UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

DEPARTAMENT OF INFORMATION ENGENEERING

MASTER'S DEGREE IN BIOENGINEERING FOR NEUROSCIENCE

Inference of cortical circuit parameters based on simulation of biophysically complex brain models

Supervisor
Prof. Atzori Manfredo

Supervisor
Prof. Pablo Martínez Cañada

Author
Sandron Alessandro

ACADEMIC YEAR 2023/2024
8 October

Apes. Together. Strong.

Abstract

In the field of neuroscience, analyzing brain signals and identifying relevant patterns pose significant challenges. The brain's complex structure and non-linear nature makes it difficult to classify and detect patterns within raw brain signals. Extracellular electrophysiology recordings capturing the activity of neuronal populations, e.g., Local Field Potentials (LFPs), have offered important insights into cortical dynamics. Yet there is still a lack of clarity about how features and characteristics of these extracellular potentials relate to the properties and function of the underlying neural populations. Mechanistic models combined with simulation-based inference (SBI) algorithms have emerged as an effective strategy for developing predictive tools that fit well with available empirical data and can be used to predict key parameters that describe neural activity. Numerous SBI techniques rely on summary statistics or interpretable features to approximate the likelihood or posterior. However, at present, a significant challenge is assessing how each feature impacts the SBI model's predictions. Here, it was developed an approach to determine feature importance in the context of cortical circuit parameter inference. it was first created a dataset that includes a million distinct simulations from a spiking cortical microcircuit model of recurrently connected excitatory and inhibitory populations. Biophysics-based causal filters were coupled with spikes to generate realistic LFP data. Then, it was extracted a set of meaningful features from simulated LFP data that were used to train an SBI algorithm. To evaluate feature importance, differents tecnichs were used like mutual information (MI), principal component analisis (PCA) and SHAP values, a prominent tool in machine learning for interpreting the contribution of eachfeature to the prediction outcomes. The results demonstrate the effectiveness of this approach in identifying the most critical features for inferring the parameters of a recurrent cortical circuit model based on electrophysiological data. These results were presented at the Brain Informatics 2024 International Conference.

Contents

List of Figures

Chapter 1

Introduction

1.1 General Background

1.1.1 General information on computational neuroscience

Computational neuroscience is a specialized field within neuroscience focused on theoretically examining the brain. Its objective is to reveal the principles and mechanisms governing the development, arrangement, processing of information, and cognitive capacities of the nervous system. This domain emerged from recognizing the imperative of interdisciplinary cooperation to grasp the intricacies of the brain. Scholars from various disciplines collaborate to tackle fundamental inquiries concerning brain function, organization, evolution, and pathology. Key investigations include understanding how the brain functions, its biological mechanisms, organization, principles of information processing, evolution, changes throughout the lifespan, effects of injury, possibilities for rehabilitation, and treatments for diseases. Computational neuroscience furnishes theoretical frameworks and analytical instruments to simulate brain processes, aiding in the interpretation of experimental evidence and advancing our comprehension of the nervous system.

It represents a multifaceted realm nestled within neuroscience, serving as a critical avenue for untangling the intricate processes of the brain. Embracing an array of methodologies and techniques, it endeavors to craft and validate hypotheses concerning the functional dynamics of the brain. Neural computation revolves around the formulation and scrutiny of models, serving as theoretical scaffolds for grasping the essence of brain function.

These models, often complex and beyond analytical tractability, are simulated using computers to simulate neural processes and interactions. The use of computational tools allows researchers to conduct carefully designed numerical experiments, enabling comparisons between model predictions and experimental data. However, computational neuroscience does not solely rely on numerical techniques; analytical studies are also employed to provide deeper insights into model features and underlying phenomena [1].

Experimental data from various disciplines within neuroscience, such as neuroanatomy, neurophysiology, and psychology, provide crucial input for computational neuroscience. Neuroanatomy offers insights into the morphology and functional connectivity of brain structures, while neurophysiology investigates the behavior of individual neurons. Additionally, psychology contributes behavioral effects through psychophysical experiments.

Integrating experimental results from various levels of research is crucial for creating unified models of brain function. Through the fusion of computational analyses and empirical data, computational neuroscience aims to clarify the workings of the brain, formulating hypotheses, building models, and validating them against experimental findings. Ultimately, the objective is to attain a thorough comprehension of brain function through interdisciplinary teamwork and meticulous scientific investigation [2].

1.1.2 The Significance of Brain Signal Analysis

Analysis of brain signals forms one of the most important in neuroscience research; it offers insights into the inner working of the human mind that are second to none. Researchers can, through the dissection of intricate neural activity patterns, disentangle the layers of complexity shrouding brain function and dysfunction, leading toward breakthroughs in discoveries of cognitive science and clinical neurology.

Brain signal analysis expands our understanding of how the brain functions. It allows scientists to explore the activity of neurons and circuits in more detail. Techniques such as electroencephalography (EEG), magnetoencephalography (MEG) and functional magnetic resonance imaging (fMRI) broaden our view of brain activity. These signals reveal how the brain processes information, focuses attention, recalls memories, and makes decisions.

Additionally, studying brain signals is vital for understanding the causes of neurological disorders. By comparing how the brain works in healthy people to those with conditions like epilepsy, Alzheimer's, or schizophrenia, scientists can pinpoint unusual brain patterns linked to these diseases. This insight not only helps us grasp what's happening in these disorders but also opens doors for new ways to diagnose and treat them.[3].

For years, EEG, MEG, and other brain imaging methods have been crucial for understanding how the brain works. EEG gives us a direct look at the brain's electrical activity, helping us map out things like neural rhythms, how the brain responds to events, and how different areas communicate. Similarly, MEG shows us the magnetic fields produced by brain cells firing, offering another layer of insight alongside EEG. By combining EEG and MEG data, scientists can get a richer picture of how the brain operates, revealing detailed patterns of activity across both space and time.[4].

This study is focused on Local field potentials (LFP), that offer another approach to study neural circuits through correlated neural activity. LFP are measured by placing an electrode within or directly on the surface of the brain. Unlike action potentials which are generated by individual neurons, LFP predominantly measure synaptic potentials pooled across groups of neurons near the recording electrode and are closely linked to trans-membrane currents. As a result, LFP signals reflect the presence of correlations in the activity of many neurons. Crucially, while LFP signals are related to EEG signals measured at the scalp because both measure voltages from pooled synaptic potentials, LFP signals can reveal quite different activity patterns than EEG signals.

1.2 SBI model (Simulation Based-Inference)

Simulation-Based Inference (SBI) has emerged as a powerful paradigm in computational neuroscience, revolutionizing the way we analyze and interpret brain signals. Initially, traditional statistical methods struggled to capture the complex dynamics of neural systems, leading researchers to explore alternative approaches.

In the early 2000s, the advent of computational models enabled researchers to simulate large-scale neural networks and generate synthetic brain signals resembling those observed experimentally. In recent years, significant strides have been made in three key areas. Firstly, the machine learning (ML) revolution has empowered us to engage with higher-dimensional data, notably advancing inference quality through deep learning breakthroughs. Secondly, active learning techniques have systematically enhanced sampling efficiency, enabling us to address increasingly intricate and computationally demanding simulations. Lastly, the deep integration of automatic differentiation and probabilistic programming into simulation codes, coupled with leveraging insights extracted from the simulator to enrich training data, has yielded substantial improvements in inference quality and sample efficiency. However, despite these advancements, the fundamental approach to simulation-based inference remains largely unchanged. Simulators are still perceived as black boxes, receiving parameters as input and generating data as output, maintaining a distinct separation from the inference engine. However, inferring the underlying parameters of these models from empirical data remained a formidable challenge.

Statistical inference operates within a statistical model framework, where in simulation-based inference (SBI), the simulator itself defines the statistical model. In SBI, a simulator is a computer program that takes a parameter vector as input, generates internal states or latent variables

$$z \sim p(z|\theta, z)$$

and outputs a data vector

$$x \sim p(x|\theta, z)$$

SBI addresses this challenge by employing advanced simulation techniques and Bayesian statistical inference to estimate model parameters directly from observed brain signals. Through iterative comparisons between simulated and experimental data, SBI refines its parameter estimates, offering insights into the underlying mechanisms of brain function.

Simulation-Based Inference (SBI) represents a paradigm shift in computational neuroscience, offering a unique avenue for unraveling the mysteries of brain function. By harnessing the power of computational modeling, SBI allows researchers to construct virtual brain networks that faithfully mimic the intricate dynamics of the human brain. These models not only capture the complex interplay between neurons but also simulate the emergence of brain signals.

By careful calibration to empirical data, SBI empowers researchers to decode the hidden mechanisms that govern neural activity and to infer the rich tapestry of brain signals that underpin cognition and behavior. Furthermore, SBI acts as a catalyst for hypothesis generation and testing by allowing researchers to navigate a huge landscape of possible neural architectures and functional configurations. Through iterative refinement and validation of computational models to experimental observations, SBI promotes a deeper understanding of brain functions, enabling transformative insights into the workings of the mind and new therapeutic interventions against neurological disorders. In all respects, SBI ushers in a new epoch of discovery in neuroscience, where by computational modeling and empirical investigation come together to illuminate the intricacies of brain mechanisms and unlock its full potential.[5].

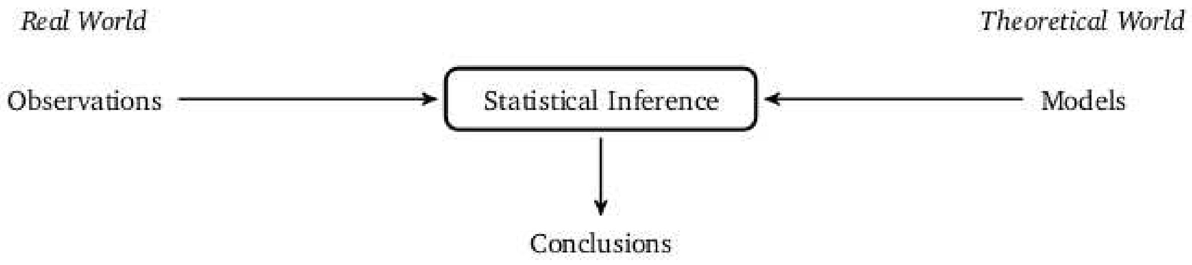


Figure 1.1: Statistical inference bridges the real world, one in which observations are made, with the theoretical world, inhabited by models.[6]

1.2.1 Statistical inference

The advancement of detailed computational models in neuroscience presents a challenge in aligning them with increasingly complex observations. Bridging this gap necessitates a method to discern which models best fit observed data and existing knowledge. Statistical inference emerges as a crucial tool in this endeavor, acting as the conduit between real-world observations and theoretical models. By employing probability as its language, statistical inference accommodates the inherent uncertainty and variability present in scientific investigations[6].

Observations are always mediated through measurement probes and experimental conditions, leading to inherent variability. Likewise, the specification of models is often incomplete, and systems may exhibit stochastic behavior at microscopic levels. Statistical methods enable principled decision-making amidst uncertainty, offering a framework to assess model consistency with observed data.

However, conventional statistical inference methods can impose stringent restrictions on viable model configurations, often requiring numerical evaluation of likelihood functions to gauge their consistency with observations. This dilemma underscores the tension between two approaches in model design within neuroscience.

On one hand, incorporating statistical considerations into model development creates a close feedback loop between data and theory, fostering a nuanced understanding of the underlying causal mechanisms. However, this approach imposes stringent constraints on the types of models that can be utilized, potentially limiting the scope of insights gained.

Conversely, opting for interpretable mechanistic models, particularly those relying on high-fidelity computer simulations, offers a deeper understanding of underlying processes. Yet, this approach often sacrifices the ability to perform likelihood-based statistical inference, making it challenging to constrain these models using observed data.

Thus, striking a balance between statistical rigor and mechanistic fidelity remains a significant challenge in advancing computational neuroscience[7].

1.2.2 Difference between Frequentism and Bayesianism

Statistical inference enables us to address the question of which model configurations align with observed data. Traditionally, all types of statistical inferences rely on the numerical evaluation of the likelihood function.The likelihood function yields a value that rises for compatible model configurations and declines for incompatible ones. In other words, it articulates the relative coherence of each model configuration with the observation x. In statistical theory, two fundamental paradigms emerge: frequentist and Bayesian inference. Both approaches frequently rely on numerical evaluation of the likelihood function

Frequentism, also known as the classical interpretation of probability, defines probability in terms of observed frequencies. According to frequentist principles, the probability $P(x)$ of an event x is the relative frequency of its occurrence in an infinite sequence of repeated trials:

$$P(x) = \lim_{n\to\infty} \frac{n_x}{n}$$

where n_x denotes the number of times event x occurs in n trials. Frequentist methods rely on sampling theory and the properties of estimators for statistical inference.

In contrast, Bayesianism adopts a subjective interpretation of probability. According to Bayesian principles, probability reflects the degree of belief or uncertainty associated with events. Central to Bayesian inference is Bayes' theorem, which provides a systematic framework for updating prior beliefs based on observed evidence:

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{\int p(x|\theta)p(\theta)d\theta}$$

where $p(\theta|x)$ denotes the posterior distribution, $p(x|\theta)$ represents the likelihood function, $p(\theta)$ is the prior distribution, and $p(x)$ is the marginal likelihood or evidence[8].

While frequentism emphasizes objective measurements and observed frequencies, Bayesianism incorporates subjective beliefs and the iterative update of prior knowledge. Bayesian inference is akin to a learning process, where prior beliefs about model configurations are updated in light of observed data. The posterior distribution represents the updated beliefs about model configurations consistent with the observed data.

Moreover, in Bayes approach we don't have to choose a probability distribution over the model configuration space arbitrarily, there is a natural procedure for updating a prior distribution that encodes our domain expertise about the model into a posterior distribution that identifies those model configurations consistent with both our domain expertise and the observed data. Inferences, decisions, and predictions all follow from natural probabilistic operations.

1.2.3 Intractable likelihood

Statistical analysis and data inference are often driven by models that rely on the assumption of a known and computable likelihood function. However, in many real-world scenarios, the likelihood may be inestimable or difficult to compute, leading to a series of challenges in the statistical inference process.

The notion of "intractable likelihood" refers to situations where the mathematical form of the likelihood is known, but its direct computation is computationally prohibitive or impractical. This can stem from various factors such as model complexity, data dimensionality, or the presence of unobservable latent variables.

In contexts like these, using traditional approaches based on direct likelihood computation becomes ineffective or even impossible. However, there are several strategies that can be adopted to address the issue of unavailable likelihood.

The core challenge in simulation-based inference is the intractability of the likelihood function defined by the simulator. This function involves integrating over all possible trajectories in the latent space, making explicit computation infeasible.

$$p(\theta|x) = \int p(x,z|\theta)dz$$

For simulators in real-world scenarios with extensive latent spaces, explicitly computing this integral is clearly impractical. Given that the likelihood function serves as a cornerstone in both frequentist and Bayesian inference, this presents a significant challenge for inference across numerous fields.

There are several scenarios where the likelihood is intractable. One such scenario is integrating over the simulator's latent state. The stochastic nature of a simulator often arises from generating random numbers that form the simulator's latent state. Calculating the likelihood involves integrating over these latent states, which is typically intractable due to the complexity and high dimensionality of the latent variables.

Another scenario is when the simulator's output is a deterministic transformation of its latent state. In such cases, calculating the integral over the delta function for the likelihood is not tractable. The joint distribution of the output and latent variables does not admit a proper density, complicating the likelihood calculation and making it impractical.

A final scenario is when the simulator's internal workings are inaccessible. Even if theoretically possible, using a likelihood-based method may be impractical if the simulator is a black box. The simulator might be provided as an executable, written in a low-level language for efficiency, or be an external library routine. This makes it difficult to access the underlying mechanisms needed for likelihood calculation, as the internal processes are hidden or not easily manipulated[9][10].

Simulation models with this problem type are particularly suitable for scientific applications, not only for neuroscience research, but also in biology[11], astronomy and cosmology[12][13], evolution and ecology[14][15].

One common strategy involves the use approximation techniques such as Taylor expansions or advanced numerical methods like Monte Carlo integration to obtaining approximate likelihood estimates while avoiding direct computation of its analytical form.

Additionally, utilizing alternative models or Bayesian inference approaches such as Sequential Neural Posterior Estimation (SNPE) can provide innovative solutions to tackle the problem of intractable likelihood. These models enable conducting statistical inferences even in the presence of challenging-to-compute likelihoods, opening up new possibilities in data analysis and statistical modeling.

1.2.4 Challenges in Complex Brain Model Inference

When exploring the domain of intricate biophysical models of the cortex, scientists face numerous hurdles that greatly affect our capacity to precisely grasp and simulate brain operations. These obstacles stem from the complex nature of neural networks and the intricacies woven into their interactions. Here, we dissect the specific obstacles faced when applying Simulation-Based Inference (SBI) to such models, highlighting high dimensionality, non-linearity, and computational cost as the primary hurdles.

The vast networks of interconnected neurons comprising neural systems contribute to the high dimensionality of complex brain models. Each neuron exhibits intricate dynamics, resulting in a parameter space that becomes exceedingly large and challenging to explore comprehensively. This dimensionality complicates the estimation of model parameters from observed data, as it amplifies issues related to model identifiability and overfitting.

The nonlinear nature of neural dynamics presents another significant challenge when inferring complex brain models. Individual neurons and their interactions often follow nonlinear trajectories, characterized by thresholding, saturation, and feedback loops. These nonlinearities give rise to emergent properties at the network level, making it difficult to predict system behavior solely based on its constituent parts. Accounting for these nonlinearities is crucial when applying SBI to complex brain models, as linear approximations may fail to capture the rich dynamics exhibited by real neural systems.

The computational cost of simulating and analyzing complex brain models poses a practical challenge for researchers. Multi-layered biophysical models require sophisticated numerical methods and extensive computational resources to simulate accurately. Additionally, inferring model parameters from experimental data involves computationally intensive optimization algorithms, often necessitating large-scale parallel computing infrastructure. These practical limitations in terms of time, computational resources, and expertise hinder the application of SBI to complex brain models.

Overcoming these obstacles requires interdisciplinary teamwork among neuroscientists, mathematicians, and computer scientists. Creative methods like reducing dimensions, implementing regularization techniques, and employing parallel computing approaches can ease the computational challenges posed by high-dimensional models. Furthermore, progress in machine learning and Bayesian inference provides encouraging paths for enhancing the precision and effectiveness of parameter estimation in nonlinear systems.

1.2.5 The Simulation Based Inference (SBI) model: Sequential Neural Posterior Estimation (SNPE)

Traditional methods for inference without tractable likelihoods have been developed to address this longstanding problem. Among these methods, Approximate Bayesian Computation (ABC) is arguably the most well-known. In its simplest form, rejection ABC involves sampling parameters from the prior distribution and then using a simulator to generate simulated data. If the simulated data are sufficiently close to the observed data, the parameters are retained as samples from the posterior distribution. The likelihood is essentially approximated by the probability that the condition is satisfied, typically using a distance measure and a tolerance threshold.1.2

However, ABC has limitations. For instance, the acceptance probability vanishes for continuous data in the limit of a small tolerance threshold, requiring a large number of simulations. Moreover, ABC's sample efficiency scales poorly with high-dimensional data, and inference for new observations necessitates repeating the entire inference algorithm [16].

