Technische Universität München – Zentrum für Mathematik

Lehrstuhl M12 – Mathematische Modellierung biologischer Systeme

Approximate Bayesian Model Selection for Local Cortical Networks at Cellular Resolution

Emmanuel G. Klinger

Vollständiger Abdruck der von der Fakultät für Mathematik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender:		Prof. Dr. Felix Krahmer
Prüfer der Dissertation:		
	1.	Prof. Dr. Dr. Fabian Theis
	2.	Prof. Dr. Moritz Helmstaedter
	3.	Prof. Dr. Jochen Triesch

Die Dissertation wurde am 17.01.2018 bei der Technischen Universität München eingereicht und durch die Fakultät für Mathematik am 24.08.2018 angenommen.

Acknowledgements

I am grateful to have worked with and to have learned from inspiring teachers and colleagues. I would like to thank Moritz Helmstaedter for conceiving and supervising the project, Fabian Theis for supervision and essential ideas, Carsten Marr for supervision, discussions and important advice in many situations. I thank Jan Hasenauer for the very friendly and extremely efficient working atmosphere. I thank Fabian Fröhlich for the time he invested to reproduce my results independently. I thank my colleagues Carolin Loos, Daniel Weindl, Lisa Bast, Benedikt Staffler, Anjali Gour, Florian Drawitsch and Manuel Berning for comments on manuscripts. I am also thankful that the Technical University Munich, the Helmholtz Zentrum München and the Max Planck Society gave me the opportunity to pursue this project. I would like to thank my wife for supporting me over the years of this thesis. Certainly, this would not have been possible without her.

Abstract

Testable hypotheses have been driving advances in the natural sciences. In physics, huge efforts were undertaken to acquire the data necessary to test some long-standing hypotheses. Neuroscience is not falling short in elaborate data acquisition schemes developed. Following data acquisition, often a plethora of hypotheses were formulated in terms of computational models aiming to explain these data. Nevertheless, it is still unknown which of the proposed computational models describes even a local cortical module best. To date, it has been difficult to compare these models systematically, not only because most of them lack analytically tractable likelihood functions, but also since the available data has been specific to certain behavioral paradigms hampering direct comparison. Neuroscience is about to enter the big data regime through the acquisition of connectomic data — structural neural network data at cellular resolution — which are independent of behavioral paradigms and will soon become available at the scale of local cortical modules. However, it is unclear whether connectomic data will possess the power to support or falsify models of local cortical computation, or not.

In this thesis, I therefore investigated whether connectomic data will be suitable for model selection of local cortical computational models. I reviewed the existing literature and compiled a broad range of relevant models proposed so far. I examined whether they already satisfied a set of common circuit constraints, corresponding to layer 4 of mouse primary somatosensory cortex, and adapted them to these constraints if necessary, controlling that they were still able to perform meaningful computations. Lacking analytically tractable likelihood functions, I propose to use a likelihood-free, approximate Bayesian model selection method on the adapted models. To cope with the algorithmic challenges and the high computational demands associated with likelihood-free inference, I developed a general-purpose, scalable and distributed, likelihood-free inference software framework, together with a scheme for the adaptive selection of sample sizes for this framework. Using this framework, I evaluated the proposed model selection method under different scenarios and designed a feasible connectomic circuit reconstruction experiment under which the method will become applicable.

The results suggest that connectomic circuit data will likely advance a mechanistic understanding of cortex through falsification of cortical models. This thesis hence, proposes a method to foster the fusion of experiment and theory in the neural sciences.

Zusammenfassung

Testbare Hypothesen haben die Naturwissenschaften seit jeher vorangebracht. In der Physik wurden immense Anstrengungen unternommen, um die notwendigen Daten zu akquirieren und damit mehrere etablierte Hypothesen zu überprüfen. Die Neurowissenschaften stehen der Physik in der Entwicklung elaborierter Datenakquisemethoden in nichts nach. Im Anschluss an die Datenakquise wurde häufig eine Vielzahl von Hypothesen als rechenfähige Modelle formuliert, um diese Daten zu interpretieren. Dennoch ist noch nicht einmal klar, welches Modell auch nur ein lokales kortikales Modul am besten beschreibt. Bis jetzt war ein systematischer Vergleich dieser Modelle sehr schwierig, nicht nur, weil die meisten keine analytisch zugängliche Likelihood-Funktion haben, sondern auch, weil die verfügbaren Daten bestimmten Verhaltensparadigmen eigen waren und somit den direkten Vergleich erschwerten. Mit der Akquise connectomischer Daten — zellulär aufgelöster struktureller neuronaler Netzwerkdaten — stehen die Neurowissenschaften kurz davor, in die Big Data Ära einzutreten. Diese connectomischen Daten sind unabhängig von Verhaltensparadigmen und bald in der Größenordnung lokaler, kortikaler Module verfügbar. Jedoch ist noch nicht klar, ob man mit connectomischen Daten Modelle lokaler kortikaler Berechnungen wird falsifizieren können oder nicht.

In dieser Dissertation habe ich deshalb untersucht, ob künftige connectomische Daten geeignet sein werden, um Modellselektion auf lokalen kortikalen rechenfähigen Modellen durchzuführen. Ich habe den derzeitigen Forschungsstand gesichtet und ein breites Spektrum bislang vorgeschlagener relevanter Modelle zusammengetragen. Ich habe dann untersucht, ob diese bereits eine Menge gemeinsamer struktureller Nebenbedingungen, Schicht 4 des primären somatosensorischen Kortex der Maus entsprechend, erfüllen und habe diese, sofern erforderlich, an die Nebenbedingungen angepasst. Dabei habe ich sicher gestellt, dass diese Modelle immer noch sinnvolle Berechnungen durchführen können. Da die meisten Modelle keine analytisch zugängliche Likelihood-Funktion besitzen, schlage ich vor, eine Likelihood-freie, approximativ Bayesianische Modellselektionsmethode auf die angepassten Modelle anzuwenden. Um der algorithmischen Herausforderungen und der hohen Anforderungen an die Rechenkapazität habhaft zu werden, entwickelte ich ein allgemein verwendbares, verteilt arbeitendes und skalierbares Software Framework zur Likelihood-freien Inferenz, kombiniert mit einer Methode zur adaptiven Selektion von Sample-Größen für dieses Framework. Mithilfe dieses Frameworks evaluierte ich die vorgeschlagene Modellselektionsmethode in verschiedenen Szenarien und entwarf ein durchführbares neuronales Schaltkreisrekonstruktionsexperiment, mit dem die Methode angewandt werden kann.

Die Resultate weisen darauf hin, dass neuronale Schaltkreisdaten durch die Falsifizierung von kortikalen Modellen ein mechanistisches Verständnis vom Kortex fördern werden. Diese Dissertation schlägt daher eine Methode vor, die eine Verknüpfung von Experiment und Theorie in den Neurowissenschaften vorantreibt.

Contents

1. Introduction

1.1. Aspects of the Relation Between Theory and Experiment

The exploration of natural phenomena through the combination of mathematical analysis and experimental measurements has shaped the natural sciences. In particular, experimental falsification of or experimental support for theoretically proposed hypotheses plays an essential role in the combination of theory and experiment (Popper 1934). A mathematical theory accompanying a certain hypothesis might of course be extremely interesting in its own right, even if it is not related to any experimental finding, but is then less useful to the natural scientist. Therefore, in some areas of physics, tremendous efforts were undertaken to support or falsify certain long-standing hypotheses. For example, the existence of a particle, now known as Higgs boson, was predicted theoretically as part of the standard model of particle physics in 1964 (Higgs 1964). Experimental support for the existence of this particle was found 48 years later, in 2012, by the European Organization for Nuclear Research (CERN) (Chatrchyan et al. 2012). Another example is given by gravitational waves. The prediction of gravitational waves was derived from Einstein's theory of general relativity in the early 20th century (Einstein 1916). It took a hundred years – until 2016 – that gravitational waves were experimentally observed by the Laser Interferometer Gravitational-Wave Observatory (LIGO) group (LIGO Scientific Collaboration and Virgo Collaboration et al. 2016).

In several ways life sciences, and in particular neuroscience, the sub-discipline concerned with the study of the nervous system, do now play a role similar to the one physics played 100 years ago; albeit neuroscience is not young at all. In fact, neuroscience dates back to ancient Egypt (Martín-Araguz et al. 2002) but has of course developed substantially since then. Nowadays it is a diverse multi-disciplinary research field incorporating methods and techniques from anatomy, chemistry, genetics and computational and mathematical modeling. One formidable challenge is trying to understand how cognitive function emerges from cells of the brain (Lisman 2015). The efforts undertaken are breathtaking: From 2005 to 2016 over 300,000 articles were published (Yeung, Goto, and Leung 2017). Over 24,000 scientists attended the largest neuroscience conference in the year 2016[1]. This is accompanied by large-scale public funding. To fund the human brain

[1] SfN, San Diego: 24,328 scientific attendees and 5,978 non-scientific attendees (https://www.sfn.org/

project (Markram 2012), the EU intends to provide one billion Euros. One of the goals of this project is to build the necessary infrastructure to enable advanced simulations and multi-scale modeling. Also the USA invest heavily in neuroscientific research. The Obama administration announced in 2013 the BRAIN initiative, a large-scale cooperative neuroscientific research initiative with the declared goal of obtaining a functional understanding of the human brain. It might rightfully be asked if all these efforts have lead to advances in the understanding of cognitive function and information processing performed in the brain? Has it been possible to support and to falsify theories through experimental measurements as it has been done in physics before?

