^\alpha\} \quad \text{and} \quad \varphi_X(-\kappa) = \exp\{-z_n\kappa^\alpha\}, \quad \kappa \geq 0 \tag{C.21}$$

where,

$$z_p = \exp(i\theta\pi/2), \quad z_n = \exp(-i\theta\pi/2). \tag{C.22}$$

Then

$$\rho(\delta)\mathbb{E}[|X|^\delta] = \frac{1}{\delta}2^{\delta-2}\Gamma\left(1-\frac{\delta}{\alpha}\right)\left(z_p^{\delta/\alpha}+z_n^{\delta/\alpha}\right). \tag{C.23}$$

Therefore, the absolute moment of order δ is given as

$$\mathbb{E}[|X|^\delta] = \frac{\Gamma\left(1-\frac{\delta}{\alpha}\right)\cos\left(\frac{\delta\pi\theta}{2\alpha}\right)}{\Gamma(1-\delta)\cos\left(\frac{\delta\pi}{2}\right)}. \tag{C.24}$$

C.1.2.3 Signed fractional order moment

Using the method provided in (Kuruoglu, 2001), the signed absolute moment or order δ for α-stable distribution is written as follows. For $\delta \in (-2,-1)\cup(-1,0)$ we have

$$\begin{aligned}
\mathbb{E}[X^{\langle\delta\rangle}] &= \frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\text{sign}(x)|x|^\delta\varphi_X(\kappa)e^{i\kappa x}dxd\kappa \\
&= \frac{i}{\pi}\int_0^\infty\int_0^\infty x^\delta\sin(\kappa x)dx\left(\varphi_X(\kappa)-\varphi_X^*(-\kappa)\right)dt \qquad \text{(C.25)} \\
&= \frac{i}{\pi}\Gamma(1+\delta)\cos\left(\frac{\delta\pi}{2}\right)\int_0^\infty\kappa^{-1-\delta}\left(e^{-\kappa^\alpha z_p}-e^{-\kappa^\alpha z_n}\right)dt \\
&= \frac{i}{\pi}\Gamma(1+\delta)\cos\left(\frac{\delta\pi}{2}\right)\left[\frac{1}{\alpha}\Gamma\left(-\frac{\delta}{\alpha}\right)\left(z_p^{\frac{\delta}{\alpha}}-z_n^{\frac{\delta}{\alpha}}\right)\right] \\
&= -\frac{\Gamma\left(1-\frac{\delta}{\alpha}\right)\sin\left(\frac{\delta\pi\theta}{2\alpha}\right)}{\Gamma(1-\delta)\sin\left(\frac{\delta\pi}{2}\right)}. \qquad \text{(C.26)}
\end{aligned}$$

As discussed in (Kuruoglu, 2001), the same result can be generalized to $\delta \in [0,\alpha]$.

C.1.3 Lévy stable stochastic processes

A one-dimensional stochastic process $\{X(t); t \geq 0\}$ said to be a Lévy process if it satisfies the following properties:

1. $X(0) \overset{a.s.}{=} 0$.
2. Disjoint increments are mutually independent. It means that for any $0 \leq t_1 < t_2 < \cdots < t_n < \infty$ the increments $(X(t_2) - X(t_1), X(t_3) - X(t_2), \ldots, X(t_n) - X(t_{n-1}))$ are mutually independent.
3. Stationary increments: for any $s < t, X(t) - X(s)$ is equal in distribution to $X(t - s)$.
4. The sample paths are Cádlág ((Billingsley, 2008)), meaning they are almost surely right-continuous and have left limits at all time points.

A process $X(t)$ is said to be a *strictly α − stable process* if it is a Lévy process which also satisfies the scaling (self-similarity) property (i.e., the process $(c\, X(t\, c^{-\alpha}); t \geq 0)$ has the same distribution as $X(t)$ for every $c > 0$ denoted as $X(t) \overset{d}{=} t^{1/\alpha} X(1)$ ((Kyprianou, 2006)).

One of the main property of a Lévy process is that its characteristic function has the following form

$$\mathbb{E}[\exp(i\, \kappa\, X(t))] = \exp(t\, \Psi(\kappa)). \tag{C.27}$$

This means that the stationary independent increments of process $X(t)$ are i.i.d samples of a stable distribution. In case of a one dimensional *strictly α − stable process* with asymmetric parameter θ, the $\Psi(\kappa)$ is equal to $\psi_\theta^\alpha(\kappa)$.

C.1.3.1 Space fractional diffusion

One group of random processes that have a space fractional diffusion are the strictly α-stable processes. Suppose the random process $X(t_*)$ is a strictly α-stable process for some $0 < \alpha \leq 2$ and $0 \leq |\theta| \leq \min(\alpha, 2-\alpha)$. Then, according to (C.27), the diffusion that is defined as $f_{\alpha,\theta}(x, t_*) = \mathcal{P}\{X(t_*) = x | X(0) = 0\}$, has a Fourier transform equal to

$$\widehat{f}_{\alpha,\theta}(\kappa, t_*) = \exp(-t_*\, \psi_\alpha^\theta(\kappa)), \tag{C.28}$$

where $\psi_\alpha^\theta(\kappa)$ is the same function defined at (C.3). Taking the Laplace transform on t_*, the fractional order PDE of the diffusion is achieved

$$\widehat{\widetilde{f}}_{\alpha,\theta}(\kappa, s_*) = \frac{1}{s_* + \psi_\alpha^\theta(\kappa)}, \tag{C.29}$$

and then

$$-\psi_{\alpha}^{\theta}(\kappa)\,\widehat{\widetilde{f}}_{\alpha,\theta}(\kappa,s_*)=s_*\,\widehat{\widetilde{f}}_{\alpha,\theta}(\kappa,s_*)-1. \tag{C.30}$$

So

$$\begin{aligned}\frac{\partial}{\partial t_*}f_{\alpha,\theta}(x,t_*)&={}_x\mathcal{D}_{\theta}^{\alpha}\{f_{\alpha,\theta}(x,t_*)\},\quad t_*\geq 0\\ f_{\alpha,\theta}(x,0)&=\delta(x),\quad x\in\mathbb{R}.\end{aligned} \tag{C.31}$$

C.1.3.2 Stable subordinator process

A Subordinator process is defined as a Lévy process with non-decreasing sample paths. Suppose $T(t_*)$ is strictly β-stable[3] process for some $0<\beta\leq 1$, and $\theta=-\beta$. It can be shown that the condition $\theta=-\beta$ (which is only feasible when $0<\beta\leq 1$) implies that the increments of the process $T(t_*)$ are almost surely non-negative, thus here we use the Laplace transform. The diffusion defined as $r_{\beta}(t,t_*)=\mathcal{P}\{T(t_*)=t|T(0)=0\}$, has a Laplace transform equal to

$$\widetilde{r}_{\beta}(s,t_*)=\exp(-t_*s^{\beta}). \tag{C.32}$$

C.1.3.3 Inverse subordinator

Because $T(t_*)$ is a monotonically increasing function, the inverse process $(T_*(t))$ is a well defined function, which could be interpreted as the first hitting time.

$$T_*(t)=\inf\{\tau\quad|T(\tau)\geq t\}, \tag{C.33}$$

which can be used to write the following properties.

$$\begin{aligned}t_2>t_1\Longrightarrow T_*(t_2)&\geq T_*(t_1),\\ \mathcal{P}(T_*(t)\leq t_*)&=\mathcal{P}(T(t_*)\geq t).\end{aligned} \tag{C.34}$$

Define $q_{\beta}(t_*,t)=\mathcal{P}\{T_*(t)=t_*|T_*(0)=0\}$. Using the property in (C.34):

$$\int_0^{t_*}q_{\beta}(t'_*,t)dt'_*=\int_t^{\infty}r_{\beta}(t',t_*)dt'. \tag{C.35}$$

So

$$q_{\beta}(t_*,t)=\frac{\partial}{\partial t_*}\int_t^{\infty}r_{\beta}(t',t_*)dt'=\int_t^{\infty}\frac{\partial}{\partial t_*}r_{\beta}(t',t_*)dt'. \tag{C.36}$$

[3] An α-stable process where $\alpha=\beta$

Then

$$\widetilde{q}_\beta(t_*, s) = \frac{-1}{s}\frac{\partial}{\partial t_*}\widetilde{r}_\beta(s, t_*) = s^{\beta-1}\exp(-t_* s^\beta). \tag{C.37}$$

C.1.4 Space-Time fractional diffusion

Suppose $X(t_*)$ is a strictly α-stable process and $T(t_*)$ is a subordinator process. The process $X(t) = X(T_*(t))$ is called a subordinated (Leonenko et al., 2014) process if $T_*(t)$ be the inverse process of the subordinator process $(T(t_*))$ [4]. Define a diffusion function $u(x,t) = \mathcal{P}\{X(t) = x\}$, hence a direct result of the definition is

$$u(x,t) = \int_0^\infty f_{\alpha,\theta}(x, t_*) q_\beta(t_*, t) dt_*, \tag{C.38}$$

where $f_{\alpha,\theta}(x, t_*)$ and $q_\beta(t_*, t)$ defined in previous section. Using the Laplace and Fourier transform, the aforementioned expression can be simplified

$$\begin{aligned}
\widehat{\widetilde{u}}(\kappa, s) &= \int_0^\infty \widehat{f}_{\alpha,\theta}(\kappa, t_*)\widetilde{q}_\beta(t_*, s) dt_* \\
&= \int_0^\infty [\exp(-\psi_\alpha^\theta(\kappa))][s^{\beta-1}\exp(-t_* s^\beta)] dt_* \\
&= \frac{s^{\beta-1}}{s^\beta + \psi_\alpha^\theta(\kappa)}.
\end{aligned} \tag{C.39}$$

Then,

$$s^\beta \widehat{\widetilde{u}}(\kappa, s) - s^{\beta-1} = -\psi_\alpha^\theta(\kappa)\widehat{\widetilde{u}}(\kappa, s). \tag{C.40}$$

Thus,

$${}_t\mathcal{D}_*^\beta u(x,t) = {}_x\mathcal{D}_\theta^\alpha u(x,t), \quad u(x,0) = \delta(x), \quad x \in \mathbb{R}, \quad t \geq 0, \tag{C.41}$$

where $0 < \alpha \leq 2, |\theta| \leq \min\{\alpha, 2-\alpha\}$ and $0 < \beta \leq 1$. One important result of (C.39) is the scaling property of the diffusion function which can be reflected by a single variable function $K_{\alpha,\beta}^\theta(x)$

$$u(x,t) = t^{-\gamma}u(x/t^\gamma, 1) = t^{-\gamma}K_{\alpha,\beta}^\theta(x/t^\gamma), \quad \gamma = \beta/\alpha. \tag{C.42}$$

The following properties of $K_{\alpha,\beta}^\theta(x)$ will be used later ((Mainardi, Luchko, and Pagnini, 2007))

$$K_{\alpha,\beta}^\theta(-x) = K_{\alpha,\beta}^{-\theta}(x) \tag{C.43}$$

[4] t_* is named operational time while t is called physical/regular time

$$\begin{cases} \int_0^{+\infty} K_{\alpha,\beta}^{\theta}(x)x^{\delta}\mathrm{d}x = \rho\frac{\Gamma(1-\delta/\alpha)\Gamma(1+\delta/\alpha)\Gamma(1+\delta)}{\Gamma(1-\rho\delta)\Gamma(1+\rho\delta)\Gamma(1+\beta\delta/\alpha)} \\ \\ \quad -\min\{\alpha,1\} < \Re(\delta) < \alpha, \quad \rho = \frac{\alpha-\theta}{2\alpha}. \end{cases} \tag{C.44}$$

C.2 Proof of Proposition 2

Proof. Using equation (C.38), we write that

$$\begin{aligned} \mathbb{E}[|X(t)|^{\delta}] &= \int_{-\infty}^{\infty} |x|^{\delta} u(x,t)dx \\ &= \int_{-\infty}^{\infty}\int_{0}^{\infty} |x|^{\delta} f_{\alpha,\theta}(x,t_*)q_{\beta}(t_*,t)dt_*dx. \end{aligned} \tag{C.45}$$

Now, using the discussion in Section C.1.2.2, and using $\varphi_{X(t_*)}(\kappa) = \exp\left(t_*\psi_{\alpha}^{\theta}(\kappa)\right)$ from (C.27) for $D = 1$, and $\varphi_{X(t_*)}(\kappa) = \exp\left(t_*D\psi_{\alpha}^{\theta}(\kappa)\right)$ for $D \neq 1$. After substituting in (C.21), we write that

$$\int_{\infty}^{\infty} |x|^{\delta} f_{\alpha,\theta}(x,t_*)dx = t_*^{\frac{\delta}{\alpha}} D^{\frac{\delta}{\alpha}} \frac{\Gamma\left(1-\frac{\delta}{\alpha}\right)\cos\left(\frac{\delta\pi\theta}{2\alpha}\right)}{\Gamma(1-\delta)\cos\left(\frac{\delta\pi}{2}\right)}. \tag{C.46}$$

Using (C.46) and (C.37), we continue with the Laplace transform of the time-varying moment expression in (C.45 as

$$\begin{aligned} \mathbb{E}[\widetilde{|X(t)|^{\delta}}] &= \int_{-\infty}^{\infty}\int_{0}^{\infty} |x|^{\delta} f_{\alpha,\theta}(x,t_*)q_{\beta}(t_*,s)dt_*dx \\ &= \int_{0}^{\infty} t_*^{\frac{\delta}{\alpha}} D^{\frac{\delta}{\alpha}} \frac{\Gamma(1-\frac{\delta}{\alpha})\cos\left(\frac{\delta\pi\theta}{2\alpha}\right)}{\Gamma(1-\delta)\cos\left(\frac{\delta\pi}{2}\right)} s^{\beta-1}\exp\left(-t_*s^{\beta}\right)dt_* \\ &= s^{-(\delta\frac{\beta}{\alpha}+1)} D^{\frac{\delta}{\alpha}} \frac{\Gamma\left(1-\frac{\delta}{\alpha}\right)\Gamma\left(1+\frac{\delta}{\alpha}\right)\cos\left(\frac{\delta\pi\theta}{2\alpha}\right)}{\Gamma(1-\delta)\cos\left(\frac{\delta\pi}{2}\right)}. \end{aligned} \tag{C.47}$$

Using the inverse Laplace relation $\mathcal{L}^{-1}\{s^{-\left(\delta\frac{\beta}{\alpha}+1\right)}\} = \frac{t^{\delta\frac{\beta}{\alpha}}}{\Gamma\left(1+\delta\frac{\beta}{\alpha}\right)}$, and taking inverse Laplace transform on both sides in (C.47), we finally write the time-varying moment

with order δ as

$$\mathbb{E}[|X(t)|^{\delta}] = t^{\delta\frac{\beta}{\alpha}} D^{\frac{\delta}{\alpha}} \frac{\Gamma\left(1-\frac{\delta}{\alpha}\right)\Gamma\left(1+\frac{\delta}{\alpha}\right)\cos\left(\frac{\delta\pi\theta}{2\alpha}\right)}{\Gamma(1-\delta)\Gamma\left(1+\delta\frac{\beta}{\alpha}\right)\cos\left(\frac{\delta\pi}{2}\right)}. \tag{C.48}$$

□

C.3 Proof of Proposition 3

Proof. Using equation (C.38), we write that

$$\begin{aligned}\mathbb{E}[X(t)^{\langle\delta\rangle}] &= \int_{-\infty}^{\infty} x^{\langle\delta\rangle} u(x,t)dx \\ &= \int_{-\infty}^{\infty}\int_{0}^{\infty} \text{sign}(x)|x|^{\delta} f_{\alpha,\theta}(x,t_*)q_{\beta}(t_*,t)dt_*dx. \end{aligned}\tag{C.49}$$

Now, using the discussion in Section C.1.2.3, and using $\varphi_{X(t_*)}(\kappa) = \exp\left(t_*\psi_{\alpha}^{\theta}(\kappa)\right)$ from (C.27) for $D = 1$, and $\varphi_{X(t_*)}(\kappa) = \exp\left(t_* D\psi_{\alpha}^{\theta}(\kappa)\right)$ for $D \neq 1$. After substituting in (C.25), we write that

$$\int_{\infty}^{\infty} \text{sign}(x)|x|^{\delta} f_{\alpha,\theta}(x,t_*)dx = -t_*^{\frac{\delta}{\alpha}} D^{\frac{\delta}{\alpha}} \frac{\Gamma\left(1-\frac{\delta}{\alpha}\right)\sin\left(\frac{\delta\pi\theta}{2\alpha}\right)}{\Gamma(1-\delta)\sin\left(\frac{\delta\pi}{2}\right)}. \tag{C.50}$$

Using (C.50) and (C.37), we continue with the Laplace transform of the time-varying signed moment expression in (C.49 as

$$\begin{aligned}\mathbb{E}[\widetilde{X(t)^{\langle\delta\rangle}}] &= \int_{-\infty}^{\infty}\int_{0}^{\infty} x^{\langle\delta\rangle} f_{\alpha,\theta}(x,t_*)q_{\beta}(t_*,s)dt_*dx \\ &= -\int_{0}^{\infty} t_*^{\frac{\delta}{\alpha}} D^{\frac{\delta}{\alpha}} \frac{\Gamma\left(1-\frac{\delta}{\alpha}\right)\sin\left(\frac{\delta\pi\theta}{2\alpha}\right)}{\Gamma(1-\delta)\sin\left(\frac{\delta\pi}{2}\right)} s^{\beta-1}\exp(-t_*s^{\beta})dt_* \\ &= -s^{-\left(\delta\frac{\beta}{\alpha}+1\right)} D^{\frac{\delta}{\alpha}} \frac{\Gamma\left(1-\frac{\delta}{\alpha}\right)\Gamma\left(1+\frac{\delta}{\alpha}\right)\sin\left(\frac{\delta\pi\theta}{2\alpha}\right)}{\Gamma(1-\delta)\sin\left(\frac{\delta\pi}{2}\right)}. \end{aligned}\tag{C.51}$$

Using the inverse Laplace relation $\mathcal{L}^{-1}\{s^{-(\delta\frac{\beta}{\alpha}+1)}\} = \frac{t^{\delta\frac{\beta}{\alpha}}}{\Gamma(1+\delta\frac{\beta}{\alpha})}$, and taking inverse Laplace transform on both sides in (C.51), we finally write the time-varying signed

moment with order δ as

$$\mathbb{E}[X(t)^{\langle\delta\rangle}] = -t^{\delta\frac{\beta}{\alpha}} D^{\frac{\delta}{\alpha}} \frac{\Gamma(1-\frac{\delta}{\alpha})\Gamma(1+\frac{\delta}{\alpha})\sin(\frac{\delta\pi\theta}{2\alpha})}{\Gamma(1-\delta)\Gamma(1+\delta\frac{\beta}{\alpha})\sin(\frac{\delta\pi}{2})}. \tag{C.52}$$

□

C.4 Proof of Proposition 4

Proof. Using equation (C.38), we write that

$$\begin{aligned}\mathbb{E}[\log|X(t)|] &= \int_{-\infty}^{\infty} \log|x| u(x,t)dx \\ &= \int_{-\infty}^{\infty}\int_{0}^{\infty} \log|x| f_{\alpha,\theta}(x,t_*) q_\beta(t_*,t) dt_* dx. \end{aligned} \tag{C.53}$$

For writing the log moments, we observe that $\mathbb{E}[\exp(t\log|X|)] = \mathbb{E}[|X|^t]$, i.e., the moment generating function of $\log|X|$ is the absolute moment of order t. Since we know the expression for absolute moments of alpha stable distributions from Section C.1.2.2, we use the following to write the log moments

$$\mathbb{E}[(\log|X|)^n] = \lim_{\delta\to 0} \frac{d^n}{d\delta^n}\mathbb{E}[|X|^\delta].$$

Using the similar procedure as mentioned in (Kuruoglu, 2001), we can write that

$$\mathbb{E}[|X|^\delta] = c(\delta)K^\delta \Gamma\left(1-\frac{\delta}{\alpha}\right)\cos(\delta R), \tag{C.54}$$

where,

$$c(\delta) = \Gamma(1+\delta)\text{sinc}\left(\frac{\pi\delta}{2}\right), \qquad K = (t_* D)^{\frac{\delta}{\alpha}}, \qquad R = \frac{\pi\theta}{2\alpha}. \tag{C.55}$$

We then write that

$$\frac{d}{d\delta}\mathbb{E}[|X(t_*)|^\delta] = h(\delta)\mathbb{E}[|X(t_*)|^\delta], \tag{C.56}$$

where,

$$h(\delta) = \frac{1}{\alpha}\log(t_* D) + \frac{c'(\delta)}{c(\delta)} - \frac{1}{\alpha}\psi^{(0)}\left(1-\frac{\delta}{\alpha}\right) - R\tan(\delta R),$$

and we have used the polygamma function as

$$\psi^{(n-1)}(x) = \frac{d^n}{dx^n} \log(\Gamma(x))$$

Therefore, the expected log moment is written as

$$\mathbb{E}[\log|X(t_*)|] = h(0) = \frac{1}{\alpha}\log(t_* D) + \psi^{(0)}(1)\left(1 - \frac{1}{\alpha}\right). \tag{C.57}$$