Moreover, Markov-chain Monte Carlo ABC (MCMC-ABC) is an alternative approach to reduce the number of rejections involves proposing parameters using Markov-chain Monte Carlo (MCMC) techniques. Instead of independently drawing parameters from the prior distribution, this method generates new parameters by making small adjustments to previously accepted ones. By keeping these adjustments small, the likelihood of accepting the new parameters increases, resulting in fewer rejections compared to directly sampling from the prior. Similarly to rejection ABC, the acceptance probability of MCMC-ABC decreases as becomes smaller or as the dimensionality of the data increases. Another challenge with MCMC-ABC is the initialization of the Markov chain: if the initial parameters θ are unlikely to generate data close to the observed data x_0, the chain may remain stuck at its initial state for an extended period[9].

Another classic approach is to create a model for estimating likelihood are frequentist [18] and Bayesian [19] inference then proceed as if the likelihood were tractable. This method, sometimes referred to as approximate frequentist computation, offers the advantage of being amortized: after an initial computational cost, new data points can be evaluated efficiently.

Both traditional approaches face challenges due to the curse of dimensionality, necessitating the use of low-dimensional summary statistics. Historically, the development of these summary statistics has been the responsibility of domain experts, and the choice of summary statistics has been predetermined prior to inference.

Running high-fidelity simulations can often be resource-intensive, involving numerous parameters and potentially generating complex, high-dimensional data. Traditional methods face difficulties in handling such scenarios effectively. Consequently, there has been a shift towards exploring alternative approaches that leverage machine learning techniques to enhance the applicability space after SBI.Their method directly addresses the posterior distribution by employing

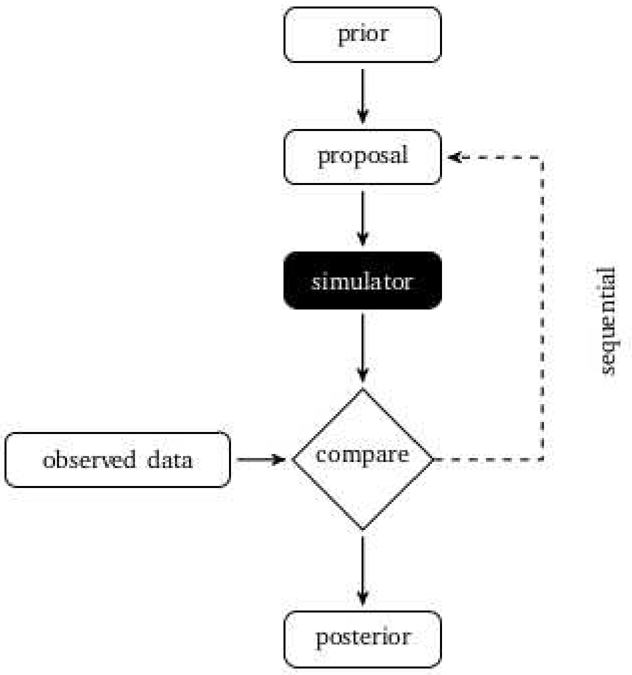


Figure 1.2: A general scheme for rejection Approximate Bayesian Computation (ABC) algorithms involves either proposing all parameters from the prior distribution or employing sequentially refined proposal distributions [17].

neural network-based conditional density estimation, offering a promising avenue for tackling the limitations of classical SBI methods in complex simulation settings.

The fundamental concept behind neural posterior estimation (NPE) approaches is straightforward: a dataset is created by sampling parameters from the prior distribution and simulating, resulting in a training set. This dataset is then utilized to optimize the parameters of a neural network-based conditional density estimator, enabling the derivation of an estimate for the posterior distribution. Compared to traditional methods, this strategy has demonstrated superior simulation efficiency.To enhance simulation space after efficiency, a sequential version of NPE (SNPE) can be employed, particularly beneficial when performing inference for a specific observation.

Sequential Neural Posterior Estimation (SNPE) is an innovative Bayesian inference method based on neural networks and machine learning techniques. The primary goal of SNPE is to efficiently and accurately estimate the posterior distribution of a statistical model's parameters, even in the presence of intractable likelihoods or high computational complexity. This method is based on recent advances in simulation-based Bayesian inference[16],[20],[9].

Unlike traditional approaches that rely on Monte Carlo sampling methods [19] or analytical likelihood approximations [18], SNPE takes a completely different approach leveraging the predictive capabilities of neural networks. Specifically, SNPE uses neural networks as generative models to approximate the joint distribution of observed data and model parameters.

The inference process in SNPE occurs sequentially , using a series of neural networks to progressively approximate the posterior distribution of parameters. Initially, neural networks are trained to approximate the prior distribution of parameters and the data distribution generated by the model. Subsequently, active learning techniques are employed to select points in the parameter space to progressively improve the posterior distribution approximation.1.3

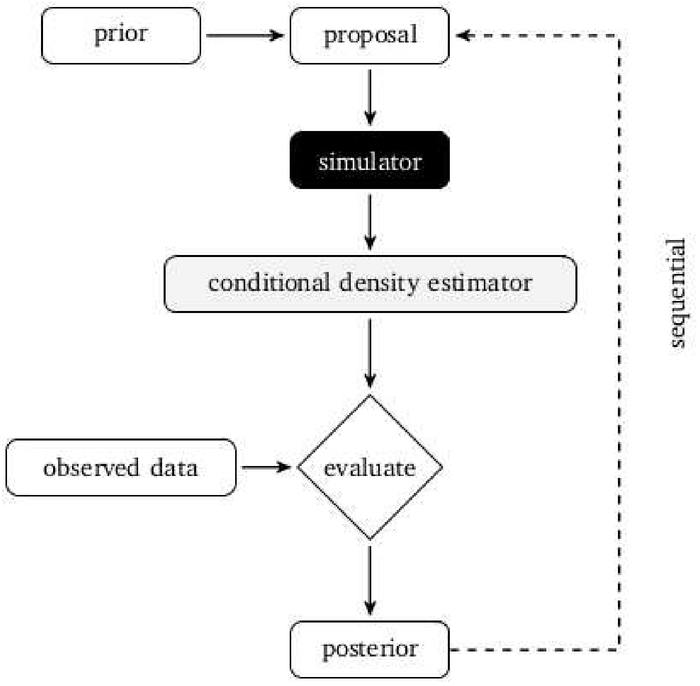


Figure 1.3: A general framework for posterior estimation algorithms entails either proposing all parameters directly from the prior distribution or employing sequentially refined proposal distributions [17].

Using neural networks as generative models allows SNPE to effectively handle intractable likelihoods or complex models, enabling accurate and reliable model parameter estimates. Furthermore, the sequential approach adopted by SNPE optimizes computational efficiency and reduces the overall computational cost of inference. Given the observed/simulation experimental data (or summary features) x_o and a mechanistic model with parameters θ, SNPE uses probability distributions to represent both prior knowledge and the range of parameters that are consistent with the data. SNPE produces a posterior distribution $p(\theta \mid x_o)$ that assigns high probability to parameters θ that align with both the observed data x_o and the prior knowledge, while assigning near-zero probability to parameters θ that do not1.4.

In order to define posterior distributions, various strategies for SNPE (Fig1.4) have been proposed, with the latest one implemented and utilized in this thesis being SNPE-C [21]. SNPE-C enables dynamically adapting the posterior estimation using arbitrary, continuously updated proposals. This method is particularly significant as it overcomes limitations of existing approaches, which are constrained by a narrow range of proposal distributions. Moreover, SNPE-C is compatible with powerful flow-based density estimators, making it flexible, scalable, and efficient. Its capability to operate directly on high-dimensional time series and image data opens up new possibilities for likelihood-free inference applications.

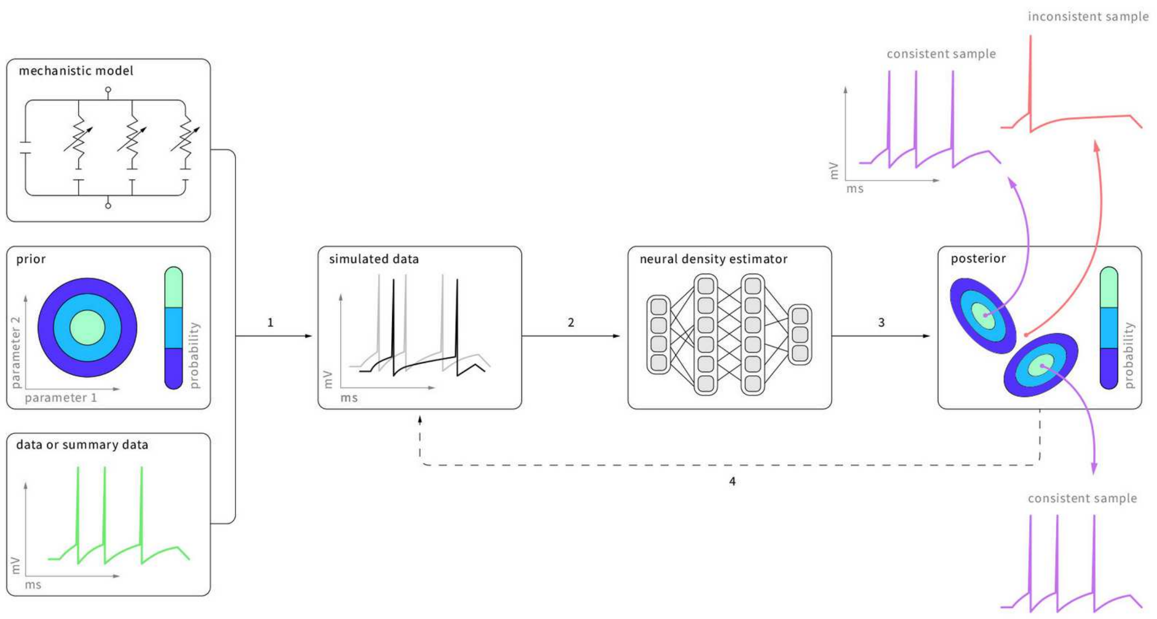


Figure 1.4: The algorithm (SNPE) takes three inputs: a candidate mechanistic model, prior knowledge or constraints on the model parameters, and data (or summary statistics). SNPE proceeds as follows: (1) it samples parameters from the prior and simulates sets of synthetic data based on these parameters; (2) it uses a deep density estimation neural network to learn the probabilistic association between the data (or data features) and the underlying parameters, i.e., to learn statistical inference from the simulated data; (3) it applies this density estimation network to empirical data to derive the entire parameter space consistent with the data and the prior, thus obtaining the posterior distribution. Parameters consistent with the data and the prior are assigned a high posterior probability, while inconsistent parameters are assigned a low posterior probability; (4) if necessary, an initial estimate of the posterior can be used to adaptively guide further simulations to produce results consistent with the data [22].

1.2.6 Model simulation and structure

Modeling neuronal networks entails simulating the intricate behavior of individual neurons and their interactions within a network. These simulations serve as a cornerstone in comprehending brain functionality and elucidating how neural activity manifests into observable phenomena, such as extracellular electric and magnetic signals.

By delving into the dynamics of neuronal networks, researchers gain insights into the complex interplay between various neuronal components, including synaptic connections, neurotransmitter release, and action potential generation. These models not only provide a theoretical framework for understanding brain function but also offer a platform for investigating the effects of different stimuli or perturbations on neural activity [5].

There are various levels of detail in neuronal network models to predict extracellular signals such as EEG. Biophysically detailed multicompartment (MC) neuron models strive to capture the intricate behaviors of neurons by considering various factors such as their electrophysiological properties, synaptic connections, and anatomical structures. These models aim to replicate the complex dynamics observed in real neurons, providing a high level of biological realism in simulations. They are valuable for predicting extracellular signals because they take into consideration the spatiotemporal distribution of transmembrane currents. By accurately simulating the activity of individual neurons, including the flow of ions across their membranes and the resulting changes in membrane potential, these models can provide insights into the generation of extracellular electrical fields. Since extracellular signals are influenced by the collective activity of neurons in a particular region, understanding the dynamics of transmembrane currents is crucial for predicting the patterns of extracellular signals observed experimentally. Nevertheless, The mechanisms governing networks of biophysically detailed multicompartment model neurons are less accessible to analysis and these models are more prone to overfitting.

On the other hand, simplified point-neuron models focus primarily on the generation of action potentials (spikes) in response to external stimuli. While these models are computationally efficient and useful for certain types of analyses, they often lack the biological detail present in multicompartment models.

Population-type models offer a broader perspective by treating groups of neurons as collective entities. Instead of simulating individual neurons, these models represent the overall activity of neuronal populations. They are valuable for understanding emergent properties and population-level dynamics within neural circuits, but they may oversimplify the behavior of individual neurons 1.5.

Spiking neuron network models offer a powerful framework to describe neural phenomena at both micro- and mesoscopic scales [24]–[27]. These models occupy a unique intermediate level of biophysical detail, allowing researchers to interpret the relationships between neuronal and synaptic changes—mediated by molecular and cellular mechanisms—and large-scale brain dynamics [28], [29].

In this study, it was simulated a generic recurrent network model consisting of excitatory (E) and inhibitory (I) populations of integrate-and-fire neurons, with 8192 excitatory neurons and 1024 inhibitory neurons. These populations were driven by external inputs modeled as

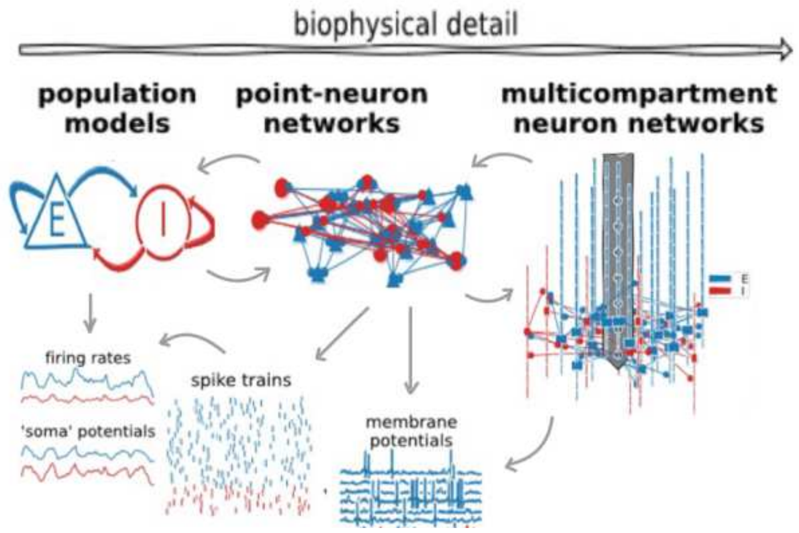


Figure 1.5: Levels of detail for neuronal network models and roadmaps for approximate predictions of brain signals. Biophysically detailed MC neuronal network models at the microscopic scale allow direct simulation of synaptic connectivity and complete cellular dynamics, including action potentials, spike trains, and extracellular signals (such as extracellular potential). Less detailed point neuron network models and continuous population models (such as neuronal mass models, mean field models, and neuronal field models) at the mesoscopic scale do not allow direct predictions of extracellular signals[23].

fixed-rate Poisson processes. The network parameters were adapted from the best-fit values reported in [23], with the exception of the synaptic weights J_{EE}, J_{EI}, J_{IE}, and J_{II} (where J_{YX} represents the synaptic weight from presynaptic population X to postsynaptic population Y), as well as the excitatory and inhibitory synaptic time constants (τ_{synE} and τ_{synI}) and the weight of external synapses (J_{ext}). These parameters were systematically varied to explore one million distinct configurations, providing a detailed understanding of how synaptic mechanisms influence cortical circuit activity and contribute to the generation of macroscopic brain signals (Fig. 1.6).

The motivation for focusing on these synaptic parameters stems from their critical role in neurodegenerative diseases [30], [31] and neurodevelopmental disorders [32]–[36], where synaptic dysfunctions are key factors. Focusing on these specific parameters focused on investigating how variations in synaptic properties shape the behavior of the neural circuit while holding other aspects of the model constant.

To compute the extracellular signals and Local Field Potentials (LFP) generated by the simulated synaptic activity, it was employed the *LFPykernels* package [23]. The current dipole moment, serving as a proxy for the LFP, this is why it was important to describe the LFP signal in the sections above. It was derived by convolving the population spike rates with spatiotemporal filter kernels. These kernels accounted for neuronal biophysics, the spatial distribution of

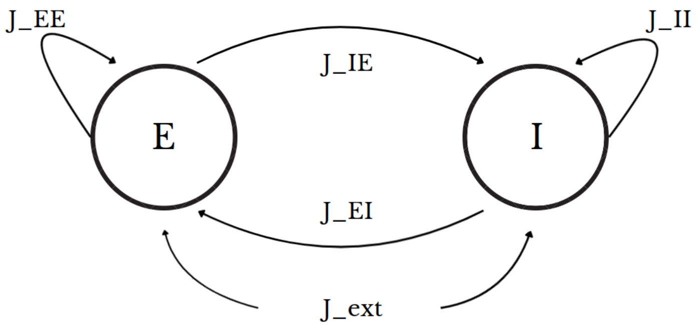


Figure 1.6: The cortical network model include an excitatory and an inhibitory population of leaky integrate-and-fire (LIF) spiking neuron models that interact through recurrent connections

cells and synapses, and network connectivity, using conductance-based multicompartment neurons. For simplicity, it was adopted the reference model of ball-and-stick neurons to represent the multicompartment network, as described in [23].

This approach allowed us to link biophysical synaptic changes to the generation of LFP signals, offering insights into how cortical circuits contribute to macroscopic brain activity.

1.3 Dataset

In the field of neuroscience research, the use of simulated datasets plays a important role in deepening our understanding of neural systems. Examining a simulated dataset created through Hagen's model [23] signifies a notable stride forward in our capacity to scrutinize brain activity.

Simulating data offers an invaluable advantage: the ability to establish a ”ground truth,” a reliable reference point that allows us to control and validate the results of analyses. This holds particular significance in comprehending intricate phenomena such as brain signals, as it grants us meticulous control over experimental conditions, thus ensuring the robustness of our findings.

Furthermore, simulating data provides us with unparalleled flexibility in exploring a wide range of scenarios and conditions that may not be easily replicable in real experimental studies. This flexibility allows us to test hypotheses, carefully examine variations in model parameters, and gain a deeper understanding of the underlying mechanisms driving the observed phenomenology. The dataset considered in this study consists of approximately 1 million brain signal simulation samples (to be precise, 948,819) along with their respective parameters from which they are generated through the model explained in the next section.

1.3.1 Features Dataset

Given the complexity and variability of time series data, it was decided to focus analysis on a limited set of features. To achieve this, it was used a filtered version of the hctsa[37] library. The 22 CAnonical Time-series CHaracteristics[38], known as catch22, were carefully selected from the time series data with the aim of significantly reducing the computational cost associated with temporal analysis without substantially compromising classification accuracy.