1.2. Experimental Neuroscience at the Micro-, Macro- and Mesoscale

Numerous experiments were performed in the last decades, characterizing the brain at the micro-, macro- and mesoscale. At the microscale, for example, the anatomy of single neurons has been described already in the 19th century (Cajal 1888; Golgi 1873). Single neurons were reconstructed and hand-drawn in great detail (Fig. 1.1a). Studies of neuronal morphology were later complemented by studies of neuronal activity. The electrical activity of single neurons has been characterized (Edwards et al. 1989; Hamill et al. 1981; Lee, Epsztein, and Brecht 2009; Margrie et al. 2003; Sakmann and Neher 1984), for example through recordings of electrical voltage traces of spiking neurons (Fig. 1.1b).

At the macroscale, brains were characterized over 100 years ago (Brodmann 1909). For instance, the mammalian brain's cellular composition (cytoarchitecture) was described to consist of six layers (Fig. 1.2a). In addition, brain size related studies were carried out. The human brain consists of around 86 billion neurons (Azevedo et al. 2009) and is essentially a scaled-up version of a primate brain (Herculano-Houzel 2012). So is size the distinguishing factor? It has, however, also been observed, that sizes of local cortical modules, which supposedly fulfill the same function vary by at least a factor of two (Meyer, R. Egger, et al. 2013). This questions whether the sheer size is the explaining factor of the computational capacity of the human brain compared to other species.

At the mescoscale, activity of small neuronal populations was described (H. Berger 1929; Blanche et al. 2005; Mittmann et al. 2011; Ogawa et al. 1990) and was, for example, recorded through Ca^{2+} imaging (Fig. 1.2b). In terms of anatomy, however, not much beyond pairwise connectivity statistics is known about smaller populations consisting of

annual-meeting/past-and-future-annual-meetings/annual-meeting-attendance-statistics, retrieved Oct. 13, 2017)

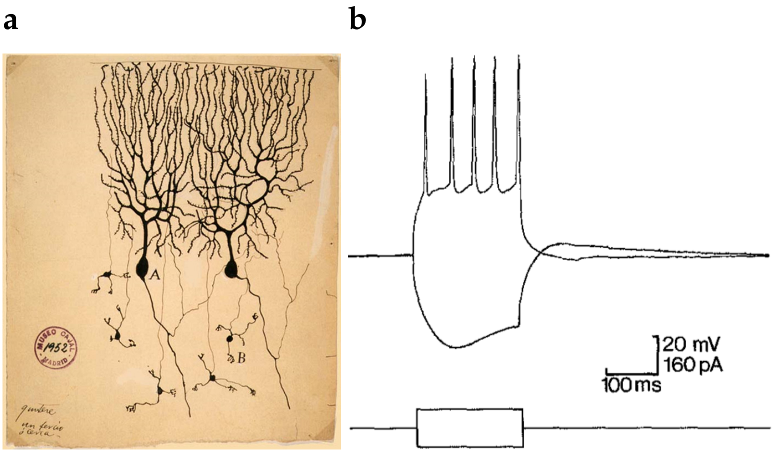

Figure 1.1. Shape and activity of single neuronal cells.
(a) Single cell morphology. Drawing of Purkinje cells (A) and granule cells (B) from pigeon cerebellum (Cajal 1888). (b) Voltage recordings from a rat hippocampal CA1 (Cornu Ammonis 1) pyramidal cell. The upper panel shows two superimposed whole-cell recordings of voltage responses evoked by a depolarizing (positive) and a hyperpolarizing (negative) current pulse (± 80 pA, as indicated in the lower panel). The depolarizing pulse resulted in a train of action potentials. Data was filtered (low-pass) at 2 kHz (-3 dB). (Edwards et al. 1989, Copyright 1989 by Springer, adapted with permission).

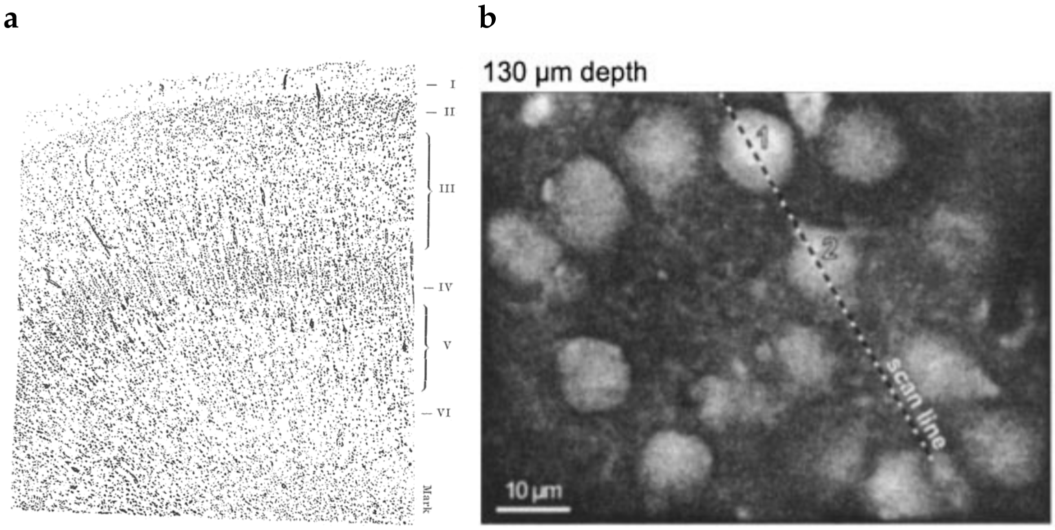

Figure 1.2. Macroscopic and mesoscopic experiments.
(a) Classification of adult human cortex into six cortical layers (Brodmann 1909). (b) In vivo recordings of Ca^{2+} transients evoked by whisker deflection. Image of layer 2/3 neurons in vivo in the barrel cortex of neuronalneuronalneuronala P13 mouse (Stosiek et al. 2003).

tens to hundreds of neurons (Feldmeyer, V. Egger, et al. 1999; Helmstaedter 2013; Koelbl et al. 2015). Only few attempts have been made to describe circuit structure beyond pairwise connectivity statistics. For example, it has been counted how often small circuit motives (sub-graphs), such as, e.g., the number of neurons connected in a triangle, occur in larger networks (Gal et al. 2017; Perin, T. K. Berger, and Markram 2011). How were these experimental findings across the scales used to advance the understanding of brain function?

1.3. Theoretical Neuroscience at the Micro-, Macro- and Mesoscale

Also theoretical modeling efforts were undertaken at the micro-, macro- and mesoscale. At the microscale, detailed models describing the dynamics of individual neurons have been proposed. A well-known example is the model by Hodgkin and Huxley (1952). These models helped to understand, for instance, the shapes of neuronal spikes (Figure 1.1b), but cannot explain cortical computations involving more than a single neuron. At the macroscale, methods from the network sciences have been applied, mostly to small graphs of entire brain modules instead of large graphs of individual neurons (Bullmore and Sporns 2009). Such graphs were commonly described by network statistics (Rubinov and Sporns 2010). For instance, it has been investigated, whether rich-club structure[2] can be related to diseases such as schizophrenia (Heuvel et al. 2013) and how costly it is for a brain to maintain such a rich-club organization (Collin et al. 2014). Other studies investigated the small-world[3] hypothesis for neural networks and asserted that whole brain networks are arranged in a fractal manner (Sporns 2006). It has also been tried to relate such structural network properties to functional network properties (Goñi et al. 2014; Sporns, Tononi, and Edelman 2000). Arguments were put forward that network function cannot be understood without structural description (Sporns 2013; Sporns, Tononi, and Kötter 2005). However, at the whole-brain scale it was mostly not possible to obtain insights how computations are performed mechanistically.

At the mescoscale, statistical physics approaches were used to explain the statistics of firing patterns, for example through mean field models (Vreeswijk and Sompolinsky 1996). Indeed, it is believed, that computation arises through interactions between individual neurons (Averbeck, Latham, and Pouget 2006). Due to the nature of these models,

[2] Informally, a graph is said to possess the rich-club property if high-degree nodes are preferentially connected to other high-degree nodes.

[3] Informally, a graph is said to possess the small-world property if the typical distance between two nodes grows only logarithimically with the number of nodes in the graph. That is, although most nodes are not directly connected to each other, the average distance between nodes is still small.

mechanistic insight into neuronal information processing were, however, scarce. It seems, that despite all these efforts, a mechanistic understanding of how neural computations are performed is still mostly lacking (Helmstaedter 2013) and that, unfortunately, "computational and theoretical neuroscience, in a similar way to some areas of physics, suffers from a falsification gap: its practitioners out-think, out-compute and out-simulate what is actually measured" (Denk, Briggman, and Helmstaedter 2012).

1.4. Connectomic Circuit Data: Decisive Information at the Mescoscale?

Recently, a new kind of neuroscientific data has emerged: so called "connectomes" or "connectomic" data. Connectomics is the field within neuroscience dedicated to the (dense) reconstruction of cortical circuits at single-cell resolution. A "connectome" is the wiring diagram or circuit diagram of a neuronal network. The oldest dense connectomic microcircuit reconstruction goes back to J. G. White et al. (1986). There, the connectome of the nematode *Caenorhabditis Elegans*, of about 1 mm length, was fully reconstructed. The effort spent was tremendous: The reconstruction was compiled from about 8000 images which were obtained from five different worms and took over a decade of effort. Much of the labor was performed manually. The complete connectome comprised 302 neurons. Its architecture was later examined in a descriptive fashion (Varshney et al. 2011), for example using techniques from spectral graph theory.