Using (C.57) and (C.37), we continue with the Laplace transform of the time-varying log moment expression in (C.53) as

$$\begin{aligned}
\mathbb{E}[\widetilde{\log|X(t)|}] &= \int_{-\infty}^{\infty}\int_{0}^{\infty} \log|x| f_{\alpha,\theta}(x, t_*) q_\beta(t_*, s) dt_* dx \\
&= \int_{0}^{\infty} \left(\frac{1}{\alpha}\log(t_* D) + \psi^{(0)}(1)\left(1 - \frac{1}{\alpha}\right)\right) s^{\beta-1} \exp\left(-t_* s^\beta\right) dt_*
\end{aligned} \tag{C.58}$$

Now, using the Euler-Mascheroni constant and the following integral expression

$$\int_0^\infty \log(x) e^{-x} dx = -\gamma,$$

we can write the Laplace of the log moment as

$$\mathbb{E}[\widetilde{\log|X(t)|}] = \left(\frac{\log(D)}{\alpha} + \psi^{(0)}(1)(1 - \frac{1}{\alpha}) - \frac{\gamma}{\alpha}\right)\frac{1}{s} - \frac{\beta}{\alpha s}\log(s). \tag{C.59}$$

Inverting the Laplace transform in (C.59) using the identity $\mathcal{L}^{-1}\left\{\frac{\log(s)}{s}\right\} = -\gamma - \log(t)$, and using the fact that $\psi^{(0)}(1) = -\gamma$, we have

$$\mathbb{E}[\log|X(t)|] = \frac{\beta}{\alpha}\log(t) + \frac{\log(D)}{\alpha} + \gamma\left(\frac{\beta}{\alpha} - 1\right). \tag{C.60}$$

□

C.5 Proof of Proposition 5

Proof. Using the similar approach as in proof of the Proposition 3 and as derived in (Kuruoglu, 2001), we can write that

$$\text{var}(\log|X(t_*)|) = \psi^{(1)}(1)\left(\frac{1}{2}+\frac{1}{\alpha}^2\right) - \left(\frac{\pi\theta}{2\alpha}\right)^2. \tag{C.61}$$

Using the expression in (C.61), we can write the Laplace of the variance of log absolute values as

$$\begin{aligned}
\text{var}(\widetilde{\log|X(t)|}) &= \int_{-\infty}^{\infty} \text{var}(\log|X(t)|) q_\beta(t_*, s) dt_* \\
&= \int_{-\infty}^{\infty} \left(\psi^{(1)}(1)\left(\frac{1}{2}+\frac{1}{\alpha}^2\right) - \left(\frac{\pi\theta}{2\alpha}\right)^2\right) s^{\beta-1}\exp\left(-t_* s^\beta\right) dt_* \\
&\overset{(a)}{=} \frac{\pi^2}{6}\left(\frac{1}{2}+\frac{1}{\alpha}^2\right) - \left(\frac{\pi\theta}{2\alpha}\right)^2,
\end{aligned} \tag{C.62}$$

where in (a) we substitute the value of polygamma function $\psi^{(1)}(1) = \pi^2/6$. □

C.6 Proof of Proposition 6

Proof. Using equation (C.38), we write that

$$\begin{aligned}
\mathbb{E}[(\log|X(t)|)^2] &= \int_{-\infty}^{\infty} (\log|x|)^2 u(x,t) dx \\
&= \int_{-\infty}^{\infty}\int_{0}^{\infty} (\log|x|)^2 f_{\alpha,\theta}(x, t_*) q_\beta(t_*, t) dt_* dx.
\end{aligned} \tag{C.63}$$

Using the similar approach as in the proof of the Proposition 3, we can write the second moment of the log absolute values of $X(t_*)$, using (C.57) and (C.61), as the following

$$\begin{aligned}
\mathbb{E}[(\log|X(t_*)|)^2] &= (\mathbb{E}[\log|X(t_*)|])^2 + \text{var}(\log|X(t_*)|) \\
&= \frac{1}{\alpha^2}(\log(t_*))^2 + \frac{2c}{\alpha}\log(t_*) + k,
\end{aligned} \tag{C.64}$$

where, $c = \frac{\log(D)}{\alpha} + \gamma(\frac{\beta}{\alpha} - 1)$ and $k = c^2 + \frac{\pi^2}{6}(\frac{1}{2} + \frac{1}{\alpha}^2) - (\frac{\pi\theta}{2\alpha})^2$. Using the expression in (C.64), we can write the Laplace of the second moment of the log absolute values as the following

$$\begin{aligned}\mathbb{E}[(\widetilde{\log|X(t)|})^2] &= \int_{-\infty}^{\infty}\int_{0}^{\infty}(\log|x|)^2 f_{\alpha,\theta}(x,t_*)q_\beta(t_*,s)dt_*dx \\ &= \int_{0}^{\infty}\left(\frac{1}{\alpha^2}(\log(t_*))^2 + \frac{2c}{\alpha}\log(t_*) + k\right)s^{\beta-1}\exp\left(-t_*s^\beta\right)dt_* \quad \text{(C.65)}\end{aligned}$$

Now, using the Euler-Mascheroni constant and the following integral expressions

$$\int_0^\infty \log(x)e^{-x}dx = -\gamma, \qquad \int_0^\infty \log^2(x)e^{-x}dx = \gamma^2 + \frac{\pi^2}{6},$$

we can write the Laplace of the second log moment as

$$\begin{aligned}\mathbb{E}[(\widetilde{\log|X(t)|})^2] &= \left(k - \frac{2c}{\gamma} + \frac{1}{\alpha^2}\left(\gamma^2 + \frac{\pi^2}{6}\right)\right)\frac{1}{s} + \left(\frac{2\beta\gamma}{\alpha^2} - \frac{2\beta c}{\alpha}\right)\frac{\log(s)}{s} \\ &\quad + \frac{\beta^2}{\alpha^2}\frac{\log^2(s)}{s}. \quad \text{(C.66)}\end{aligned}$$

Inverting the Laplace transform in (C.66) using the identity $\mathcal{L}^{-1}\left\{\frac{\log(s)}{s}\right\} = -\gamma - \log(t)$, $\mathcal{L}\left\{\log^2(t)\right\} = \frac{1}{s}\left(\gamma^2 + \frac{\pi^2}{6}\right) + 2\gamma\frac{\log(s)}{s} + \frac{\log^2(s)}{s}$, we have

$$\mathbb{E}[(\log|X(t)|)^2] = \frac{\beta^2}{\alpha^2}\log^2(t) + 2\frac{\beta\gamma}{\alpha}\left(\frac{\beta}{\alpha} - 1\right)\log(t) + c_1, \quad \text{(C.67)}$$

where $c_1 = \frac{\pi^2}{6}\left(\frac{1}{\alpha^2} + \frac{1}{2}\right) - \left(\frac{\pi\theta}{2\alpha}\right)^2 + \left(\frac{\log(D)}{\alpha} + \gamma\left(\frac{\beta}{\alpha} - 1\right)\right)^2 + \frac{\pi^2}{6\alpha^2}(1-\beta^2)$. □

Appendix D

Multiwavelet-based Operator

D.1 Preliminaries

D.1.1 Orthogonal Polynomials

The next set of ingredients that are useful to us are the family of orthogonal polynomials (OPs). Specifically, the OPs in the current work will serve as the mother wavelets or span the 'mother subspace' (see Section D.1.2). Therefore, we are interested in the OPs that are non-zero over a finite domain, and are zero almost everywhere (a.e.). For a given measure μ that defines the OPs, a sequence of OPs $P_0(x), P_1(x), \ldots$ satisfy $\deg(P_i) = i$, and $\langle P_i, P_j \rangle_\mu = 0, \forall i \neq j$, where $\langle P_i, P_j \rangle_\mu = \int P_i(x) P_j(x) d\mu$. Therefore, sequence of OPs are particularly useful as they can act as a set of basis for the space of polynomials with degree ¡ d by using $P_0, \ldots, P_{d-1}(x)$.

The popular set of OPs are hypergeometric polynomials (also known as Jacobi polynomials). Among them, the common choices are Legendre, Chebyshev, and Gegenbauer (which generalize Legendre and Chebyshev) polynomials. These polynomials are defined on a finite interval of $[-1, 1]$ and are useful for the current work. The other set of OPs are Laguerre, and Hermite polynomials which are defined over non-finite domain. Such OPs could be used for extending the current work to non-compact wavelets. We now review some defining properties of the Legendre and Chebyshev polynomials.

D.1.1.1 Legendre Polynomials

The Legendre polynomials are defined with respect to (w.r.t.) a uniform weight function $w_L(x) = 1$ for $-1 \leq x \leq 1$ or $w_L(x) = \mathbf{1}_{[-1,1]}(x)$ such that

$$\int_{-1}^{1} P_i(x)P_j(x)dx = \begin{cases} \frac{2}{2i+1} & i = j, \\ 0 & i \neq j. \end{cases} \tag{D.1}$$

For our purpose, we shift and scale the Legendre polynomials such that they are defined over $[0, 1]$ as $P_i(2x-1)$, and the corresponding weight function as $w_L(2x-1)$.

Derivatives: The Legendre polynomials satisfy the following recurrence relationships

$$\begin{aligned} iP_i(x) &= (2i-1)xP_{i-1}(x) - (i-1)P_{i-2}(x), \\ (2i+1)P_i(x) &= P'_{i+1}(x) - P'_{i-1}(x), \end{aligned}$$

which allow us to express the derivatives as a linear combination of lower-degree polynomials itself as follows:

$$P'_i(x) = (2i-1)P_{i-1}(x) + (2i-3)P_{i-1}(x) + \ldots, \tag{D.2}$$

where the summation ends at either $P_0(x)$ or $P_1(x)$, with $P_0(x) = 1$ and $P_1(x) = x$.

Basis: A set of orthonormal basis of the space of polynomials with degree $< d$ defined over the interval $[0, 1]$ is obtained using shifted Legendre polynomials such that

$$\phi_i = \sqrt{2i+1}P_i(2x-1),$$

w.r.t. weight function $w(x) = w_L(2x-1)$, such that

$$\langle \phi_i, \phi_j \rangle_\mu = \int_0^1 \phi_i(x)\phi_j(x)dx = \delta_{ij}.$$

D.1.1.2 Chebyshev Polynomials

The Chebyshev polynomials are two sets of polynomial sequences (first, second order) as T_i, U_i. We take the polynomial of the first order $T_i(x)$ of degree i which is defined w.r.t. weight function $w_{Ch}(x) = 1/\sqrt{1-x^2}$ for $-1 \leq x \leq 1$ as

$$\int_{-1}^{1} T_i(x)T_j(x)\frac{1}{\sqrt{1-x^2}}dx = \begin{cases} \pi & i = j = 0, \\ \pi/2 & i = j > 0, \\ 0 & i \neq j. \end{cases} \tag{D.3}$$

After applying the scale and shift to the Chebyshev polynomials such that their domain is limited to $[0, 1]$, we get $T_i(2x-1)$ and the associated weight function as $w_{Ch}(2x-1)$ such that $T_i(2x-1)$ are orthogonal w.r.t. $w_{Ch}(2x-1)$ over the interval $[0, 1]$.

Derivatives: The Chebyshev polynomials of the first order satisfy the following recurrence relationships

$$\begin{aligned} 2T_i(x) &= \frac{1}{i+1}T'_{i+1}(x) - \frac{1}{i-1}T'_{i-1}(x), \quad i > 1, \\ T_{i+1}(x) &= 2xT_i(x) - T_{i-1}(x), \end{aligned}$$

The derivative of the $T_i(x)$ can be written as the following summation of sequence of lower degree polynomials

$$T'_i(x) = i(2T_{i-1}(x) + 2T_{i-3}(x) + \ldots),$$

where the series ends at either $T_0(x) = 1$, or $T_1(x) = x$. Alternatively, the derivative of $T_i(x)$ can also be written as $T'_i(x) = iU_{i-1}(x)$, where $U_i(x)$ is the second-order Chebyshev polynomial of degree i.

Basis: A set of orthonormal basis of the space of polynomials of degree up to d and domain $[0, 1]$ is obtained using Chebyshev polynomials as

$$\phi_i = \begin{cases} \frac{2}{\sqrt{\pi}} T_i(2x-1) & i > 0, \\ \sqrt{\frac{2}{\pi}} & i = 0. \end{cases}$$

w.r.t. weight function $w_{Ch}(2x-1)$, or

$$\langle \phi_i, \phi_j \rangle_\mu = \int_0^1 \phi_i(x)\phi_j(x) w_{Cb}(2x-1)dx = \delta_{ij}.$$

Roots: Another useful property of Chebyshev polynomials is that they can be expressed as trigonometric functions; specifically, $T_n(\cos\theta) = \cos(n\theta)$. The roots of such are also well-defined in the interval $[-1, 1]$. For $T_n(x)$, the n roots $x_1, \ldots, x_n$ are given by

$$x_i = \cos\left(\frac{\pi}{n}\left(i - \frac{1}{2}\right)\right).$$

D.1.2 Multiwavelets

The multiwavelets, as introduced in (Alpert, 1993), exploit the advantages of both wavelets as well as OPs (Section D.1.1). For a given function f, instead of projecting the function onto a single wavelet function (wavelet transform), the multiwavelets go one step further and projects the function onto a subspace of degree-restricted polynomials. Along the essence of the wavelet-transform, in multiwavelets, a sequence of wavelet bases are constructed which are scaled/shifted version of the basis of the coarsest scale polynomial subspace.

In this work, we present a measure-version of the multiwavelets which opens-up a family of the multiwavelet-based models for the operator learning. In Section D.2.1, we provide a detailed mathematical formulation for developing multiwavelets using any set of OPs with measures which can be non-uniform. To be able to develop compactly supported multiwavelets, we have restricted ourself to the family of OPs which are non-zero only over a finite interval. The extension to non-compact wavelets could be done by using OPs which are non-zero over complete/semi range of the real-axis (for example, Laguerre, Hermite polynomials). As an example, we present the expressions for Legendre polynomials which use uniform measure in Section D.2.2, and Chebyshev polynomials which use non-uniform measure in Section D.2.3. The work can be readily extended to other family of OPs like Gegenbauer polynomials.

D.1.3 Measures, Basis, and Projections

Measures: The functions are expressed w.r.t. basis usually by using measures μ which could be non-uniform in-general. Intuitively, the measure provides weights to different locations over which the specified basis are defined. For a measure μ, let us consider a Radon-Nikodym derivative as $w(x) \coloneqq \frac{d\mu}{d\lambda}(x)$, where, $d\lambda \coloneqq dx$ is the Lebesgue measure. In other words, the measure-dependent integrals $\int f d\mu(x)$, can now be defined as $\int f(x)w(x)dx$.

Basis: A set of orthonormal basis w.r.t. measure μ, are $\phi_0, \ldots, \phi_{k-1}$ such that $\langle \phi_i, \phi_j \rangle_\mu = \delta_{ij}$. With the weighting function $w(x)$, which is a Radon-Nikodym derivative w.r.t. Lebesgue measure, the orthonormality condition can be re-written as $\int \phi_i(x)\phi_j(x)w(x)dx = \delta_{ij}$.

The basis can also be appended with a multiplicative function called *tilt* $\chi(x)$ such that for a set of basis ϕ_i which is orthonormal w.r.t. μ with weighting function $\frac{d\mu}{d\lambda}(x) = w(x)$, a new set of basis $\phi_i\chi$ are now orthonormal w.r.t. a measure having weighting function w/χ^2. We will see that for OPs like Chebyshev in Section D.2.3, a proper choice of tilt $\chi(x)$ simplifies the analysis.

Projections: For a given set of basis ϕ_i defined w.r.t. measure μ and corresponding weight function $w(x)$, the inner-products are defined such that they induce a measure-dependent Hilbert space structure $\mathcal{H}_\mu$. Next, for a given function f such that $f \in \mathcal{H}_\mu$, the projections onto the basis polynomials are defined as $c_i = \int f(x)\phi_i(x)w(x)dx$.

D.1.4 Gaussian Quadrature

The Gaussian quadrature are the set of tools which are useful in approximating the definite integrals of the following form

$$\int_a^b f(x)w(x)dx \approx \sum_{i=1}^{n} \omega_i f(x_i), \tag{D.4}$$

where, ω_i are the scalar weight coefficients, and x_i are the n locations chosen appropriately. For a n-point quadrature, the eq. (D.4) is exact for the functions f that are polynomials of degree $\leq 2n-1$. This is particularly useful to us, as we see in the Section D.2.

From the result in (Stoer et al., 2002), it can be argued that, for a class of OPs P_i defined w.r.t. weight function $w(x)$ over the interval $[a, b]$ such that $x_1, x_2, \ldots, x_n$ are the roots of P_n, if

$$\sum_{i=1}^{n} \omega_i P_k(x_i) = \begin{cases} ||P_0||_\mu^2 & k = 0, \\ 0 & k > 0, \end{cases}$$

then,

$$\sum_{i=1}^{n} \omega_i f(x_i) = \int_a^b f(x)w(x)dx,$$

for any f such that f is a polynomial of degree $\leq 2n-1$. The weight coefficients can also be written in a closed-form expression (Abramowitz and Stegun, 1965) as follows

$$\omega_i = \frac{a_n}{a_{n-1}} \frac{\int_a^b P_{n-1}^2(x)w(x)dx}{P_n'(x_i)P_{n-1}(x_i)}, \tag{D.5}$$

where, a_n is the coefficient of x^n in P_n. Thus, the integral in (D.4) can be computed using family of OPs defined w.r.t. weight function $w(x)$. Depending on the class of OPs chosen, the Gaussian quadrature formula can be derived accordingly using eq. (D.5). For a common choice of OPs, the corresponding name for the Quadrature is 'Gaussian-Legendre', 'Gaussian-Chebyshev', 'Gaussian-Laguerre', etc.

D.1.5 Gram-Schmidt Orthogonalization

The Gram-Schmidt Orthogonalization (GSO) is a common technique for deriving a (i) set of vectors in a subspace, orthogonal to an (ii) another given set of vectors. We briefly write the GSO procedure for obtaining a set of orthonormal polynomials w.r.t. measures which in-general is different for polynomials in set (i) and (ii). Specifically, we consider that for a given subspace of polynomials with degree $< k$ as V_0 and another subspace of polynomials with degree $< k$ V_1, such that $V_0 \subset V_1$, we wish to obtain a set of orthonormal basis for the subspace of polynomials with degree $< k$ W_0, such that $V_0 \perp W_0$ and $W_0 \subset V_1$. It is apparent that, if $\dim(W_0) = n$, $\dim(V_0) = m$ and $\dim(V_1) = p$, then $m + n \leq p$.

Let $(\psi_0, \ldots, \psi_{n-1})$ be a set of basis of the polynomial subspace W_0, $(\phi_0^{(0)}, \ldots, \phi_{m-1}^{(0)})$ be a set of basis for V_0, and $(\phi_1^{(0)}, \ldots, \phi_{p-1}^{(1)})$ be a set of basis for V_1. We take that basis ψ_i and $\phi_i^{(0)}$ are defined w.r.t. same measure μ_0, while $\phi_i^{(1)}$ are defined w.r.t. a different measure μ_1. A set of ψ_i can be obtained by iteratively applying the following procedure for $i = 0, 1, \ldots, n-1$

$$\begin{aligned} \psi_i &\leftarrow \phi_i^{(1)} - \textstyle\sum_{j=0}^{m-1} \langle \phi_i^{(1)}, \phi_j^{(0)} \rangle_{\mu_0} \phi_j^{(0)} - \sum_{l=0}^{i-1} \langle \phi_i^{(i)}, \psi_l \rangle_{\mu_0} \psi_l, \\ \psi_i &\leftarrow \frac{\psi_i}{||\psi_i||_{\mu_0}}. \end{aligned} \tag{D.6}$$

The procedure in (D.6) results in a set of orthonormal basis of W_0 such that $\langle \psi_i, \psi_j \rangle_{\mu_0} = \delta_{ij}$ as well as $\langle \psi_i, \phi_j^{(0)} \rangle_{\mu_0}$, $\forall\, 0 \leq i < n, 0 \leq j < m$. We will see in Section D.2 that the inner-product integrals in eq. (D.6) can be efficiently computed using the Gaussian Quadrature formulas (as discussed in Section D.1.4).

D.2 Derivations for Multiwavelet Filters

Using the mathematical preliminaries and tools discussed in the Sections D.1, we are now in shape to present a detailed derivations for the measure dependent multi-wavelet filters. We start with deriving the general filters expressions in Section D.2.1. Particular expressions for Legendre polynomials are presented in Section D.2.2, and then for Chebyshev polynomials in Section D.2.3.