#	Feature name	Short name	Category
1	DN_HistogramMode_5	mode_5	Distribution shape
2	DN_HistogramMode_10	mode_10	Distribution shape
3	DN_OutlierInclude_p_001_mdrmd	outlier_timing_pos	Extreme event timing
4	DN_OutlierInclude_n_001_mdrmd	outlier_timing_neg	Extreme event timing
5	first1e_acf_tau	acf_timescale	Linear autocorrelation
6	firstMin_acf	acf_first_min	Linear autocorrelation
7	SP_Summaries_welch_rect_area_5_1	low_freq_power	Linear autocorr
8	SP_Summaries_welch_rect_centroid	centroid_freq	Linear autocorr
9	FC_LocalSimple_mean3_stderr	forecast_error	Simple forecasting
10	FC_LocalSimple_mean1_tauresrat	whiten_timescale	Incremental differences
11	MD_hrv_classic_pnn40	high_fluctuation	Incremental differences
12	SB_BinaryStats_mean_longstretch1	stretch_high	Symbolic
13	SB_BinaryStats_diff_longstretch0	stretch_decreasing	Symbolic
14	SB_MotifThree_quantile_hh	entropy_pairs	Symbolic
15	CO_HistogramAMI_even_2_5	ami2	Nonlinear autocorr
16	CO_trev_1_num	trev	Nonlinear autocorr
17	IN_AutoMutualInfoStats_40_gaussian_fmmi	ami_timescale	Nonlinear autocorr
18	SB_TransitionMatrix_3ac_sumdiagcov	transition_variance	Symbolic
19	PD_PeriodicityWang_th001	periodicity	Linear autocorr
20	CO_Embed2_Dist_tau_d_expfit_meandiff	embedding_dist	Other
21	SC_FluctAnal_2_rsrangefit_50_1_logi_prop_r1	rs_range	Self-affine scaling
22	SC_FluctAnal_2_dfa_50_1_2_logi_prop_r1	dfa	Self-affine scaling

Table 1.1: Series of numbered features along with a concise description and their corresponding short names

Below is a specific description of each feature regarding its informative content:

- **Mode_5 and Mode_10**: These features analyze the shape of the distribution of time-series values by identifying the most common values within the data. Mode_5 divides the time series into 5 bins, while Mode_10 divides it into 10 bins. They provide insights into the positioning of the most probable values relative to the mean, highlighting whether the distribution is concentrated around the mean or dispersed.

- **Outlier_timing_pos and outlier_timing_neg**: Timing of Extreme Events Features assess the timing of extreme events concerning the start and end of the time series. They measure whether over-threshold events, which could be positive or negative deviations from the mean, tend to occur near the start of the time series, are approximately equally likely to happen anywhere throughout the time series, or are more frequently observed near the end of the time series. By analyzing these patterns, these features capture aspects related to the stationarity of over-threshold events, indicating whether their occurrence remains consistent or changes over time.

- **Acf_timescale**: This feature quantifies the scale of autocorrelation within a time series, indicating how far into the future a value at the current point remains substantially correlated with a future point ($>\frac{1}{e}$ correlation).

- **Acf_first_min**: this features computes the first minimum of the autocorrelation function. It exhibits similar behavior.

- **Periodicity**: This feature detects periodic patterns in the time series, indicating regular repetitions of values at consistent intervals. Such periodic behavior reveals recurring patterns within the data, offering insights into the cyclic nature of the observed phenomenon.

- **Low_freq_power**: This feature calculates the relative power in the lowest 20% of frequencies in a time series. High values indicate dominance of low frequencies, while low values suggest higher frequency dominance. The power spectrum is estimated using Welch's method with a rectangular window

- **Centroid_freq**: This feature calculates the frequency based on the higher predominance frequencies.

- **Trev**: This feature calculates the average of the cube of successive differences in the time series. If increases tend to be larger in magnitude, it yields a positive value; if decreases are larger, it yields a negative value. Essentially, it qualitatively reveals the asymmetry of the time series.

- **Ami2**: This feature represents a nonlinear variation of the autocorrelation function. Instead of employing a traditional linear correlation metric, it utilizes a nonlinear correlation metric, specifically mutual information. The evaluation is conducted using a histogram with 5 bins and at a time delay of $\tau = 2$.

- **Ami_timescale**: This feature offers an assessment of the timescale at which the time series exhibits significant (potentially nonlinear) autocorrelation. It is calculated as the minimum value of the automutual information function, which evaluates the level of informativeness between observations at a given time interval compared to observations at subsequent time intervals. This measurement provides an estimate of the characteristic timescale where the time series displays substantial autocorrelation structure.

- **Entropy_pairs**: This feature divides the time series into three equal parts ('A', 'B', 'C') based on value ranges. Then, it calculates the entropy of the probabilities of all two-letter sequences ('AA', 'AB', 'BB', etc.). Low entropy suggests predictable sequences, while high entropy indicates less predictable sequences.

- **Transition_variance**: This feature divides the time series into three equal parts ('A', 'B', 'C') based on their quantiles. It then identifies the time delay, denoted as τ, by locating the first zero-crossing of the autocorrelation function. This feature quantifies the specificity of transitioning from one state to another over the timescale determined by τ. Higher values suggest more ordered series with distinct state transition rules, while lower values indicate noisier series with more uniform transition probabilities.

- **Stretch_decreasing and Stretch_high**: This two features measure the maximum length of time during which similar consecutive local patterns are observed. The former calculates the longest sequence of successive steps in the time series that decrease, while the latter computes the longest sequence of successive values in the time series that are greater than the mean.

- **High_fluctuation**: This feature assigns low values to time series with periods where the values remain approximately constant (within 0.04 standard deviations), and high values to series that do not exhibit such constant periods. Additionally, it monitors and identifies sudden increases in amplitude within the time series.

- **Whiten_timescale**: This feature computes the ratio between the autocorrelation of incremental differences between successive pairs of time-series values and the autocorrelation for the original time series. A higher ratio suggests that the residuals are more consistent with the original time series, while a lower ratio indicates greater divergence between the two. Suggesting a metric for assessing the predictability of fluctuations between successive time points in the time series.

- **Forecast_error**: This feature computes the error from using the mean of the previous 3 values to predict the next value in the time series. It measures the standard deviation of the residuals from a set of simple 1-step forecasts. A low value indicates that the time series is easy to predict, as the mean of the previous 3 time steps serves as a good predictor of the current value.

- **Rs_range**: This features aim to capture potential long-range correlations in time series by analyzing fluctuations. it utilizes rescaled range analysis to estimate points in a logarithmic timescale–fluctuation space. Time series that exhibit rapid changes over short timescales will receive low values for this feature.

- **Dfa**: This features also aim to capture potential long-range correlations in time series through fluctuation analysis. One feature measures the same property as before but from a timescale–fluctuation curve estimated using detrended fluctuation analysis, employing linear detrending in each window after down-sampling the time series by a factor of 2.

- **Embedding_dist**: This feature begins by embedding the time series into a two-dimensional space using a time delay determined by the first zero-crossing of the autocorrelation function. It then calculates the distances between successive points in this embedding space and examines the probability distribution of these distances. The feature outputs the mean absolute error of an exponential fit to this distribution.

1.3.2 Parameters

Each neuron receives connections from other neurons in the network, both excitatory and inhibitory, and also receives an external input, that is the same for both excitatory and inhibitory neurons. These parameters play a critical role in defining the behavior of the network and its response to external stimuli.

- J_{EE} **(Excitatory-Excitatory Synapses)**: This parameter represents the synaptic efficacy of excitatory connections transmitting signals from excitatory neurons to other excitatory neurons within the same population. It indicates the strength of excitatory synapses and influences the network's ability to generate intrinsic excitatory activity and propagate signals through the network.

- J_{II} **(Inhibitory-Inhibitory Synapses)**: This parameter represents the synaptic efficacy of inhibitory connections transmitting signals from inhibitory neurons to other inhibitory neurons within the same population. It controls inhibitory activity in the network and regulates the balance between excitation and inhibition.

- J_{IE} **(Excitatory-Inhibitory Synapses)**: This parameter indicates the synaptic efficacy of excitatory connections transmitting signals from excitatory neurons to inhibitory neurons. It regulates the influence of excitatory inputs on the inhibitory population and can have a significant impact on the network dynamics.

- J_{EI} **(Inhibitory-Excitatory Synapses)**: This parameter represents the synaptic efficacy of inhibitory connections transmitting signals from inhibitory neurons to excitatory neurons. It controls inhibitory activity that modulates excitatory activity in the network and can influence the network's ability to generate and regulate neuronal spikes.

- τ_E **(Excitatory Time Constant)**: This parameter represents the characteristic response time of neurons integrate excitatory synaptic inputs.

- τ_I **(Inhibitory Time Constant)**: This parameter represents the characteristic response time of inhibitory postsynaptic potentials.

- $J_{\textbf{ext}}$ **(External Input)**: This parameter represents the external input that affects both populations of excitatory and inhibitory neurons. It can represent external stimuli or inputs from other regions of the brain or nervous system.

Further, it was decided The $[E/I]_{net}$ like a new metric.

$$[E/I]_{net} = \left(\frac{J_{EE}}{J_{IE}}\right) / \left(\frac{J_{II}}{J_{EI}}\right)$$

It is defined as the ratio between predicted EI of excitatory and inhibitory populations, respectively. This summary measure quantifies the net effect of excitatory and inhibitory processes in the circuit model and may be seen as a global EI ratio from model simulations. It is a convenient metric because the synaptic parameters of recurrent connections (J_YX) are difficult to be interpreted alone as they are strongly coupled and interrelated. Additionally, in the real world, we don't know about any brain disease or artificial manipulation of brain activity that modifies only one synaptic parameter alone. For example, when you chemogenetically inhibit pyramidal neurons, both J_EE and J_IE are directly affected. However, for interpreting results, it is complicated to analyze a change of a parameter without considering a change in the other parameters simultaneously.

Overall, these parameters define the properties and dynamics of the neural network, including how it responds to external inputs, how it self-organizes, and how it generates and transmits signals within the network.

1.3.3 Software and hardware

The SBI (Simulation-Based Inference) library is a Python toolbox [39] designed for simulation-based inference. Instead of relying on complex analytical models, SBI enables users to formulate models through direct computational simulations. Additionally, SBI provides a set of tools and techniques optimized for simulation-based inference, allowing users to efficiently explore the parameter space of the model and obtain robust probabilistic estimates. It has been supported by the German Federal Ministry of Education and Research (BMBF) through project ADIMEM (FKZ 01IS18052 A-D), project SiMaLeSAM (FKZ 01IS21055A) and the Tübingen AI Center (FKZ 01IS18039A).

For the calculation of brain activity simulations to construct the dataset, NEST [40] was employed for running the simulations. NEST is a powerful tool designed for simulating large-scale neural networks, enabling the modeling of complex neuronal interactions and dynamics across extensive populations of neurons. LFPykernels [41] were used to calculate kernels derived from simulated brain activity, which are critical to understanding how simulated neural activity translates into measurable electrical signals.

Due to the significant data load and consistent computational demand for calculating results and executing algorithms, it was necessary to utilize the supercomputers of University of Granada. It is composed into different parts, with the most recent one known as 'ALBAICÍN' [42]. Its architecture is a powerful server infrastructure designed to deliver high performance across a wide range of computational applications. Here are some additional specifications:

- **168 Nodes with 56 Cores and 192 GB of RAM**: These nodes provide substantial computing power for medium to large-scale workloads. Each node is equipped with 56 cores and 192 GB of RAM.

- **2 Nodes with 56 Cores and 1 TB of RAM**: These two nodes boast a large amount of RAM, totaling 1 TB each, making them suitable for applications requiring management of large datasets or complex computational simulations.

- **Infiniband HDR200 Interconnection**: The Infiniband HDR200 connectivity ensures high bandwidth and low latency, facilitating fast and efficient communication between cluster nodes.

- **822 TFLOPS Linpack Rmax**: This measurement indicates the maximum computing power achievable by the system, assessed using the Linpack benchmark. With 822 TFLOPS, the system offers significant computing capacity, ideal for applications requiring intense computational performance.

It offers a total of 9520 cores, 184 terabytes of hard drive memory, and 35 terabytes of RAM.

1.3.4 Multiprocessing and Multithreads

The operating system plays a crucial role in resource allocation to processes, and to improve CPU utilization, multiprocessing and multithreading are two fundamental approaches.

Multiprocessing involves executing multiple processes using two or more processors simultaneously. This approach can significantly enhance system performance by distributing the workload across multiple processing units.

- **Symmetric Multiprocessing (SMP)**: In this model, all processors have uniform access to memory and I/O resources. Each processor performs tasks independently without any hierarchy.
- **Asymmetric Multiprocessing (AMP)**: Here, processors follow a master-slave architecture. The master processor manages process allocation and resource management, while the slave processors execute the tasks assigned by the master. This model can simplify task management but may introduce bottlenecks if the master becomes overloaded.

Multithreading, on the other hand, involves dividing a single process into multiple threads that run concurrently. This approach allows a process to handle multiple tasks in parallel without the need to create separate processes, making it more resource-efficient. Specifically, the key differences are:

- **Multiprocessing**: Involves executing multiple processes on two or more processors. Each process is independent and requires separate resources.
- **Multithreading**: Involves executing multiple threads within a single process. Threads share common resources, reducing overhead and improving efficiency.

In this study, due to the high computational load caused by the SBI model, it became necessary to use these techniques to optimize and reduce code execution time. Initially, a multithreading approach was attempted, but it proved inefficient because it did not allow for separate control signal handling for the interruption of each created thread. Therefore, an asymmetric multiprocessing (AMP) approach was adopted, made possible by utilizing the UGR server, which allows for distributing the computational load across up to four nodes, including the master node [43].

1.4 Posterior Diagnostics

1.4.1 Interquartile range (IQR)

In statistics, the interquartile range (IQR) is a measure of statistical dispersion and is defined as the difference between the third quartile (Q3) and the first quartile (Q1) of a dataset. A quartile is one of three markers that divide a dataset into four equally sized groups, each containing approximately 25% of the data points. The first quartile, or lower quartile (Q1), marks the 25th percentile, while the third quartile, or upper quartile (Q3), marks the 75th percentile 1.7.

The IQR is useful because it quantifies the spread of the middle 50% of the data, providing a measure of central dispersion that is not influenced by extreme values. This makes the IQR particularly robust in the presence of outliers, unlike other measures of dispersion such as the range or standard deviation, which can be significantly affected by outliers.

To calculate the IQR, you must first determine the median of the dataset. The median is the midpoint value that separates the higher half from the lower half of the data. Once the median is identified, the dataset is divided into two halves: the lower half (below the median) and the upper half (above the median).

The first quartile (Q1) is calculated as the median of the lower half of the dataset:

$$Q1 = \text{Median of lower half of data}$$

Similarly, the third quartile (Q3) is calculated as the median of the upper half of the dataset:

$$Q3 = \text{Median of upper half of data}$$

The interquartile range (IQR) is then computed as:

$$IQR = Q3 - Q1$$

Despite its usefulness, the IQR has limitations. It does not provide information about the outliers or the overall shape of the data distribution; it only reflects the spread of the middle 50% of the data. Additionally, the IQR can be biased by small sample sizes, potentially not capturing the true variation within a population. However, it remains a valuable tool for understanding data dispersion, particularly when dealing with datasets containing extreme values [44].

1.4.2 Parameter Recovery Error (PRE)

When working with a posterior approximation Φ, it is beneficial to examine its behavior across different regions of the parameter space. For a given configuration of parameters θ_0 and a simulated output $x_0 \sim p(x|\theta_0)$, we measure how concentrated $\Phi(\theta|x_0)$ is around θ_0. This measure is known as the Parameter Recovery Error (PRE)[45].

To calculate the PRE, we compare the marginal distribution $\Phi(\theta|x_0)$ with a Dirac delta centered at $\theta_0[k]$. Practically, we estimate the PRE as follows:

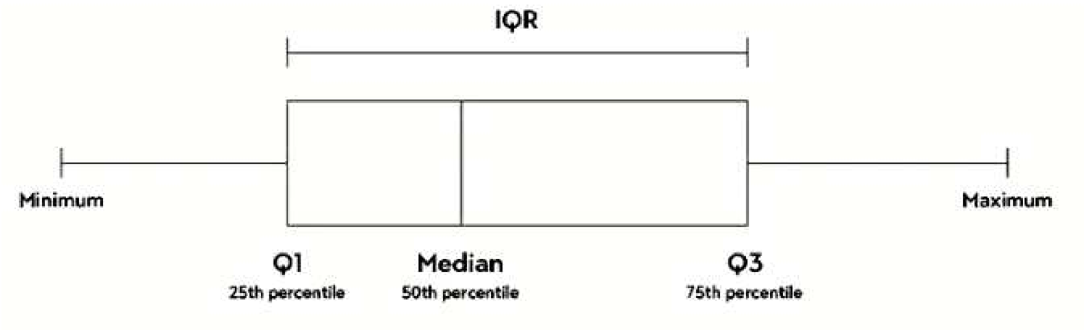


Figure 1.7: Rappresentation of the interquartile range (IQR)like the difference between the third quartile of your data and the first quartile of your data on boxplot.

$$\text{PRE}_k(\Phi, \theta_0) = \frac{1}{N}\sum_{i=1}^{N}(\theta_i[k] - \theta_0[k])^2$$

where N samples $\{\theta_1, \ldots, \theta_N\} \sim \Phi(\theta|x_0)$ are generated from our posterior approximation conditioned on $x_0 \sim p(x|\theta_0)$, which is an observation from the simulator at the true value θ_0.

In a model with multiple parameters, θ_0 represents a vector of k distinct parameters, i.e.,

$$\theta_0 = (\theta_0[1], \theta_0[2], \ldots, \theta_0[k])$$

This means that each component of θ_0 (i.e., each $\theta_0[k]$) represents a specific parameter of the model.

To evaluate how well the posterior approximation $\Phi(\theta|x_0)$ recovers each individual parameter $\theta_0[k]$, a PRE value is calculated for each parameter. Thus, we obtain k distinct PRE values, one for each parameter.

In other words, each parameter $\theta_0[k]$ is mapped from its defined interval in the prior distribution to the interval $(0, 1)$. This means that regardless of the original range of the parameters, they are transformed into a standardized interval from 0 to 1.

- A maximum PRE of 1.0 indicates the worst possible parameter recoverability, meaning that the posterior samples are far from the true value $\theta_0[k]$.
- A PRE of 0.0 indicates perfect recoverability, meaning that the posterior samples are exactly equal to the true value $\theta_0[k]$.

This normalization provides a standardized measure of error, regardless of the different scales of the original parameters.

This mapping allows for the comparison of estimated parameter values with true values on a common, normalized scale. Here's why this transformation is necessary:

- **Consistency in Measures**: Mapping both $\theta_0[k]$ and the samples from the posterior distribution $\Phi(\theta|x_0)$ to the interval $(0, 1)$ ensures that we are comparing values on the same scale. This makes the PRE measure meaningful and comparable across different parameters.

- **Comparability between Parameters**: Different parameters may have vastly different value ranges. For instance, one parameter might range from 0 to 1000, while another ranges from 0 to 1. Without normalization, PRE values would not be comparable, as an error of 0.1 has very different implications across different ranges. By mapping all parameters to the interval $(0, 1)$, we can meaningfully compare PREs across different parameters.

- **Interpretation of PRE**: When all parameters are mapped to the interval $(0, 1)$, the PRE has a clear and interpretable meaning. A PRE of 1.0 indicates the maximum possible error (i.e., the samples are at the extreme of the interval relative to the true value), while a PRE of 0.0 indicates perfect parameter recovery.