Today, state of the art connectomics relies heavily on electron microscopy (EM). Traditionally, EM was used for two-dimensional investigations (see for example Gray (1959)), but is now also used to reconstruct three-dimensional volumes of nervous tissue (Helmstaedter 2013). One method which leverages EM for connectomics reconstruction is serial section block face electron microscopy (SBEM) (Denk and Horstmann 2004) (Fig. 1.3a). In SBEM a block of nervous tissue is imaged slice by slice using scanning electron microscopy (SEM) (Fig. 1.3b). A thin slice is cut with a diamond knife (Fig. 1.3a). The surface of the block is then imaged (Fig. 1.3a). The process of cutting and imaging is repeated several thousand times. Such EM based methods were used to examine the connectivity in mouse primary visual cortex (Bock et al. 2011), the direction selectivity in the mouse retina (Briggman, Helmstaedter, and Denk 2011), the Drosophila visual motion detection circuit (Takemura et al. 2013) and the mammalian retina (Helmstaedter, Briggman, Turaga, et al. 2013; Kim et al. 2014). But this is only the dawn of the new neuroscientific sub-discipline connectomics: a research strategy for achieving the still far fetched goal of obtaining a human connectome has already been discussed (Sporns, Tononi, and Kötter 2005).

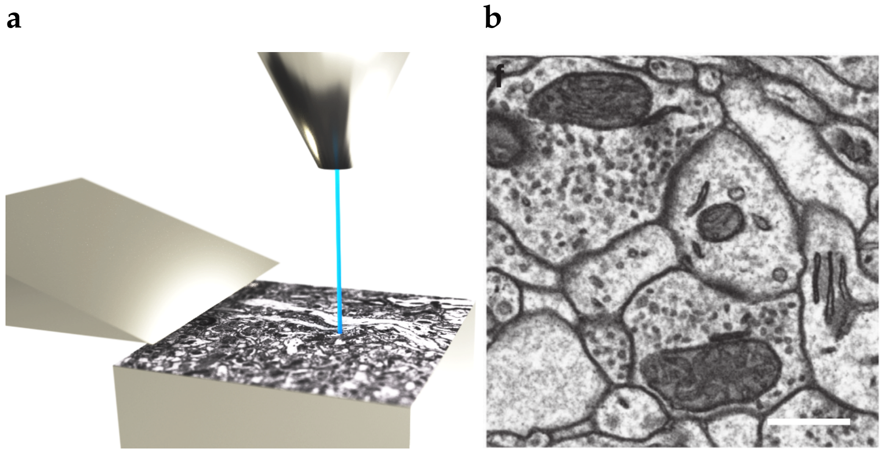

Figure 1.3. Serial section block face electron microscopy.
(a) Serial section block face electron microscopy. The surface of a block of nervous tissue is imaged with scanning electron microscopy (SEM). A thin slice of approximately 25 nm thickness is then cut from the block with a diamond knife. This process is repeated. (b) Electron microscopy data. Imaged tissue, Scale bar 200 μm (Hua, Laserstein, and Helmstaedter 2015, licensed under the Creative Commons Attribution 4.0 International license, `https://creativecommons.org/licenses/by/4.0/`)

Connectomics is challenging in several ways. Although image acquisition already poses substantial, mostly technical, challenges, the real bottleneck lies in data analysis (Helmstaedter 2013). The three dimensional image has to be segmented and individual neurites have to be reconstructed to infer the synaptic connections between neurons. Current machine learning techniques are still too error prone such that reconstruction has to be performed with the help of human annotators, which is time consuming, costly and does not scale well (Helmstaedter, Briggman, and Denk 2011). Substantial improvement is to be made in EM data analysis to scale connectomics to larger volumes and to eventually reach the goal of a human connectome.

Despite all these difficulties, the young field of connectomics is developing rapidly. This begs the question of how connectomic data can be used. Is the data useful at all? The range of positions is quite extreme. It has been stated that “there is no way to read the wiring diagram” (Bargmann 2012) and that connectomics was therefore mostly useless (Bargmann and Marder 2013). At the other extreme, it has been stated, that knowledge of the brain’s circuit structure at single-cell resolution is indispensable to understand cortical computations (Denk, Briggman, and Helmstaedter 2012). It seems therefore necessary to further examine this question and phrase it as concrete research problem. Can connectomic data be used to falsify or to support models of cortical computation? Can connectomics help to bridge the falsification gap neuroscience is experiencing (Denk, Briggman, and Helmstaedter 2012)?

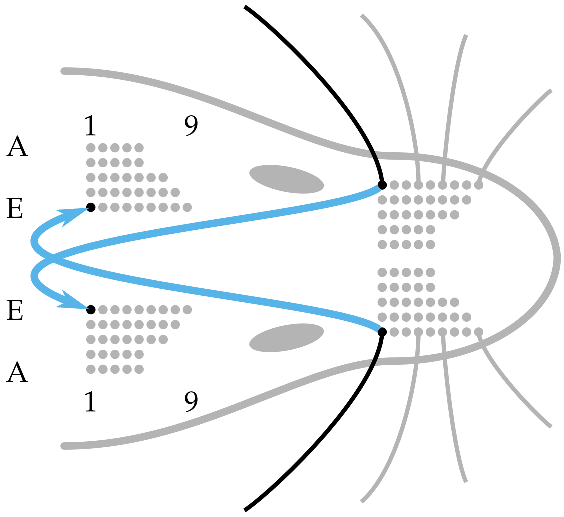

Figure 1.4. Rodent whisker system.
The whisker pattern is replicated in cortex on the contralateral side. Each whisker is associated with one barrel. The cortical L4 barrels are analogous to the vibrissae arranged in a grid consisting of five rows A-E. Each row contains up to nine barrels.

1.5. Layer 4 of Primary Somatosensory Cortex as Model System

Due to availability of data this question is best treated at a concrete, well-examined model system. A system of utmost importance is certainly the mammalian cortex, which is thought to be responsible for high-level cognitive functions (Singer 2013). Within cortex, a particularly well-examined model system is primary somatosensory cortex (S1), layer 4 (L4), "barrel cortex" (Diamond et al. 2008; Feldmeyer, Brecht, et al. 2013). Its architecture is modular, consisting of 34 alike barrels on each side (Figure 1.4). Each of the barrels receives primarily input from one associated whisker. This structure is present in rodents such as mice and rats and is the first cortical region receiving whisker input (Diamond et al. 2008; Feldmeyer 2012; Feldmeyer, Brecht, et al. 2013; Helmstaedter, Kock, et al. 2007). Rodents use their whiskers to explore their environment and to sense and discriminate surface textures (Morita et al. 2011). It is one of the major model systems of tactile sensation (Morita et al. 2011).

1.6. Outline and Main Scientific Contributions

In this thesis, I examined if and how connectomic data can be used to falsify or support hypotheses of local cortical computational models in a concrete model system, a single barrel of mouse L4, S1. Since connectomic reconstruction experiments are extremely expensive and labor-intensive it is necessary to design a useful and reliable analysis–experiment pairing pre hoc. Ideally, it could be analytically evaluated how useful and reliable a given analysis–experiment paring is, but in the examined scenario this is impossible. Instead, simulations were employed, which are still cheap compared to the efforts involving real experiments. Here, I focused on the long-standing issue of local cortical model selection, proposing a method for such model selection and assessing through numerical simulations with known ground truth which kind of experiment makes this method reliably applicable.

I assembled seven competing hypotheses and investigated whether they have concrete implementations satisfying a common set of already known L4, S1 circuit constraints, or whether some of the hypotheses have to be rejected based on prior knowledge (Chap. 2). I developed and implemented adaptations of the models to the common circuit constraints, turning rather abstract hypotheses into concrete mechanistic model. Moreover, I controlled the functional viability of these models after adaptation. For model selection, I pursued a Bayesian approach. Again, an anaytical treatment was impossible, lacking efficiently computable expressions for the likelihood-functions. I therefore employed a likelihood-free Approximate Bayesian Computation - Sequential Monte Carlo (ABC-SMC) scheme. However, it was computationally challenging to apply such a scheme, primarily because the investigated models have to be simulated repeatedly. In addition, such a scheme has many tunable options and involves algorithmic challenges as well as challenges of data organization, making its application rather intricate. Since no available ABC framework was able to employ distributed hardware for model selection, and none of the available frameworks was extensible in this respect, it was necessary to develop a framework, capable to deal with the computational challenges faced here. I therefore developed a scalable software framework for distributed, likelihood-free ABC-SMC inference (Klinger, Rickert, and Hasenauer 2017) and a scheme for automated selection of population sizes – a difficult-to-tune option – for this framework (Klinger and Hasenauer 2017) (Chap. 3). This framework is to date the one which scales best with distributed hardware, and also the most user-friendly and extensible one, allowing not only application of ABC-SMC schemes, but also experimentation with new ones. I applied this ABC-SMC framework to model selection for connectomic circuit data (Chap. 4). The proposed method proofed robust to noisy experimental conditions allowing to design a concrete cortical reconstruction experiment (Chap. 4). In this thesis, I hence propose a method to advance a mechanistic understanding of cortical computation in a concrete local cortical circuit module to foster the fusion of experiment and theory in the neural sciences.