D.2.1 Filters as Subspace Projection Coefficients

The 'multiwavelet filters' play the role of transforming the multiwavelet coefficients from one scale to another. Let us revisit the Section 5.1.2, where we defined a space of piecewise polynomial functions, for $k \in \mathbb{N}$ and $n \in \mathbb{Z}^+ \cup \{0\}$ as, $\mathbf{V}_n^k$. The $\dim(\mathbf{V}_n^k) = 2^n k$, and for subsequent n, each subspace is contained in another, i.e, $V_{n-1}^k \subset V_n^k$. Now, if $\phi_0, \ldots, \phi_{k-1}$ are a set of basis polynomials for V_0^k w.r.t. measure μ_0, then we know that a set of basis for V_1^k can be obtained by scale and shift of ϕ_i as $\phi_{jl}^1 = 2^{1/2}\phi_j(2x-l)$ $l = 0, 1$, and the measure accordingly as μ_1. For a given function f, its multiwavelet coefficients for projections over V_0^k are taken as $s_{0i}^0 = \langle f, \phi_i \rangle)\mu_0$ and for V_1^k is taken as $s_{li}^1 = \langle f, \phi_{il}^1 \rangle_{\mu_1}$, and, we are looking for filter coefficients (H) such that a transformation between projections at these two consecutive scale exists, or

$$s_{0i}^0 = \sum\nolimits_{l=0,1} \sum\nolimits_{j=0}^{k-1} H_{ij}^{(l)} s_{lj}^1. \tag{D.7}$$

Let us begin by considering a simple scenario. Since, $V_0^k \subset V_1^k$, the basis are related as

$$\phi_i = \sum\nolimits_{j=0}^{k-1} \alpha_{ij}^{(0)} \sqrt{2}\phi_j(2x) + \sum\nolimits_{j=0}^{k-1} \alpha_{ij}^{(1)} \sqrt{2}\phi_j(2x-1). \tag{D.8}$$

It is straightforward to see that if ϕ_i and ϕ_{il}^1 are defined w.r.t. same measure, or $\mu_0 = \mu_1$ almost everywhere (a.e.), then the filters transforming the multiwavelet coefficients from higher to lower scale, are exactly equal to the subspace mapping coefficients $\alpha_{ij}^{(0)}, \alpha_{ij}^{(1)}$ (by taking inner-product with f on both sides in (D.8)). However, this is not the case in-general, i.e., the measures w.r.t. which the basis are defined at each scale are not necessarily same. To remedy this issue, and to generalize the multiwavelet filters, we now present a general measure-variant version of the multiwavelet filters.

We note that, solving for filters H that satisfy eq. (D.7) indeed solves the general case of $n+1 \to n$ scale, which can be obtained by a simple change of variables as $s_{l,i}^n = \sum_{j=0}^{k-1} H_{ij}^{(0)} s_{2l,j}^{n+1} + \sum_{j=0}^{k-1} H_{ij}^{(1)} s_{2l+1,j}^{n+1}$. Now, for solving (D.7), we consider the following equation

$$\phi_i(x)\frac{d\mu_0}{d\lambda}(x) = \sum_{j=0}^{k-1} H_{ij}^{(0)} \sqrt{2}\phi_j(2x)\frac{d\mu_1}{d\lambda}(x) + \sum_{j=0}^{k-1} H_{ij}^{(1)} \sqrt{2}\phi_j(2x-1)\frac{d\mu_1}{d\lambda}(x), \tag{D.9}$$

where $\frac{d\mu}{d\lambda}$ is the Radon-Nikodym derivative as discussed in Section D.1.3, and we have also defined $d\lambda \coloneqq dx$. We observe that eq. (D.7) can be obtained from (D.9) by simply integrating with f on both sides.

Next, we observe an important fact about multiwavelets (or wavelets in-general) that the advantages offered by multiwavelets rely on their ability to project a function *locally*. One way to achieve this is by computing basis functions which are dilation/translations of a fixed mother wavelet. However, the idea can be generalized by

projecting a given function onto any set of basis as long as they capture the *locality*. A possible approach to generalize is by using a tilt variant of the basis at higher scales, i.e., using $\sqrt{2}\tilde{\phi}_i(2x) = \sqrt{2}\phi_i(2x)\chi_0(x)$, and $\sqrt{2}\tilde{\phi}_i(2x-1) = \sqrt{2}\phi_i(2x-1)\chi_1(x)$ such that $\sqrt{2}\tilde{\phi}_i(2x)$ are now orthonormal w.r.t. weighting function $w(2x)/\chi_0^2(x)$, and similarly $\sqrt{2}\tilde{\phi}_i(2x-1)$ w.r.t. $w(2x-1)/\chi_1^2(x)$. By choosing $\chi_0(x) = w(2x)/w(x)$, and $\chi_1(x) = w(2x-1)/w(x)$, and taking the new tilted measure $\tilde{\mu}_1$ such that

$$\tilde{\mu}_1([0,1]) = \int_0^{1/2} \frac{w(2x)}{\chi_0^2(x)} d\lambda(x) + \int_{1/2}^1 \frac{w(2x-1)}{\chi_1^2(x)} d\lambda(x),$$

or,

$$\frac{d\tilde{\mu}_1}{d\lambda}(x) = \begin{cases} \frac{w(2x)}{\chi_0^2(x)} & 0 \le x \le 1/2, \\ \frac{w(2x-1)}{\chi_1^2(x)} & 1/2 < x \le 1. \end{cases}$$

We re-write the eq. (D.9), by substituting $\phi_i(2x) \leftarrow \tilde{\phi}_i(2x), \phi_i(2x-1) \leftarrow \tilde{\phi}_i(2x-1)$ and $\mu_1 \leftarrow \tilde{\mu}_1$, in its most useful form for the current work as follows

$$\phi_i(x)w(x) = \sum_{j=0}^{k-1} H_{ij}^{(0)} \sqrt{2}\phi_j(2x)w(x) + \sum_{j=0}^{k-1} H_{ij}^{(1)} \sqrt{2}\phi_j(2x-1)w(x),$$

or,

$$\phi_i(x) = \sum_{j=0}^{k-1} H_{ij}^{(0)} \sqrt{2}\phi_j(2x) + \sum_{j=0}^{k-1} H_{ij}^{(1)} \sqrt{2}\phi_j(2x-1), \qquad (a.e.). \tag{D.10}$$

Thus, *filter coefficients can be looked upon as subspace projection coefficients*, with a proper choice of *tilted* basis. Note that eq.(D.14) is now equivalent to (D.8) but is an outcome of a different back-end machinery. Since, $\sqrt{2}\phi_i(2x), \sqrt{2}\phi_i(2x-1)$ are orthonormal basis for V_1^k, we have

$$2\int_0^{1/2} \phi_i(2x)\phi_j(2x)w(2x)dx = \delta_{ij},$$
$$2\int_{1/2}^1 \phi_i(2x-1)\phi_j(2x-1)w(2x-1)dx = \delta_{ij},$$

and hence we obtain the filter coefficients as follows

$$H_{ij}^{(0)} = \sqrt{2}\int_0^{1/2} \phi_i(x)\phi_j(2x)w(2x)dx, \tag{D.11}$$
$$H_{ij}^{(1)} = \sqrt{2}\int_{1/2}^1 \phi_i(x)\phi_j(2x-1)w(2x-1)dx. \tag{D.12}$$

For a given set of basis of V_0^k as $\phi_0, \ldots, \phi_{k-1}$ defined w.r.t. measure/weight function $w(x)$, the filter coefficients H can be derived by solving eq. (D.10). In a similar

way, if $\psi_0, \ldots, \psi_{k-1}$ is the basis for the multiwavelet subspace W_0^k w.r.t. measure μ_0 such that $V_0^k \bigoplus W_0^k = V_1^k$, and the projection of function f over W_0^k is denoted by $d_{0,i}^0 = \langle f, \psi_i \rangle_{\mu_0}$, then the filter coefficients for obtaining the multiwavelet coefficients is written as

$$d_{0i}^0 = \sum\nolimits_{l=0,1} \sum\nolimits_{j=0}^{k-1} G_{ij}^{(l)} s_{lj}^1. \tag{D.13}$$

Again using a change of variables, we get $d_{l,i}^n = \sum_{j=0}^{k-1} G_{ij}^{(0)} s_{2l,j}^{n+1} + \sum_{j=0}^{k-1} G_{ij}^{(1)} s_{2l+1,j}^{n+1}$. To solve for G in (D.13), similar to eq. (D.10), the measure-variant multiwavelet basis transformation (with appropriate tilt) is written as

$$\psi_i(x) = \sum_{j=0}^{k-1} G_{ij}^{(0)} \sqrt{2} \phi_j(2x) + \sum_{j=0}^{k-1} G_{ij}^{(1)} \sqrt{2} \phi_j(2x-1), \qquad (a.e.). \tag{D.14}$$

Similar to eq. (D.11)-(D.12), the filter coefficients G can be obtained from (D.14) as follows

$$G_{ij}^{(0)} = \sqrt{2} \int_0^{1/2} \psi_i(x) \phi_j(2x) w(2x) dx, \tag{D.15}$$

$$G_{ij}^{(1)} = \sqrt{2} \int_{1/2}^{1} \psi_i(x) \phi_j(2x-1) w(2x-1) dx. \tag{D.16}$$

Since $\langle \phi_i, \phi_j \rangle_{\mu_0} = \delta_{ij}$, $\langle \psi_i, \psi_j \rangle_{\mu_0} = \delta_{ij}$ and $\langle \phi_i, \psi_j \rangle_{\mu_0} = 0$, therefore, using (D.10), (D.14), we can write that

$$\begin{aligned} \int_0^1 \phi_i(x) \phi_j(x) w(x) dx = & \, 2 \sum_{l=0}^{k-1} \sum_{l'=0}^{k-1} H_{il}^{(0)} H_{jl'}^{(0)} \int_0^{1/2} \phi_l(2x) \phi_{l'}(2x) w(x) dx \\ & + 2 \sum_{l=0}^{k-1} \sum_{l'=0}^{k-1} H_{il}^{(1)} H_{jl'}^{(1)} \int_{1/2}^{1} \phi_l(2x-1) \phi_{l'}(2x-1) w(x) dx, \end{aligned} \tag{D.17}$$

$$\begin{aligned} \int_0^1 \psi_i(x) \psi_j(x) w(x) dx = & \, 2 \sum_{l=0}^{k-1} \sum_{l'=0}^{k-1} G_{il}^{(0)} G_{jl'}^{(0)} \int_0^{1/2} \phi_l(2x) \phi_{l'}(2x) w(x) dx \\ & + 2 \sum_{l=0}^{k-1} \sum_{l'=0}^{k-1} G_{il}^{(1)} G_{jl'}^{(1)} \int_{1/2}^{1} \phi_l(2x-1) \phi_{l'}(2x-1) w(x) dx, \end{aligned} \tag{D.18}$$

$$0 = 2 \sum_{l=0}^{k-1} \sum_{l'=0}^{k-1} H_{il}^{(0)} G_{jl'}^{(0)} \int_0^{1/2} \phi_l(2x) \phi_{l'}(2x) w(x) dx$$

$$+ 2\sum_{l=0}^{k-1}\sum_{l'=0}^{k-1} H_{il}^{(1)} G_{jl'}^{(1)} \int_{1/2}^{1} \phi_l(2x-1)\phi_{l'}(2x-1)w(x)dx. \tag{D.19}$$

Let us define filter matrices as $H^{(l)} = [H_{ij}^{(l)}] \in \mathbb{R}^{k\times k}$ and $G^{(l)} = [G_{ij}^{(l)}] \in \mathbb{R}^{k\times k}$ for $l = 0, 1$. Also, we define correction matrices as $\Sigma^{(0)} = [\Sigma_{ij}^{(0)}], \Sigma^{(1)} = [\Sigma_{ij}^{(1)}]$ such that

$$\begin{aligned}
\Sigma_{ij}^{(0)} &= 2\int_{0}^{1/2} \phi_i(2x)\phi_j(2x)w(x)dx, \\
\Sigma_{ij}^{(1)} &= 2\int_{1/2}^{1} \phi_i(2x-1)\phi_j(2x-1)w(x)dx.
\end{aligned} \tag{D.20}$$

Now, we can write that

$$\begin{aligned}
H^{(0)}\Sigma^{(0)}H^{(0)\,T} + H^{(1)}\Sigma^{(1)}H^{(1)\,T} &= I, \\
G^{(0)}\Sigma^{(0)}G^{(0)\,T} + G^{(1)}\Sigma^{(1)}G^{(1)\,T} &= I, \\
H^{(0)}\Sigma^{(0)}G^{(0)\,T} + H^{(1)}\Sigma^{(1)}G^{(1)\,T} &= 0.
\end{aligned} \tag{D.21}$$

Rearranging eq. we can finally express the relationships between filter matrices and correction matrices as follows

$$\begin{bmatrix} H^{(0)} & H^{(1)} \\ G^{(0)} & G^{(1)} \end{bmatrix} \begin{bmatrix} \Sigma^{(0)} & 0 \\ 0 & \Sigma^{(1)} \end{bmatrix} \begin{bmatrix} H^{(0)} & H^{(1)} \\ G^{(0)} & G^{(1)} \end{bmatrix}^T = I. \tag{D.22}$$

The discussion till now is related to 'decomposition' or transformation of multiwavelet transform coefficients from higher to lower scale. However, the other direction, i.e., 'reconstruction' or transformation from lower to higher scale can also be obtained from (D.22). First, note that the general form of eq. (D.7), (D.13) can be written in the matrix format as

$$\begin{aligned}
\mathbf{s}_l^n &= H^{(0)}\mathbf{s}_{2l}^{n+1} + H^{(1)}\mathbf{s}_{2l+1}^{n+1}, \\
\mathbf{d}_l^n &= G^{(0)}\mathbf{s}_{2l}^{n+1} + G^{(1)}\mathbf{s}_{2l+1}^{n+1}.
\end{aligned} \tag{D.23}$$

Next, we observe that $\Sigma^{(0)}, \Sigma^{(1)} \succ 0$, which follows from their definition. Therefore, eq. (D.22) can be inverted to get the following form

$$\begin{bmatrix} H^{(0)} & H^{(1)} \\ G^{(0)} & G^{(1)} \end{bmatrix} \begin{bmatrix} H^{(0)} & H^{(1)} \\ G^{(0)} & G^{(1)} \end{bmatrix}^T = \begin{bmatrix} \Sigma^{(0)\,-1} & 0 \\ 0 & \Sigma^{(1)\,-1} \end{bmatrix}. \tag{D.24}$$

Finally, by using (D.24), we can essentially invert the eq. (D.23) to get

$$\begin{aligned}
\mathbf{s}_{2l}^{n+1} &= \Sigma^{(0)}(H^{(0)\,T}\mathbf{s}_l^n + G^{(0)\,T}\mathbf{d}_l^n), \\
\mathbf{s}_{2l+1}^{n+1} &= \Sigma^{(1)}(H^{(1)\,T}\mathbf{s}_l^n + G^{(1)\,T}\mathbf{d}_l^n).
\end{aligned} \tag{D.25}$$

In the following Section D.2.2,D.2.3 we see the the filters H, G in (D.23), (D.25) for different polynomial basis.

D.2.2 Multiwavelets using Legendre Polynomials

The basis for V_0^k are chosen as normalized shifted Legendre polynomials of degree upto k w.r.t. weight function $w_L(2x-1) = \mathbf{1}_{[0,1]}(x)$ from Section D.1.1.1. For example, the first three bases are

$$\begin{aligned} \phi_0(x) &= 1, \\ \phi_1(x) &= \sqrt{3}(2x-1), \\ \phi_2(x) &= \sqrt{5}(6x^2-6x+1), \quad 0 \le x \le 1. \end{aligned} \tag{D.26}$$

For deriving a set of basis ψ_i of W_0^k using GSO, we need to evaluate the integrals which could be done efficiently using Gaussian quadrature.

Gaussian-Legendre Quadrature: The integrals involved in GSO procedure, and the computations of H, G can be done efficiently using the Gaussian quadrature as discussed in Section D.1.4. Since the basis functions ϕ_i, ψ_i are polynomials, therefore, the quadrature summation would be *exact*. For a given k basis of the subspace V_0^k, the deg $(\phi_i\phi_j) < 2k-1$, as well as deg $(\phi_i\psi_j) < 2k-1$, therefore a k-point quadrature would be sufficient for expressing the integrals. Next, we take the interval $[a,b] = [0,1]$, and the OPs for approximation in Gaussian quadrature as shifted Legendre polynomials $P_k(2x-1)$. The weight coefficients ω_i can be written as

$$\begin{aligned} \omega_i &= \frac{a_k}{a_{k-1}} \frac{\int_0^1 P_{k-1}^2(2x-1)w(2x-1)dx}{P_k'(2x_i-1)P_{k-1}(2x_i-1)} \\ &= \frac{2k-1}{k}.\frac{1}{2k-1}\frac{1}{P_k'(2x_i-1)P_{k-1}(2x_i-1)} = \frac{1}{kP_k'(2x_i-1)P_{k-1}(2x_i-1)}, \end{aligned} \tag{D.27}$$

where x_i are the k roots of $P_k(2x-1)$ and a_k can be expressed in terms of a_{k-1} using the recurrence relationship of Legendre polynomials from Section D.1.1.1.

A set of basis for V_1^k is $\sqrt{2}\phi_i(2x)$ and $\sqrt{2}\phi_i(2x-1)$ with weight functions $w_L(4x-1) = \mathbf{1}_{[0,1/2]}(x)$ and $w_L(4x-3) = \mathbf{1}_{(1/2,1]}(x)$, respectively. We now use GSO procedure as outlined in Section D.1.5 to obtain set of basis $\psi_0, \ldots, \psi_{k-1}$ for W_0^k. We use Gaussian-Legendre quadrature formulas for computing the inner-products. As an example, the inner-products are computed as follows

$$\langle \sqrt{2}\phi_i, \phi_j \rangle_{\mu_0} = \int_0^1 \sqrt{2}\phi_i(2x)\phi_j(x)w_L(2x-1)dx$$

$$= \sqrt{2} \sum_{i=1}^{k} \omega_i \phi_i(2x_i) \phi_j(x_i),$$

where $\phi_i(2x_i) = 0$ for $x_i > 0.5$.

With shifted Legendre polynomials as basis for V_0^3, the multiwavelet bases for W_0^3 are

$$\begin{aligned}
\psi_0(x) &= \begin{cases} 6x - 1 & 0 \le x \le 1/2, \\ 6x - 5 & 1/2 < x \le 1, \end{cases} \\
\psi_1(x) &= \begin{cases} \sqrt{3}(30x^2 - 14x + 1) & 0 \le x \le 1/2, \\ \sqrt{3}(30x^2 - 46x + 17) & 1/2 < x \le 1, \end{cases} \\
\psi_2(x) &= \begin{cases} \sqrt{5}(24x^2 - 12x + 1) & 0 \le x \le 1/2, \\ \sqrt{5}(-24x^2 + 36x - 13) & 1/2 < x \le 1. \end{cases}
\end{aligned} \tag{D.28}$$

Next, we compute the filter matrices, but first note that since the weighting function for Legendre polynomials basis are $w_L(x) = \mathbf{1}_{[0,1]}(x)$, therefore, $\Sigma^{(0)}, \Sigma^{(1)}$ in eq. (D.20) are just identity matrices because of orthonormality of the basis $\sqrt{2}\phi_i(2x)$ and $\sqrt{2}\phi_i(2x-1)$ w.r.t. $\mathbf{1}_{[0,1/2]}(x)$ and $\mathbf{1}_{[1/2,1]}(x)$, respectively. The filter coefficients can be computed using Gaussian-Legendre quadrature as follows

$$\begin{aligned}
H_{ij}^{(0)} &= \sqrt{2} \int_0^{1/2} \phi_i(x) \phi_j(2x) w_L(2x-1) dx \\
&= \frac{1}{\sqrt{2}} \int_0^1 \phi_i(x/2) \phi_j(x) dx \\
&= \frac{1}{\sqrt{2}} \sum_{i=1}^{k} \omega_i \phi_i\left(\frac{x_i}{2}\right) \phi_j(x_i),
\end{aligned}$$

and similarly other coefficients can be obtained in eq. (D.11)-(D.12), (D.15)-(D.16). As an example, for $k = 3$, following the outlined procedure, the filter coefficients are derived as follows

$$H^{(0)} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & 0 \\ -\frac{\sqrt{3}}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & 0 \\ 0 & -\frac{\sqrt{15}}{4\sqrt{2}} & \frac{1}{4\sqrt{2}} \end{bmatrix}, \qquad H^{(1)} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{\sqrt{3}}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & 0 \\ 0 & \frac{\sqrt{15}}{4\sqrt{2}} & \frac{1}{4\sqrt{2}} \end{bmatrix},$$

$$G^{(0)} = \begin{bmatrix} \frac{1}{2\sqrt{2}} & \frac{\sqrt{3}}{2\sqrt{2}} & 0 \\ 0 & \frac{1}{4\sqrt{2}} & \frac{\sqrt{15}}{4\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix}, \qquad G^{(1)} = \begin{bmatrix} -\frac{1}{2\sqrt{2}} & \frac{\sqrt{3}}{2\sqrt{2}} & 0 \\ 0 & -\frac{1}{4\sqrt{2}} & \frac{\sqrt{15}}{4\sqrt{2}} \\ 0 & 0 & -\frac{1}{\sqrt{2}} \end{bmatrix}.$$

D.2.3 Multiwavelets using Chebyshev Polynomials

We choose the basis for V_0^k as shifted Chebyshev polynomials of the first-order from degree 0 to $k-1$. The weighting function for shifted Chebyshev polynomials is $w_{Ch}(2x-1) = 1\sqrt{1-(2x-1)^2}$ from Section D.1.1.2. The first three bases using Chebyshev polynomials are as follows

$$\begin{aligned}
\phi_0(x) &= \sqrt{2/\pi}, \\
\phi_1(x) &= \frac{2}{\sqrt{\pi}}(2x-1), \\
\phi_2(x) &= \frac{2}{\sqrt{\pi}}(8x^2-8x+1), 0 \leq x \leq 1.
\end{aligned}$$

The Gaussian quadrature for the Chebyshev polynomials is used to evaluate the integrals that appears in the GSO procedure as well as in the computations of filters H, G.