1.5 Objective & Motivation

Understanding the intricate wiring patterns of neurons within the brain and their function is essential for unraveling the complex mechanisms underlying brain function. In particular, studying the interplay between excitatory and inhibitory dynamics within cortical circuits provides crucial insights into how synaptic mechanisms contribute to overall brain activity. The role of synaptic dysfunction in neurodegenerative diseases and neurodevelopmental disorders underscores the importance of focusing on the balance of excitation and inhibition within neural circuits. For example, in Alzheimer's disease and other neurodegenerative conditions, alterations in synaptic strength and connectivity often precede the onset of clinical symptoms, highlighting the need for early detection methods that target these underlying synaptic changes [30], [31]. Similarly, in neurodevelopmental disorders such as autism spectrum disorder and schizophrenia, disruptions in the excitatory/inhibitory (E/I) balance are thought to contribute to the core cognitive and behavioral symptoms observed in these conditions [32]–[34].

Recent advancements in the field of connectomics have enabled researchers to gather vast datasets on neuronal connectivity, facilitating detailed mapping of the brain's structural networks.However, despite the tremendous progress in data acquisition, translating this wealth of connectivity data into actionable insights about the functional principles governing neuronal communication remains a formidable challenge. The brain's connectivity is highly complex, involving a dense web of excitatory and inhibitory connections that interact dynamically across multiple spatial and temporal scales. Extracting meaningful patterns from these connections requires not only advanced computational tools but also a deeper understanding of how these connectivity patterns influence the emergent properties of brain networks, such as oscillatory dynamics, synchronization, and signal propagation.

Simulation-based Bayesian inference (SBI) has emerged as a promising approach to address this challenge by leveraging computational models to infer the parameters of hypothesized wiring rules. By simulating neuronal networks and comparing the resulting synthetic data with empirical observations, SBI enables researchers to systematically explore the space of possible wiring rule parameters and identify those that best replicate real-world connectivity patterns.

The objective of this study was to develop a robust approach for determining feature importance in the context of cortical circuit parameter inference. To achieve this, we generated a dataset comprising one million distinct simulations from a spiking cortical microcircuit model, featuring recurrently connected excitatory and inhibitory neuronal populations. Realistic local field potential (LFP) data were obtained by applying biophysics-based causal filters to the simulated spike activity. From these LFP simulations, we extracted a comprehensive set of meaningful features that were used to train a SBI algorithm. To assess the significance of each feature in predicting model parameters, we employed differents tecnich like mutual information (MI), principal component analysis (PCA) and SHAP values, a widely-used method in machine learning that provides detailed insights into the contribution of each feature to the prediction outcomes.

Chapter 2

Methods

2.1 Features Analysis

2.1.1 Correlation

Correlation is a fundamental concept in science, particularly in statistics and scientific research, as it provides a way to measure and understand the degree of relationship between two variables. In simple terms, it indicates whether and how much two variables tend to vary together. There are different types of correlation [46], but one of the most common is Spearman's correlation.

Spearman's correlation is a non-parametric measure of correlation that assesses the monotonic relationship between two variables. This means that, instead of being based on the actual values of the variables, it is based on the ordinal classifications of observations. In other words, it evaluates whether there is a general trend between the variables without assuming a specific functional relationship.

Spearman's correlation is often used when the data do not meet the requirements for the application of Pearson's correlation, for example, when the data are nominal or ordinal rather than continuous, or when the relationship between the variables is nonlinear [47].

Spearman's correlation provides a correlation coefficient, known as Spearman's correlation coefficient or rho (ρ), which ranges from -1 to +1. A value of +1 indicates a perfect positive correlation, a value of -1 indicates a perfect negative correlation, and a value of 0 indicates no correlation. Spearman's correlation is used in a wide range of scientific fields.[48]. It is employed to examine relationships between variables in observational studies, to assess the reliability of measures or tests, and to identify potential associations between phenomena.

Spearman's correlation is an important statistical technique for measuring and understanding the relationship between two variables, especially when the data do not meet the requirements for other forms of correlation or when the relationship between the variables is nonlinear.

2.1.2 Mutual Information

Mutual Information (MI) between two random variables measures the nonlinear relationships between them and indicates how much information can be obtained from one random variable by observing another.

It is closely related to the concept of entropy, as it can be interpreted as the reduction of uncertainty of a random variable if another is known. Another use is for feature selection. When having a big dataset with a big range of features, mutual information can help to select a subset of those features in order to discard the irrelevant ones.

The formula for calculating the mutual information for two discrete random variables is:

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(X,Y)(x,y) \cdot \log\left(\frac{p(X,Y)(x,y)}{p_X(x) \cdot p_Y(y)}\right)$$

Where p_X and p_Y are the marginal probability density functions and p_{XY} is the joint probability density function. To compute the mutual information for continuous random variables, the summations are replaced by integrals As explained before, it is related to entropy. This relation is shown in the following formula:

$$I(X;Y) = H(X) + H(Y) - H(X,Y) = H(X,Y) - H(X|Y) - H(Y|X)$$

Entropy (H) measures the level of expected uncertainty in a random variable. Therefore, $H(X)$ is approximately how much information can be learned from the random variable X by observing just one sample.

$$H(X) = -\sum_{x_i \in X} P(X = x_i) \cdot \log(P(X = x_i))$$

The joint entropy measures the uncertainty when considering two random variables together. The conditional entropy$H(X,Y)$ measures how much uncertainty the random variable X has when the value of Y is known. In other words if X and Y are independent then MI will be zero and greater than zero if they are dependent. This implies that one variable can provide information about the other thus proving dependency. The definitions provided above are given for discrete variables and the same can be obtained for continuous variables by replacing the summations with integrations [49].

Mutual information is a useful measure for quantifying nonlinear relationships between random variables and also between variable and target for assessing how much information one random variable provides about another random variable or specific target. Its interpretation is crucial for better understanding the nonlinear relationships intrinsic to the dataset, which could be essential for data analysis and statistical modeling.

2.2 Evaluation of Features Importance

A fundamental property of a unique feature is its ability to contain useful information about the different classes within the data. This property, known as feature relevance, measures a feature's usefulness in discriminating between different classes [50].

The issue of feature relevancy is crucial, and numerous publications[50][51][52] have proposed various definitions and measurements for the relevance of a variable. A general definition states that a feature can be regarded as irrelevant if it is conditionally independent of the model output. Essentially, this means that while a feature can be independent of the input data, it cannot be independent of the model output. In other words, a feature that does not influence the class labels could be discarded. By focusing on relevant features, we can reduce the dimensionality of the data, mitigate the risk of overfitting, and improve the interpretability and efficiency of the model.

Moreover, understanding and quantifying feature relevancy is not just about eliminating irrelevant features but also about recognizing the interplay between features. Highly correlated features may share redundant information, suggesting that only one of them might be sufficient for the model. Conversely, features that provide unique, non-redundant information are crucial because they capture different aspects of the data's underlying structure.

However, it is also important to consider that a variable, which may appear completely useless by itself, can provide a significant performance improvement when taken in combination with other variables due to the interactions between features[50][53]. These interactions can have a substantial impact on the model's performance, highlighting the complexity and interdependence of features within the dataset.

2.2.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a fundamental technique in the realm of dimensionality reduction, widely employed to simplify large datasets by transforming a voluminous set of variables into a more manageable and concise representation. The essence of PCA lies in its ability to reduce the number of variables while retaining the majority of the original data's information. This reduction facilitates easier exploration, visualization, and analysis of data, enhancing both efficiency and clarity in data processing.

In PCA, ”information” is quantified as the total variability present in the original input variables, essentially the sum of the variances of these variables. The core of PCA is the spectral decomposition (also known as eigendecomposition) of the sample variance-covariance matrix. This decomposition yields the eigenvalues and eigenvectors of the covariance matrix. The eigenvalues, ordered in descending magnitude, represent the amount of total variability in the original variables "explained" or "expressed" by each principal component. The corresponding eigenvectors denote the orthogonal directions of maximum variability derived from the principal components.

Principal components are constructed as new variables through linear combinations of the original variables. These combinations are designed to ensure that the new variables (i.e., principal components) are uncorrelated and encapsulate most of the information from the initial variables within the first few components. By organizing information in this manner, PCA enables dimensionality reduction with minimal loss of information, achieved by discarding components with low information content and retaining those with significant variance [54].

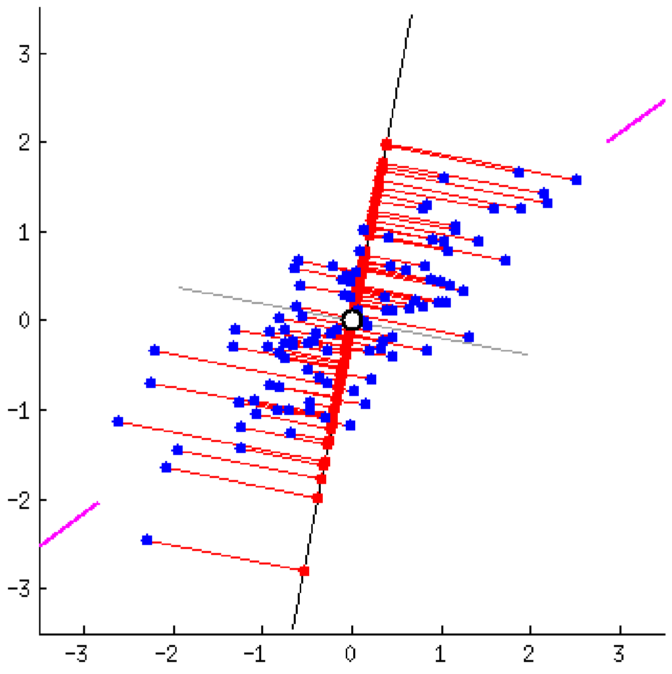


Figure 2.1: The purple line represents the first principal components at which explain the highest variance.

However, it is crucial to acknowledge that principal components, being linear combinations of the original variables, are less interpretable and do not possess intrinsic meaning. Geometrically, principal components represent the directions in the data that account for the maximum variance, essentially the axes along which data dispersion is greatest. The relationship between variance and information implies that a line carrying larger variance exhibits greater data dispersion along it, hence more information. Conceptually, principal components can be envisioned as new axes providing optimal angles to view and evaluate data, enhancing the visibility of differences between observations.

The construction of principal components matches the number of variables in the dataset, with the first principal component capturing the maximum possible variance. Mathematically, this component is the line that maximizes the variance, represented as the average of the squared distances from the projected points to the origin(Fig. 2.1). The second principal component, orthogonal to the first, accounts for the next highest variance and is calculated similarly.

Standardization of variables is often essential in PCA, particularly when there are significant differences in the variances of the original variables. Without standardization, variables with higher variance may dominate the principal components, skewing the analysis. Consequently, principal components should typically be derived from the covariance matrix only when all original variables operate on approximately the same scale. In datasets where variables differ in type and origin, the structure of principal components will depend on the arbitrary choice of measurement units, underscoring the importance of standardization [55].

PCA offers a robust methodology for dimensionality reduction, preserving critical information while simplifying data structures. Its mathematical and geometric foundations ensure that it remains a pivotal tool in the analysis of complex datasets, facilitating clearer insights and more efficient data processing.

2.2.2 Shapley Additive Explanations (SHAP)

Shapley Additive Explanations (SHAP) is a powerful method for interpreting and explaining machine learning model predictions by attributing importance scores to input features. Machine learning models have become increasingly complex, making it difficult for users to understand and trust their predictions. SHAP addresses this issue by providing a way to explain the contributions of each feature to a model's prediction for a specific instance.

SHAP is a model-agnostic method (i.e., it can be applied to different types of machine learning models) and belongs to the class of additive feature attribution methods; meaning that it attributes an effect of a feature x_i on the prediction of a model $f(x)$. Such methods construct a simple additive explanation model, g—which is a linear function of binary variables—to represent the complex original model, f. In the SHAP framework, the explanation model is expressed as a ”conditional expectation function of the original model” [20].

Given simplified inputs x', the original input can be mapped through a mapping function h_x, where $x = h_x(x')$. This ensures that $g(z') \approx g(h_x(z'))$ whenever $z' \approx x'$, with $z' \in \{0,1\}^M$ and M representing the number of simplified input features.

Consequently, an effect ϕ_i, where $\phi_i \in R$, is attributed to each feature, and the sum of these effects approximates $f(x)$ as follows:

$$f(x) \approx \sum_{i=1}^{M} \phi_i$$

This quantification is based on the concept of Shapley values, which are a concept from cooperative game theory that provides a fair way to distribute rewards among players (or features).

They are calculated by considering all possible permutations of players and determining the marginal contribution of each player to the total reward. This ensures that each player's contribution is fairly recognized, taking into account the interactions between players and their individual impact on the game's outcome. At its core, the Shapley Value is a fair value allocation method, a way to distribute total gains (or losses) among participants in a multi-player setting, ensuring each one gets their due based on their contribution.

The Shapley value for a player i is the average of their marginal contributions to all possible coalitions they can join. The formal definition is:

$$\phi(i) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} [v(S \cup \{i\}) - v(S)]$$

In this equation:

- $\phi(i)$ represents the Shapley value for player i.
- The sum is taken over all subsets S of the set N that don't include player i.
- $|S|$ is the size of subset S, and $|N|$ is the total number of players.
- $v(S)$ is the value function, which assigns a value to each coalition of players.
- The formula effectively averages the marginal contributions of player i across all possible coalitions.

This equation may look complicated, but it's essentially a method to average all possible contributions of a player (or feature) across different coalitions [56].

In this setting, Shapley values help to fairly distribute a model's prediction among its input features, providing insight into the importance of each feature. To define the influence of features on the model they focus and take into account certain proprieties

- **Local Accuracy (Additivity)**: This property ensures that the sum of the individual feature attributions equals the original model prediction. If a model's prediction for a particular instance is $f(x)$, then the sum of the Shapley values of all features for that instance should also be $f(x)$.

  Mathematically, for a model f and input instance x:

  $$f(x) = \sum_{i=1}^{m} \phi_i$$

  where ϕ_i is the Shapley value corresponding to feature i for the instance x.

- **Consistency (Symmetry)**: Consistency ensures that if a feature contributes more to a model's prediction in one model compared to another, then its Shapley value should reflect this difference. If changing a feature value has a larger impact on model A than on model B, the Shapley value (feature importance) for that feature should be higher in model A than in model B.

- **Nonexistence (Null Effect)**: If a feature does not contribute to the model's prediction regardless of the presence of other features, its Shapley value should be zero. This means that for any feature that has no impact on the output of the model, the Shapley value will reflect that it has no importance.

The SHAP framework leverages the properties of Shapley values to provide a unified approach to feature attribution in machine learning models. SHAP values are derived from Shapley values and adapted for interpreting the output of complex models.

These properties can be examined by representing feature weights as Shapley values. A crucial aspect involves weighting artificial samples appropriately to link Shapley values with the LIME approach [57][20]. LIME is an approach that explains model predictions by approximating the model locally around the instance to be explained.

The LIME methodology generates the explanation x_i of an instance x according to Equation:

$$\xi(x) = \underset{g \in G}{\arg\min}\, \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

where:

- G is a class of interpretable (linear) models.

- $\mathcal{L}$ is the loss function to minimize.

- π_x is the proximity measure between an instance z(approximation) and x (kernel defining locality).

- $\Omega(g)$ is an optional regularization term to control (limit) model complexity.

While LIME uses heuristic choices to select the model and weighting function, SHAP refines this approach by introducing a kernel function directly related to the definition of Shapley values, ensuring that the feature weights follow the axioms of local accuracy and consistency. Kernel SHAP approximates feature contributions using Shapley values, whereas the original LIME approach defines the locality of an instance heuristically. Kernel SHAP necessitates a background dataset for training, simulating feature absence by replacing feature values with prevalent values from the training data. It then trains a weighted linear regression model, g, as the explanation model based on artificial samples created by toggling features on or off, which corresponds to considering various feature sets. The coefficients of model g are the SHAP values that determine feature importance [58].

Specifically, SHAP uses the following procedure for interpreting an instance:

- **Training Data Clustering**: The training data is organized using k-means clustering to create a set of k representative samples. These samples are weighted according to the number of training instances they represent, forming a background data set with typical feature values.

- **Generating Artificial Samples**: Artificial samples are generated by systematically replacing features of the test instance x with values from the background data set. This process creates variations of the test instance with different combinations of original and background feature values.

- **Weighting Artificial Samples**: These artificial samples are weighted using the SHAP kernel function. The kernel function is designed to assign weights in a manner consistent with the Shapley value definition, ensuring that the importance of each feature is accurately captured.

- **Training a Linear Model**: A weighted linear regression model g is trained using the weighted artificial samples. The goal of this model is to approximate the original model f locally around the instance x. The coefficients of this linear model correspond to the Shapley values, providing feature importance estimates.

The SHAP method offers significant advantages for machine learning models, particularly in terms of transparency and trust. By explaining the contributions of each feature to a model's prediction, SHAP helps users understand and trust complex models, which is crucial for deployment in sensitive areas such as healthcare, finance, and legal systems. In this study, we investigated which features impact a single parameter and how they do so using two metrics: the mean and the interquartile range (IQR). The mean identifies how the central, high-probability value of the parameter shifts in response to changes in the features, providing insight into the general direction and magnitude of the parameter's response. In contrast, the IQR focuses on the variability or variance within the parameter's distribution, revealing the spread of the parameter values and how this spread changes with different feature values. By employing both metrics, we obtain a comprehensive understanding of how features influence the parameter, considering both central tendency and distribution variability.

Chapter 3

Results and Discussion

3.1 Features Analysis

3.1.1 Data visualization

In this section, with the aid of graphs and visualizations, it will be explore the dataset to gain a preliminary understanding of the collected data.

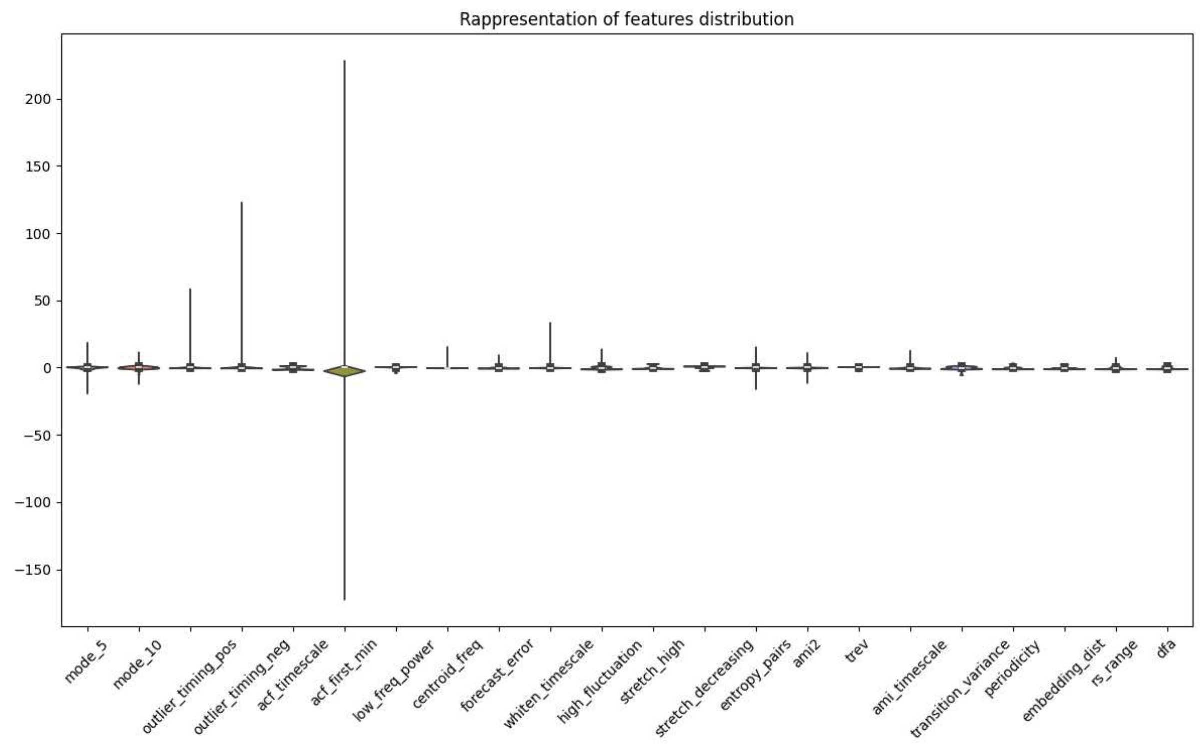


Figure 3.1: Violin plot of normalized features

By employing graphical representations, the relationships was analyzed between different variables and assess their significance and impact within the dataset.