The main scientific contributions are

- Proposal, implementation and analysis of the first method which uses structural

circuit data at cellular resolution for model selection of local cortical models. The approach is new in several ways: the to-be-employed data has never been examined before in such a context, the analysis is pre hoc specified thus possesses greater power and enforces the formulation of testable hypotheses, and the approach is mathematically well-grounded.
- Proposal of concrete implementations and adaptations of rather abstract hypotheses of cortical processing. It is examined whether several long-standing hypotheses are compatible with known cortical constraints, or whether they had to be dismissed based on available knowledge.
- Demonstration that structural circuit data combined with the here proposed method is decisive for local cortical models. This shows that connectomics, a big-data neuroscientific approach, has the potential to advance a mechanistic understanding of cortical computations, falsifying and supporting theories of cortical computation via the proposed method instead of merely collecting data.
- Design of a concrete connectomic reconstruction experiment for model selection of local cortical models. It is quantified with which precision and which fraction of a circuit need to be reconstructed. The requirements are experimentally satisfiable, the method is thus practically applicable.
- Design and development of the pyABC general-purpose software framework for distributed, likelihood-free parameter inference and model selection. According to benchmarks, this is currently the framework which scales best with distributed infrastructure. This framework was already used by several groups worldwide. For example, it was applied to parameterize stochastic models of HIV spread on a single cell level.
- Development and analysis of a scheme for adaptive selection of population sizes in Approximate Bayesian Computation - Sequential Monte Carlo (ABC-SMC). This facilitates practical application of ABC-SMC, reducing the need to manually tune options.

These contributions are partially already published at peer-reviewed journals or are currently under review. If applicable, it is indicated at the beginning of chapters, sections, or in figure captions which manuscripts cover similar content. The relevant manuscripts are

- Klinger, Emmanuel, and Jan Hasenauer. "A Scheme for Adaptive Selection of Population Sizes in Approximate Bayesian Computation - Sequential Monte Carlo." In Computational Methods in Systems Biology, 128–44. Lecture Notes in Computer Science. Springer, Cham, 2017.
- Klinger, Emmanuel, Dennis Rickert, and Jan Hasenauer. "pyABC: distributed,

likelihood-free inference." Available at BioRxiv, July 17, 2017, 162552. Currently under review at Oxford Bioinformatics.

- Klinger, Emmanuel, Carsten Marr, Fabian J. Theis, and Moritz Helmstaedter. "Cellular connectomes as arbiters of local computational models in the cerebral cortex". Currently under review at Nature Communications.

None of these manuscripts has a shared first authorship. In addition I have been supervising the preparation of a manuscript:

- Waibel, Ricardo, Emmanuel Klinger, and Moritz Helmstaedter. "A topologically invariant error metric for the comparison of connectomic segmentations". Currently in preparation.

2. Adaptation to Known Constraints and Functional Viability of Local Cortical Models

Section is similar to Klinger, Marr, Theis, Helmstaedter (under review)

In this chapter, the graph theoretical notation used throughout the chapter is first introduced. Next, the circuit constraints which all candidate models of layer 4 (L4) of primary somatosensory cortex (S1) have to satisfy are described. The candidate models were extracted from or inspired by existing literature where they were mostly proposed in the form of rather abstract computational concepts. These models are only now, in this thesis, systematically compiled and their compatibility with known circuit constraints of L4 of S1 is examined. Each model is concretely adapted (if necessary) and implemented (if possible) to satisfy the constraints of L4, S1. Additionally, it is verified that the adapted models are still functional. These models were not until this thesis implemented as concrete candidates models for L4, S1. Here, rather abstract concepts are finally turned into concrete and falsifiable candidate models.

The compiled and adapted models cover a range of structural and computational hypotheses.

- The feature vector recombination (FEVER) model (Sect. 2.3) is a model of short-term memory. The short-term memory property is achieved through a connectivity rule, which aims to form outgoing projections such that the postsynaptic targets' feature vectors additively recombine to resemble the projecting neuron's own feature vector.
- The Erdős-Rényi echo state network (ER-ESN) model (Sect. 2.5) is a model of general cortical computation. In this model, the synaptic weights of L4 are not trained. Instead, only the readout neurons' weights are trained. Connectivity is pairwise random.
- The exponentially decaying liquid state machine (EXP-LSM) model (Sect. 2.6) is also a model of general cortical computation. Also in this model, only the readout neurons' weights are trained. In contrast to the Erdős-Rényi echo state network (ER-ESN) model, connectivity is distance dependent.
- The layered network (LAYERED) model (Sect. 2.7) is again a model of general cor-

tical computation. It is composed of sequentially stacked recurrent neural networks. In this model, also the L4 neurons are trained. It represents the long-standing hypothesis of hierarchical cortical processing organized in layers.

- The anti phase inhibition (API) model (Sect. 2.8) aims to perform stimulus tuning or sharpening. Connectivity is formed such that excitatory neurons project preferentially to other neurons which are thought to represent a similar stimulus, while inhibitory neurons project preferentially to other neurons which are thought to represent a dissimilar stimulus.
- The self-organizing recurrent neural network (SORN) model (Sect. 2.9) shapes cortical connections through a set of rules which modify synaptic weights in an activity dependent manner. For example, connections between neurons which fire together are strengthened.
- The synfire chain (SYNFIRE) model (Sect. 2.10) examines if and how activity can be propagated in a synchronous fashion through overlapping chain structures. The network is recurrent but incorporates aspects of feed-forward processing.

2.1. Graph Theoretical Notation

The number of excitatory neurons is denoted by n_e, the number of inhibitory neurons by n_i. The size of a network is denoted by $n = n_\mathrm{e} + n_\mathrm{i}$. Adjacency matrices are denoted by $W \in \mathbb{R}^{n \times n}$. The excitatory part of the adjacency matrix W is denoted by $W_\mathrm{E} \in \mathbb{R}^{n \times n_\mathrm{e}}$, $(W_\mathrm{E})_{k,l} = W_{k,l}$. The inhibitory part of the adjacency matrix W is denoted by $W_\mathrm{I} \in \mathbb{R}^{n \times n_\mathrm{i}}$, $(W_\mathrm{I})_{k,l} = W_{k,l+n_\mathrm{e}}$.

A directed graph G is also represented as a pair $G = (V, E)$ of a vertex set V and an edge set $E \subset V \times V$. In this notation, directed edges are therefore tuples $(u, v) \in E$, meaning that a directed edge, originating at vertex u and connecting to vertex v exists in the graph. The vertex u is called the outgoing vertex, the vertex v the ingoing vertex of the edge (u, v). Using neuroscientific terminology, the vertex u is also called presynaptic and the vertex v postsynaptic.

2.2. Circuit Constraints

A C2 barrel (Fig. 1.4) is composed of approximately 2000 neurons (Feldmeyer 2012; Meyer, Schwarz, et al. 2011). Of these neurons, roughly 90% are excitatory and 10% are inhibitory (Feldmeyer 2012; Meyer, Schwarz, et al. 2011). These neurons establish about 2 million chemical synapses within a single barrel. The fraction of connected cell pairs – the average connectivity – has often been estimated based on data obtained from whole-cell recordings as listed in Table 2.1 (Beierlein, Gibson, and Connors 2003; Feldmeyer,

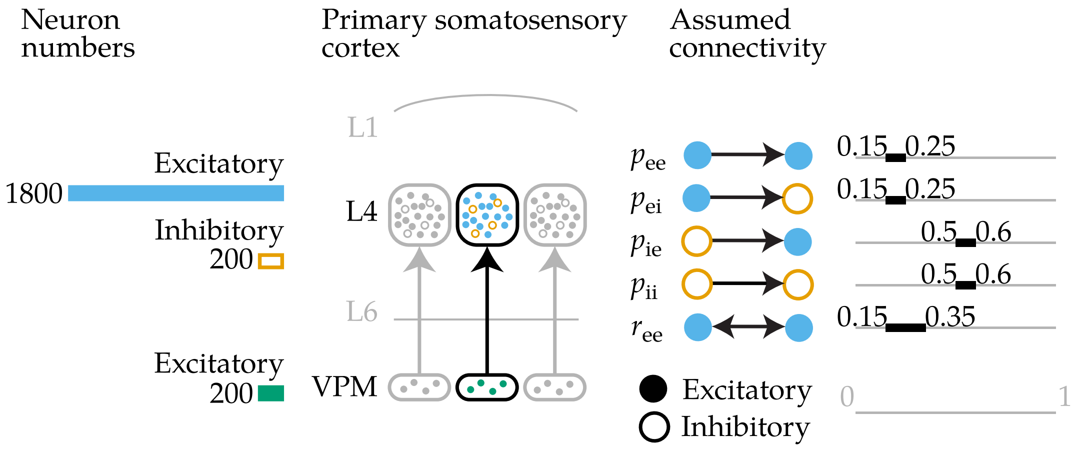

Figure 2.1. Assumed circuit constraints.
A layer 4 (L4) barrel is assumed to consist of 1800 excitatory neurons ● and 200 inhibitory neurons ○. A single barreloid of the ventral posteromedial nucleus (VPM), providing input to its corresponding barrel, is assumed to consists of 200 excitatory neurons ●. The excitatory-excitatory connectivity (p_{ee}) and excitatory-inhibitory connectivity (p_{ei}) are assumed to lie in the inverval $[0.15, 0.25]$. The inhibitory-excitatory connectivity (p_{ie}) and inhibitory-inhibitory connectivity (p_{ii}) are assumed to lie in the inverval $[0.5, 0.6]$. The excitatory-excitatory reciprocity (r_{ee}) is assumed to lie in the inverval $[0.15, 0.35]$. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

V. Egger, et al. 1999; Gibson, Beierlein, and Connors 1999; Gibson, Beierlein, and Connors 2005; Koelbl et al. 2015; Lübke et al. 2000). It was found that excitatory neurons connect on average to about 20% of other neurons within the same barrel, inhibitory neurons connect to about 50%-60% of other neurons within the same barrel. Note the substantial variance in measured connectivities as obtained by the various studies (Table 2.1) which might be due to the limited accuracy of whole-cell recordings for connectivity statistics. These measurements are obtained in slices where the typical dimension of a slice is much smaller than that of a barrel. Therefore, many connections are cut off. The values listed in Table 2.1 are therefore to be taken with care and are to be considered as indicators of a biologically plausible regime, rather than as precise, trustworthy values. Also, these numbers vary across individual rodents and across barrels (see Fig. 1.4 for the layout of the individual barrels in primary somatosensory cortex). The numbers taken here as circuit constraints (Fig. 2.1) constitute therefore what is considered a prototypical barrel. Once an actual barrel connectome is measured, these numbers are updated with the actually obtained values. To judge whether connectomes can be used for model discrimination or not, it is only important to use values roughly in the correct regime. The results obtained later in this work are robust to the precise chosen values. That is, if the connectivity is assumed to be 15% or 30% has almost no influence on the results.