Gaussian-Chebyshev Quadrature: The basis functions ϕ_i, ψ_i resulting from the use of shifted Chebyshev polynomials are also polynomials with degree of their products such that $\deg(\phi_i\phi_j) < 2k-1$ and $\deg(\phi_i\psi_i) < 2k-1$, therefore a k-point quadrature would be sufficient for evaluating the integrals that have products of bases. Upon taking the interval $[a,b]$ as $[0,1]$, and using the canonical OPs as shifted Chebyshev polynomials, the weight coefficients are written as

$$\begin{aligned}
\omega_i &= \frac{a_k}{a_{k-1}} \frac{\int_0^1 T_{k-1}^2(2x-1) w_{Ch}(2x-1)dx}{T_k'(2x_i-1)T_{k-1}(2x_i-1)} \\
&\overset{(a)}{=} 2\frac{\pi}{4} \frac{1}{T_k'(2x_i-1)T_{k-1}(2x_i-1)} \\
&\overset{(b)}{=} \frac{\pi}{2k},
\end{aligned} \tag{D.29}$$

where x_i are the k roots of $T_k(2x-1)$, (a) is using the fact that $a_n/a_{n-1} = 2$ by using the recurrence relationship of Chebyshev polynomials from Section D.1.1.2, and assumes $k > 1$ for the squared integral. For (b), we first note that $T_k(\cos\theta) = \cos(k\theta)$, hence, $T_k'(\cos\theta) = n\sin(n\theta)/\sin(\theta)$. Since x_i are the roots of $T_k(2x-1)$, therefore, $2x_i - 1 = \cos(\frac{\pi}{n}(i-1/2))$. Substituting the x_i, we get $T_k'(2x_i-1)T_{k-1}(2x_i-1) = k$.

A set of basis for V_1^k is $\sqrt{2}\phi_i(2x)$ and $\sqrt{2}\phi_i(2x-1)$ with weight functions $w_{Ch}(4x-1) = 1/\sqrt{1-(4x-1)^2}$ and $w_{Cb}(4x-3) = 1/\sqrt{1-(4x-3)^2}$, respectively. We now use GSO procedure as outlined in Section D.1.5 to obtain set of basis $\psi_0, \ldots, \psi_{k-1}$ for W_0^k. We use Gaussian-Chebyshev quadrature formulas for computing the inner-products.

As an example, the inner-products are computed as follows

$$\begin{aligned}\langle\sqrt{2}\phi_i,\phi_j\rangle_{\mu_0} &= \int_0^1 \sqrt{2}\phi_i(2x)\phi_j(x)w_{Ch}(2x-1)dx \\ &= \sqrt{2}\frac{\pi}{2k}\sum_{i=1}^{k}\phi_i(2x_i)\phi_j(x_i),\end{aligned}$$

where $\phi_i(2x_i)=0$ for $x_i > 0.5$.

With shifted Chebyshev polynomials as basis for V_0^3, the multiwavelet bases for W_0^3 are derived as

$$\begin{aligned}\psi_0(x) &= \begin{cases}4.9749x-0.5560 & 0\le x\le 1/2,\\ 4.9749x-4.4189 & 1/2<x\le 1,\end{cases}\\ \psi_1(x) &= \begin{cases}58.3516x^2-22.6187x+0.9326 & 0\le x\le 1/2,\\ 58.3516x^2-94.0846x+36.6655 & 1/2<x\le 1,\end{cases}\\ \psi_2(x) &= \begin{cases}59.0457x^2-23.7328x+1.0941 & 0\le x\le 1/2,\\ -59.0457x^2+94.3586x-36.4070 & 1/2<x\le 1.\end{cases}\end{aligned} \tag{D.30}$$

Next, we compute the filter and the correction matrices. The filter coefficients can be computed using Gaussian-Chebyshev quadrature as follows

$$\begin{aligned}H_{ij}^{(0)} &= \sqrt{2}\int_0^{1/2}\phi_i(x)\phi_j(2x)w_{Ch}(4x-1)dx\\ &= \frac{1}{\sqrt{2}}\int_0^1\phi_i(x/2)\phi_j(x)w_{Ch}(2x-1)dx\\ &= \frac{\pi}{2\sqrt{2}k}\sum_{i=1}^{k}\phi_i\left(\frac{x_i}{2}\right)\phi_j(x_i),\end{aligned}$$

and similarly, other coefficients can be obtained in eq. (D.11)-(D.12), (D.15)-(D.16). Using the outlined procedure for Chebyshev based OP basis, for $k=3$, the filter and the corrections matrices are derived as

$$H^{(0)} = \begin{bmatrix}\frac{1}{\sqrt{2}} & 0 & 0\\ -\frac{1}{2} & \frac{1}{2\sqrt{2}} & 0\\ -\frac{1}{4} & -\frac{1}{\sqrt{2}} & \frac{1}{4\sqrt{2}}\end{bmatrix}, \qquad H^{(1)} = \begin{bmatrix}\frac{1}{\sqrt{2}} & 0 & 0\\ \frac{1}{2} & \frac{1}{2\sqrt{2}} & 0\\ -\frac{1}{4} & \frac{1}{\sqrt{2}} & \frac{1}{4\sqrt{2}}\end{bmatrix},$$

$$G^{(0)} = \begin{bmatrix}0.6094 & 0.7794 & 0\\ 0.6632 & 1.0272 & 1.1427\\ 0.6172 & 0.9070 & 1.1562\end{bmatrix}, \qquad G^{(1)} = \begin{bmatrix}-0.6094 & 0.7794 & 0\\ 0.6632 & -1.0272 & 1.1427\\ -0.6172 & 0.9070 & -1.1562\end{bmatrix},$$

$$\Sigma^{(0)} = \begin{bmatrix} 1 & -0.4071 & -0.2144 \\ -0.4071 & 0.8483 & -0.4482 \\ -0.2144 & -0.4482 & 0.8400 \end{bmatrix}, \qquad \Sigma^{(1)} = \begin{bmatrix} 1 & 0.4071 & -0.2144 \\ 0.4071 & 0.8483 & 0.4482 \\ -0.2144 & 0.4482 & 0.8400 \end{bmatrix}.$$

Appendix E

Approximate Submodularity and Sensor Selection

E.1 Proof of Theorem 6.3

Proof. Let S_G be the output of the greedy algorithm and therefore $f(\Omega^*) \geq f(S_G)$. At the ith stage of the algorithm, if S_G^i is the selected set then $f(\Omega^* \cup S_G^i)$ can be expanded in two ways. First we have

$$\begin{aligned} f(\Omega^* \cup S_G^i) &= f(\Omega^*) + f_{\Omega^*}(S_G^i) = f(\Omega^*) + \sum_{s_j \in S_G^i \setminus \Omega^*} f_{\Omega^* \cup S_G^{j-1}}(s_j) \\ &\geq f(\Omega^*) + (1 - \alpha_G) \sum_{s_j \in S_G^i \setminus \Omega^*} f_{S_G^{j-1}}(s_j), \end{aligned} \tag{E.1}$$

where the inequality is written using the definition of greedy curvature from (6.7). On the other hand, the expansion is as follows

$$f(\Omega^* \cup S_G^i) = f(S_G^i) + f_{S_G^i}(\Omega^*). \tag{E.2}$$

After combining (E.1) and (E.2), we can write that

$$f(\Omega^*) + (1 - \alpha_G) \sum_{s_j \in S_G^i \setminus \Omega^*} f_{S_G^{j-1}}(s_j) \leq \sum_{s_j \in S_G^i} f_{S_G^{j-1}}(s_j) + f_{S_G^i}(\Omega^*),$$

which can be re-written as

$$f(\Omega^*) \leq \alpha_G \sum_{s_j \in S_G^i} f_{S_G^{j-1}}(s_j) + (1-\alpha_G) \sum_{s_j \in S_G^i \cap \Omega^*} f_{S_G^{j-1}}(s_j) + f_{S_G^i}(\Omega^*).$$

At this point, it should be noted that out of $\text{ROS}(f, \delta)$, we have chosen g which has total curvature of α_δ. We can upper bound the last term of the above inequality using (6.3) and use the diminishing returns property of submodular function g to write that

$$\begin{aligned} f(\Omega^*) &\leq \alpha_G f(S_G^i) + (1-\alpha_G) \sum_{s_j \in S_G^i \cap \Omega^*} f_{S_G^{j-1}}(s_j) + (1+\delta) \sum_{\omega \in \Omega^* \setminus S_G^i} g_{S_G^i}(\omega) \\ &\leq \alpha_G f(S_G^i) + (1-\alpha_G) \sum_{s_j \in S_G^i \cap \Omega^*} f_{S_G^{j-1}}(s_j) + \frac{1+\delta}{1-\delta} \sum_{\omega \in \Omega^* \setminus S_G^i} f_{S_G^i}(\omega). \end{aligned}$$

The greedy algorithm at the $(i+1)$th step would select s_{i+1} according to the following

$$A(i+1) = \max_{a \in \Omega \setminus S_G^i} f_{S_G^i}(a) = f_{S_G^i}(s_{i+1}), \tag{E.3}$$

where $A(i+1)$ is the gain at the $(i+1)$th step. The last term can be upper bounded by $A(i+1)$ and let us denote the size of set $S_G^i \cap \Omega^*$ as t_i. Subsequently, we can write that

$$\begin{aligned} f(\Omega^*) &\leq \alpha_G f(S_G^i) + (1-\alpha_G) \sum_{s_j \in S_G^i \cap \Omega^*} f_{S_G^{j-1}}(s_j) + \frac{1+\delta}{1-\delta}(k-t_i) f_{S_G^i}(s_{i+1}) \\ &\leq \alpha_G f(S_G^i) + \sum_{s_j \in S_G^i \cap \Omega^*} \left\{ (1-\alpha_G) f_{S_G^{j-1}}(s_j) - f_{S_G^i}(s_{i+1}) \right\} + k \frac{1+\delta}{1-\delta} f_{S_G^i}(s_{i+1}). \end{aligned}$$

where we have used the fact that δ is feasible according to Lemma 6.1 and hence $\frac{1+\delta}{1-\delta} \geq \frac{1}{\gamma_f} \geq 1$. The summation term in the above inequality can be upper-bounded by 0 using the definition of greedy curvature from (6.7) and we obtain

$$f(\Omega^*) \leq \alpha_G f(S_G^i) + k \frac{1+\delta}{1-\delta} f_{S_G^i}(s_{i+1}). \tag{E.4}$$

We shall now upperbound the greedy curvature in terms of α_δ. In that process, we can write that

$$\begin{aligned}\min_{a\in S_G\setminus(S_G^{i-1}\cup\Omega^*)}\frac{f_{S_G^{i-1}\cup\Omega^*}(a)}{f_{S_G^{i-1}}(a)} &\geq \frac{1-\delta}{1+\delta}\min_{a\in S_G\setminus(S_G^{i-1}\cup\Omega^*)}\frac{g_{S_G^{i-1}\cup\Omega^*}(a)}{g_{S_G^{i-1}}(a)}\\ &\geq \frac{1-\delta}{1+\delta}\min_{a\in\Omega}\frac{g_{\Omega\setminus a}(a)}{g(a)} = \frac{1-\delta}{1+\delta}(1-\alpha_\delta), \qquad \text{(E.5)}\end{aligned}$$

where the first inequality is written using (6.3) and second inequality is using the property of submodular functions. The last equality is using the definition of total curvature from (6.5). Using the similar approach, we can again write that

$$\begin{aligned}\min_{\substack{a\in(S_G\cap\Omega^*)\setminus S_G^{i-1}\\ i\leq j\leq k}}\frac{f_{S_G^{j-1}}(s_j)}{f_{S_G^{i-1}}(a)} &\geq \min_{\substack{a\in(S_G\cap\Omega^*)\setminus S_G^{i-1}\\ i\leq j\leq k}}\frac{f_{S_G^{j-1}}(a)}{f_{S_G^{i-1}}(a)}\\ &\geq \frac{1-\delta}{1+\delta}\min_{\substack{a\in(S_G\cap\Omega^*)\setminus S_G^{i-1}\\ i\leq j\leq k}}\frac{g_{S_G^{j-1}}(a)}{g_{S_G^{i-1}}(a)}\\ &\geq \frac{1-\delta}{1+\delta}\min_{\substack{a\in(S_G\cap\Omega^*)\setminus S_G^{i-1}\\ i\leq j\leq k}}\frac{g_{\Omega\setminus a}(a)}{g(a)}\\ &\geq \frac{1-\delta}{1+\delta}\min_{a\in\Omega}\frac{g_{\Omega\setminus a}(a)}{g(a)} = \frac{1-\delta}{1+\delta}(1-\alpha_\delta). \qquad \text{(E.6)}\end{aligned}$$

Using the equation (E.5), (E.6) and (6.7), we can now conclude that $\alpha_G \leq \frac{2\delta}{1+\delta}+\frac{1-\delta}{1+\delta}\alpha_\delta$. Therefore, equation (E.4) can now be written as

$$f(\Omega^*) \leq \left(\frac{2\delta}{1+\delta}+\frac{1-\delta}{1+\delta}\alpha_\delta\right)f(S_G^i)+k\frac{1+\delta}{1-\delta}f_{S_G^i}(s_{i+1}). \qquad \text{(E.7)}$$

The equation (E.7) is of the form of $\lambda u_{i+1} \geq c-\sum_{j=1}^{i}u_j$, which has the solution of the form $\sum_{i=1}^{k}u_i \geq c(1-(1-\frac{1}{\lambda})^k)$ using simple mathematical induction. Therefore, we can write

$$\begin{aligned}f(S_G) &= \sum_{i=1}^{k}A(i)\\ &\geq \frac{1}{\frac{2\delta}{1+\delta}+\frac{1-\delta}{1+\delta}\alpha_\delta}\left(1-\left(1-\frac{1}{k}\left(\frac{2\delta}{1+\delta}+\frac{1-\delta}{1+\delta}\alpha_\delta\right)\left(\frac{1-\delta}{1+\delta}\right)\right)^k\right)OPT\\ &\geq \frac{1}{\frac{2\delta}{1+\delta}+\frac{1-\delta}{1+\delta}\alpha_\delta}\left(1-e^{-\left(\frac{2\delta}{1+\delta}+\frac{1-\delta}{1+\delta}\alpha_\delta\right)\frac{1-\delta}{1+\delta}}\right)OPT. \qquad \text{(E.8)}\end{aligned}$$

□

E.2 Proofs of the Propositions in Section 6.2

The proofs of the Propositions stating δ-approximation of non-submodular functions are described in this Section. In each proof, with f denoting non-submodular function and g being submodular, we will establish the inequality of the form of

$$g_S(a)\delta_l \leq f_S(a) \leq \delta_u g_S(a),$$

to get the expressions for δ_l and δ_u.

E.2.1 Proposition 7

Proof. For the matrix of the form of $W_S = \Lambda_0 + \sum_{s\in S} \mathbf{x}_s \mathbf{x}_s^T$ or any Gramian, with ordered eigenvalues $\lambda_n \geq \ldots \geq \lambda_2 \geq \lambda_1$, we consider a submodular function $g(S) = \log\det(W_S)$. The marginal of g can be written as

$$\begin{aligned} g_S(a) &= \log\det(W_{S\cup\{a\}}) - \log\det(W_S) \\ &= \sum_{i=1}^{n} \log\left(\frac{\lambda_i(W_{S\cup\{a\}})}{\lambda_i(W_S)}\right). \end{aligned} \tag{E.9}$$

The marginal of the considered non-submodular function, $f(S) = -\mathrm{tr}(W_S^{-1})$ can be written as

$$\begin{aligned} f_S(a) &= -\mathrm{tr}(W_{S\cup\{a\}}^{-1}) + \mathrm{tr}(W_S^{-1}) \\ &= \sum_{i=1}^{n} -\frac{1}{\lambda_i(W_{S\cup\{a\}})} + \frac{1}{\lambda_i(W_S)} = \sum_{i=1}^{n} \frac{\lambda_i(W_{S\cup\{a\}}) - \lambda_i(W_S)}{\lambda_i(W_{S\cup\{a\}})\lambda_i(W_S)}. \end{aligned} \tag{E.10}$$

The expression in (E.10) can be upper bounded using the relation $1-\frac{1}{x} \leq \log(x), x > 0$ as

$$\begin{aligned} f_S(a) \leq \sum_{i=1}^{n} \frac{1}{\lambda_i(W_S)} \log\left(\frac{\lambda_i(W_{S\cup\{a\}})}{\lambda_i(W_S)}\right) &\leq \frac{1}{\lambda_1(W_\phi)} \sum_{i=1}^{n} \log\left(\frac{\lambda_i(W_{S\cup\{a\}})}{\lambda_i(W_S)}\right) \\ &= \frac{1}{\lambda_1(W_\phi)} g_S(a), \end{aligned} \tag{E.11}$$

where we have used (E.9) in the last equality. The marginal of f can be lower bounded using the relation $x - 1 \geq \log(x), x > 0$ as follows.

$$\begin{aligned} f_S(a) \geq \sum_{i=1}^{n} \frac{1}{\lambda_i(W_{S\cup\{a\}})} \log\left(\frac{\lambda_i(W_{S\cup\{a\}})}{\lambda_i(W_S)}\right) &\geq \frac{1}{\lambda_n(W_\Omega)} \sum_{i=1}^{n} \log\left(\frac{\lambda_i(W_{S\cup\{a\}})}{\lambda_i(W_S)}\right) \\ &= \frac{1}{\lambda_n(W_\Omega)} g_S(a). \end{aligned}$$

□

E.2.2 Proposition 8

Proof. For the matrix of the form of $W_S = \Lambda_0 + \sum_{s\in S} x_s x_s^T$ or any Gramian, we denote the ordered eigenvalues of X_S as $\lambda_n \geq \ldots \geq \lambda_2 \geq \lambda_1$. The Weyl's inequality for matrices can be written as $\lambda_i(W_S) + \lambda_1(W_{\{a\}}) \leq \lambda_i(W_{S\cup\{a\}}) \leq \lambda_i(W_S) + \lambda_n(W_{\{a\}})$. We consider the submodular function as $g_1(S) = \lambda_n(W_S)$. The considered non-submodular function, $f(S) = \lambda_1(W_S)$ has the marginals which can be bounded using the Weyl's inequality as follows:

$$\lambda_1(W_{\{a\}}) \leq f_S(a) = \lambda_1(W_{S\cup\{a\}}) - \lambda_1(W_S) \leq \lambda_n(W_{\{a\}}). \tag{E.12}$$

The marginal of the submodular function g_1 can be lower bounded as follows.

$$\begin{aligned} g_{1\,S}(a) = \lambda_n(W_{S\cup\{a\}}) - \lambda_n(W_S) &\geq \lambda_1(W_{\{a\}}) \\ &= \frac{\lambda_1(W_{\{a\}})}{\lambda_n(W_{\{a\}})} \lambda_n(W_{\{a\}}) \\ &\geq \left(\min_{\omega\in\Omega} \frac{\lambda_1(W_{\{\omega\}})}{\lambda_n(W_{\{\omega\}})}\right) \lambda_n(W_{\{a\}}) \\ &\geq \left(\min_{\omega\in\Omega} \frac{\lambda_1(W_{\{\omega\}})}{\lambda_n(W_{\{\omega\}})}\right) f_S(a). \end{aligned}$$

where in the last inequality we have used (E.12). Similarly, we can upperbound the marginal of g_1 as follows.

$$\begin{aligned} g_{1\,S}(a) = \lambda_n(W_{S\cup\{a\}}) - \lambda_n(W_S) &\leq \lambda_n(W_{\{a\}}) \\ &= \frac{\lambda_n(W_{\{a\}})}{\lambda_1(W_{\{a\}})} \lambda_1(W_{\{a\}}) \\ &\leq \left(\max_{\omega\in\Omega} \frac{\lambda_n(W_{\{\omega\}})}{\lambda_1(W_{\{\omega\}})}\right) \lambda_1(W_{\{a\}}) \end{aligned}$$

$$\leq \left(\max_{\omega\in\Omega}\frac{\lambda_n(W_{\{\omega\}})}{\lambda_1(W_{\{\omega\}})}\right) f_S(a).$$