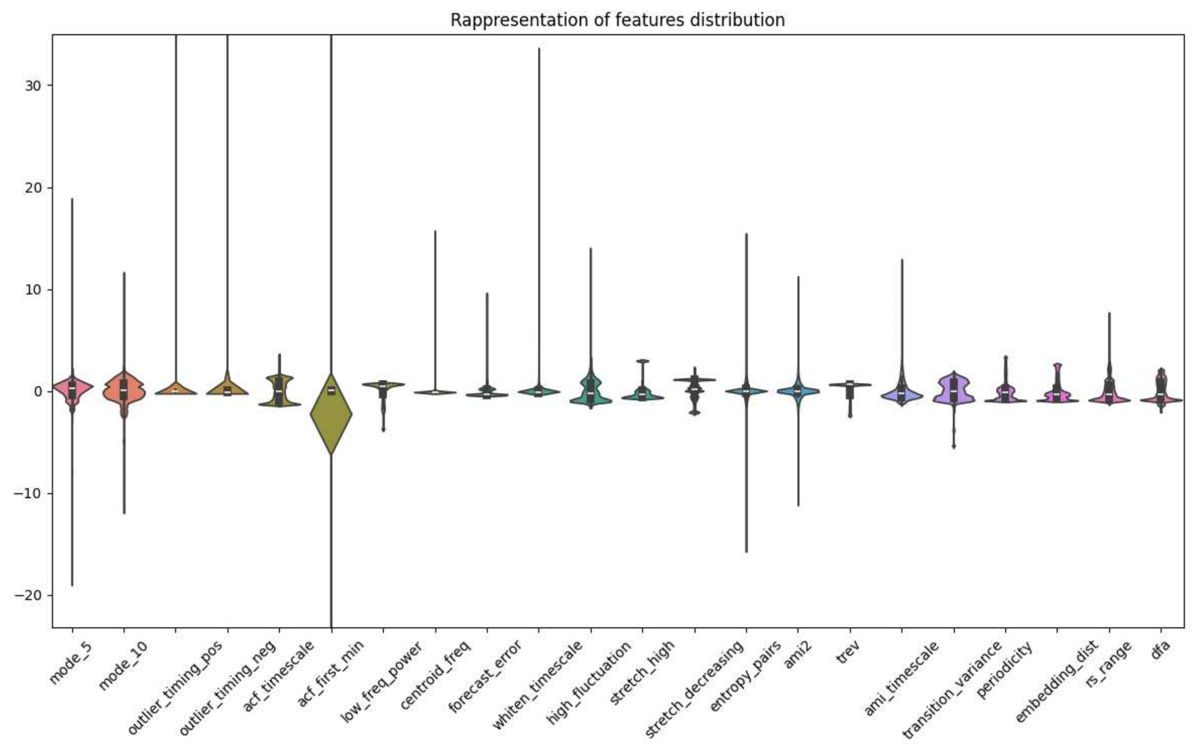


Figure 3.2: Zoom of violin plot of normalized features

In Figure 3.2 The violin plot of the 22 normalized features extracted reveals significant insights into the distribution and variability of these features. Features like *mode_5* and *mode_10* display broad distributions, indicating a wide range of common values, with *mode_10* shows more detail due to finer binning.

Features such as *outlier_timing_pos* and *outlier_timing_neg* capture the timing of extreme events, with their spread suggesting variability in the occurrence of such events, indicating potential non-stationarity. Autocorrelation-related features like *acf_timescale* and *acf_first_min* show distributions that highlight the persistence and initial decay of correlations within the series, showing a greatly variability.

Features such as *low_freq_power* and *centroid_freq* provide insights into the frequency domain, where broader distributions in *low_freq_power* suggest varying dominance of low frequencies across samples. The *trev* feature reveals asymmetry in the time series, with its distribution indicating variability in the magnitude of increases versus decreases.

Trend-related features such as *stretch_decreasing* and *stretch_high* measure the persistence of trends, with their distributions reflecting how often certain patterns persist. The *high_fluctua-*

tion feature highlights the stability of the series, with higher values indicating less stable periods. Predictability measures like *forecast_error* and *whiten_timescale* offer insights into the ease of forecasting the series. Long-term correlation features such as *rs_range* and *dfa* reveal potential long-term dependencies, with their distributions indicating a sustained maintenance of the patterns.

3.1.2 Relationship between parameters of a cortical model and features

Prior to investigating the impact of features on the outcomes of the SBI prediction model, the initial focus was on analyzing the relationships between cortical model parameters and LFP-derived features. To identify potential associations between the model parameters and these features, it was computed the mutual information between each catch22 feature and each parameter of the model.

To detect the relationship intra-features dataset it can be considered Fig. 3.3, it is noted that the features *Mode_5* and *Mode_10* show a strong positive correlation (0.68), suggesting that dividing the time series into 5 or 10 bins provides similar information about the distribution of values.

Similarly, *Outlier_timing_pos* and *Outlier_timing_neg* are highly correlated (0.96), indicating that the timings of extreme positive and negative events tend to occur at similar moments.

The *transition_variance* exhibits a strong negative correlation with *outlier_timing_pos* (-0.96), *outlier_timing_neg* (-0.95), *whiten_timescale* (-0.92), *high_fluctuation* (-0.92), and *stretch_high* (-0.92). This suggests that variable transitions are inversely proportional to the timings of extreme events and autocorrelation timescales, while the positive correlation with *low_freq_power* indicates that greater variability and noise in transitions are associated with lower fluctuations and stretching in the data.

Both *rs_range* and *dfa* are strongly negatively correlated with *outlier_timing_pos* (-0.98), *outlier_timing_neg* (-0.99), *high_fluctuation* (-0.79), and *stretch_high* (-0.95), suggesting an association with extreme event timings, high fluctuations, and data stretching. Additionally, they are negatively correlated with *acf_timescale* (-0.94) and *whiten_timescale* (-0.93), indicating that a larger RS range is associated with shorter autocorrelation timescales. Examining the correlations of *entropy_pairs*, *ami2*, *periodicity*, and *embedding_dist*, it is observe a low correlation with most other features. This suggests that these metrics are relatively independent and not strongly influenced by other aspects of the system. The structural complexity captured by these metrics is not directly related to the variations and dynamics captured by the other features, thus providing unique and complementary information.

By analyzing both the correlation matrix Fig. 3.3 and the Mutual Information (MI) Fig. 3.4 matrix among the features, it can be seen similar patterns of interdependence. This sug-

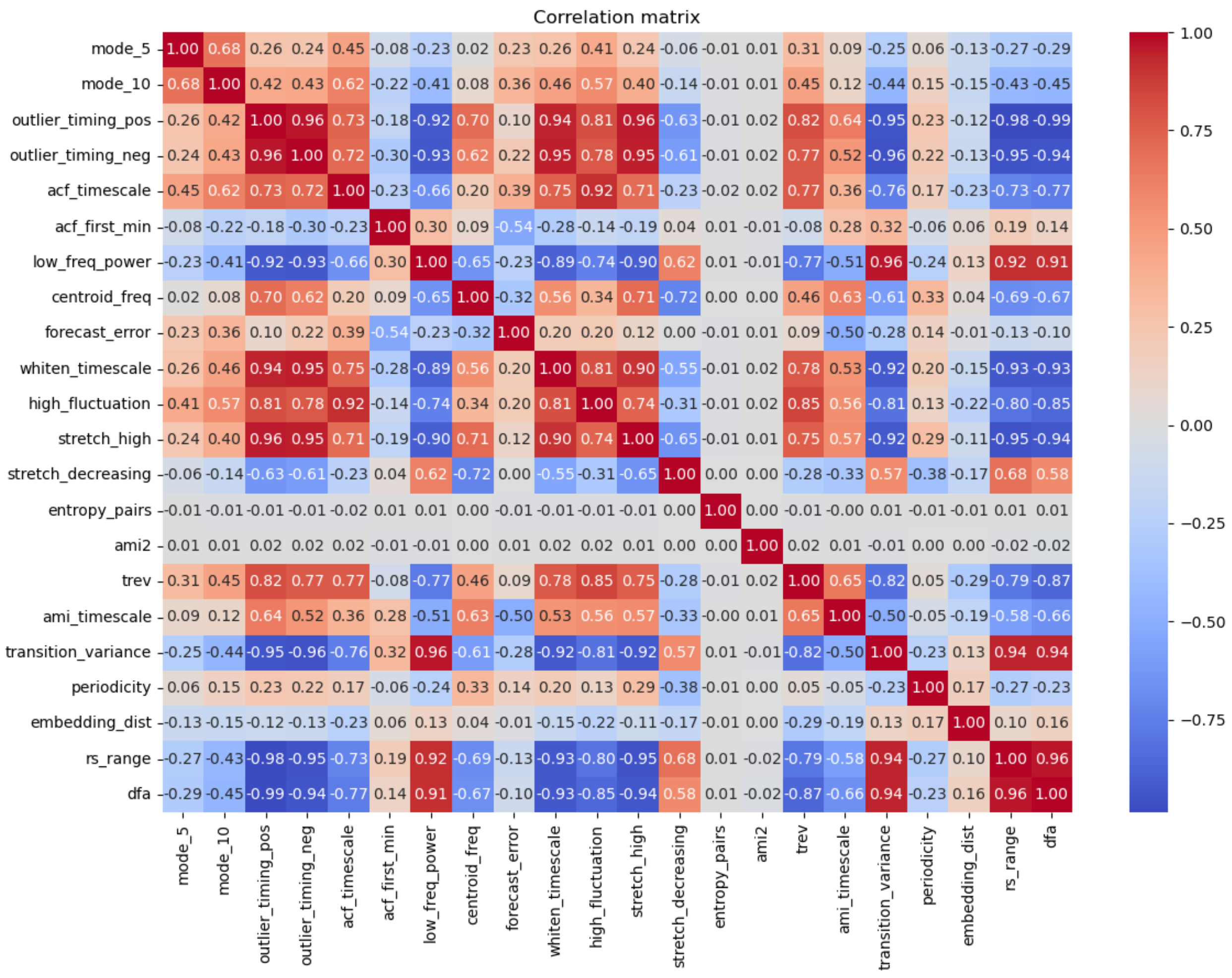


Figure 3.3: Spearman's correlation between features

gests that the same pairs of variables that exhibit strong correlation also tend to have high MI values, indicating a close dependency. *Transition_variance*, for example, shows a high correlation with *dfa* and *rs_range*. This elevated correlation suggests that *transition_variance*, *dfa*, and *rs_range* share a substantial amount of information. The redundancy among these variables could lead to multicollinearity, negatively impacting the model's stability and performance. Thus, reducing or transforming these variables might be necessary. Nevertheless they show a strong and significant correlation with parameters. In addition, *Mode_10* and *Mode_5* have an high MI value, indicating a strong dependency, it might be appropriate to remove one of them. *Entropy_pairs*, *ami2*, *periodicity*, and *embedding_dist* exhibit weak relationships with other variables. *Entropy_pairs* shows relatively weak relationships with other variables, potentially providing unique information not captured by other features. Similarly, *Periodicity* and

Embedding_dist have low MI values with most other variables, suggesting it might contribute unique information to the model.

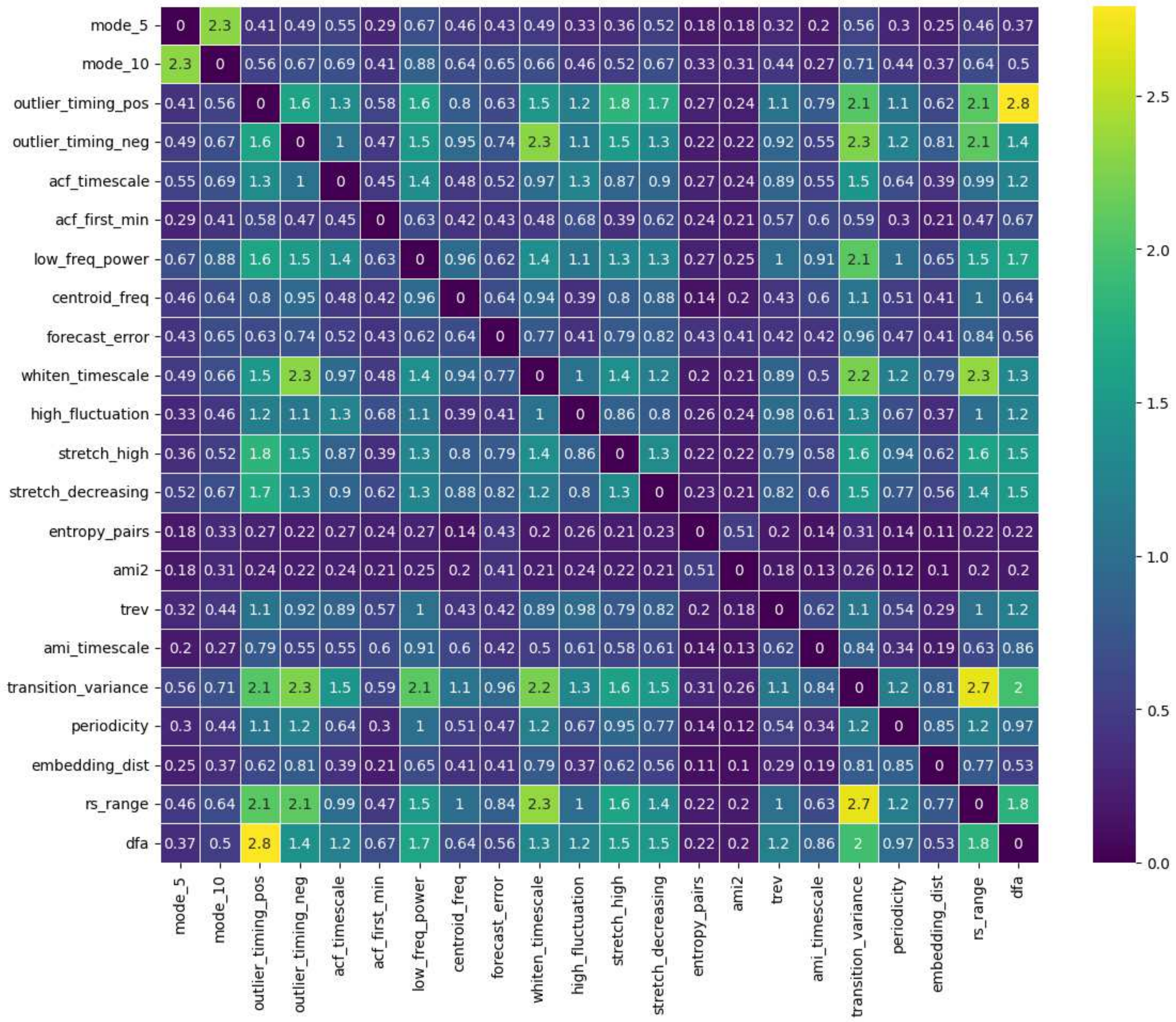
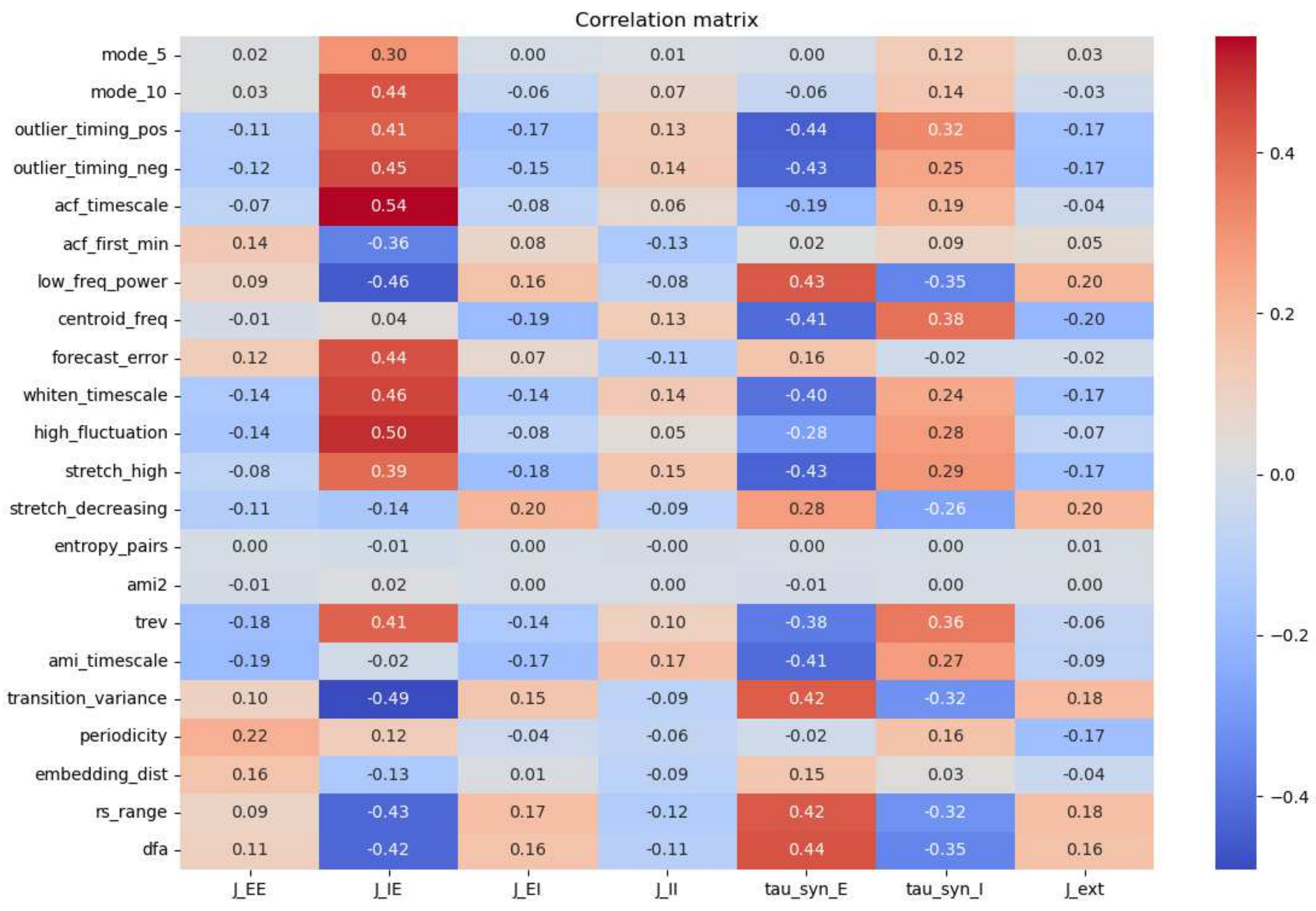


Figure 3.4: Mutual information(MI) between features

To explore potential associations between model parameters and features it was computed the correlations between key parameters and various system features (Fig. 3.5), and it can be see a low general correlation, in particular in the first four coloums. However, it can be seen in figure 3.6 like the new parameters adopted $[E/I]_{net}$ is more correlated on the parameter respect to his components.