Table 2.1. Pairwise connectivity and reciprocity statistics from paired whole-cell recordings. **Columns.** Pub.: the literature reference. Spec.: the species. Pre: the presynaptic neuronal type. Post: the presynaptic neuronal type. p: the connectivity, $p = n_{\text{conn}}/n_{\text{test}}$. r: the reciprocity, $r = n_{\text{rec}}/n_{\text{conn}}$. n_{test}: the number of tested pairs. n_{conn}: the number of pairs found to be connected. n_{rec}: the number of pairs found to be reciprocally connected. Distance: the maximal inter-soma distance of the recorded neurons. **Pubs.** 1: Koelbl et al. (2015). 2: Gibson, Beierlein, and Connors (1999). 3: Beierlein, Gibson, and Connors (2003). 4: Gibson, Beierlein, and Connors (2005). **Cell types.** SS: Spiny stellate (excitatory). RS: Regular spiking (excitatory). FS: Fast spiking (inhibitory). LTS: Low threshold spiking (inhibitory). **Connectivity.** Excitatory connectivity is visualized in blue (0.5 means 50% excitatory connectivity), inhibitory connectivity in orange (0.5 means 50% inhibitory connectivity).

Pub.	Spec.	Pre	Post	p	r	n_{test}	n_{conn}	n_{rec}	Distance
1	Rat	FS	SS	0.67	0.69	43	29	20	
2	Rat	RS	RS	0.11	0.00	80	9	0	
2	Rat	RS	FS+LTS	0.41	0.23	54	22	5	
2	Rat	LTS	LTS	0.15	0.00	13	2	0	
2	Rat	FS	LTS	0.34	0.64	32	11	7	
2	Rat	FS+LTS	RS	0.52	0.14	71	37	5	
2	Rat	LTS	FS	0.53	0.41	32	17	7	
2	Rat	FS	FS	0.62	0.25	39	24	6	
3	Rat	RS	RS	0.12	0.09	89	11	1	$< 50\,\mu m$
3	Rat	RS	FS	0.43	0.46	172	74	34	$< 50\,\mu m$
3	Rat	RS	LTS	0.57	0.36	63	36	13	$< 50\,\mu m$
3	Rat	LTS	RS	0.35	0.50	74	26	13	$< 50\,\mu m$
3	Rat	FS	RS	0.44	0.41	190	83	34	$< 50\,\mu m$
4	Rat	LTS	LTS	0.08		24	2		$< 100\,\mu m$
4	Rat	FS	LTS	0.36		55	20		$< 100\,\mu m$
4	Rat	LTS	FS	0.62		55	34		$< 100\,\mu m$
4	Rat	FS	FS	0.74		57	42		$< 100\,\mu m$

2.3. FEVER: Feature Vector Recombination

2.3.1. Key Concepts

Information is persistently represented in cortex over time scales much larger than the dynamics of an individual neuron permits (Hogan and Flash 1987). Such persistent representation is possibly explained by an encoding of the information in the population activity of a cortical network. A recurrently connected network is able to maintain activity over larger time scales than individual neurons. The feature vector recombination (FEVER) network was proposed by Druckmann and Chklovskii (2012) to explain persistent cortical representations despite time varying neuronal activity. In the FEVER model, each neuron $i \in \{1, \dots, n\}$ is associated with a feature vector $f_i \in \mathbb{R}^{d_\mathrm{f}}$ with feature space dimension d_f. The overall represented state s of the network is additively obtained as the sum over all the feature vectors, weighted by the corresponding neuron's activity. In matrix notation,

$$s = Da, \tag{2.1}$$

in which $a = (a_1, \dots, a_n)^t$ denotes the vector of internal neuronal activity and $D = (f_1, \dots, f_n)$ the matrix[1] of feature vectors. The internal neuronal activity $a_i \in \mathbb{R}$ evolves over time according to the system of ordinary differential equations

$$\tau \dot{a} = -\sigma(a) + W\sigma(a) + u(t) \tag{2.2}$$

in which $W \in \mathbb{R}^{n \times n}$ denotes the adjacency matrix, $\tau > 0$ is the neuronal time constant, $\sigma : \mathbb{R}^n \to \mathbb{R}^{+n}$ an activation function mapping internal neuronal activity to neuronal output and $u(t)$ is the external input to the network. The state s is persistently represented when

$$\dot{s} = 0 \tag{2.3}$$

is satisfied. By elementary calculation, it follows from Eqs. (2.1) to (2.3) for the adjacency matrix the requirement

$$D = DW, \tag{2.4}$$

to achieve persistent state representation Eq. (2.3). That is, for each neuron, the sum over its postsynaptic targets' feature vectors, weighted by the synaptic weights, recombines to the neuron's own feature vector (Fig. 2.2). Notably, this connectivity rule is local in the sense that each neuron chooses its postsynaptic partners independent of the other neurons' postsynaptic partners. In general, Eq. (2.4) has no unique solution. A trivial solution is given by $W = I$. However, biologically, this corresponds to a network in which

[1] The matrix D is often called "dictionary" (of feature vectors) in the sparse coding and dictionary learning literature and is therefore abbreviated by "D" (Aharon, Elad, and Bruckstein 2006; Mairal et al. 2009).

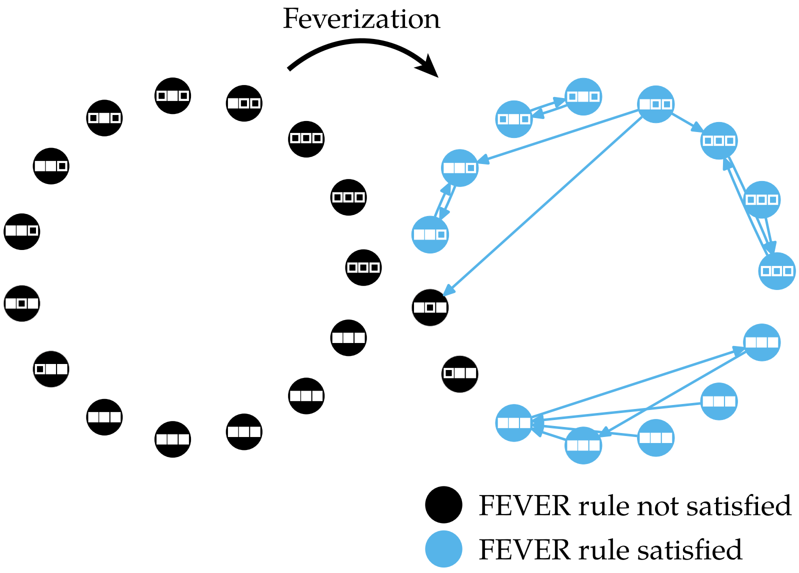

Figure 2.2. The FEVER rule.
For each neuron, the sum of its postsynaptic neurons' feature vectors is intended to approximate the neuron's own feature vector. The filled squares "■" represent +1, the empty squares "□" represent −1. The FEVER rule is not necessarily perfectly satisfied for each neuron.

each neuron makes a single self-connection – an autapse – only. Such self-connections are extremely rare in biological neuronal networks and are therefore discarded as biologically not meaningful. To obtain more specific solutions, a sparsity constraint is imposed: a solution with as few connections as possible is sought. This is biologically meaningful as it minimizes the cost to build and maintain the network structure. This sparsity constraint translates to the minimization problem

$$\min_W \sum_{i,k:W_{i,k}\neq 0} 1 \quad \text{with} \quad D = DW, \; \operatorname{diag} W = 0, \tag{2.5}$$

which, however, cannot be efficiently solved. Instead, the relaxed minimization problem

$$\min_W \|D - DW\|_2^2 + \lambda \|W\|_1, \quad \operatorname{diag} W = 0 \tag{2.6}$$

is considered for fixed $\lambda \geq 0$. The parameter λ governs the sparsity of the solution. For $\lambda = 0$, the solution is generally not sparse but fully populated $W_{i,k} \neq 0 \; \forall i \neq k$. The larger λ, the more entries of W are exactly zero. Opposed to Eq. (2.5), the relaxed minimization problem Eq. (2.6) can be efficiently numerically solved with the least absolute shrinkage and selection operator (LASSO) algorithm (Tibshirani 1996). Note that as result of the minimization, the FEVER rule is usually not perfectly satisfied, i.e. $\|D - DW\|_2^2 > 0$. This is acceptable, as the proper functioning of a FEVER network is not evaluated according

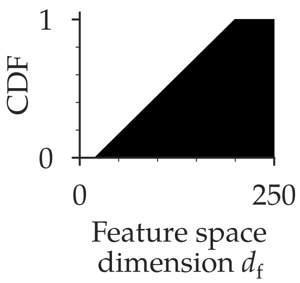

Figure 2.3. Prior over the feature space dimension of the FEVER model.
The feature space dimension $d_\mathrm{f} \sim \mathcal{UI}(20, 200)$ of the FEVER model as cumulative distribution function (CDF). (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

to the value achieved for $\|D - DW\|_2^2 > 0$, but by the quality of persistent stimulus representation (Sect. 2.3.3).