Therefore, we can bound the marginal of f on the both sides as follows:

$$\frac{1}{\max\limits_{\omega\in\Omega}\frac{\lambda_n(W_{\{\omega\}})}{\lambda_1(W_{\{\omega\}})}} g_{1\,S}(a) \leq f_S(a) \leq \frac{1}{\min\limits_{\omega\in\Omega}\frac{\lambda_1(W_{\{\omega\}})}{\lambda_n(W_{\{\omega\}})}} g_{1\,S}(a),$$

or,

$$\min_{\omega\in\Omega}\frac{\lambda_1(W_{\{\omega\}})}{\lambda_n(W_{\{\omega\}})} g_{1\,S}(a) \leq f_S(a) \leq \max_{\omega\in\Omega}\frac{\lambda_n(W_{\{\omega\}})}{\lambda_1(W_{\{\omega\}})} g_{1\,S}(a).$$

□

E.2.3 Proposition 9

Proof. Similar to the Proof of the Proposition 8, we consider the matrix of the form of $W_S = \Lambda_0 + \sum_{s\in S} x_s x_s^T$ or any Gramian, and we denote the ordered eigenvalues of X_S as $\lambda_n \geq \ldots \geq \lambda_2 \geq \lambda_1$. Therefore, the Weyl's inequality for matrices can be written as $\lambda_i(W_S) + \lambda_1(W_{\{a\}}) \leq \lambda_i(W_{S\cup\{a\}}) \leq \lambda_i(W_S) + \lambda_n(W_{\{a\}})$. Also, we have taken the case of $\Lambda_0 = \beta^2 I_N$. We consider the submodular (or modular in this case) function as $g_2(S) = \mathrm{tr}(W_S)$. The marginal of g_2 can be written as $g_{2\,S}(a) = \mathrm{tr}(W_{\{a\}})$. The marginal of the considered non-submodular function, $f(S) = \lambda_1(W_S)$ can be upper bounded as

$$\begin{aligned}
f_S(a) = \lambda_1(W_{S\cup\{a\}}) - \lambda_1(W_S) &\leq \lambda_n(W_{\{a\}}) \\
&= \mathrm{tr}(W_{\{a\}})\left(1 - \frac{\sum\limits_{i=1}^{n-1}\lambda_i(W_{\{a\}})}{\sum\limits_{i=1}^{n}\lambda_i(W_{\{a\}})}\right) \\
&\leq \mathrm{tr}(W_{\{a\}})\left(1 - \frac{n-1}{n}\frac{\lambda_1(W_{\{a\}})}{\lambda_n(W_{\{a\}})}\right) \\
&\leq \mathrm{tr}(W_{\{a\}})\left(1 - \frac{n-1}{n}\min_{\omega\in\Omega}\frac{\lambda_1(W_{\{\omega\}})}{\lambda_n(W_{\{\omega\}})}\right) \\
&= g_{2\,S}(a)\left(1 - \frac{n-1}{n}\min_{\omega\in\Omega}\frac{\lambda_1(W_{\{\omega\}})}{\lambda_n(W_{\{\omega\}})}\right),
\end{aligned}$$

where in the first inequality we have used the Weyl's inequality. The marginal of f can again written to be lower bounded as

$$f_S(a) = \lambda_1(W_{S\cup\{a\}}) - \lambda_1(W_S) \geq \lambda_1(W_{\{a\}})$$

$$
\begin{aligned}
&\geq \frac{\lambda_1(W_{\{a\}})}{n}\left(1+\frac{\lambda_{n-1}(W_{\{a\}})}{\lambda_n(W_{\{a\}})}+\ldots+\frac{\lambda_1(W_{\{a\}})}{\lambda_n(W_{\{a\}})}\right)\\
&= \frac{\lambda_1(W_{\{a\}})\mathrm{tr}(W_{\{a\}})}{n\lambda_n(W_{\{a\}})}\\
&\geq \left(\frac{1}{n}\min_{\omega\in\Omega}\frac{\lambda_1(W_{\{\omega\}})}{\lambda_n(W_{\{\omega\}})}\right)\mathrm{tr}(W_{\{a\}})\\
&= \left(\frac{1}{n}\min_{\omega\in\Omega}\frac{\lambda_1(W_{\{\omega\}})}{\lambda_n(W_{\{\omega\}})}\right)g_{2\,S}(a).
\end{aligned}
$$

□

Appendix F

Fractional Reinforcement Learning

F.1 Proof of Theorem 7.1

For the notational purpose of the proof, we define the non-Markovian environment as M, and approximation to the environment as $\hat{M}$. The non-Markovian MPC based policy is $\hat{\pi}$, and the optimal policy is π^*. The approximation quality of the HDP dynamics are $||\hat{P}_{h'}(s'|s,a) - P_h(s'|s,a)||_1 \leq \mathcal{O}(t^q)$, $\forall h, h' \in \mathcal{H}_t$ with $h(t) = h'(t) = s$, and the approximation for cost $||c(s,a) - \hat{c}(s,a)||_\infty \leq \varepsilon$. We also assume that the range of cost function is $[c_{min}, c_{max}]$. The initial history information provided is taken as $h_0 \in \mathcal{H}_0$, for example, initial state s_0. We define $\hat{h}(h^*)$ as the trajectories taken by policy $\hat{\pi}$ (π^*), respectively. We define the value function at step-t ($0 \leq t \leq T-1$), with history h using the model M, and policy π as

$$V^{\pi}_{t,M}(h) = \mathbb{E}_{h,\pi,M} \sum_{k=t}^{T-1} \gamma^k(s_k, a_k), \tag{F.1}$$

where $h \in \mathcal{H}_t$ and $h(t) = s_t$. We first present the simulation lemma for non-Markovian HDP as follows.

Lemma F.1. *Given an approximation $\hat{M}$ of the environment M as, $||\hat{P}_{h'}(s'|s,a) - P_h(s'|s,a)||_1 \leq \mathcal{O}(t^q)$, $\forall h, h' \in \mathcal{H}_t$ with $h(t) = h'(t) = s$, and the approximation for cost $||c(s,a) - \hat{c}(s,a)||_\infty \leq \varepsilon$, and any policy π and history $h_t \in \mathcal{H}_t$,*

$$\begin{aligned} &||\mathbb{E}_{h,\pi,\hat{M}} \sum_{k=t}^{t+H-1} \gamma^k(s_k, a_k) - \mathbb{E}_{h,\pi,\hat{M}} \sum_{k=t}^{t+H-1} \gamma^k(s_k, a_k)||_\infty \\ &\qquad \leq \gamma^t H \left(\frac{c_{max} - c_{min}}{2}\right) \frac{1-\gamma^H}{1-\gamma} \mathcal{O}((t+H)^q) + \varepsilon\gamma^t \frac{1-\gamma^H}{1-\gamma}. \end{aligned} \tag{F.2}$$

Proof. The difference in the value functions can be written as

$$\begin{aligned}
&\mathbb{E}_{h_t,\pi,\hat{M}} \sum_{k=t}^{t+H-1} \gamma^k \hat{c}(s_k,a_k) - \mathbb{E}_{h_t,\pi,M} \sum_{k=t}^{t+H-1} \gamma^k c(s_k,a_k) \\
&= \mathbb{E}_{h_t,\pi,\hat{M}} \sum_{k=t}^{t+H-1} \gamma^k \hat{c}(s_k,a_k) - \mathbb{E}_{h_t,\pi,\hat{M}} \sum_{k=t}^{t+H-1} \gamma^k c(s_k,a_k) \\
&+ \mathbb{E}_{h_t,\pi,\hat{M}} \sum_{k=t}^{t+H-1} \gamma^k c(s_k,a_k) - \mathbb{E}_{h_t,\pi,M} \sum_{k=t}^{t+H-1} \gamma^k c(s_k,a_k).
\end{aligned}$$

With $||c(s,a) - \hat{c}(s,a)||_\infty \leq \varepsilon$, we can write that

$$\begin{aligned}
&||\mathbb{E}_{h_t,\pi,\hat{M}} \sum_{k=t}^{t+H-1} \hat{c}(s_k,a_k) - \mathbb{E}_{h_t,\pi,\hat{M}} \sum_{k=t}^{t+H-1} c(s_k,a_k)||_\infty \\
&\leq \mathbb{E}_{h_t,\pi,\hat{M}} \textstyle\sum_{k=t}^{t+H-1} \gamma^k \varepsilon = \varepsilon \dfrac{1-\gamma^H}{1-\gamma}.
\end{aligned}$$

For the remaining terms, we can write that

$$\begin{aligned}
&\mathbb{E}_{h_t,\pi,\hat{M}} \sum_{k=t}^{t+H-1} \gamma^k c(s_k,a_k) - \mathbb{E}_{h_t,\pi,M} \sum_{k=t}^{t+H-1} \gamma^k c(s_k,a_k) \\
&= \left(\sum_{s_k \sim h_t,\pi,\hat{M}} P(s_t,\ldots,s_{t+H-1}) - \sum_{s_k \sim h_t,\pi,M} P(s_t,\ldots,s_{t+H-1}) \right) \times \\
&\quad (\sum_{k=t}^{t+H-1} \gamma^k c(s_k,a_k) - \delta),
\end{aligned}$$

where in the last equality, we can insert δ as

$$\sum_{s_k \sim h_t,\pi,\hat{M}} P(s_t,\ldots,s_{t+H-1}) = \sum_{s_k \sim h_t,\pi,M} P(s_t,\ldots,s_{t+H-1}) = 0.$$

Next,

$$\begin{aligned}
&||\mathbb{E}_{h_t,\pi,\hat{M}} \sum_{k=t}^{t+H-1} \gamma^k c(s_k,a_k) - \mathbb{E}_{h_t,\pi,M} \sum_{k=t}^{t+H-1} \gamma^k c(s_k,a_k)||_\infty \\
&\leq ||(\sum_{s_k \sim h_t,\pi,\hat{M}} P(s_t,\ldots,s_{t+H-1}) - \sum_{s_k \sim h_t,\pi,M} P(s_t,\ldots,s_{t+H-1}))||_\infty \times \\
&\quad || \sum_{k=t}^{t+H-1} \gamma^k c(s_k,a_k) - \delta||_\infty.
\end{aligned}$$

By choosing $\delta = \sum_{k=t}^{t+H-1} \gamma^k (\frac{c_{max}+c_{min}}{2})$, and upper-bounding difference of transition dynamics as $\mathcal{O}((t+H)^q)$, we get the final expression. □

Now, we begin the proof of the Theorem 1 as follows.

$$V^{\hat{\pi}}_{t,M}(\hat{h}_t) - V^{\pi^*}_{t,M}(h^*_t) = \mathbb{E}_{\hat{h}_t,\hat{\pi},M} \sum_{k=t}^{t+H-1} \gamma^k c(s_k,a_k) - \mathbb{E}_{h^*_t,\pi^*,M} \sum_{k=t}^{t+H-1} \gamma^k c(s_k,a_k) + V^{\hat{\pi}}_{t+H,M}(\hat{h}_{t+H}) - V^{\pi^*}_{t+H,M}(h^*_{t+H}).$$

The first set of terms can be expanded as follows.

$$\begin{aligned}
&\mathbb{E}_{\hat{h}_t,\hat{\pi},M} \sum_{k=t}^{t+H-1} \gamma^k c(s_k,a_k) - \mathbb{E}_{h^*_t,\pi^*,M} \sum_{k=t}^{t+H-1} \gamma^k c(s_k,a_k) \\
&= \mathbb{E}_{\hat{h}_t,\hat{\pi},M} \sum_{k=t}^{t+H-1} \gamma^k c(s_k,a_k) - \mathbb{E}_{\hat{h}_t,\hat{\pi},\hat{M}} \sum_{k=t}^{t+H-1} \gamma^k \hat{c}(s_k,a_k) \\
&\quad + \mathbb{E}_{\hat{h}_t,\hat{\pi},\hat{M}} \sum_{k=t}^{t+H-1} \gamma^k \hat{c}(s_k,a_k) - \mathbb{E}_{\hat{h}_t,\pi^*,\hat{M}} \sum_{k=t}^{t+H-1} \gamma^k \hat{c}(s_k,a_k) \\
&\quad + \mathbb{E}_{\hat{h}_t,\pi^*,\hat{M}} \sum_{k=t}^{t+H-1} \gamma^k \hat{c}(s_k,a_k) - \mathbb{E}_{\hat{h}_t,\pi^*,\hat{M}} \sum_{k=t}^{t+H-1} \gamma^k c(s_k,a_k) \\
&\quad + \mathbb{E}_{\hat{h}_t,\pi^*,\hat{M}} \sum_{k=t}^{t+H-1} \gamma^k c(s_k,a_k) - \mathbb{E}_{h^*_t,\pi^*,M} \sum_{k=t}^{t+H-1} \gamma^k c(s_k,a_k).
\end{aligned}$$

Since the $\hat{\pi}$ is greedy policy that optimize (7.5), therefore,

$$\mathbb{E}_{\hat{h}_t,\hat{\pi},\hat{M}} \sum_{k=t}^{t+H-1} \gamma^k \hat{c}(s_k,a_k) \leq \mathbb{E}_{\hat{h}_t,\pi^*,\hat{M}} \sum_{k=t}^{t+H-1} \gamma^k \hat{c}(s_k,a_k). \tag{F.3}$$

Now, using lemma 1 and the approximation quality of costs, we get

$$\begin{aligned}
||V^{\hat{\pi}}_{t,M}(\hat{h}_t) - V^{\pi^*}_{t,M}(h^*_t)||_\infty &\leq 2\gamma^t H \left(\frac{c_{max}-c_{min}}{2}\right) \frac{1-\gamma^H}{1-\gamma} \mathcal{O}((t+H)^q) \\
&\quad + 2\gamma^t \varepsilon \frac{1-\gamma^H}{1-\gamma} + ||V^{\hat{\pi}}_{t+H,M}(\hat{h}_{t+H}) - V^{\pi^*}_{t+H,M}(h^*_{t+H})||_\infty.
\end{aligned}$$

By adding the terms from $t=0$ till $T-1$, the terms that are H apart cancels out in a telescopic sum fashion. Finally, we can write that

$$||V^{\hat{\pi}}_{0,M}(h_0) - V^{\pi^*}_{0,M}(h_0)||_\infty \leq \sum_{t=0}^{T-1} 2\gamma^t H \left(\frac{c_{max}-c_{min}}{2}\right) \frac{1-\gamma^H}{1-\gamma} \mathcal{O}((t+H)^q)$$

$$
\begin{aligned}
&+ 2\gamma^t \varepsilon \frac{1-\gamma^H}{1-\gamma} \\
&\leq 2\frac{1-\gamma^H}{1-\gamma}\left(\frac{c_{max}-c_{min}}{2}\right) H\mathcal{O}(T^q) \\
&+ 2\varepsilon \frac{1-\gamma^H}{1-\gamma}\frac{1-\gamma^T}{1-\gamma}.
\end{aligned}
$$

□

F.2 Fractional MPC

The fractional MPC in Section 7.2.2 has linear constraints. Define the optimization variable $x = [\bar{s}[k+H]^T, \bar{s}[k+H-1]^T, \ldots, \bar{s}[0]^T, a[k+H-1]^T, \ldots, a[k]^T]^T$. The first set of constraints can then be written as $\Theta\, x = b$, where

$$
\Theta = \begin{bmatrix}
I & D(\alpha,1)+A & D(\alpha,2) & \ldots & & & \ldots \\
0 & I & D(\alpha,1)+A & D(\alpha,2) & & & \ldots \\
\vdots & \vdots & \ddots & \ldots & I & D(\alpha,1)+A & \ldots \\
 & & D(\alpha,k+H) & B & 0 & \ldots & 0 \\
 & & \ldots & 0 & B & \ldots & 0 \\
 & & D(\alpha,k+1) & 0 & 0 & \ldots & B
\end{bmatrix},
$$

using,

$$
D(\alpha, j) = diag(\psi(\alpha_1, j), \ldots, \psi(\alpha_n, j)). \tag{F.4}
$$

and $b = [e[k+H-1]^T, \ldots, e[k]^T, 0^T, \ldots, 0^T]^T$. The history equality constraint can be set as $\Phi\, x = d$ with $\Phi = [0, I, 0]$ of appropriate size, and $d = [0^T, \ldots, s[k]^T, \ldots, s[0]^T, 0^T, \ldots, 0^T]^T$. Using these two equality constraints, and boundary limits for $\bar{s}[k]$, we get a quadratic programming with approximated costs $\hat{c}$ as quadratic function. For other convex versions of the approximated cost, a convex optimization with the above linear constraints can be formulated.

Appendix G

Neuron Particles

G.1 Cantor Spikes

The Cantor spikes concept is used for demonstrating the fractal memory in the spike trains. The Cantor set is constructed by repeatedly deleting 1/3 of the unit interval. The extension of Cantor set, C_r is defined such that at any iteration-n, rth ratio is removed from each subsection. The extended Cantor set C_r with ratio r at each depth level N has deleted intervals of size $r, r\delta, r\delta^2, \ldots, r\delta^{N-1}$, where $\delta = (1-r)/2$. For example, at depth $N = 2$, the sequence of deleted intervals is $T = r\delta, r, r\delta$. For the computational purpose, at any depth N, the sequence is generated by constructing a binary tree with root node having weight r, and each child (left and right) is assigned a weight of $(1-r)/2$ times the weight of its parent. The sequence T is obtained by traversing the binary tree with left child first rule. We note that, for each depth N, the neuron particles can be analytically identified as $r, r\delta, r\delta^2, \ldots, r\delta^{N-1}$. The fractality of Cantor sets (and hence cantor spikes) is analytically captured via box counting dimension (35). The box counting dimension for set A is defined as

$$\dim_{\text{box}} = \lim_{\epsilon \to 0} \frac{-\log(N(\epsilon))}{\log(\epsilon)}, \tag{G.1}$$

where, $N(\epsilon)$ are the number of boxes with diameter ϵ required to cover the set A. At iteration-n, there are $2n$ resulting sub-intervals, each of size $((1-r)/2)^n$, therefore, the box-dimension is written as

$$\dim_{\text{box}} = \lim_{n \to \infty} \frac{\log 2^n}{-\log(\frac{1-r}{2})^n} \tag{G.2}$$

$$= -\frac{\log 2}{\log((1-r)/2)} \tag{G.3}$$

Bibliography

Abramowitz, M. and I.A. Stegun (1965). *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*. Applied mathematics series. Dover Publications. ISBN: 9780486612720.

Acheson, David J (1991). *Elementary fluid dynamics*.

Adler, Jonas and Ozan Öktem (Nov. 2017). "Solving ill-posed inverse problems using iterative deep neural networks". In: *Inverse Problems* 33.12. ISSN: 1361-6420. DOI: 10.1088/1361-6420/aa9581. URL: http://dx.doi.org/10.1088/1361-6420/aa9581.

Agarwal, Mridul and Vaneet Aggarwal (2021). *Reinforcement Learning for Joint Optimization of Multiple Rewards*. arXiv: 1909.02940 [cs.LG].

Alpert, B., G. Beylkin, R. Coifman, and V. Rokhlin (1993). "Wavelet-Like Bases for the Fast Solution of Second-Kind Integral Equations". In: *SIAM Journal on Scientific Computing* 14.1, pp. 159–184. DOI: 10.1137/0914010.

Alpert, B., G. Beylkin, D. Gines, and L. Vozovoi (2002). "Adaptive Solution of Partial Differential Equations in Multiwavelet Bases". In: *Journal of Computational Physics* 182.1, pp. 149–190. ISSN: 0021-9991.

Alpert, Bradley K. (1993). "A Class of Bases in L^2 for the Sparse Representation of Integral Operators". In: *SIAM Journal on Mathematical Analysis* 24.1, pp. 246–262. DOI: 10.1137/0524016.

Alpert, Bradley K. and Vladimir Rokhlin (1991). "A Fast Algorithm for the Evaluation of Legendre Expansions". In: *SIAM Journal on Scientific and Statistical Computing* 12.1, pp. 158–179. DOI: 10.1137/0912009.

Amaratunga, Kevin and John Williams (July 2001). "Wavelet based Green's function approach to 2D PDEs". In: *Engineering Computations* 10. DOI: 10.1108/eb023913.

Anandkumar, Animashree, Daniel Hsu, Adel Javanmard, and Sham M. Kakade (2013). "Learning Linear Bayesian Networks with Latent Variables". In: *International Conference on Machine Learning*.

Arafa, A. A. M. (July 2013). "Fractional Differential Equations in Description of Bacterial Growth". In: *Differential Equations and Dynamical Systems* 21.3, pp. 205–214.

Arel, Itamar, Cong Liu, Tom Urbanik, and Airton G Kohls (2010). "Reinforcement learning-based multi-agent system for network traffic signal control". In: *IET Intelligent Transport Systems* 4.2, pp. 128–135.

Bach, Francis (2013). *Learning with Submodular Functions: A Convex Optimization Perspective*. arXiv:1111.6453.

Balaban, Valeriu, Sean Lim, Gaurav Gupta, James Boedicker, and Paul Bogdan (Aug. 2018). "Quantifying emergence and self-organisation of Enterobacter cloacae microbial communities". In: *Scientific Reports* 8.1, p. 12416. ISSN: 2045-2322. DOI: 10.1038/s41598-018-30654-9. URL: https://doi.org/10.1038/s41598-018-30654-9.