These parameters ($[E/I]_{net}$, τ_{synE}, τ_{synI}) show strong correlations with various features, highlighting how they significantly modulate the behavior and characterization of the simulated output signal, influencing the system's complexity and variability. In particular, it is noticeable how some features generally appear to be more correlated than others across all parameters like *low_freq_power*, *trancition_variance*, *rs_range* and *dfa*

The high correlations, both positive and negative, indicate the presence of underlying complex dynamics affecting the system's behavior. This suggests that the system is highly interdependent and that the measured metrics often represent complex and intertwined aspects of the system's behavior.



	J_EE	J_IE	J_EI	J_II	tau_syn_E	tau_syn_I	J_ext
mode_5	0.02	0.30	0.00	0.01	0.00	0.12	0.03
mode_10	0.03	0.44	-0.06	0.07	-0.06	0.14	-0.03
outlier_timing_pos	-0.11	0.41	-0.17	0.13	-0.44	0.32	-0.17
outlier_timing_neg	-0.12	0.45	-0.15	0.14	-0.43	0.25	-0.17
acf_timescale	-0.07	0.54	-0.08	0.06	-0.19	0.19	-0.04
acf_first_min	0.14	-0.36	0.08	-0.13	0.02	0.09	0.05
low_freq_power	0.09	-0.46	0.16	-0.08	0.43	-0.35	0.20
centroid_freq	-0.01	0.04	-0.19	0.13	-0.41	0.38	-0.20
forecast_error	0.12	0.44	0.07	-0.11	0.16	-0.02	-0.02
whiten_timescale	-0.14	0.46	-0.14	0.14	-0.40	0.24	-0.17
high_fluctuation	-0.14	0.50	-0.08	0.05	-0.28	0.28	-0.07
stretch_high	-0.08	0.39	-0.18	0.15	-0.43	0.29	-0.17
stretch_decreasing	-0.11	-0.14	0.20	-0.09	0.28	-0.26	0.20
entropy_pairs	0.00	-0.01	0.00	-0.00	0.00	0.00	0.01
ami2	-0.01	0.02	0.00	0.00	-0.01	0.00	0.00
trev	-0.18	0.41	-0.14	0.10	-0.38	0.36	-0.06
ami_timescale	-0.19	-0.02	-0.17	0.17	-0.41	0.27	-0.09
transition_variance	0.10	-0.49	0.15	-0.09	0.42	-0.32	0.18
periodicity	0.22	0.12	-0.04	-0.06	-0.02	0.16	-0.17
embedding_dist	0.16	-0.13	0.01	-0.09	0.15	0.03	-0.04
rs_range	0.09	-0.43	0.17	-0.12	0.42	-0.32	0.18
dfa	0.11	-0.42	0.16	-0.11	0.44	-0.35	0.16

Figure 3.5: Spearman's correlation between features and parameters

In examining the role of *rs_range* and *dfa*, it can be said to capture the rapid fluctuations of time series. These metrics serve as general aggregators of other features describing such behavior, rather than individually defining each characteristic of the signal's fluctuations.

Moreover, ir was conducted a mutual information (MI) analysis to measure the dependency between the *catch22* features and each model parameter, specifically $[E/I]net$, $\tau synE$, τ_{synI}, and J_{ext} (Fig. 3.7). This initial step in the analysis was crucial for identifying and quantifying the relevance of various features in a machine learning regression framework, offering insights

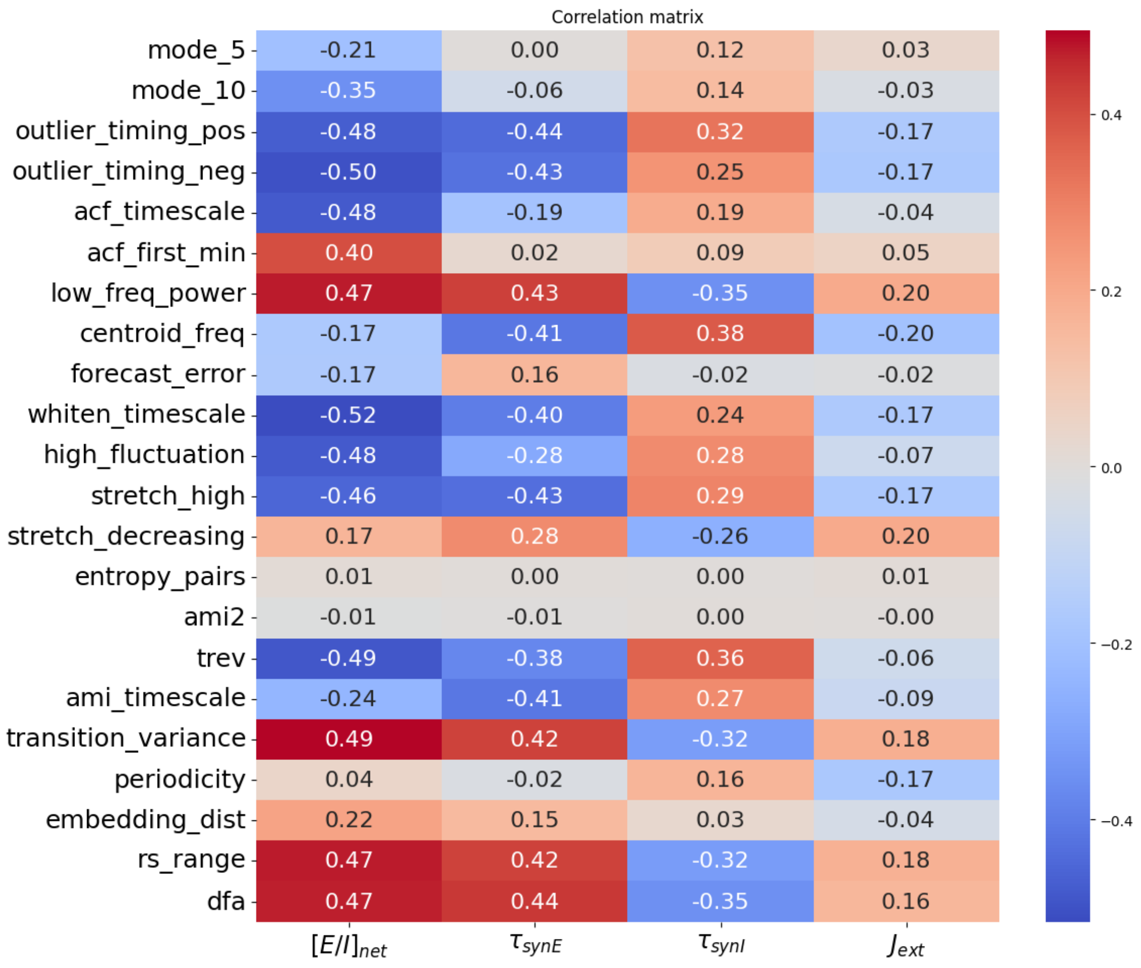


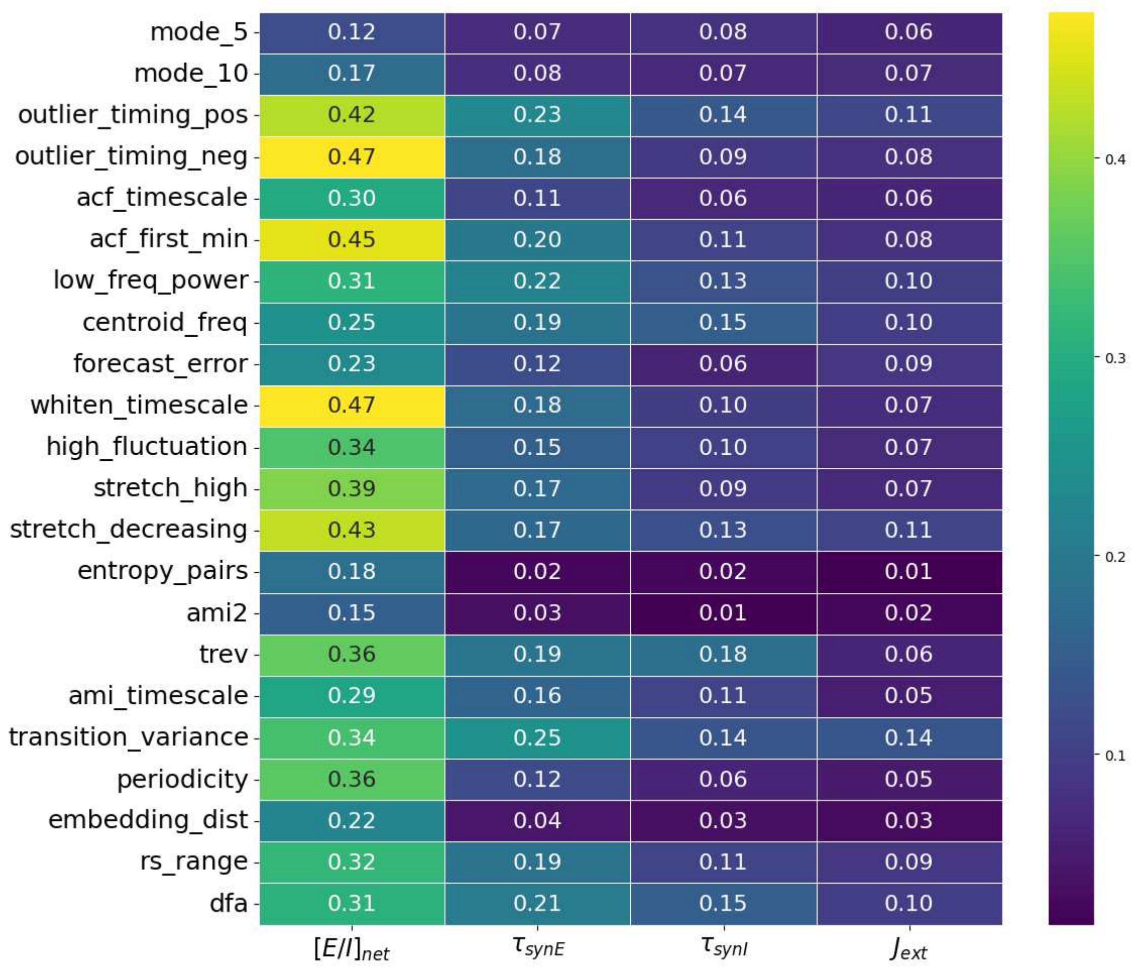
Figure 3.6: Spearman's correlation between features and parameters

into their potential predictive power.

The MI analysis revealed that among the parameters, $[E/I]net$ exhibited the highest MI scores with a majority of the features. This suggests a strong relationship between $[E/I]net$ and many of the *catch22* features, indicating that these features might be particularly effective in predicting or modeling this specific parameter. However, it is noteworthy that some features, such as *mode_5*, *mode_10*, *entropy_pairs*, and *ami2*, were found to be less informative in relation to $[E/I]net$. These features consistently demonstrated lower MI values, implying a weaker association and, consequently, a limited contribution to the prediction of $[E/I]net$.

On the other hand, J_{ext} generally showed low MI values across the majority of features, with only a few exceptions, such as *transition_variance*. This pattern suggests that J_{ext} may be less influenced by the features examined, or that the relevant features for this parameter might not be fully captured by the *catch22* set. Nonetheless, the presence of certain exceptions like *transition_variance* indicates that specific features still hold predictive potential for J_{ext}.

Interestingly, some features demonstrated consistently high MI values across all parameters.



	$[E/I]_{net}$	τ_{synE}	τ_{synI}	J_{ext}
mode_5	0.12	0.07	0.08	0.06
mode_10	0.17	0.08	0.07	0.07
outlier_timing_pos	0.42	0.23	0.14	0.11
outlier_timing_neg	0.47	0.18	0.09	0.08
acf_timescale	0.30	0.11	0.06	0.06
acf_first_min	0.45	0.20	0.11	0.08
low_freq_power	0.31	0.22	0.13	0.10
centroid_freq	0.25	0.19	0.15	0.10
forecast_error	0.23	0.12	0.06	0.09
whiten_timescale	0.47	0.18	0.10	0.07
high_fluctuation	0.34	0.15	0.10	0.07
stretch_high	0.39	0.17	0.09	0.07
stretch_decreasing	0.43	0.17	0.13	0.11
entropy_pairs	0.18	0.02	0.02	0.01
ami2	0.15	0.03	0.01	0.02
trev	0.36	0.19	0.18	0.06
ami_timescale	0.29	0.16	0.11	0.05
transition_variance	0.34	0.25	0.14	0.14
periodicity	0.36	0.12	0.06	0.05
embedding_dist	0.22	0.04	0.03	0.03
rs_range	0.32	0.19	0.11	0.09
dfa	0.31	0.21	0.15	0.10

Figure 3.7: Mutual information between parameters and features

Features such as *outlier_timing_pos*, *low_freq_power*, *transition_variance*, and *dfa* stood out for their strong associations with all parameters under investigation. These features appear to be robust indicators with significant predictive relevance, making them valuable for a broad range of parameter estimations within the model.

Conversely , the removal of redundant variables such as *transition_variance*, *dfa*, *rs_range* and *Mode_5* could be beneficial due to their high interdependence. Given their high correlation, these variables might be considered redundant, and removing or transforming them could help reduce multicollinearity. This careful consideration of both redundant and unique features is essential for enhancing overall model performance, ensuring stability, and preserving the richness of the dataset. It will be check on the next section.

3.2 Evaluation of Features Importance

3.2.1 Principal Component Analysis (PCA)

Subsequently, to investigate the informativeness of the features and their importance, in line with the results of previous section, of correlation and mutual information (MI) analyses, a principal component analysis (PCA) was performed. The scatter plot of the variance explained by each principal component provides an in-depth analysis of the distribution of variance within the dataset. From this plot (Fig.3.8), two key observations emerge:

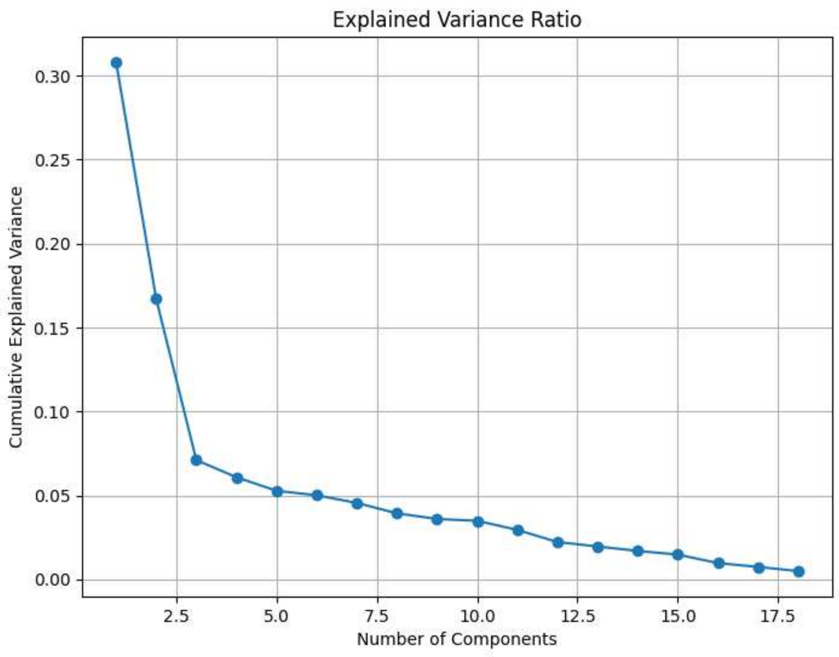


Figure 3.8: The variance scatter plot shows the variance explained for each principal component with all features.

Firstly, the first three principal components stand out as dominant, as they explain a significant portion of the total variance, approximately 60%. This suggests that a substantial amount of the information contained in the original data can be effectively captured using just these three components. In other words, most of the distinctive features and variations present in the dataset can be adequately represented through these three principal components, allowing for dimensionality reduction without substantial information loss.

Secondly, there is a rapid decrease in the variance explained by the subsequent components. This trend indicates that each additional component contributes less and less to the explanation of the total variance. As more principal components are considered, their informational contribution diminishes significantly, suggesting that the inclusion of further components may not be justified in terms of improving data interpretation or representation.

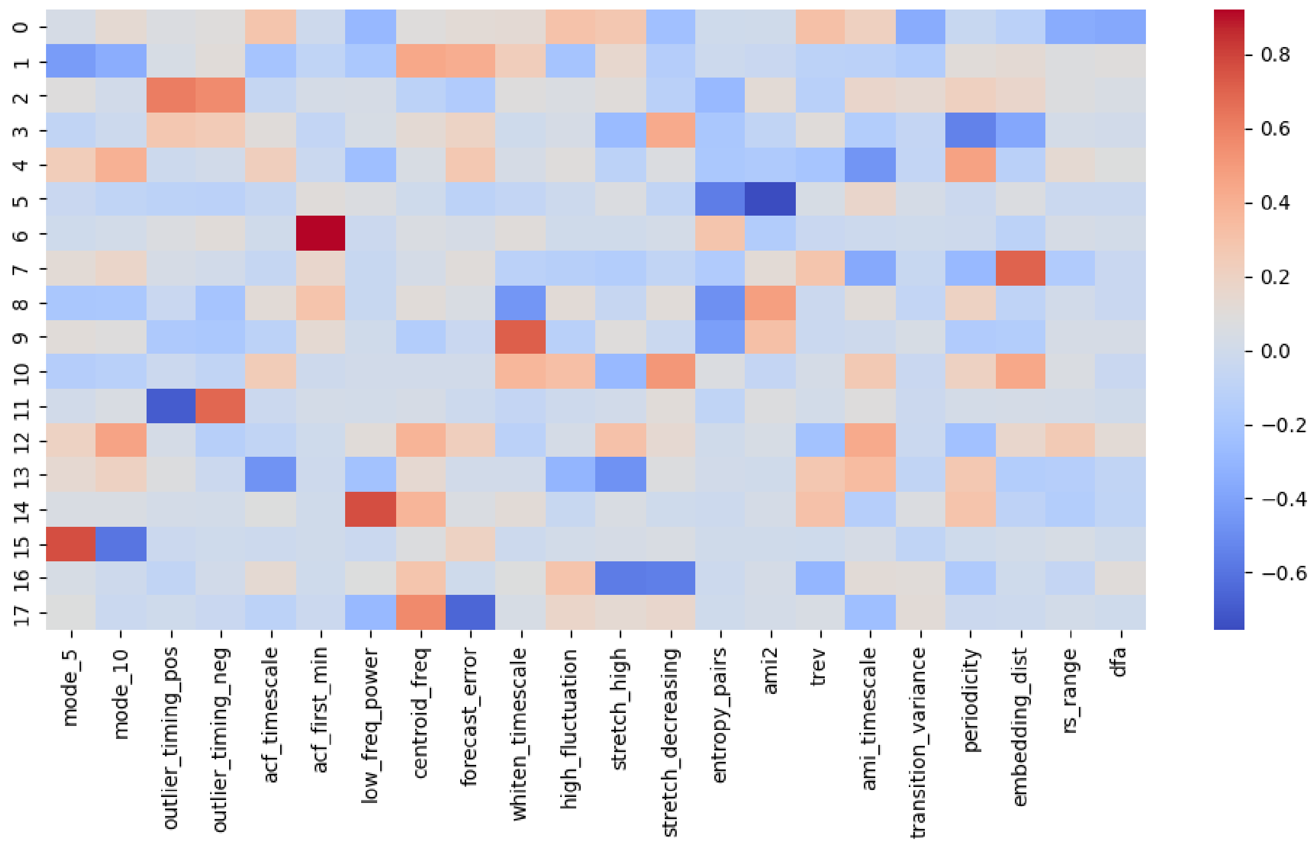


Figure 3.9: The heatmap of the 17 principal components shows the correlation coefficients between the original features and the principal components.