2.3.2. Implementation and Adaptation

The model described in Sect. 2.3.1 is not yet sufficient to realize networks. The following problems arise:

- excitatory and inhibitory neurons are not distinguished, which is not biologically plausible,
- the feature space dimension d_f and the feature vectors f_i themselves are not defined,
- the network does not necessarily satisfy the circuit constraints of L4, S1,
- the parameter λ has to be chosen.

To distinguish between excitatory and inhibitory neurons, the weights of the excitatory neurons have to be restricted to be positive, the weights of the inhibitory neurons to be negative.

An upper bound on the feature space dimension d_f is determined from the dimension of the inputs to the networks. It is unlikely that d_f is larger than the dimension of the input space. Input to a single barrel is primarily provided by about 200 ventral posteromedial nucleus (VPM) neurons (Bale et al. 2013; Maravall and Diamond 2014), translating to the upper bound $200 \geq d_\mathrm{f}$. The lower bound should be a small integer number. Here, 20 is assumed. The prior $d_\mathrm{f} \sim \mathcal{UI}(20, 200)$, denoting by $\mathcal{UI}(a, b)$ the uniform distribution on the integers $a, \dots, b$ (including the boundaries), is imposed on the feature space dimension d_f to reflect the uncertainty involved in choosing it (Fig. 2.3). The feature vectors f_i are assumed to be uniformly distributed on a unit sphere of dimension d_f,

$$f_i = \frac{\xi_i}{\|\xi_i\|}, \quad \xi_i \sim \mathcal{N}(0, I_{d_\mathrm{f}}), \tag{2.7}$$

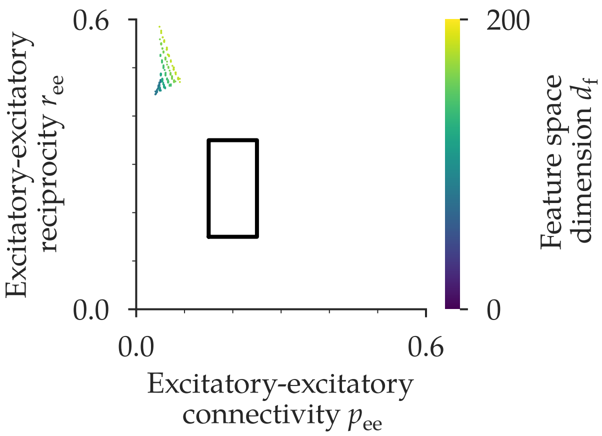

Figure 2.4. Connectivity and reciprocity of the original FEVER model.
Excitatory-excitatory reciprocity r_{ee} over excitatory-excitatory connectivity p_{ee} for the original FEVER model. The feature space dimension d_{f} is color coded. The black rectangle indicates the circuit constraints. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

in which ξ_i denote iid. multivariate normally distributed random variables with covariance matrix $I_{d_{\mathrm{f}}}$, which is the identity matrix in $\mathbb{R}^{d_{\mathrm{f}}}$.

The parameter λ governs the overall number of connections. By only adjusting λ, however, only network samples with too small p_{ee} and too high r_{ee} are realized (Fig. 2.4). The overrepresentation of reciprocated connections is a consequence of the symmetry of Eq. (2.6): outgoing connections of neuron i are preferentially made to those neurons k whose feature vectors f_k are similar to the feature vector f_i and vice versa (Fig. 2.2, e.g., the group of neurons with feature vector ■■■ in the lower right of the circle, and, e.g., the bidirectional motive ⇄ of the two neurons with feature vectors □■□ on the top of the circle). Moreover, the achievable connectivity is upper bounded by $d_{\mathrm{f}}/n \approx 0.1$, which is too small. To remedy these problems, a three-stage process is proposed as concrete cortical implementation of the FEVER model (Fig. 2.5). Initially (Fig. 2.5, left), the neurons do not yet possess feature vectors. The network structure is pairwise random. Each excitatory connection is at this stage realized with initial connection probability $p_{0,\mathrm{e}}$ and each inhibitory connection with initial connection probability $p_{0,\mathrm{i}}$, yielding an initial matrix W_0. Then (Fig. 2.5, middle), feature vectors are generated according to Eq. (2.7). Lastly (Fig. 2.5, right), connections are made attempting to recombine additively the postsynaptic neuron's feature vectors to resemble the neuron's own feature vector. In the sparse optimization, the initial connectivity and the signs of the entries of the excitatory submatrix W_{E} and the inhibitory submatrix W_{I} of W are now taken into account as well:

$$\min_{\Delta W} \|D - D(\Delta W + W_0)\|_2^2 + \lambda \|\Delta W\|_1, \quad \operatorname{diag} \Delta W = 0, \Delta W_{\mathrm{E},k,l} \geq 0, \Delta W_{\mathrm{I},k,l} \leq 0, \quad (2.8)$$

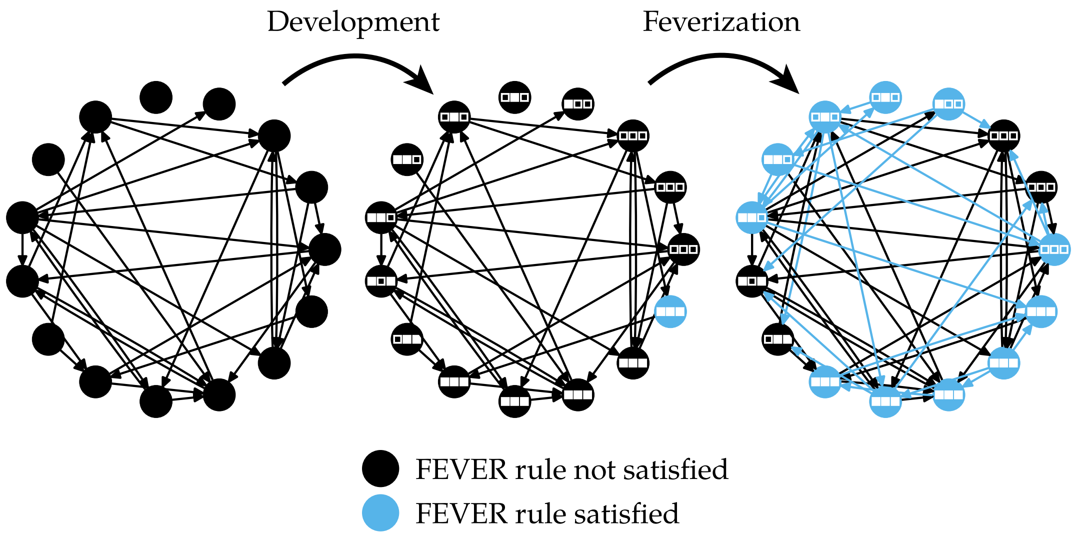

Figure 2.5. Adaptation of the FEVER model.
The adapted FEVER model is a process in which connectivity is unorganized at the beginning and then becomes organized. Left: in the first stage a network is randomly pre-filled with connections; the neurons do not yet posess feature vectors. Middle: in the second stage, neurons are equipped with feature vectors. Right: in the third state, the FEVER recombination is applied, but now also taking the initial connectivity into account. Note that not all neurons have to satisfy the FEVER rule perfectly for a functioning FEVER network.

in which $\Delta W = \Delta W_E + \Delta W_I$ is the change of the adjacency matrix due to feverization, denoting by ΔW_E the excitatory submatrix of the change and by ΔW_I the inhibitory submatrix, and the final network is $W = \Delta W + W_0$. The sparseness constraint is applied to the deviation ΔW of W from the initial network W_0. Equation (2.8) is numerically solved with Alg. 1, a problem specific version of the LASSO algorithm. Algorithm 1 takes the initial connectivity and the excitatory and inhibitory sign constraints into account. The inputs are the (dicrionary of) feature vectors D, the initial adjacency matrix W_0 and λ. The output is the optimized FEVER network W. In Alg. 1, the sign s is positive if an excitatory neuron is optimized and negative if an inhibitory neuron is optimized. As no self-connections are allowed, a dictionary $\widetilde{D}$ of feature vectors omitting a neuron's k own feature vectors is created. Similarly, the weight vector $\widetilde{w}$ of dimension $n-1$, omitting the optimized neuron k is generated. The effective target feature vector y is a modification to the actual target vector d_k, taking into account the neuron's sign and the initial connectivity. A coordinate descent algorithm is then used to optimize $\tilde{w}$. Note, that in the assignment of $\tilde{w}_l$, a positivity constraints is enforced. The resulting matrix W is created from the vectors $\tilde{w}$, reintroducing the omitted entries and the neuronal signs.