Balan, Radu, Bernhard G. Bodmann, Peter G. Casazza, and Dan Edidin (2009). "Painless Reconstruction from Magnitudes of Frame Coefficients". In: *Journal of Fourier Analysis and Applications* 16, 488–501.

Balankin, Alexander S (2018). "Mapping physical problems on fractals onto boundary value problems within continuum framework". In: *Physics Letters A* 382.4, pp. 141–146.

Baleanu, Dumitru, José António Tenreiro Machado, and Albert CJ Luo (2011). *Fractional dynamics and control*. Springer Science & Business Media.

Banstola, Ashik, Carlos Silva, Katharina Ulrich, Ming Ruan, Lindsay Robertson, and Neil McNaughton (2020). "Construction of simple, customised, brain-spanning, multi-channel, linear microelectrode arrays". In: *Journal of Neuroscience Methods* 348, p. 109011. ISSN: 0165-0270.

Bär, Markus, Rainer Hegger, and Holger Kantz (1999). "Fitting partial differential equations to space-time dynamics". In: *Physical Review E* 59.1, p. 337.

Barabasi, A. L. and R. Albert (1999). "Emergence of scaling in random networks". In: *Science* 286.5439, pp. 509–12. ISSN: 1095-9203. DOI: 10.1126/science.286.5439.509. URL: https://www.ncbi.nlm.nih.gov/pubmed/10521342.

Bargmann, Cornelia I and Eve Marder (2013). "From the connectome to brain function". In: *Nature methods* 10.6, p. 483. ISSN: 1548-7105.

Barkai, Eli, Yuval Garini, and Ralf Metzler (2012). "Strange kinetics of single molecules in living cells". In: *Physics Today* 65 (8), pp. 29 –35.

Barrat, Alain, Marc Barthélemy, and Alessandro Vespignani (2008). *Dynamical Processes on Complex Networks*. Cambridge University Press.

Bartheld, Christopher S von, Jami Bahney, and Suzana Herculano-Houzel (2016). "The search for true numbers of neurons and glial cells in the human brain: A review of 150 years of cell counting". In: *Journal of Comparative Neurology* 524.18, pp. 3865–3895. ISSN: 0021-9967.

Bassett, Danielle S and Michael S Gazzaniga (2011). "Understanding complexity in the human brain". In: *Trends in cognitive sciences* 15.5, pp. 200–209. ISSN: 1364-6613.

Batchelor, Cx K and GK Batchelor (2000). *An introduction to fluid dynamics*. Cambridge university press.

Bellman, Richard (1957). "A Markovian Decision Process". In: *Journal of Mathematics and Mechanics* 6.5, pp. 679–684.

Benedetto, J.J. (1993). *Wavelets: Mathematics and Applications*. Studies in Advanced Mathematics. Taylor & Francis. ISBN: 9780849382710.

Berry II, Michael J and Markus Meister (1998). "Refractoriness and neural precision". In: *Advances in Neural Information Processing Systems*, pp. 110–116.

Bertsekas, Dimitri P (2019). *Reinforcement learning and optimal control*. Athena Scientific Belmont, MA.

Beylkin, G. (1992). "On the Representation of Operators in Bases of Compactly Supported Wavelets". In: *SIAM Journal on Numerical Analysis* 29.6, pp. 1716–1740. DOI: 10.1137/0729097.

Beylkin, G., R. Coifman, and V. Rokhlin (1991). "Fast wavelet transforms and numerical algorithms I". In: *Communications on Pure and Applied Mathematics* 44.2, pp. 141–183. DOI: https://doi.org/10.1002/cpa.3160440202.

Beylkin, Gregory and James M. Keiser (1997). "An Adaptive Pseudo-Wavelet Approach for Solving Nonlinear Partial Differential Equations". In: *Multiscale Wavelet Methods for Partial Differential Equations*. Vol. 6. Wavelet Analysis and Its Applications. Academic Press, pp. 137–197.

Bhardwaj, Mohak, Sanjiban Choudhury, and Byron Boots (2020). *Blending MPC & Value Function Approximation for Efficient Reinforcement Learning*. arXiv: 2012.05909 [cs.LG].

Bhatnagar, Saakaar, Yaser Afshar, Shaowu Pan, Karthik Duraisamy, and Shailendra Kaushik (June 2019). "Prediction of aerodynamic flow fields using convolutional neural networks". In: *Computational Mechanics* 64.2, 525–545. ISSN: 1432-0924. DOI: 10.1007/s00466-019-01740-0.

Bhattacharya, Kaushik, Bamdad Hosseini, Nikola B. Kovachki, and Andrew M. Stuart (2020). *Model Reduction and Neural Networks for Parametric PDEs*. arXiv: 2005.03180 [math.NA].

Bian, Andrew An, Joachim M. Buhmann, Andreas Krause, and Sebastian Tschiatschek (2017). *Guarantees for Greedy Maximization of Non-submodular Functions with Applications*. arXiv:1703.02100.

Billingsley, Patrick (2008). *Probability and measure*. John Wiley & Sons.

Blankertz, B., K. R. Muller, D. J. Krusienski, G. Schalk, J. R. Wolpaw, A. Schlogl, G. Pfurtscheller, J. R. Millan, M. Schroder, and N. Birbaumer (June 2006). "The BCI competition III: validating alternative approaches to actual BCI problems". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 14.2, pp. 153–159.

Bogdan, Paul and Radu Marculescu (2011). "A fractional calculus approach to modeling fractal dynamic games". In: *2011 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE, pp. 255–260.

Borst, Alexander and Frédéric E Theunissen (1999). "Information theory and neural coding". In: *Nature neuroscience* 2.11, pp. 947–957. ISSN: 1546-1726.

Bouchaud, Jean-Philippe and Antoine Georges (1990). "Anomalous diffusion in disordered media: Statistical mechanisms, models and physical applications". In: *Physics Reports* 195.4, pp. 127 –293. ISSN: 0370-1573. DOI: https://doi.org/10.1016/0370-1573(90)90099-N. URL: http://www.sciencedirect.com/science/article/pii/037015739090099N.

Boussinesq, Joseph (1877). *Essai sur la théorie des eaux courantes*. Impr. nationale.

Brangwynne, Clifford P, Gijsje H Koenderink, Frederick C MacKintosh, and David A Weitz (2009). "Intracellular transport by active diffusion". In: *Trends in cell biology* 19.9, pp. 423–427.

Breakspear, M. (2017). "Dynamic models of large-scale brain activity". In: *Nat Neurosci* 20.3. Breakspear, Michael 2017/2/24, pp. 340–352. ISSN: 1546-1726. DOI: 10.1038/nn.4497. URL: https://www.ncbi.nlm.nih.gov/pubmed/28230845.

Brown, Emery N (2005). "Theory of point processes for neural systems". In: *Les Houches*. Vol. 80. Elsevier, pp. 691–727.

Brown, Emery N, Robert E Kass, and Partha P Mitra (2004). "Multiple neural spike train data analysis: state-of-the-art and future challenges". In: *Nature neuroscience* 7.5, p. 456.

Bu, Xiangping, Jia Rao, and Cheng-Zhong Xu (2009). "A reinforcement learning approach to online web systems auto-configuration". In: *2009 29th IEEE International Conference on Distributed Computing Systems*. IEEE, pp. 2–11.

Buchbinder, N., M. Feldman, J. Naor, and R. Schwartz (2012). "A tight (1/2) linear-time approximation to unconstrained submodular maximization". In: *FOCS*.

Cabreira, R Gomes, A da Silva Ferreira, M Kouyaté, G Demouchy, G Mériguet, R Aquino, E Dubois, S Nakamae, M Roger, J Depeyrot, et al. (2018). "Thermodiffusion of repulsive charged nanoparticles-the interplay between single-particle and thermoelectric contributions." In: *Physical chemistry chemical physics: PCCP* 20.24, pp. 16402–16413.

Caputo, Michele (1967). "Linear models of dissipation whose Q is almost frequency independent—II". In: *Geophysical Journal International* 13.5, pp. 529–539.

Chandrasekaran, Venkat, Pablo A. Parrilo, and Alan S. Willsky (Aug. 2012). "Latent variable graphical model selection via convex optimization". In: *Ann. Statist.* 40.4, pp. 1935–1967.

Chang, R. Y. and M. L. Wang (Feb. 1983). "Shifted Legendre direct method for variational problems". In: *Journal of Optimization Theory and Applications* 39.2, pp. 299–307. ISSN: 1573-2878. DOI: 10.1007/BF00934535.

Chen, Ricky T. Q., Yulia Rubanova, Jesse Bettencourt, and David Duvenaud (2019). *Neural Ordinary Differential Equations*. arXiv: 1806.07366 [cs.LG].

Chen, Zhengdao, Jianyu Zhang, Martin Arjovsky, and Léon Bottou (2020). "Symplectic Recurrent Neural Networks". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=BkgYPREtPr.

Chornoboy, ES (1986). *Maximum likelihood techniques for the identification of neural point processes*. Ph.D. Thesis. The Johns Hopkins School of Medicine, Baltimore, Md.

Chornoboy, ES, LP Schramm, and AF Karr (1988). "Maximum likelihood identification of neural point process systems". In: *Biological cybernetics* 59.4-5, pp. 265–275.

Chou, Kenneth C. and Gary S. Guthart (2000). "Representation of Green's Function Integral Operators Using Wavelet Transforms". In: *Journal of Vibration and Control* 6.1, pp. 19–48. DOI: 10.1177/107754630000600102.

Chua, Kurtland, Roberto Calandra, Rowan McAllister, and Sergey Levine (2018). "Deep reinforcement learning in a handful of trials using probabilistic dynamics models". In: *Advances in Neural Information Processing Systems*, pp. 4754–4765.

Chuong, N.M. (2018). *Pseudodifferential Operators and Wavelets over Real and p-adic Fields*. Springer International Publishing. ISBN: 9783319774732.

Clarke, W. and B. Kovatchev (June 2009). "Statistical tools to analyze continuous glucose monitor data". In: *Diabetes Technol. Ther.* 11 Suppl 1, pp. 45–54.

Coffen, Ronald D and Lynnda M Dahlquist (2009). "Magnitude of type 1 diabetes self-management in youth health care needs diabetes educators". In: *The Diabetes Educator* 35.2, pp. 302–308.

Conforti, Michele and Gerard Cornuejols (1984). "Submodular set functions, matroids and the greedy algorithm: tight worst-case bounds and some generalizations of the Rado-Edmonds theorem". In: *Discrete Applied Math* 7.3, pp. 251–274.

Cotronei, M., L.B. Montefusco, and L. Puccio (1998). "Multiwavelet analysis and signal processing". In: *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 45.8, pp. 970–987.

Couceiro, Micael and Seenith Sivasundaram (2016). "Novel fractional order particle swarm optimization". In: *Applied Mathematics and Computation* 283, pp. 36 –54.

Couceiro, Micael S., Rui P. Rocha, N. M. Fonseca Ferreira, and J. A. Tenreiro Machado (Sept. 2012). "Introducing the fractional-order Darwinian PSO". In: *Signal, Image and Video Processing* 6.3, pp. 343–350.

Darcy, Henry (1856). *Les fontaines publiques de la ville de Dijon: exposition et application...* Victor Dalmont.

Darrigol, Olivier (2005). *Worlds of flow: A history of hydrodynamics from the Bernoullis to Prandtl*. Oxford University Press.

Das, Abhimanyu and David Kempe (2008). "Algorithms for Subset Selection in Linear Regression". In: *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, pp. 45–54.

– (2011). "Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection". In: *ICML*.

Daubechies, Ingrid (1988). "Orthonormal bases of compactly supported wavelets". In: *Communications on Pure and Applied Mathematics* 41.7, pp. 909–996. DOI: https://doi.org/10.1002/cpa.3160410705.

– (1992). *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics. DOI: 10.1137/1.9781611970104.

Demirhan, Haydar and Canan Hamurkaroglu (2011). "On a multivariate log-gamma distribution and the use of the distribution in the Bayesian analysis". In: *Journal of Statistical Planning and Inference* 141.3, pp. 1141–1152.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 39.1, pp. 1–38.

Deng, D., Y. Meyer, and Y. Han (2008). *Harmonic Analysis on Spaces of Homogeneous Type*. Lecture Notes in Mathematics. Springer Berlin Heidelberg. ISBN: 9783540887447.

Derouich, Mohammed and Abdesslam Boutayeb (2002). "The effect of physical exercise on the dynamics of glucose and insulin". In: *Journal of biomechanics* 35.7, pp. 911–917.

Diabetes Control and Complications Trial Research Group and others (1995). "Resource utilization and costs of care in the Diabetes Control and Complications Trial". In: *Diabetes Care* 18.11, pp. 1468–1478.

Dornhege, Guido, Benjamin Blankertz, Gabriel Curio, and Klaus-Robert Müller (June 2004). "Boosting bit rates in non-invasive EEG single-trial classifications by feature combination and multi-class paradigms". In: *IEEE Trans. Biomed. Eng.* 51.6, pp. 993–1002.

Dorogovtsev, S. N., J. F. Mendes, and A. N. Samukhin (2000). "Structure of growing networks with preferential linking". In: *Phys Rev Lett* 85.21, pp. 4633–6. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.85.4633. URL: https://www.ncbi.nlm.nih.gov/pubmed/11082614.

Driscoll, T. A, N. Hale, and L. N. Trefethen (2014). *Chebfun Guide*. Pafnuty Publications.

Dua, Dheeru and Casey Graff (2017). *UCI Machine Learning Repository*. URL: http://archive.ics.uci.edu/ml.

Dzielinski, Andrzej and Dominik Sierociuk (2005). "Adaptive Feedback Control of Fractional Order Discrete State-Space Systems". In: *CIMCA-IAWTIC*.

Einstein, A. (1905). "Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen". In: *Annalen der Physik* 322.8, pp. 549–560. DOI: 10.1002/andp.19053220806. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/andp.19053220806. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/andp.19053220806.

Elidan, Gal, Iftach Nachman, and Nir Friedman (2007). ""Ideal Parent" Structure Learning for Continuous Variable Bayesian Networks". In: *Journal of Machine Learning Research* 8, pp. 1799–1833.

Erdos, Paul and Alfred Renyi (1960). "On the evolution of random graphs". In: *Publ. Math. Inst. Hungary. Acad. Sci.* 5, pp. 17–61.

Fan, Yuwei, Jordi Feliu-Faba, Lin Lin, Lexing Ying, and Leonardo Zepeda-Nunez (2019). *A multiscale neural network based on hierarchical nested bases*. arXiv: 1808.02376 [math.NA].

Feige, U., V. S. Mirrokni, and J. Vondrak (2011). "Maximizing non-monotone submodular functions". In: *SIAM J. Comput* 40.4, pp. 1133–1153.

Feliu-Fabà, Jordi, Yuwei Fan, and Lexing Ying (May 2020). "Meta-learning pseudo-differential operators with deep neural networks". In: *Journal of Computational Physics* 408, p. 109309. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2020.109309.

Feller, W. (1962). *On a Generalization of Marcel Riesz' Potentials and the Semi-groups Generated by Them*. Gleerup. URL: https://books.google.com/books?id=UZTqjwEACAAJ.

Filip, Silviu-Ioan, Aurya Javeed, and Lloyd Trefethen (Jan. 2019). "Smooth Random Functions, Random ODEs, and Gaussian Processes". In: *SIAM Review* 61, pp. 185–205. DOI: 10.1137/17M1161853.

Flandrin, P. (Mar. 1992). "Wavelet analysis and synthesis of fractional Brownian motion". In: *IEEE Transactions on Information Theory* 38.2, pp. 910–917.

Fox, Colin and GK Nicholls (2001). *Statistical estimation of the parameters of a pde*.

Fricke, E. C. and J. C. Svenning (2020). "Accelerating homogenization of the global plant-frugivore meta-network". In: *Nature* 585.7823. Fricke, Evan C Svenning, Jens-Christian 2020/9/4, pp. 74–78. ISSN: 1476-4687. DOI: 10.1038/s41586-020-2640-y. URL: https://www.ncbi.nlm.nih.gov/pubmed/32879498.

Gao, Jianxi, Yang-Yu Liu, Raissa M. D'Souza, and Albert-László Barabási (2014). "Target control of complex networks". In: *Nature communications*.

Gaon, Maor and Ronen I. Brafman (2019). *Reinforcement Learning with Non-Markovian Rewards*. arXiv: 1912.02552 [cs.AI].

Gefen, Yuval, Amnon Aharony, and Shlomo Alexander (Jan. 1983). "Anomalous Diffusion on Percolating Clusters". In: *Phys. Rev. Lett.* 50 (1), pp. 77–80. DOI: 10.1103/PhysRevLett.50.77. URL: https://link.aps.org/doi/10.1103/PhysRevLett.50.77.

Geiger, Philipp, Kun Zhang, Mingming Gong, Dominik Janzing, and Bernhard Schölkopf (2015). "Causal Inference by Identification of Vector Autoregressive Processes with Hidden Components". In: *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*.

Ghorbani, M. and P. Bogdan (2014). "Reducing risk of closed loop control of blood glucose in artificial pancreas using fractional calculus". In: *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 4839–4842.

Goldberger, Ary L., Luis A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and H. Eugene Stanley (2000). "PhysioBank, PhysioToolkit, and PhysioNet". In: *Circulation* 101.23, e215–e220.

Gorenflo, Rudolf, Asaf Iskenderov, and Yuri Luchko (2000). "Mapping between solutions of fractional diffusion-wave equations". In: *Fractional Calculus and Applied Analysis* 3.1, pp. 75–86.

Gorenflo, Rudolf and F. Mainardi (1999). "Approximation of Levy-Feller Diffusion by Random Walk". In: *Zeitschrift für Analysis und ihre Anwendungen* 18.2, pp. 231–246.

Gorenflo, Rudolf and Francesco Mainardi (2012). "Parametric subordination in fractional diffusion processes". In: *arXiv preprint arXiv:1210.8414*.

Gorenflo, Rudolf, Francesco Mainardi, Daniele Moretti, Gianni Pagnini, and Paolo Paradisi (2002). "Discrete random walk models for space–time fractional diffusion". In: *Chemical Physics* 284.1. Strange Kinetics, pp. 521 –541. ISSN: 0301-0104. DOI: https://doi.org/10.1016/S0301-0104(02)00714-0. URL: http://www.sciencedirect.com/science/article/pii/S0301010402007140.

Goychuk, Igor (Jan. 2015). "Anomalous transport of subdiffusing cargos by single kinesin motors: the role of mechano–chemical coupling and anharmonicity of tether". In: *Physical Biology* 12.1, p. 016013. DOI: 10.1088/1478-3975/12/1/016013. URL: https://doi.org/10.1088%2F1478-3975%2F12%2F1%2F016013.

Granger, C. W. J. (1969). "Investigating Causal Relations by Econometric Models and Cross-spectral Methods". In: *Econometrica* 37.3. Full publication date: Aug., 1969, pp. 424–438. DOI: 10.2307/1912791. URL: https://doi.org/10.2307/1912791.

Greenfeld, Daniel, Meirav Galun, Ron Kimmel, Irad Yavneh, and Ronen Basri (2019). *Learning to Optimize Multigrid PDE Solvers*. arXiv: 1902.10248 [math.NA].

Grmela, Miroslav and Hans Christian Öttinger (1997). "Dynamics and thermodynamics of complex fluids. I. Development of a general formalism". In: *Physical Review E* 56.6, p. 6620.

Gu, Albert, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Re (2020). *HiPPO: Recurrent Memory with Optimal Polynomial Projections*. arXiv: 2008.07669 [cs.LG].

Guo, Xiaoxiao, Wei Li, and Francesco Iorio (2016). "Convolutional Neural Networks for Steady Flow Approximation". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. Association for Computing Machinery, 481–490.

Gupta, Gaurav and Paul Bogdan (2017). "Distributed Placement of Power Generation Resources in Uncertain Environments". In: *2017 ACM/IEEE 8th International Conference on Cyber-Physical Systems (ICCPS)*, pp. 71–80.

Gupta, Gaurav, Meghana Kshirsagar, Ming Zhong, Shahrzad Gholami, and Juan Lavista Ferres (Aug. 2021a). "Comparing recurrent convolutional neural networks for large scale bird species classification". In: *Scientific Reports* 11.1, p. 17085. ISSN: 2045-2322. DOI: 10.1038/s41598-021-96446-w. URL: https://doi.org/10.1038/s41598-021-96446-w.

Gupta, Gaurav, Sergio Pequito, and Paul Bogdan (2018a). *Approximate Submodular Functions and Performance Guarantees*. arXiv: 1806.06323 [cs.DS].

Gupta, Gaurav, Sérgio Pequito, and Paul Bogdan (2018b). "Dealing with Unknown Unknowns: Identification and Selection of Minimal Sensing for Fractional Dynamics with Unknown Inputs". In: *in American Control Conference*. arXiv:1803.04866.