However, noting the arrangement of the principal components in the heat map in Fig.3.10, it can be seen that the components are generally independent of each other. This is evidenced by the fact that each component is necessary to explain each feature individually. The independence of the features is supported by the fact that the first 3-4 principal components generally try to explain the variance of all features, without being able to capture a significant percentage of variability. Indeed, we can achieve an acceptable and significant explained variance around 80% by considering at least 10 principal components. It is interesting to note that through the analysis of the correlation matrix (Fig. 3.3), we can see that features showing a correlation with each other tend to be explained by the same principal component. This suggests that the principal components are effective in capturing the internal structure of the data, highlighting the relationships among correlated features (*timig_timing_pos*,*Outlier_timing_neg*, *rs_range*,*dfa*). In figure 3.10, it can be observe that *Mod_5* and *Mode_10* exhibit varying correlations with different principal components, indicating that the patterns of distribution in their temporal values contribute to multiple dimensions of variation within the data. Similarly, *Outlier_timing_pos* and *Out-*

lier_timing_neg demonstrate significant correlations with the principal components, suggesting that the timings of extreme events, both positive and negative, play a substantial role in the data's variability. The differing correlations between positive and negative outliers imply that they might be influenced by distinct factors or have varying impacts on the overall pattern. Additionally, *Acf_timescale* and *Acf_first_min*, which are features related to autocorrelation, show considerable variability. Specifically, *Acf_first_min* has a high degree of variability, indicating that the first minimum of the autocorrelation differs significantly across the data. This suggests variations in periodicity patterns or short-term memory in the time series, necessitating a separate component for its explanation. Furthermore, *Low_freq_power* and *Centroid_freq* contribute differently to the principal components, reflecting the diversity in the frequency characteristics of the time series.

However, the low percentage of variability explained by the first principal components indicates that the dataset is complex and the features are quite independent of each other. This implies that there are multiple dimensions and factors contributing to the variability in the data, and it is not possible to significantly reduce the dimensionality without losing a considerable amount of information.

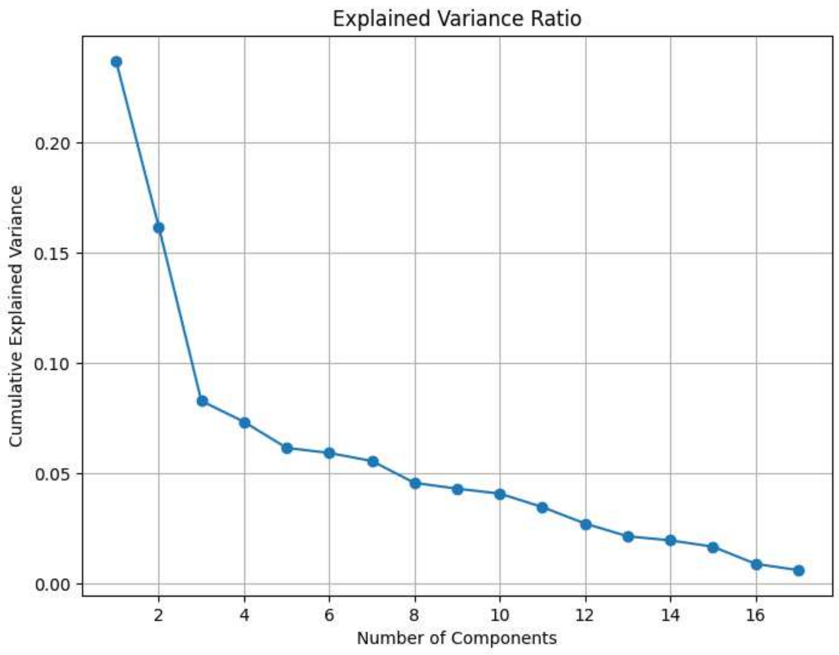


Figure 3.10: The variance scatter plot shows the variance explained for each principal component with remove features.

When a feature is correlated with multiple principal components, it indicates that this feature contributes to various dimensions of variability, capturing different and likely more significant aspects of the data. This suggests that a single principal component cannot fully explain the variability of the entire set of features, as the dataset's variability is spread across multiple levels. Consequently, some features are exclusively associated with a single principal component, highlighting their independence and unique contribution to the data's structure.

Subsequently, the PCA was recalculated on the reduced dataset to observe the impact on the explanation of the dataset with respect to the principal components (PCs) (Fig. 3.10). It is noted that the first principal component (PC1) decreases the amount of explained variance compared to the complete dataset, form 0.32 to 0.24 of explained variance. This is because the eliminated features were highly correlated with PC1. Generally, we see that some components increase their explained variance. However, the overall explained variance by the PCs remains low, necessitating the use of at least 10 PCs to achieve an explained variance around 85-90%.

In addition, to confirm the results, the SBI model was performed using two datasets: one containing all features and another excluding the selected features (*mode_5*, *transition_variance*, *rs_range*, *dfa*). The analysis was conducted using 10-fold cross-validation with 2 repetitions. For each iteration, half of the fold (approximately 50,000 samples) was used to calculate the posterior distribution of the parameters. This result can be by Table 3.2 shows that there is significant radical change compared to the model trained on the full dataset versus the reduced dataset, after removing the previously mentioned features, demonstrate a significant impact on the variability of the parameter posterior. Basically, these features prove to be very important for explaining the posterior distribution of the parameters, likely thanks to the SBI model, which is able to extract more hidden information from them.

Data	$[E/I]_{net}$	τ_{synE}	τ_{synI}	J_{ext}
Interquartile Range (IQR)				
Original	1.967	0.457	1.310	12.414
Remove	3.874	0.617	2.528	16.135
Parameter Recovery Error (PRE)				
Original	0.009	0.000	0.004	0.126
Remove	0.014	0.000	0.007	0.163

Table 3.2: The results of the SBI model with the original dataset and with the features removed dataset, after 10-fold cross-validation with 2 repetitions. it was taken half of test set sampled randomlyto expedite the progress.

3.2.2 Analysis of feature importance in SBI prediction models through SHAP values

The next step was to evaluate the significance of features within the framework of SBI predictions. SHAP values were exploited to provide insights into how each features affects the predictions made by SBI methods.To describe the posterior distribution estimated by SBI tools, it was calculated its mean and IQR. SHAP values were employed to explain how features affect these two metrics for all four parameters of the cortical model (Figs. 3.11, 3.12, 3.13 and 3.14). These bar plots of mean SHAP values show the average impact of each feature on the SBI model's predictions. Features with higher mean SHAP values are more important, meaning they have a greater impact on the model's output.

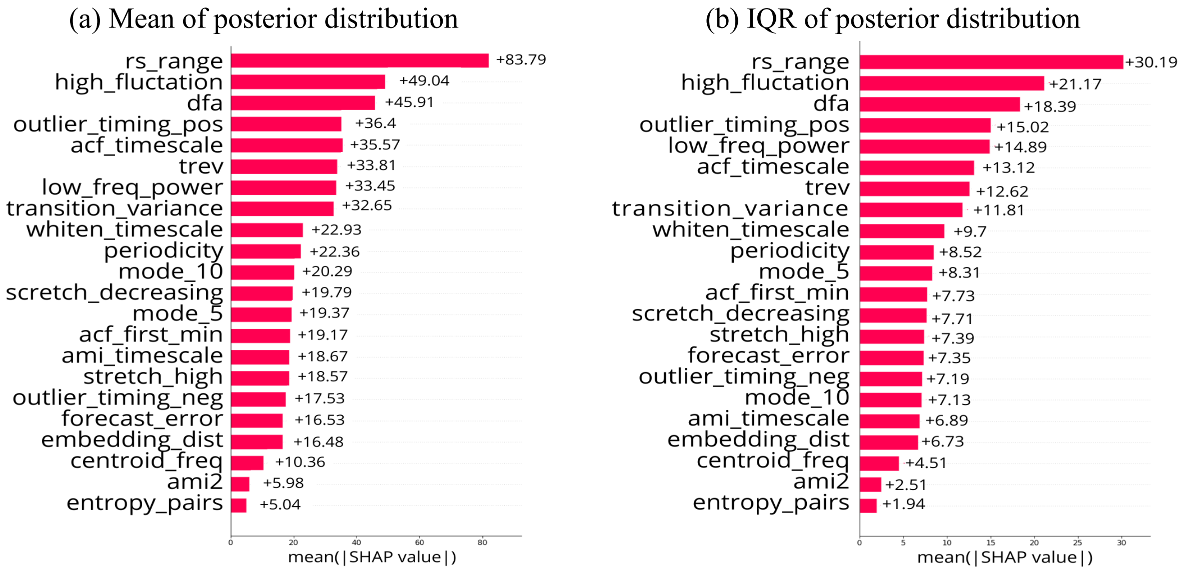


Figure 3.11: Mean SHAP values for $[E/I]_{net}$ calculated with two different metrics: (a) mean and (b) IQR of the posterior distribution. The mean SHAP values were obtained by averaging all SHAP values across the 10 fold evaluations.

The two bar (Fig. 3.11a) charts provided elucidate the influence of various features on the parameter $[E/I]_{net}$ using SHAP values. It is evident that the order of feature importance is largely consistent between the two metrics, namely means and IQR. The feature *rs_range* exhibits the highest impact on the mean of $[E/I]_{net}$, suggesting that variations in the range of residuals significantly influence this parameter. Furthermore, the feature *high_fluctuation* demonstrates a substantial effect, indicating that fluctuations in the data considerably impact $[E/I]_{net}$. Additionally, *dfa* emerges as a critical feature affecting $[E/I]_{net}$. This analysis underscores that a consistent set of features exerts a significant influence on $[E/I]_{net}$ across different metrics, thereby highlighting their essential role in the model.The charts represents (Fig. 3.12) demonstrates that the four most significant features remain consistent across both metrics. The most influential feature is *dfa*, indicating its crucial role in predicting tau_syn_E. Another important feature is *low_frequency_power*, which significantly impacts the mean of tau_syn_E, underscoring the importance of this frequency range. Additionally, the feature *mode_10* exhibits a considerable effect on tau_syn_E, highlighting the impact of specific modes on this parameter. Transition variance is also identified as a key factor, reflecting the significance of variance during transitions in influencing τ_{syn_E}. Furthermore, *high_fluctuations* in the data are shown to play an essential role. However, it is noteworthy that the mean SHAP value is not as high as those for the previously mentioned parameters.In the bar plot(Fig. 3.13), it is evident that the results of the most informative and significant features vary across the two matrices. Notably, the feature *outlier_timing_neg* consistently emerges as the most significant for the τ_{syn_I} parameter, while *dfa* and *low_freq_lower*also demonstrate considerable influence. However, it is important to note that the mean shape values of these features are somewhat lower compared to other parameters. In Figure 3.14, the features associated with the J_{ext} parameter demonstrate a consistent ranking across both charts, albeit with some variation in the absolute SHAP values depending on the context (mean vs. IQR). The *rs_range* feature emerges as the most influential in both contexts, though it exhibits greater influence when considering the mean compared to the IQR. Additionally, *transition_variance* and *high_fluctuation* are identified as highly influential across both metrics, though their relative rankings differ slightly. Notably, *transition_variance* exerts a marginally higher influence in the IQR context compared to the mean. Moreover, the mean value consistently presents higher SHAP values across features in comparison to other parameters.

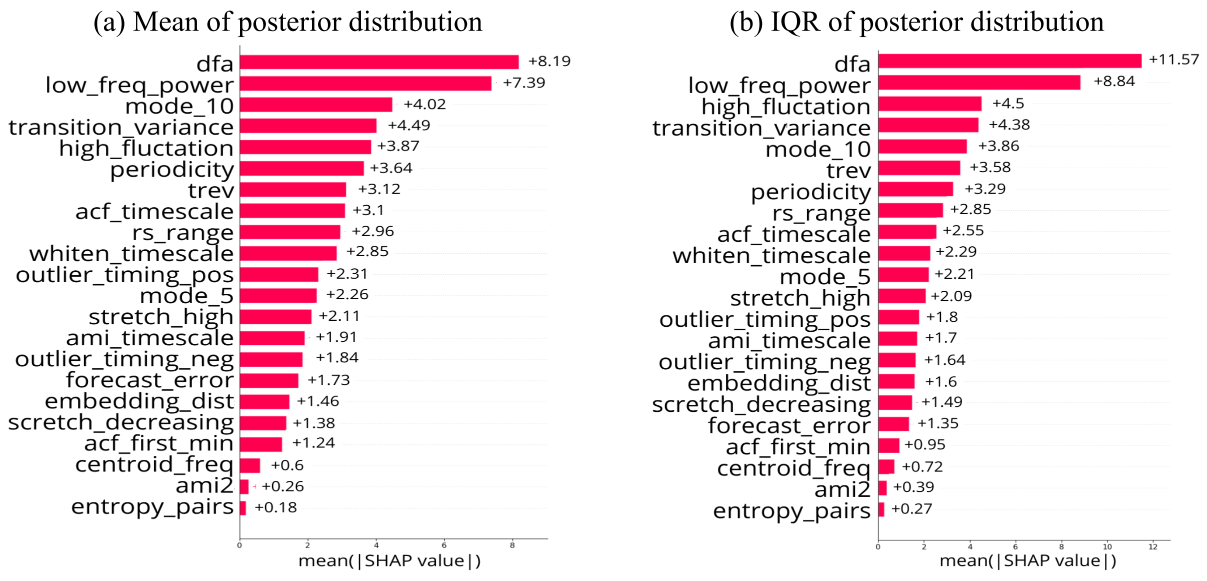


Figure 3.12: Mean SHAP values for τ_{synE}.

(a) Mean of posterior distribution (b) IQR of posterior distribution

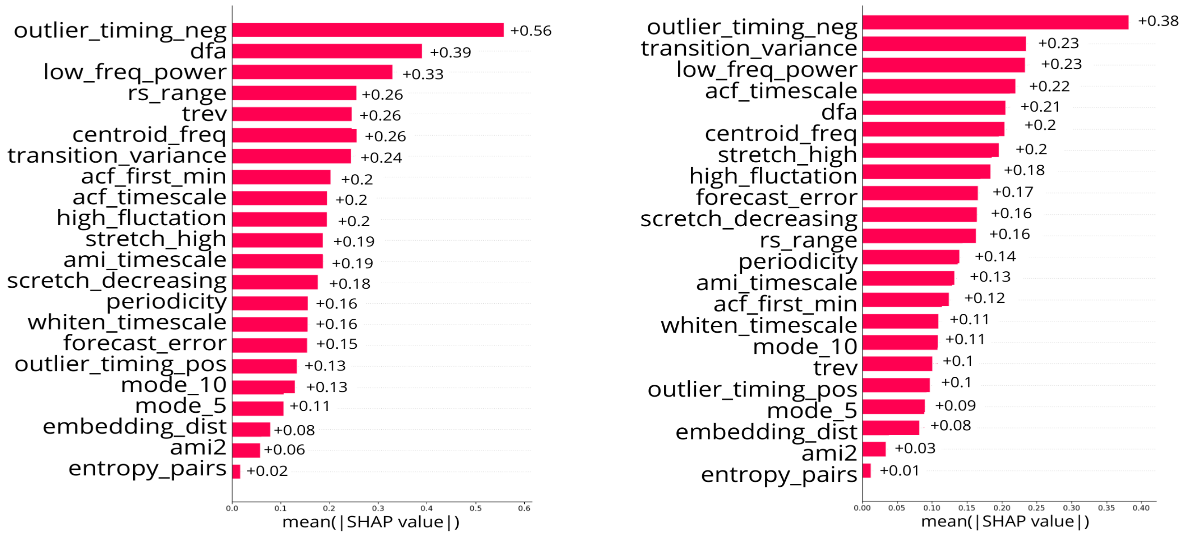


Figure 3.13: Mean SHAP values for τ_{synI}.

The analysis of the parameters $[E/I]_{net}$, τ_{syn_E}, τ_{syn_I}, and J_{ext} highlights the critical importance of specific characteristics in determining the behavior of the neural network and its response to external stimuli. In particular, the parameter [E/I]net, defined as the ratio between the product of excitatory and inhibitory synapses (JEE / JEI) and that of inhibitory and excitatory synapses (JIE / JII), quantifies the net effect of excitatory and inhibitory processes within the circuit model. This ratio can be interpreted as a global measure of E/I balance, providing an overall indication of how the two neuronal populations (excitatory and inhibitory) interact and influence the network's dynamics. Regarding $[E/I]_{net}$, the characteristics *rs_range*, *high_fluctuation*, and *dfa* emerge as the most influential, with a significant impact on both the central tendency (mean) and the variability (IQR) of the parameter. These characteristics, which capture the system's ability to detect long-range correlations and significant fluctuations, suggest that variations in these metrics can lead to substantial changes in the E/I balance, thereby affecting the overall equilibrium between excitatory and inhibitory processes. In the context of τ_{syn_E}, *dfa*, *low_freq_power*, and *mode_10* are identified as essential, with consistent influence on both metrics. The presence of *low_freq_power*, which measures the power of low frequencies in the signal, and *mode_10*, which assesses the distribution of the most frequent values in the time series, underscores the importance of modulating low_frequency components and the shape of the distribution in influencing the temporal response of excitatory neurons. For τ_{syn_I}, the analysis shows that *outlier_timing_neg* is the most significant feature, indicating that extreme negative events (deviations below the mean) play a crucial role in regulating the response time of inhibitory neurons. The persistence of characteristics such as *dfa* and *low_freq_power*, albeit with a different impact on mean and variability, highlights the complexity of inhibitory dynamics, where the network's response depends not only on temporal synchrony but also on the distribution of dominant frequencies. Finally, for J_{ext}, *rs_range* dominates the influence on the mean of the parameter, indicating its importance in regulating the network's global response to external inputs. However, its influence is slightly less pronounced on variability (IQR), suggesting that while *rs_range* plays a key role in modulating the network's average activity, characteristics such as *transition_variance* and *high_fluctuation* might be more relevant when the goal is to manage the variability of the response. This observation is confirmed by the 3.2, where a significant increase in IQR is observed once the features of interest are removed. In addition, it is noteworthy that *ami2* and *entropy_pairs*, despite being completely uncorrelated with the other features, prove to be entirely irrelevant with respect to all parameters, adding no significant information for the prediction of the posterior. This comprehensive analysis allows for the precise identification of the most influential characteristics on the network parameters, providing an effective guide for the selection and prioritization of features in predictive models. Focusing on the differences in the impact of features on mean and variability enables the optimization of models according to the specific objectives of the analysis, thereby improving predictive performance and the ability to explain the network's behavior.