The parameter λ, together with the parameters $p_{0,\mathrm{e}}$ and $p_{0,\mathrm{i}}$ are obtained from the

Algorithm 1: Coordinate descent for the adapted FEVER.

```
Input: $D$, $W_0$, $\lambda$
Output: $W$

for $k = 1, \dots, n$ do
    $s \leftarrow 1$ if $k \leq n_\mathrm{e}$ else $-1$
    $\widetilde{D} \leftarrow (d_1, \dots, d_{k-1}, d_{k+1}, \dots, d_n)$
    $\widetilde{w} \leftarrow (\widetilde{D}^t \widetilde{D} + \lambda)^{-1} \widetilde{D}^t d_k$
    $y \leftarrow s \cdot (d_k - D w_{0,k})$

    repeat
        for $l = 1, \dots, n-1$ do
            $a \leftarrow 2 \langle \tilde{d}_l, \tilde{d}_l \rangle$
            $b \leftarrow 2 \left\langle \tilde{d}_l, y - \widetilde{D}\widetilde{w} + \widetilde{w}_l \tilde{d}_l \right\rangle$
            $\widetilde{w}_l \leftarrow \max\{0, b - \lambda\}/a$
        end
    until converged;
    $w_k \leftarrow s \cdot (\widetilde{w}_1, \dots, \widetilde{w}_{k-1}, 0, \widetilde{w}_k, \dots, \widetilde{w}_{n-1})^t$
end
$W \leftarrow (w_1, \dots, w_n)$
```

newly introduced parameter feverization $f_\mathrm{r} \in [0, 1]$ governing the amount of initial connectivity and the amount of connections added via the minimization Eq. (2.8). The initial connectivity is obtained from the feverization f_r as

$$p_{0,\mathrm{e}} = p_\mathrm{e} - f_\mathrm{r} \frac{d_\mathrm{f}}{n}, \quad p_{0,\mathrm{i}} = p_\mathrm{i} - f_\mathrm{r} \frac{d_\mathrm{f}}{n}, \tag{2.9}$$

the remaining connections are obtained from minimizing Eq. (2.8). The parameter λ is set through a bisection method on a logarithmic scale such that the overall target connectivities p_e and p_i are satisfied (Alg. 2). To accommodate for the different excitatory and inhibitory connectivities, λ was in the actual implementation different for the excitatory and inhibitory population, i.e. $\lambda = (\lambda_e, \lambda_i)$. In Alg. 2, the input P is a monotonically decreasing function $P : \mathbb{R}^+ \to [0, 1]$, which maps λ to a connectivity. The target connectivity is denoted by $p_\mathrm{target} \in [0, 1]$ and $\epsilon = 0.03$ is the accuracy with which p_target is to be matched. For efficiency reason, this optimization is only performed on a subset of the network and then extrapolated. This optimization is done with the pre-filled network W_0 already in place[2]. The maximal number of connections a neuron can make is d_f, assuming $n \geq d_\mathrm{f}$. This translates to a maximally obtainable connectivity fraction d_f/n which

[2] Equation (2.9) is not valid for the addition of two random networks. Connectivities do not simply add up.

Algorithm 2: Interval Bisection for Connectivity Optimization

```
Input: $P$, $p_{\mathrm{target}}$, $\epsilon$
Output: $\lambda$

$P_{\log} \leftarrow (x \mapsto P(10^{-x}))$
$l, u \leftarrow$ -2, 20
$c \leftarrow u$
$p_{\mathrm{current}} \leftarrow P_{\log}(c)$
while $|p_{\mathrm{current}}/p_{\mathrm{target}} - 1| > \epsilon$ do
    $c \leftarrow \frac{l+u}{2}$
    $p_{\mathrm{current}} \leftarrow P_{\log}(c)$
    if $p_{\mathrm{current}} > p_{\mathrm{target}}$ then
        $u \leftarrow c$
    else
        $l \leftarrow c$
    end
end
$\lambda \leftarrow 10^{-c}$
```

can be obtained from minimizing Eq. (2.8). For $f_{\mathrm{r}} = 1$, the maximally possible fraction of connections is realized in the third stage, the minimization Eq. (2.8). For $f_{\mathrm{r}} = 0$, no connection is realized via the minimization Eq. (2.8). The network is then a directed ER graph. The feverization (f_{r}) is upper bounded by $1 \geq f_{\mathrm{r}}$. The lower bound will be determined by functional validation of the FEVER network (see Sect. 2.3.3). It is expected that a persistent state representation Eq. (2.3) cannot be maintained for small f_{r}. This should yield a lower bound on f_{r}.

2.3.3. Functional Viability

The FEVER model implements the dynamics in Eq. (2.2) aiming to realize the persistent state representation Eq. (2.3). As a functional test, it is evaluated whether the concrete implementation proposed here is capable of maintaining a stimulus representation s over 20 neuronal time constants τ, which is much larger than the timespan over which a single neuron could maintain its activity (namely τ). The ordinary differential equation (ODE) system Eq. (2.2) is numerically integrated with the Newton method,

$$a(t + \Delta t) = \frac{\Delta t}{\tau}(-\sigma(a(t)) + W\sigma(a(t)) + u(t)),$$

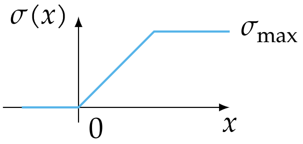

Figure 2.6. Activation function of the FEVER model.
The activation function is a rectified linear function which saturates at $\sigma_{\max}$.

with $\Delta t = \tau/100$. The activation function σ acts elementwise

$$\sigma(x) = \min\{\max\{0, x\}, \sigma_{\max}\},$$

with $\sigma_{\max} = 0.2$ (Fig. 2.6). The external input $u(t) = \sigma(r/\|r\|)I(t < 0.05\,t_{\max})$ is obtained from a random vector r with iid. components $r_k = e_k\delta_k, e_k \sim \mathcal{U}(-1, 1)$ and $\delta_k \sim \mathrm{Ber}(1/10)$. The length of the trial is $t_{\max} = 20\,\tau$. The input is only switched on for the first 5% of the trial duration to imprint the state to be represented and is then switched off. The state $FV_0 = s(0)$ at the beginning of the trial and the state $FV_{\mathrm{end}} = s(t_{\mathrm{end}})$ at the end of the trial are then compared. Ideally, if the stimulus representation was perfectly maintained, these two would coincide (Fig. 2.7a). The comparison between the two represented states is quantitatively performed in terms of their cosine similarity $C_{\mathrm{sim}}(FV_0, FV_{\mathrm{end}})$, which is defined by

$$C_{\mathrm{sim}} : \mathbb{R}^{d_{\mathrm{f}}} \times \mathbb{R}^{d_{\mathrm{f}}} \to [-1, 1], \quad (x, y) \mapsto \frac{\langle x, y\rangle}{\|x\|\,\|y\|}.$$

For $f_{\mathrm{r}} > 1/2$ the cosine similarity is, independently of d_{f}, close to 1, indicating almost perfect stimulus representation (Fig. 2.7b). This yields the remaining lower bound for f_{r} to be approximately 1/2. As $f_{\mathrm{r}} = 1$ corresponds to $\lambda = 0$, which in turn means that no sparsity constraint is enforced at all, a small margin to 1 is left. This small margin also helps to avoid numerical instabilities in the LASSO algorithm at the boundary case of $\lambda \approx 0$. The prior is therefore chosen to be uniform between 1/2 and 9/10, $f_{\mathrm{r}} \sim \mathcal{U}(1/2, 9/10)$ (Fig. 2.7c). Importantly, network samples realized according to the so defined prior distribution lie within the circuit constraints (Fig. 2.7d).

2.4. Texture Discrimination Task

The Erdős-Rényi echo state network (ER-ESN) model, Sect. 2.5, the exponentially decaying liquid state machine (EXP-LSM) model, Sect. 2.6 and the layered network (LAYERED) model, Sect. 2.7, are tested to discriminate surface texture profiles. This is a typical task rodents undergo in behavioral experiments (Diamond et al. 2008). The texture profiles are obtained from seven natural images (Fig. 2.8a); each image corresponds to one class.