Gupta, Gaurav, Sérgio Pequito, and Paul Bogdan (2018c). "Learning Latent Fractional dynamics with Unknown Unknowns". In: *CoRR* abs/1811.00703. arXiv: 1811.00703. URL: http://arxiv.org/abs/1811.00703.

Gupta, Gaurav, Sérgio Pequito, and Paul Bogdan (2018d). "Re-thinking EEG-based Non-invasive Brain Interfaces: Modeling and Analysis". In: *Proceedings of the 9th ACM/IEEE International Conference on Cyber-Physical Systems*. ICCPS '18. IEEE Press, pp. 275–286.

Gupta, Gaurav, Justin Rhodes, Roozbeh Kiani, and Paul Bogdan (2021b). "Neuron particles capture network topology and behavior from single units". In: *bioRxiv*. DOI: 10.1101/2021.12.03.471160. eprint: https://www.biorxiv.org/content/early/2021/12/05/2021.12.03.471160.full.pdf. URL: https://www.biorxiv.org/content/early/2021/12/05/2021.12.03.471160.

Gupta, Gaurav, Anit Kumar Sahu, and Wan-Yi Lin (2020). *Noisy Batch Active Learning with Deterministic Annealing*. arXiv: 1909.12473 [cs.LG].

Gupta, Gaurav, Xiongye Xiao, and Paul Bogdan (2021). "Multiwavelet-based Operator Learning for Differential Equations". In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan. URL: https://openreview.net/forum?id=LZDiWaC9CGL.

Gupta, Gaurav, Chenzhong Yin, Jyotirmoy V. Deshmukh, and Paul Bogdan (2021c). *Non-Markovian Reinforcement Learning using Fractional Dynamics*. arXiv: 2107.13790 [cs.LG].

Hajij, Mustafa, Ghada Zamzmi, Matthew Dawson, and Greg Muller (2021). *Algebraically-Informed Deep Networks (AIDN): A Deep Learning Approach to Represent Algebraic Structures*. arXiv: 2012.01141 [cs.LG].

Harvill, Jane L. (Sept. 2009). "Intermediate Probability: A Computational Approach". In: *Journal of the American Statistical Association* 104.487, pp. 1285–1286.

Heydari, M.H., M.R. Hooshmandasl, and F. Mohammadi (2014). "Legendre wavelets method for solving fractional partial differential equations with Dirichlet boundary conditions". In: *Applied Mathematics and Computation* 234, pp. 267–276. ISSN: 0096-3003. DOI: https://doi.org/10.1016/j.amc.2014.02.047.

Holtmaat, A. and P. Caroni (2016). "Functional and structural underpinnings of neuronal assembly formation in learning". In: *Nat Neurosci* 19.12, pp. 1553–1562. ISSN: 1546-1726. DOI: 10.1038/nn.4418. URL: https://www.ncbi.nlm.nih.gov/pubmed/27749830.

Householder, AS (1964). "The theory of matrices in numerical analysis, Blaisdell Publ". In: *Co., New York*.

Hutter, Frank, Holger H Hoos, and Kevin Leyton-Brown (2010). "Automated configuration of mixed integer programming solvers". In: *International Conference on Integration of Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming*. Springer, pp. 186–202.

Hwang, Chyi and Yen-Ping Shih (1982). "Solution of integral equations via Laguerre polynomials". In: *Computers & Electrical Engineering* 9.3, pp. 123–129. ISSN: 0045-7906. DOI: https://doi.org/10.1016/0045-7906(82)90018-0.

Ito, Shinya, Fang-Chin Yeh, Emma Hiolski, Przemyslaw Rydygier, Deborah E. Gunning, Pawel Hottowy, Nicholas Timme, Alan M. Litke, and John M. Beggs (Aug. 2014). "Large-Scale, High-Resolution Multielectrode-Array Recording Depicts Functional Network Differences of Cortical and Hippocampal Cultures". In: *PLOS ONE* 9.8, pp. 1–16.

Ito, Shinya, Fang-Chin Yeh, Nicholas M. Timme, Pawel Hottowy, Alan M. Litke, and John M. Beggs (2016). *Spontaneous spiking activity of hundreds of neurons in mouse somatosensory cortex slice cultures recorded using a dense 512 electrode array*. CRCNS.org. URL: http://dx.doi.org/10.6080/K07D2S2F.

Iyer, R. K., S. Jegelka, and J. A. Bilmes (2013). "Curvature and optimal algorithms for learning and minimizing submodular functions". In: *NIPS*, pp. 2742–2750.

Jagtap, Ameya D., Yeonjong Shin, Kenji Kawaguchi, and George Em Karniadakis (2021). *Deep Kronecker neural networks: A general framework for neural networks with adaptive activation functions*. arXiv: 2105.09513 [cs.LG].

Jalali, Ali and Sujay Sanghavi (2012). "Learning the Dependence Graph of Time Series with Latent Factors". In: *International Conference on Machine Learning*.

Jeon, Jae-Hyung, Vincent Tejedor, Stas Burov, Eli Barkai, Christine Selhuber-Unkel, Kirstine Berg-Sørensen, Lene Oddershede, and Ralf Metzler (Jan. 2011). "In Vivo Anomalous Diffusion and Weak Ergodicity Breaking of Lipid Granules". In: *Phys. Rev. Lett.* 106 (4), p. 048103. DOI: 10.1103/PhysRevLett.106.048103. URL: https://link.aps.org/doi/10.1103/PhysRevLett.106.048103.

Johnson, N. F., R. Leahy, N. J. Restrepo, N. Velasquez, M. Zheng, P. Manrique, P. Devkota, and S. Wuchty (2019). "Hidden resilience and adaptive dynamics of the global online hate ecology". In: *Nature* 573.7773, pp. 261–265. ISSN: 1476-4687. DOI: 10.1038/s41586-019-1494-7. URL: https://www.ncbi.nlm.nih.gov/pubmed/31435010.

Kajani, M. Tavassoli, A. Hadi Vencheh, and M. Ghasemi (2009). "The Chebyshev wavelets operational matrix of integration and product operation matrix". In: *International Journal of Computer Mathematics* 86.7, pp. 1118–1125. DOI: 10.1080/00207160701736236.

Kakade, Sham, Michael Kearns, and John Langford (2003). "Exploration in Metric State Spaces". In: *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*. ICML'03. Washington, DC, USA, 306–312.

Karr, Alan (2017). *Point processes and their statistical inference*. Routledge.

Khellat, F. and S.A. Yousefi (2006). "The linear Legendre mother wavelets operational matrix of integration and its application". In: *Journal of the Franklin Institute* 343.2, pp. 181–190. ISSN: 0016-0032. DOI: https://doi.org/10.1016/j.jfranklin.2005.11.002.

Khoo, Yuehaw, Jianfeng Lu, and Lexing Ying (July 2020). "Solving parametric PDE problems with artificial neural networks". In: *European Journal of Applied Mathematics* 32.3, 421–435. ISSN: 1469-4425.

Kiani, R., C. J. Cueva, J. B. Reppas, and W. T. Newsome (2014). "Dynamics of neural population responses in prefrontal cortex indicate changes of mind on single trials". In: *Curr Biol* 24.13, pp. 1542–7. ISSN: 1879-0445. DOI: 10.1016/j.cub.2014.05.049. URL: https://www.ncbi.nlm.nih.gov/pubmed/24954050.

Kidger, Patrick, James Morrill, James Foster, and Terry Lyons (2020). *Neural Controlled Differential Equations for Irregular Time Series*. arXiv: 2005.08926 [cs.LG].

Kim, Sanggyun, David Putrino, Soumya Ghosh, and Emery N Brown (2011). "A Granger causality measure for point process models of ensemble neural spiking activity". In: *PLoS computational biology* 7.3, e1001110.

Kispersky, T., G. J. Gutierrez, and E. Marder (2011). "Functional connectivity in a rhythmic inhibitory circuit using Granger causality". In: *Neural Syst Circuits* 1.1, p. 9. ISSN: 2042-1001. DOI: 10.1186/2042-1001-1-9. URL: https://www.ncbi.nlm.nih.gov/pubmed/22330428.

Klafter, J., A. Blumen, and M. F. Shlesinger (Apr. 1987). "Stochastic pathway to anomalous diffusion". In: *Phys. Rev. A* 35 (7), pp. 3081–3085. DOI: 10.1103/PhysRevA.35.3081. URL: https://link.aps.org/doi/10.1103/PhysRevA.35.3081.

Klafter, Joseph and Igor M Sokolov (Aug. 2005). "Anomalous diffusion spreads its wings". In: *Physics World* 18.8, pp. 29–32. DOI: 10.1088/2058-7058/18/8/33. URL: https://doi.org/10.1088%2F2058-7058%2F18%2F8%2F33.

Klages, Rainer, Günter Radons, and Igor M Sokolov (2008). *Anomalous transport: foundations and applications*. John Wiley & Sons.

Kochkov, Dmitrii, Jamie A. Smith, Ayya Alieva, Qing Wang, Michael P. Brenner, and Stephan Hoyer (2021). "Machine learning–accelerated computational fluid dynamics". In: *Proceedings of the National Academy of Sciences* 118.21. ISSN: 0027-8424. DOI: 10.1073/pnas.2101784118.

Kolev, Vasil, Todor Cooklev, and Fritz Keinert (Aug. 2019). "Design of a Simple Orthogonal Multiwavelet Filter by Matrix Spectral Factorization". In: *Circuits, Systems, and Signal Processing* 39.4, 2006–2041.

Koorehdavoudi, Hana, Paul Bogdan, Guopeng Wei, Radu Marculescu, Jiang Zhuang, Rika Wright Carlsen, and Metin Sitti (2017). "Multi-fractal characterization of bacterial swimming dynamics: a case study on real and simulated Serratia marcescens". In: *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 473.2203.

Kopell, Nancy J, Howard J Gritton, Miles A Whittington, and Mark A Kramer (2014). "Beyond the connectome: the dynome". In: *Neuron* 83.6, pp. 1319–1328. ISSN: 0896-6273.

Kovatchev, B. P., M. Breton, C. D. Man, and C. Cobelli (Jan. 2009). "In silico preclinical trials: a proof of concept in closed-loop control of type 1 diabetes". In: *J Diabetes Sci Technol* 3.1, pp. 44–55.

Krause, A., A. Singh, and C. Guestrin (June 2008). "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies". In: *Journal of Machine Learning Research* 9, pp. 235–284.

Krause, Andreas and Volkan Cevher (2010). "Submodular dictionary selection for sparse representation". In: *ICML*, pp. 567–574.

Kronqvist, Jan, David E Bernal, Andreas Lundell, and Ignacio E Grossmann (2019). "A review and comparison of solvers for convex MINLP". In: *Optimization and Engineering* 20.2, pp. 397–455.

Krumin, Michael and Shy Shoham (2010). "Multivariate autoregressive modeling and Granger causality analysis of multiple spike trains". In: *Computational intelligence and neuroscience* 2010, p. 10.

Kulkarni, Jayant E and Liam Paninski (2007). "Common-input models for multiple neural spike-train data". In: *Network: Computation in Neural Systems* 18.4, pp. 375–407.

Kuruoglu, E. E. (Oct. 2001). "Density parameter estimation of skewed /spl alpha/-stable distributions". In: *IEEE Transactions on Signal Processing* 49.10, pp. 2192–2201.

Kyprianou, Andreas E (2006). *Introductory lectures on fluctuations of Lévy processes with applications*. Springer Science & Business Media.

Lanthaler, Samuel, Siddhartha Mishra, and George Em Karniadakis (2021). *Error estimates for DeepOnets: A deep learning framework in infinite dimensions*. arXiv: 2102.09618 [math.NA].

Lefebvre, Julie L., Yifeng Zhang, Markus Meister, Xiaozhong Wang, and Joshua R. Sanes (2008). "Gamma Protocadherins regulate neuronal survival but are dispensable for circuit formation in retina". In: *Development* 135.24, pp. 4141–4151. DOI: 10.1242/dev.027912.

Leike, Jan (2016). *Nonparametric General Reinforcement Learning*. arXiv: 1611.08944 [cs.AI].

Leonenko, Nikolai N., Mark M. Meerschaert, René L. Schilling, and Alla Sikorskii (Oct. 2014). "Correlation Structure of Time-Changed Lévy Processes". In: *Communications in Applied and Industrial Mathematics* 6.1.

Lewicki, Michael S (1998). "A review of methods for spike sorting: the detection and classification of neural action potentials". In: *Network: Computation in Neural Systems* 9.4, R53–R78.

LI, Yuanlu (2010). "Solving a nonlinear fractional differential equation using Chebyshev wavelets". In: *Communications in Nonlinear Science and Numerical Simulation* 15.9, pp. 2284–2292. ISSN: 1007-5704. DOI: https://doi.org/10.1016/j.cnsns.2009.09.020.

Li, Yuanlu and Weiwei Zhao (2010). "Haar wavelet operational matrix of fractional order integration and its applications in solving the fractional order differential equations". In: *Applied Mathematics and Computation* 216.8, pp. 2276–2285. ISSN: 0096-3003. DOI: https://doi.org/10.1016/j.amc.2010.03.063.

Li, Zongyi, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar (2020a). *Fourier Neural Operator for Parametric Partial Differential Equations*. arXiv: 2010.08895 [cs.LG].

– (2020b). *Neural Operator: Graph Kernel Network for Partial Differential Equations*. arXiv: 2003.03485 [cs.LG].

Li, Zongyi, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Andrew Stuart, Kaushik Bhattacharya, and Anima Anandkumar (2020c). "Multipole Graph Neural Operator for Parametric Partial Differential Equations". In: *Advances in Neural Information Processing Systems*. Vol. 33, pp. 6755–6766.

Liang, Hua and Hulin Wu (2008). "Parameter estimation for differential equation models using a framework of measurement error in regression models". In: *Journal of the American Statistical Association* 103.484, pp. 1570–1583.

Litke, A. M. et al. (Aug. 2004). "What does the eye tell the brain?: Development of a system for the large-scale recording of retinal output activity". In: *IEEE Transactions on Nuclear Science* 51.4, pp. 1434–1440.

Lomholt, Michael A, Tobias Ambjörnsson, and Ralf Metzler (2005). "Optimal target search on a fast-folding polymer chain with volume exchange". In: *Physical review letters* 95.26, p. 260603.

Lopes, Fabrício M., Roberto M. Cesar, and Luciano Da F. Costa (2011). "Gene Expression Complex Networks: Synthesis, Identification, and Analysis". In: *Journal of Computational Biology* 18.10, pp. 1353–1367.

Lu, Lu, Pengzhan Jin, and George Em Karniadakis (2020). *DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators*. arXiv: 1910.03193 [cs.LG].

Lu, Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis (Mar. 2021). "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators". In: *Nature Machine Intelligence* 3.3, pp. 218–229. ISSN: 2522-5839.

Luchko, Yuri (2012). "Models of the neutral-fractional anomalous diffusion and their analysis". In: *AIP Conference Proceedings* 1493.1, pp. 626–632. DOI: 10.1063/1.4765552.

Lundstrom, B. N., M. H. Higgs, W. J. Spain, and A. L. Fairhall (Nov. 2008). "Fractional differentiation by neocortical pyramidal neurons". In: *Nature Neuroscience* 11.11, pp. 1335–1342.

Magin, Richard L (2006). *Fractional calculus in bioengineering*. Vol. 2. 6. Begell House Redding.

– (2012). "Fractional calculus in bioengineering". In: *2012 13th International Carpathian Control Conference, ICCC 2012*.

Mainardi, Francesco, Yuri Luchko, and Gianni Pagnini (Feb. 2007). "The fundamental solution of the space-time fractional diffusion equation". In: *arXiv preprint cond-mat/0702419*.

Majeed, Sultan Javed and Marcus Hutter (July 2018). "On Q-learning Convergence for Non-Markov Decision Processes". In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, pp. 2546–2552.

Mandelbrot, B.B., W.H. Freeman, and Company (1983). *The Fractal Geometry of Nature*. Einaudi paperbacks. Henry Holt and Company. ISBN: 9780716711865. URL: https://books.google.com/books?id=SWcPAQAAMAAJ.

Mao, Hongzi, Mohammad Alizadeh, Ishai Menache, and Srikanth Kandula (2016). "Resource management with deep reinforcement learning". In: *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*, pp. 50–56.

Marsili, Matteo, Iacopo Mastromatteo, and Yasser Roudi (2013). "On sampling and modeling complex systems". In: *Journal of Statistical Mechanics: Theory and Experiment* 2013.09, P09003.

McKeown, Ryan, Rodolfo Ostilla-Mónico, Alain Pumir, Michael P. Brenner, and Shmuel M. Rubinstein (2020). "Turbulence generation through an iterative cascade of the elliptical instability". In: *Science Advances* 6.9. DOI: 10.1126/sciadv.aaz2717.

McKinley, Scott A and Hung D Nguyen (2018). "Anomalous diffusion and the generalized Langevin equation". In: *SIAM Journal on Mathematical Analysis* 50.5, pp. 5119–5160.

McLachlan, Geoffrey and Thriyambakam Krishnan (1996). *The EM Algorithm and Extensions*. John Wiley & Sons, New York.

Metzler, Ralf, Walter G. Glöckle, and Theo F. Nonnenmacher (1994). "Fractional model equation for anomalous diffusion". In: *Physica A: Statistical Mechanics and its Applications* 211.1, pp. 13 –24. ISSN: 0378-4371. DOI: https://doi.org/10.

1016/0378-4371(94)90064-7. URL: http://www.sciencedirect.com/science/article/pii/0378437194900647.

Metzler, Ralf, Jae-Hyung Jeon, Andrey G. Cherstvy, and Eli Barkai (2014). "Anomalous diffusion models and their properties: non-stationarity, non-ergodicity, and ageing at the centenary of single particle tracking". In: *Phys. Chem. Chem. Phys.* 16, pp. 24128 –24164.

Metzler, Ralf and Joseph Klafter (2000). "The random walk's guide to anomalous diffusion: a fractional dynamics approach". In: *Physics reports* 339.1, pp. 1–77.

– (2004). "The restaurant at the end of the random walk: recent developments in the description of anomalous transport by fractional dynamics". In: *Journal of Physics A: Mathematical and General* 37.31, R161.

Metzler, Ralf and Theo F. Nonnenmacher (2002). "Space- and time-fractional diffusion and wave equations, fractional Fokker–Planck equations, and physical motivation". In: *Chemical Physics* 284.1. Strange Kinetics, pp. 67 –90. ISSN: 0301-0104. DOI: https://doi.org/10.1016/S0301-0104(02)00537-2. URL: http://www.sciencedirect.com/science/article/pii/S0301010402005372.

Meyer, Y., R. Coifman, and D. Salinger (1997). *Wavelets: Calderón-Zygmund and Multilinear Operators*. Cambridge Studies in Advanced Mathematics. Cambridge University Press. ISBN: 9780521420013.

Meyer, Y. and D.H. Salinger (1992). *Wavelets and Operators: Volume 1*. Cambridge Studies in Advanced Mathematics. Cambridge University Press. ISBN: 9780521458696.

Micciche, Salvatore (2009). "Modeling long-range memory with stationary Markovian processes". In: *Physical Review E* 79.3, p. 031116.

Moon, F. (1992). *Chaotic and Fractal Dynamics: An Introduction for Applied Scientists and Engineers*. Wiley.

Morgado, Rafael, Fernando A. Oliveira, G. George Batrouni, and Alex Hansen (Aug. 2002). "Relation between Anomalous and Normal Diffusion in Systems with Memory". In: *Phys. Rev. Lett.* 89 (10), p. 100601. DOI: 10.1103/PhysRevLett.89.100601. URL: https://link.aps.org/doi/10.1103/PhysRevLett.89.100601.

Müller, TG and Jens Timmer (2004). "Parameter identification techniques for partial differential equations". In: *International Journal of Bifurcation and Chaos* 14.06, pp. 2053–2060.

Müller, Thorsten G and Jens Timmer (2002). "Fitting parameters in partial differential equations from partially observed noisy data". In: *Physica D: Nonlinear Phenomena* 171.1-2, pp. 1–7.

Murphy, Kevin P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press. ISBN: 0262018020, 9780262018029.

Nagabandi, Anusha, Gregory Kahn, Ronald S. Fearing, and Sergey Levine (May 2018). "Neural Network Dynamics for Model-Based Deep Reinforcement Learning with Model-Free Fine-Tuning". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. DOI: 10.1109/icra.2018.8463189. URL: http://dx.doi.org/10.1109/ICRA.2018.8463189.

Nemhauser, G., L. Wolsey, and M. Fisher (1978). "An analysis of approximations for maximizing submodular set functions". In: *Mathematical Programming* 14.1, pp. 265–294.

Nigam, Sunny et al. (2016). "Rich-Club Organization in Effective Connectivity among Cortical Neurons". In: *Journal of Neuroscience* 36.3, pp. 670–684.

Nolan, J.P. (2003). *Stable Distributions: Models for Heavy-Tailed Data*. Springer New York. ISBN: 9780817641597.

Norden, Andrew D and Hal Blumenfeld (2002). "The role of subcortical structures in human epilepsy". In: *Epilepsy & Behavior* 3.3, pp. 219–231.