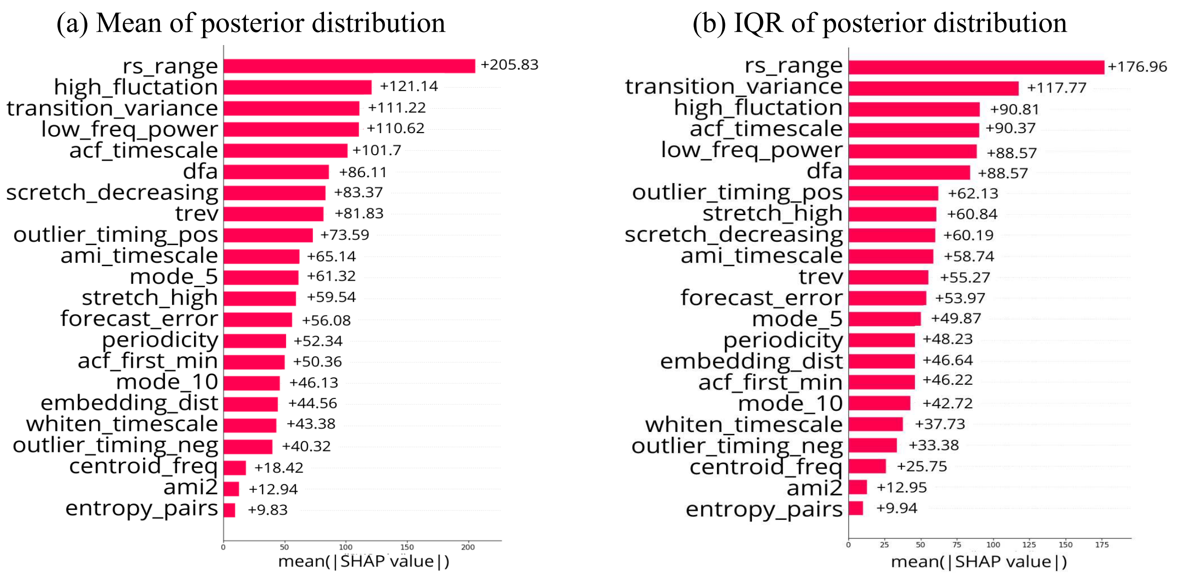


Figure 3.14: Mean SHAP values for J_{ext}.

3.3 Discussion

This study aimed to assess feature importance in SBI-based predictions using SHAP values to interpret and explain model outcomes. It was evaluated the effectiveness of this approach for predicting parameters of a spiking cortical circuit model using simulated LFP data. These results highlight the efficacy of this method in identifying crucial features from electrophysiological data, such as *dfa*, *rs_range*, *low_freq_power*, and *transition_variance*, which are essential for accurately predicting key parameters of cortical circuit activity. The significance of these features stems from their robust association with the dynamics of neural circuits. For instance, *dfa* is widely used to quantify long-range temporal correlations in time-series data, a characteristic frequently observed in brain activity. The prominence of *dfa* in these findings suggests that the temporal structure of fluctuations in neural signals holds critical information about the balance between excitatory and inhibitory processes in the brain. Similarly, *low_freq_power* is known to reflect slower brain oscillations, which play key roles in coordinating neural network activity and may directly correspond to changes in synaptic strength or network connectivity, as captured by parameters like $[E/I]_{net}$, τ_{synE}, and τ_{synI} in this study. The use of mutual information (MI) and correlation techniques allowed us to identify both linear and non-linear relationships between the features of the LFP and the parameters of interest, reinforcing the results obtained through SHAP and highlighting the most relevant features. At the same time, the PCA analysis confirmed the crucial role of these features by demonstrating their significant contribution to explaining the dataset's variance. This further validated their importance in describing the variability of the posterior distribution during parameter prediction using SBI.

Moreover, by incorporating SHAP values into this analysis, it was provided a clear and interpretable framework for examining how features influence the model's predictions. This approach allowed us to identify not only the most influential features but also those that had a negligible impact, such as *ami2* and *entropy_pairs*, which consistently received lower SHAP scores. This outcome aligns with mutual information (MI) analysis between features and parameters, reinforcing the reliability of these results and the consistency between different metrics.

However, while this study demonstrates the utility of SHAP values for feature importance assessment in SBI models, it also opens the door for further refinement. One important direction would be to compare this SHAP-based approach with other techniques specifically tailored for SBI models. For example, the FSLM algorithm[59], combined with a greedy feature selection process, is designed to evaluate how individual features influence the posterior uncertainty of parameters in SBI models. A comparative analysis between SHAP and FSLM could offer deeper insights into the nuances of feature importance, particularly regarding posterior uncertainty and its impact on model predictions. Such a comparison may reveal complementary strengths of these methods and guide researchers in selecting the most appropriate tools for different modeling scenarios.

Furthermore, this analysis could benefit from expanding the set of features used in the prediction models. While the *catch22* library provides a powerful summary of key time-series properties, other feature sets may capture complementary aspects of neural dynamics. For example, features from the *hctsa* toolbox, which offers an extensive range of over 7000 time-series features, could provide a more comprehensive picture of the underlying processes driving cortical circuit activity. Additionally, specific features known to reflect key aspects of neural signals, such as the 1/f slope, which is linked to the fractal nature of brain oscillations, or microstates, which represent quasi-stable patterns of brain activity over short periods, could further improve the predictive power of our models. By incorporating these additional feature sets, it was able to improve our ability to fully capture the complexity of neural dynamics and improve the accuracy of predictions for circuit parameters.

In conclusion, while this study has successfully demonstrated the most informative features for predictive parameters with the use of different techniques such as MI, PCA, and SHAP values to interpret the importance of features in SBI models, it also lays the foundation for future research. By refining this analysis with more specialized tools, applying the models to real-world data, and expanding the range of features considered, we can improve the robustness and interpretability of predictions in complex neural systems. This work highlights the importance of leveraging advanced computational techniques to decode the intricate relationships between neural dynamics and underlying circuit parameters, ultimately contributing to a deeper understanding of brain function.

Bibliography

[1] D. M. Kaplan, "Explanation and description in computational neuroscience," *Synthese*, vol. 183, no. 3, pp. 339–373, 2011.

[2] T. Trappenberg, *Fundamentals of computational neuroscience*. OUP Oxford, 2009.

[3] J.-Q. Kang, "Epileptic mechanisms shared by alzheimer's disease: Viewed via the unique lens of genetic epilepsy," *International Journal of Molecular Sciences*, vol. 22, no. 13, p. 7133, 2021.

[4] F. Darvas, D Pantazis, E Kucukaltun-Yildirim, and R. Leahy, "Mapping human brain function with meg and eeg: Methods and validation," *NeuroImage*, vol. 23, S289–S299, 2004.

[5] J. Boelts, P. Harth, R. Gao, *et al.*, "Simulation-based inference for efficient identification of generative models in computational connectomics," *PLOS Computational Biology*, vol. 19, no. 9, e1011406, 2023.

[6] J.-M. Lückmann, "Simulation-based inference for neuroscience and beyond," Ph.D. dissertation, Universität Tübingen, 2022.

[7] R. E. Kass, "Statistical inference: The big picture," *Statistical science: a review journal of the Institute of Mathematical Statistics*, vol. 26, no. 1, p. 1, 2011.

[8] M. Betancourt. "Probabilistic modeling and statistical inference." (2019), [Online]. Available: `https://betanalpha.github.io/assets/case_studies/modeling_and_inference.html`.

[9] G. Papamakarios, "Neural density estimation and likelihood-free inference," *arXiv preprint arXiv:1910.13233*, 2019.

[10] M. T. Moores, A. N. Pettitt, and K. L. Mengersen, "Bayesian computation with intractable likelihoods," *Case Studies in Applied Bayesian Data Science: CIRM Jean-Morlet Chair, Fall 2018*, pp. 137–151, 2020.

[11] R. D. Wilkinson, “Approximate bayesian computation (abc) gives exact results under the assumption of model error,” *Statistical applications in genetics and molecular biology*, vol. 12, no. 2, pp. 129–141, 2013.

[12] J. Alsing, B. Wandelt, and S. Feeney, “Massive optimal data compression and density estimation for scalable, likelihood-free inference in cosmology,” *Monthly Notices of the Royal Astronomical Society*, vol. 477, no. 3, pp. 2874–2885, 2018.

[13] C. M. Schafer and P. E. Freeman, “Likelihood-free inference in cosmology: Potential for the estimation of luminosity functions,” in *Statistical Challenges in Modern Astronomy V*, Springer, 2012, pp. 3–19.

[14] K. Csilléry, M. Blum, O. Gaggiotti, and O. François, “Approximate bayesian computation (abc) in practice.,” *Trends in ecology evolution*, vol. 25 7, pp. 410–8, 2010. doi: `10.1016/j.tree.2010.04.001`.

[15] O. Ratmann, O. Jørgensen, T. Hinkley, M. Stumpf, S. Richardson, and C. Wiuf, “Using likelihood-free inference to compare evolutionary dynamics of the protein networks of h. pylori and p. falciparum,” *PLoS Computational Biology*, vol. 3, no. 11, e230, 2007.

[16] K. Cranmer, J. Brehmer, and G. Louppe, “The frontier of simulation-based inference,” *Proceedings of the National Academy of Sciences*, vol. 117, no. 48, pp. 30 055–30 062, 2020.

[17] J.-M. Lückmann, “Simulation-based inference for neuroscience and beyond,” Ph.D. dissertation, Universität Tübingen, 2022.

[18] S. N. Wood, “Statistical inference for noisy nonlinear ecological dynamic systems,” *Nature*, vol. 466, no. 7310, pp. 1102–1104, 2010.

[19] L. F. Price, C. C. Drovandi, A. Lee, and D. J. Nott, “Bayesian synthetic likelihood,” *Journal of Computational and Graphical Statistics*, vol. 27, no. 1, pp. 1–11, 2018.

[20] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” *Advances in neural information processing systems*, vol. 30, 2017.

[21] D. Greenberg, M. Nonnenmacher, and J. Macke, “Automatic posterior transformation for likelihood-free inference,” in *International Conference on Machine Learning*, PMLR, 2019, pp. 2404–2414.

[22] P. J. Gonçalves, J.-M. Lueckmann, M. Deistler, *et al.*, “Training deep neural density estimators to identify mechanistic models of neural dynamics,” *Elife*, vol. 9, e56261, 2020.

[23] E. Hagen, S. H. Magnusson, T. V. Ness, *et al.*, “Brain signal predictions from multi-scale networks using a linearized framework,” *PLOS Computational Biology*, vol. 18, no. 8, e1010353, 2022.

[24] Y. Zerlaut, S. Zucca, S. Panzeri, and T. Fellin, "The spectrum of asynchronous dynamics in spiking networks as a model for the diversity of non-rhythmic waking states in the neocortex," *Cell reports*, vol. 27, no. 4, pp. 1119–1132, 2019.

[25] P. Martínez-Cañada, M. H. Mobarhan, G. Halnes, *et al.*, "Biophysical network modeling of the dlgn circuit: Effects of cortical feedback on spatial response properties of relay cells," *PLOS Computational Biology*, vol. 14, no. 1, e1005930, 2018.

[26] A. Mazzoni, S. Panzeri, N. K. Logothetis, and N. Brunel, "Encoding of naturalistic stimuli by local field potential spectra in networks of excitatory and inhibitory neurons," *PLoS computational biology*, vol. 4, no. 12, e1000239, 2008.

[27] N. Brunel, "Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons," *Journal of computational neuroscience*, vol. 8, pp. 183–208, 2000.

[28] P. Martínez-Cañada, S. Noei, and S. Panzeri, "Methods for inferring neural circuit interactions and neuromodulation from local field potential and electroencephalogram measures," *Brain Informatics*, vol. 8, no. 1, p. 27, 2021.

[29] P. Martínez-Cañada, T. V. Ness, G. T. Einevoll, T. Fellin, and S. Panzeri, "Computation of the electroencephalogram (eeg) from network models of point neurons," *PLOS Computational Biology*, vol. 17, no. 4, e1008893, 2021.

[30] F. Maestú, W. de Haan, M. A. Busche, and J. DeFelipe, "Neuronal excitation/inhibition imbalance: Core element of a translational perspective on alzheimer pathophysiology," *Ageing Research Reviews*, vol. 69, p. 101 372, 2021.

[31] S. S. Harris, F. Wolf, B. De Strooper, and M. A. Busche, "Tipping the scales: Peptide-dependent dysregulation of neural circuit dynamics in alzheimer's disease," *Neuron*, vol. 107, no. 3, pp. 417–435, 2020.

[32] N. Bertelsen, G. Mancini, D. Sastre-Yagüe, *et al.*, "Electrophysiologically-defined excitation-inhibition autism neurosubtypes," *medRxiv*, pp. 2023–11, 2023.

[33] T. Stavros, R. Federico, C. Carola, *et al.*, "Intrinsic excitation-inhibition imbalance affects medial prefrontal cortex differently in autistic men versus women," *Elife*, vol. 9, 2020.

[34] V. S. Sohal and J. L. Rubenstein, "Excitation-inhibition balance as a framework for investigating mechanisms in neuropsychiatric disorders," *Molecular psychiatry*, vol. 24, no. 9, pp. 1248–1257, 2019.

[35] E. Lee, J. Lee, and E. Kim, "Excitation/inhibition imbalance in animal models of autism spectrum disorders," *Biological psychiatry*, vol. 81, no. 10, pp. 838–847, 2017.

[36] S. B. Nelson and V. Valakh, "Excitatory/inhibitory balance and circuit homeostasis in autism spectrum disorders," *Neuron*, vol. 87, no. 4, pp. 684–698, 2015.

[37] B. D. Fulcher and N. S. Jones, “Hctsa: A computational framework for automated time-series phenotyping using massive feature extraction,” *Cell systems*, vol. 5, no. 5, pp. 527–531, 2017.

[38] C. H. Lubba, S. S. Sethi, P. Knaute, S. R. Schultz, B. D. Fulcher, and N. S. Jones, “Catch22: Canonical time-series characteristics: Selected through highly comparative time-series analysis,” *Data Mining and Knowledge Discovery*, vol. 33, no. 6, pp. 1821–1852, 2019.

[39] A. Tejero-Cantero, J. Boelts, M. Deistler, *et al.*, “Sbi: A toolkit for simulation-based inference,” *Journal of Open Source Software*, vol. 5, no. 52, p. 2505, 2020. doi: `10.21105/joss.02505`. [Online]. Available: `https://doi.org/10.21105/joss.02505`.

[40] M. Schmidt, R. Bakker, C. C. Hilgetag, M. Diesmann, and S. J. Van Albada, “Multi-scale account of the network structure of macaque visual cortex,” *Brain Structure and Function*, vol. 223, no. 3, pp. 1409–1435, 2018.

[41] E. Hagen, S. Næss, T. V. Ness, and G. T. Einevoll, “Multimodal modeling of neural network activity: Computing lfp, ecog, eeg, and meg signals with lfpy 2.0,” *Frontiers in neuroinformatics*, vol. 12, p. 92, 2018.

[42] “Supercomputación,” Universidad de Granada. (), [Online]. Available: `https://supercomputacion.ugr.es/`.

[43] K. Muthineni, U. K. Manchikatla, and R. Malisetty, “Performing multithreading and multitasking on linux operating system,” 2020. doi: `10.31237/osf.io/5zmg6`.

[44] M. Riaz, “On enhanced interquartile range charting for process dispersion,” *Quality and Reliability Engineering International*, vol. 31, pp. 389 –398, 2015. doi: `10.1002/qre.1598`.

[45] N. Tolley, P. L. Rodrigues, A. Gramfort, and S. R. Jones, “Methods and considerations for estimating parameters in biophysically detailed neural models with simulation based inference,” *PLOS Computational Biology*, vol. 20, no. 2, e1011108, 2024.

[46] P. Schober, C. Boer, and L. Schwarte, “Correlation coefficients: Appropriate use and interpretation,” *Anesthesia Analgesia*, vol. 126, pp. 1763–1768, 2018. doi: `10.1213/ANE.0000000000002864`.

[47] M. Mukaka, “Statistics corner: A guide to appropriate use of correlation coefficient in medical research.,” *Malawi medical journal : the journal of Medical Association of Malawi*, vol. 24 3, pp. 69–71, 2012. doi: `10.4314/MMJ.V24I3`.

[48] C. Shi, Q. Deng, P. Lu, M. Zhou, and G. Xiong, “Statistical analysis of medical experiment data for discovering groups of correlated symptoms,” *African Journal of Pharmacy and Pharmacology*, vol. 6, 2012. doi: `10.5897/AJPP12.493`.

[49] G. Chandrashekar and F. Sahin, “A survey on feature selection methods,” *Computers & electrical engineering*, vol. 40, no. 1, pp. 16–28, 2014.

[50] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[51] R. Kohavi and G. H. John, “Wrappers for feature subset selection,” *Artificial intelligence*, vol. 97, no. 1-2, pp. 273–324, 1997.

[52] G. H. John, R. Kohavi, and K. Pfleger, “Irrelevant features and the subset selection problem,” in *Machine learning proceedings 1994*, Elsevier, 1994, pp. 121–129.

[53] Z. Xu, I. King, M. R.-T. Lyu, and R. Jin, “Discriminative semi-supervised feature selection via manifold regularization,” *IEEE Transactions on Neural networks*, vol. 21, no. 7, pp. 1033–1047, 2010.

[54] S. M. Holland, “Principal components analysis (pca),” *Department of Geology, University of Georgia, Athens, GA*, vol. 30602, p. 2501, 2008.

[55] V. Shrivastava, S. S. Damodaran, and M. Kamble, “Adalward: A deep-learning framework for multi-class malicious webpage detection,” *Journal of Cyber Security Technology*, vol. 4, no. 3, pp. 153–195, 2020.

[56] L. Shapley, “A value for n-person games,” *Contributions to the Theory of Games*, pp. 69–79, 1953. doi: `https://doi.org/10.1515/9781400881970-018`.

[57] R. Rodríguez-Pérez and J. Bajorath, “Interpretation of compound activity predictions from complex machine learning models using local approximations and shapley values,” *Journal of Medicinal Chemistry*, vol. 63, no. 16, pp. 8761–8777, 2020, PMID: 31512867. doi: `10.1021/acs.jmedchem.9b01101`.

[58] R. Rodríguez-Pérez and J. Bajorath, “Interpretation of machine learning models using shapley values: Application to compound potency and multi-target activity predictions,” *Journal of computer-aided molecular design*, vol. 34, no. 10, pp. 1013–1026, 2020.

[59] J. Beck, M. Deistler, Y. Bernaerts, J. H. Macke, and P. Berens, “Efficient identification of informative features in simulation-based inference,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 19 260–19 273, 2022.