Okatan, Murat, Matthew A Wilson, and Emery N Brown (2005). "Analyzing functional connectivity using a network likelihood model of ensemble neural spiking activity". In: *Neural computation* 17.9, pp. 1927–1961.

Oldham, K.B. and J. Spanier (2006). *The Fractional Calculus: Theory and Applications of Differentiation and Integration to Arbitrary Order*. Dover books on mathematics. Dover Publications.

Oliveira, Fernando A., Rogelma M. S. Ferreira, Luciano C. Lapas, and Mendeli H. Vainstein (2019). "Anomalous Diffusion: A Basic Mechanism for the Evolution of Inhomogeneous Systems". In: *Frontiers in Physics* 7, p. 18. ISSN: 2296-424X. DOI: 10.3389/fphy.2019.00018. URL: https://www.frontiersin.org/article/10.3389/fphy.2019.00018.

Otoom, Mwaffaq, Hussam Alshraideh, Hisham M Almasaeid, Diego López-de Ipiña, and José Bravo (2013). "A Real-Time Insulin Injection System". In: *International Workshop on Ambient Assisted Living*. Springer, pp. 120–127.

Ovarlez, Guillaume, Anh Vu Nguyen Le, Wilbert J. Smit, Abdoulaye Fall, Romain Mari, Guillaume Chatté, and Annie Colin (2020). "Density waves in shear-thickening suspensions". In: *Science Advances* 6.16. DOI: 10.1126/sciadv.aay5589.

Pal, Akash and Lin Tian (2020). "Imaging voltage and brain chemistry with genetically encoded sensors and modulators". In: *Current Opinion in Chemical Biology* 57, pp. 166–176. ISSN: 1367-5931.

Palla, G., L. Lovász, and T. Vicsek (2010). "Multifractal network generator". In: *Proc Natl Acad Sci U S A* 107.17, pp. 7640–5. ISSN: 1091-6490. DOI: 10.1073/pnas.0912983107. URL: https://www.ncbi.nlm.nih.gov/pubmed/20385847.

Palmieri, Benoit, Yony Bresler, Denis Wirtz, and Martin Grant (2015). "Multiple scale model for cell migration in monolayers: Elastic mismatch between cells enhances motility". In: *Scientific reports* 5, p. 11745.

Papo, David (2014). "Functional significance of complex fluctuations in brain activity: from resting state to cognitive neuroscience". In: *Frontiers in Systems Neuroscience* 8, p. 112. ISSN: 1662-5137. DOI: 10.3389/fnsys.2014.00112. URL: https://www.frontiersin.org/article/10.3389/fnsys.2014.00112.

Parr, T., D. Markovic, S. J. Kiebel, and K. J. Friston (2019). "Neuronal message passing using Mean-field, Bethe, and Marginal approximations". In: *Sci Rep* 9.1, p. 1889. ISSN: 2045-2322. DOI: 10.1038/s41598-018-38246-3. URL: https://www.ncbi.nlm.nih.gov/pubmed/30760782.

Pasqualetti, F., S. Zampieri, and F. Bullo (Mar. 2014). "Controllability Metrics, Limitations and Algorithms for Complex Networks". In: *IEEE Transactions on Control of Network Systems* 1.1, pp. 40–52.

Patel, Ravi G., Nathaniel A. Trask, Mitchell A. Wood, and Eric C. Cyr (2021). "A physics-informed operator regression framework for extracting data-driven continuum models". In: *Computer Methods in Applied Mechanics and Engineering* 373, p. 113500. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2020.113500.

Pattern, Duda RO Hart PE (1973). *Classification and Scene Analysis*. John Wiley and Sons, New York.

Perez, Julien and Tomi Silander (2017). *Non-Markovian Control with Gated End-to-End Memory Policy Networks*. arXiv: 1705.10993 [stat.ML].

Pernice, V., B. Staude, S. Cardanobile, and S. Rotter (2012). "Recurrent interactions in spiking networks with arbitrary topology". In: *Phys Rev E Stat Nonlin Soft Matter Phys* 85.3 Pt 1, p. 031916. ISSN: 1550-2376. DOI: 10.1103/PhysRevE.85.031916. URL: https://www.ncbi.nlm.nih.gov/pubmed/22587132.

Peters, Charles and William A Coberly (1976). "The numerical evaluation of the maximum-likelihood estimate of mixture proportions". In: *Communications in Statistics-Theory and Methods* 5.12, pp. 1127–1135.

Petersen, Rasmus S, Stefano Panzeri, and Mathew E Diamond (2001). "Population coding of stimulus location in rat somatosensory cortex". In: *Neuron* 32.3, pp. 503–514. ISSN: 0896-6273.

Raiffa, Howard (1974). *Applied statistical decision theory*.

Raissi, M., P. Perdikaris, and G.E. Karniadakis (2019). "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: *Journal of Computational Physics* 378, pp. 686–707. ISSN: 0021-9991.

Rasmussen, C.E. and C.K.I. Williams (2006). *Gaussian Processes for Machine Learning*. Adaptative computation and machine learning series. University Press Group Limited.

Razzaghi, Mohsen and Samira Yousefi (Apr. 2001). "The Legendre wavelets operational matrix of integration". In: *International Journal of Systems Science* 32, pp. 495–502. DOI: 10.1080/00207720120227.

Rigotti, M., O. Barak, M. R. Warden, X. J. Wang, N. D. Daw, E. K. Miller, and S. Fusi (2013). "The importance of mixed selectivity in complex cognitive tasks". In: *Nature* 497.7451, pp. 585–90. ISSN: 1476-4687. DOI: 10.1038/nature12160. URL: https://www.ncbi.nlm.nih.gov/pubmed/23685452.

Rihan, Fathalla A, Dumitru Baleanu, S Lakshmanan, and R Rakkiyappan (2014). "On fractional SIRC model with salmonella bacterial infection". In: *Abstract and Applied Analysis*. Vol. 2014. Hindawi Publishing Corporation.

Rolls, Edmund T and Alessandro Treves (2011). "The neuronal encoding of information in the brain". In: *Progress in neurobiology* 95.3, pp. 448–490. ISSN: 0301-0082.

Ross, Stephane and J. Andrew Bagnell (2012). *Agnostic System Identification for Model-Based Reinforcement Learning*. arXiv: 1203.1007 [cs.LG].

Réjou-Méchain, M. et al. (2021). "Unveiling African rainforest composition and vulnerability to global change". In: *Nature* 593.7857, pp. 90–94. ISSN: 1476-4687. DOI: 10.1038/s41586-021-03483-6. URL: https://www.ncbi.nlm.nih.gov/pubmed/33883743.

Saichev, Alexander I. and George M. Zaslavsky (Dec. 1997). "Fractional kinetic equations: solutions and applications". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 7.4, pp. 753–764.

Sato, K., S. Ken-Iti, B. Bollobas, Cambridge University Press, A. Katok, W. Fulton, F. Kirwan, and P. Sarnak (1999). *Lévy Processes and Infinitely Divisible Distributions*. Cambridge Studies in Advanced Mathematics. Cambridge University Press. ISBN: 9780521553025. URL: https://books.google.com/books?id=tbZPLquJjSoC.

Scalas, Enrico (2006). "The application of continuous-time random walks in finance and economics". In: *Physica A: Statistical Mechanics and its Applications* 362.2, pp. 225–239.

Scalas, Enrico, Rudolf Gorenflo, and Francesco Mainardi (2000). "Fractional calculus and continuous-time finance". In: *Physica A: Statistical Mechanics and its Applications* 284.1-4, pp. 376–384.

Schalk, G., D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw (June 2004). "BCI2000: a general-purpose brain-computer interface (BCI) system". In: *IEEE Transactions on Biomedical Engineering* 51.6, pp. 1034–1043.

Shimono, Masanori and John M. Beggs (2015). "Functional Clusters, Hubs, and Communities in the Cortical Microconnectome". In: *Cerebral Cortex* 25.10, pp. 3743–3757.

Shlesinger, M. F., B. J. West, and J. Klafter (Mar. 1987). "Lévy dynamics of enhanced diffusion: Application to turbulence". In: *Phys. Rev. Lett.* 58 (11), pp. 1100–1103. DOI: 10.1103/PhysRevLett.58.1100.

Sierociuk, Dominik and Andrzej Dzieliński (2006). "Fractional Kalman filter algorithm for states, parameters and order of fractional system estimation". In: *International Journal of Applied Mathematics and Computer Science*, pp. 129–140.

Silverman, B. W., J. C. Vassilicos, and Nick Kingsbury (1999). "Image processing with complex wavelets". In: *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 357.1760, pp. 2543–2560. DOI: 10.1098/rsta.1999.0447.

Smallwood, Richard D. and Edward J. Sondik (1973). "The Optimal Control of Partially Observable Markov Processes Over a Finite Horizon". In: *Operations Research* 21.5, pp. 1071–1088.

Sokolov, Igor M. (2012). "Models of anomalous diffusion in crowded environments". In: *Soft Matter* 8, pp. 9043 –9052.

Sporns, Olaf (2002). "Network analysis, complexity, and brain function". In: *Complexity* 8.1, pp. 56–60. ISSN: 1076-2787.

– (2011). "The human connectome: a complex network". In: *Annals of the New York Academy of Sciences* 1224.1, pp. 109–125. ISSN: 0077-8923.

– (2012). *Discovering the human connectome*. Discovering the human connectome. Cambridge, MA, US: MIT Press, pp. xii, 232–xii, 232. ISBN: 0-262-01790-3 (Hardcover); 978-0-262-01790-9 (Hardcover).

Srivastava, K. H., C. M. Holmes, M. Vellema, A. R. Pack, C. P. Elemans, I. Nemenman, and S. J. Sober (2017). "Motor control by precisely timed spike patterns". In: *Proc Natl Acad Sci U S A* 114.5, pp. 1171–1176. ISSN: 1091-6490. DOI: 10.1073/pnas.1611734114. URL: https://www.ncbi.nlm.nih.gov/pubmed/28100491.

Stoer, J., R. Bartels, W. Gautschi, R. Bulirsch, and C. Witzgall (2002). *Introduction to Numerical Analysis*. Texts in Applied Mathematics. Springer New York. ISBN: 9780387954523.

Strehl, Alexander L, Lihong Li, and Michael L Littman (2009). "Reinforcement Learning in Finite MDPs: PAC Analysis." In: *Journal of Machine Learning Research* 10.11.

Summers, Tyler H., Fabrizio L. Cortesi, and John Lygeros (2016). "On Submodularity and Controllability in Complex Dynamical Networks". In: *IEEE Transactions on Control of Network Systems* 3, pp. 91–101.

Sutton, Richard S, Doina Precup, and Satinder Singh (1999). "Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning". In: *Artificial intelligence* 112.1-2, pp. 181–211.

Sviridenko, M. (2004). "A note on maximizing a submodular set function subject to a knapsack constraint". In: *Operations Research Letters* 32.1, pp. 41–43.

Taherkhani, Aboozar, Ammar Belatreche, Yuhua Li, Georgina Cosma, Liam P Maguire, and T Martin McGinnity (2020). "A review of learning in biologically plausible spiking neural networks". In: *Neural Networks* 122, pp. 253–272. ISSN: 0893-6080.

Takayasu, Hideki (1990). *Fractals in the physical sciences*. Manchester University Press.

Tarasov, Vasily E. (2019). *Handbook of Fractional Calculus with Applications: Applications in Physics (Part 2)*. Berlin; Boston:De Gruyter.

Tejedor, Miguel, Ashenafi Zebene Woldaregay, and Fred Godtliebsen (2020). "Reinforcement learning application in diabetes blood glucose control: A systematic review". In: *Artificial Intelligence in Medicine*, p. 101836.

Thiel, Felix, Franziska Flegel, and Igor M Sokolov (2013). "Disentangling sources of anomalous diffusion". In: *Physical review letters* 111.1, p. 010601.

Thurner, S., C. Windischberger, E. Moser, P. Walla, and M. Barth (2003). "Scaling laws and persistence in human brain activity". In: *Physica A: Statistical Mechanics and its Applications* 326.3-4, pp. 511–521.

Timme, Nicholas, Shinya Ito, Maxym Myroshnychenko, Fang-Chin Yeh, Emma Hiolski, Pawel Hottowy, and John M. Beggs (Dec. 2014). "Multiplex Networks of Cortical and Hippocampal Neurons Revealed at Different Timescales". In: *PLOS ONE* 9.12, pp. 1–43.

Timme, Nicholas M., Shinya Ito, Maxym Myroshnychenko, Sunny Nigam, Masanori Shimono, Fang-Chin Yeh, Pawel Hottowy, Alan M. Litke, and John M. Beggs (May

2016). “High-Degree Neurons Feed Cortical Computations”. In: *PLOS Computational Biology* 12.5, pp. 1–31.

Timoshenko, S. (1983). *History of Strength of Materials: With a Brief Account of the History of Theory of Elasticity and Theory of Structures*. Dover Civil and Mechanical Engineering Series. Dover Publications. ISBN: 9780486611877.

Todorov, Emanuel, Tom Erez, and Yuval Tassa (2012). “Mujoco: A physics engine for model-based control”. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 5026–5033.

Tolić-Nørrelykke, Iva Marija, Emilia-Laura Munteanu, Genevieve Thon, Lene Oddershede, and Kirstine Berg-Sørensen (2004). “Anomalous diffusion in living yeast cells”. In: *Physical Review Letters* 93.7, p. 078102.

Towns, J. et al. (Sept. 2014). “XSEDE: Accelerating Scientific Discovery”. In: *Computing in Science & Engineering* 16.5, pp. 62–74. ISSN: 1521-9615. DOI: 10.1109/MCSE.2014.80. URL: doi.ieeecomputersociety.org/10.1109/MCSE.2014.80.

Trousdale, J., Y. Hu, E. Shea-Brown, and K. Josić (2012). “Impact of network structure and cellular response on spike time correlations”. In: *PLoS Comput Biol* 8.3, e1002408. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1002408. URL: https://www.ncbi.nlm.nih.gov/pubmed/22457608.

Truccolo, Wilson, Uri T Eden, Matthew R Fellows, John P Donoghue, and Emery N Brown (2005). “A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects”. In: *Journal of neurophysiology* 93.2, pp. 1074–1089.

Turcott, R. G. and M. C. Teich (1996). “Fractal character of the electrocardiogram: Distinguishing heart-failure and normal patients”. In: *Annals of Biomedical Engineering* 24.2, pp. 269–293.

Vallin, Robert W. (2013). “Cantor Sets and Continued Fractions”. In: *The Elements of Cantor Sets*, pp. 51–66. DOI: https://doi.org/10.1002/9781118548745.ch4. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118548745.ch4.

Voss, Henning U, Paul Kolodner, Markus Abel, and Jürgen Kurths (1999). “Amplitude equations from spatiotemporal binary-fluid convection data”. In: *Physical review letters* 83.17, p. 3422.

Vázquez-Guardado, Abraham, Yiyuan Yang, Amay J Bandodkar, and John A Rogers (2020). “Recent advances in neurotechnologies with broad potential for neuroscience research”. In: *Nature neuroscience*, pp. 1–15. ISSN: 1546-1726.

Wang, Le-Zhi, Ri-Qi Su, Zi-Gang Huang, Xiao Wang, Wen-Xu Wang, Celso Grebogi, and Ying-Cheng Lai (2016). "A geometrical approach to control and controllability of nonlinear dynamical networks". In: *Nature Communications* 7.

Wang, Lifeng, Yunpeng Ma, and Zhijun Meng (2014). "Haar wavelet method for solving fractional partial differential equations numerically". In: *Applied Mathematics and Computation* 227, pp. 66–76. ISSN: 0096-3003. DOI: https://doi.org/10.1016/j.amc.2013.11.004.

Wang, Q., J. Xie, P. Molenaar, and J. Ulbrecht (2015). "Model predictive control for type 1 diabetes based on personalized linear time-varying subject model consisting of both insulin and meal inputs: In Silico evaluation". In: *2015 American Control Conference (ACC)*, pp. 5782–5787.

Wang, Sifan, Hanwen Wang, and Paris Perdikaris (2021). *Learning the solution operator of parametric partial differential equations with physics-informed DeepOnets*. arXiv: 2103.10974 [cs.LG].

Wang, Yanxin and Qibin Fan (2012). "The second kind Chebyshev wavelet method for solving fractional differential equations". In: *Applied Mathematics and Computation* 218.17, pp. 8592–8601. ISSN: 0096-3003. DOI: https://doi.org/10.1016/j.amc.2012.02.022.

Werner, G. (2010). "Fractals in the nervous system: Conceptual implications for theoretical neuroscience". In: *Frontiers in Physiology* 1, p. 15.

West, Bruce (2010). "Fractal Physiology and the Fractional Calculus: A Perspective". In: *Frontiers in Physiology* 1, p. 12. ISSN: 1664-042X. DOI: 10.3389/fphys.2010.00012. URL: https://www.frontiersin.org/article/10.3389/fphys.2010.00012.

Wigner, Eugene P. (1960). "The unreasonable effectiveness of mathematics in the natural sciences". In: *Communications on Pure and Applied Mathematics*.

Wilson, Matthew A and Bruce L McNaughton (1993). "Dynamics of the hippocampal ensemble code for space". In: *Science* 261.5124, pp. 1055–1058.

Wu, C. F. Jeff (Mar. 1983). "On the Convergence Properties of the EM Algorithm". In: *Ann. Statist.* 11.1, pp. 95–103.

Xie, Jinyu (2018). *Simglucose v0.2.1*. URL: https://github.com/jxx123/simglucose.

Xiong, Shiying, Xingzhe He, Yunjin Tong, and Bo Zhu (2020). *Neural Vortex Method: from Finite Lagrangian Particles to Infinite Dimensional Eulerian Dynamics*. arXiv: 2006.04178 [physics.comp-ph].

Xue, Yuankun and Paul Bogdan (Apr. 2017). "Constructing Compact Causal Mathematical Models for Complex Dynamics". In: *2017 ACM/IEEE 8th International Conference on Cyber-Physical Systems (ICCPS)*, pp. 97–108.

Xue, Yuankun, Sergio Pequito, Joana R. Coelho, Paul Bogdan, and George J. Pappas (2016). "Minimum Number of Sensors to Ensure Observability of Physiological Systems: a Case Study". In: *Allerton*.

Xue, Yuankun, Saul Rodriguez, and Paul Bogdan (2016). "A Spatio-Temporal Fractal Model for a CPS Approach to Brain-Machine-Body Interfaces". In: *DATE*.

Xun, Xiaolei, Jiguo Cao, Bani Mallick, Arnab Maity, and Raymond J Carroll (2013). "Parameter estimation of partial differential equation models". In: *Journal of the American Statistical Association* 108.503, pp. 1009–1020.

Yang, R. and P. Bogdan (2020). "Controlling the Multifractal Generating Measures of Complex Networks". In: *Sci Rep* 10.1, p. 5541. ISSN: 2045-2322. DOI: 10.1038/s41598-020-62380-6. URL: https://www.ncbi.nlm.nih.gov/pubmed/32218468.

Yang, Ruochen, Gaurav Gupta, and Paul Bogdan (2019). "Data-Driven Perception of Neuron Point Process with Unknown Unknowns". In: *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*. New York, NY, USA: Association for Computing Machinery, 259–269. ISBN: 9781450362856. DOI: 10.1145/3302509.3311056.

Yin, Chenzhong, Gaurav Gupta, and Paul Bogdan (2020). "Discovering Laws from Observations: A Data-Driven Approach". In: *Dynamic Data Driven Applications Systems*. Ed. by Frederica Darema, Erik Blasch, Sai Ravela, and Alex Aved. Cham: Springer International Publishing, pp. 302–310. ISBN: 978-3-030-61725-7.

You, Wen-Peng and Maciej Henneberg (2016). "Type 1 diabetes prevalence increasing globally and regionally: the role of natural selection and life expectancy at birth". In: *BMJ open diabetes research and care* 4.1.

Zhang, Yi-Feng, Hiroki Asari, and Markus Meister (2014). *Multi-electrode recordings from retinal ganglion cells*. CRCNS.org. URL: http://dx.doi.org/10.6080/K0RF5RZT.

Zheng, Guanjie, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li (2018). "DRN: A deep reinforcement learning framework for news recommendation". In: *Proceedings of the 2018 World Wide Web Conference*, pp. 167–176.

Zhu, Li and Yanxin Wang (Aug. 2017). "Solving fractional partial differential equations by using the second Chebyshev wavelet operational matrix method". In: *Nonlinear Dynamics* 89.3, pp. 1915–1925. ISSN: 1573-269X. DOI: 10.1007/s11071-017-3561-7.

Zhu, Yinhao and Nicholas Zabaras (Aug. 2018). "Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification". In: *Journal of Computational Physics* 366, 415–447. ISSN: 0021-9991.

Znaidi, Mohamed Ridha, Gaurav Gupta, Kamiar Asgari, and Paul Bogdan (2020). "Identifying Arguments of Space-Time Fractional Diffusion: Data-Driven Approach". In: *Frontiers in Applied Mathematics and Statistics* 6, p. 14. ISSN: 2297-4687. DOI: 10.3389/fams.2020.00014.