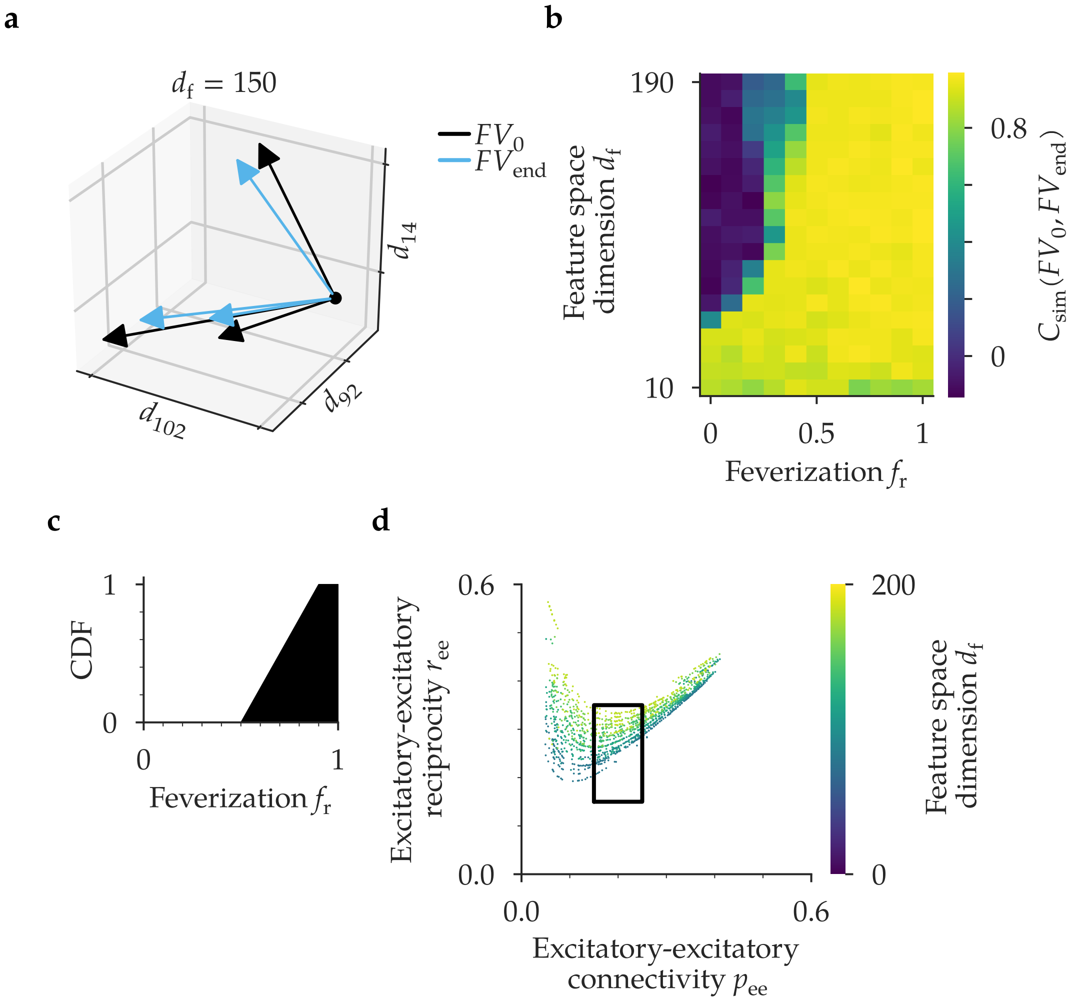

Figure 2.7. Persistent state representation under the circuit constraints.
(a) State representation of the FEVER model illustrated for two independent trials. A memory represented by a state vector FV_0 of dimension d_f is imprinted on the network at the beginning of a trial (only 3 dimensions shown). The state vector FV_end at the end of the trial is then compared to FV_0. These two vectors would coincide, $FV_\mathrm{end} = FV_0$, if the memory was perfectly preserved. (b) Evaluation of the stimulus representation in terms of the cosine similarity. Comparison of the imprinted state vector FV_0 at the beginning of a trial to the state vector FV_end represented by the network at the end of a trial in terms of their cosine similarity (C_sim), which is color coded. The horizontal axis shows the feverization (f_r), the vertical axis the feature space dimension (d_f). (c) Prior over the feverization (f_r) of the FEVER model as cumulative distribution function (CDF). (d) Excitatory-excitatory reciprocity r_ee over excitatory-excitatory connectivity p_ee for the adapted FEVER model. The feature space dimension d_f is color coded. The black rectangle indicates the circuit constraints. The adapted model realizes networks satisfying the circuit constraints. Compare to Fig. 2.4.
(Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

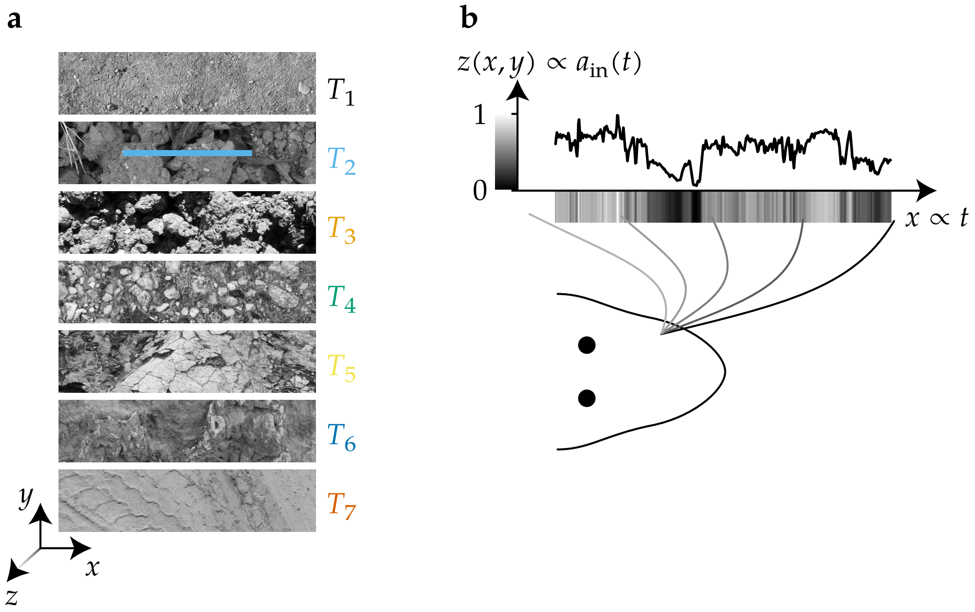

Figure 2.8. Texture discrimination.
(a) Seven images of natural textures are used to create texture profiles of classes T_1 to T_7. The images are linearized in a zigzag fashion $\Longrightarrow$ to maintain continuity. Texture profiles are obtained from parts (e.g., blue bar, T_2) of the linearized images. (Textures are from Mayang's free texture library, www.mayang.com/textures). (b) Texture elevation $z(x, y)$ is translated into a temporal profile $a_{\text{in}}(t)$ by whisker sweeping. The depicted profile (blue) corresponds to the region of the blue bar in (a), T_2. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

Pixel intensity is assumed to encode surface elevation z, which is in turn assumed to be transformed into a temporal profile by whisker sweeps along the surface (Fig. 2.8b). To simulate this process, the images are normalized to $[0, 1]$ by an affine linear transformation and then linearized in a zigzag fashion $\Longrightarrow$ yielding linear texture profiles. These profiles are modeled as ventral posteromedial nucleus (VPM) activity $a_{\text{in}}(t)$ (Figs. 2.9a and 2.9b, bottom). VPM provides input to L4 mediated by the thalamo-cortical projection w_{tc} (Figs. 2.9a and 2.9b, middle). The entries of the projection w_{tc} are iid. random variables $w_{\text{tc},k} \sim \delta_k u_k$ with $\delta_k \sim \text{Ber}(p_{\text{tc}})$, $p_{\text{tc}} = 0.5$ and $u_k \sim \mathcal{U}(0, 1/n_{\text{in}})$, in which n_{in} denotes the number of neurons in the thalamic input-receiving sub-layer of L4. The network's temporally evolving a_{L4} L4 activity (Fig. 2.9b, middle) is described by the discrete time dynamics

$$a_{\text{L4}}(t+1) = (1-\alpha)a_{\text{L4}}(t) + \alpha\,\sigma\left(W a_{\text{L4}}(t) + w_{\text{tc}}\,a_{\text{in}}(t) + b\right),$$

with activation function $\sigma : \mathbb{R}^n \to \mathbb{R}^{+n}, \quad x \mapsto \max\{0, x\}$, max acting elementwise, $\alpha = 0.2$ and bias vector $b \in \mathbb{R}^n$. The weighted adjacency matrix W (Fig. 2.9a, middle) is obtained from the network's binary representation δ_{kl}. Let $\omega_{kl} \sim \text{Lognormal}(0, 5^2)$

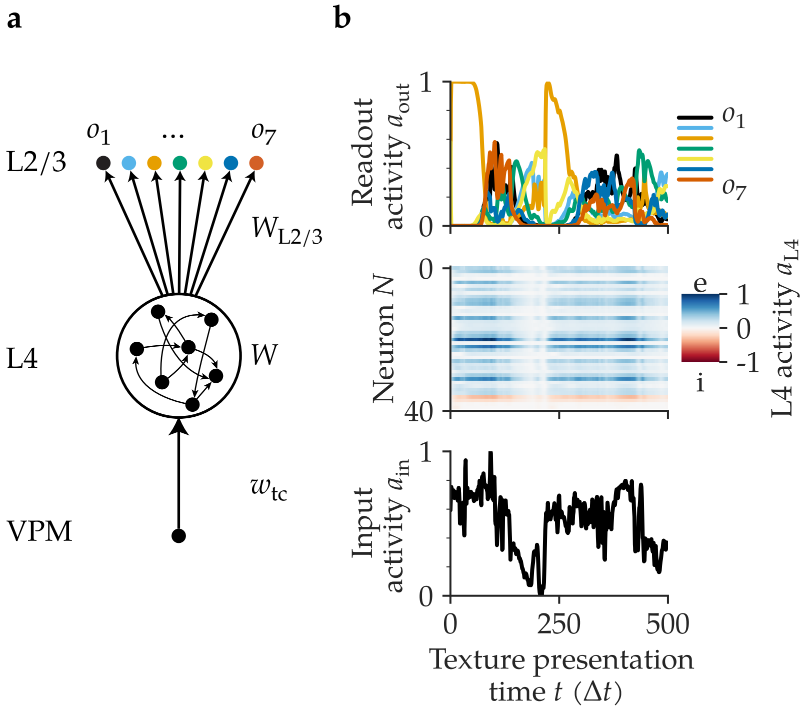

Figure 2.9. Network architecture and activity during texture classification.
(a) network architecture and (b) activity. Bottom: input activity (a_{in}) is received first in the ventral posteromedial nucleus (VPM), then projected to layer 4 (L4) via the thalamo-cortical projection w_{tc}. Middle: in layer 4 (L4) activity is transformed through recurrent connectivity W. L4 activity (a_{L4}) is color coded. Blue ("e") is excitatory activity; red ("i") is inhibitory activity. Top: layer 2/3 (L2/3) neurons are activated by the output projection $W_{\mathrm{L2/3}}$, yielding the readout activity (a_{out}) of the seven readout neurons $o_1, \dots, o_7$ representing the seven texture classes. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

iid., then $W_{kl} = \omega_{kl}\delta_{kl}$. The incoming excitatory weights of each neuron of L4 are then normalized to 1, the inhibitory weights to -1. L4 activity is read out via a projection $W_{\mathrm{L2/3}}$ (Figs. 2.9a and 2.9b, top). Each texture class is represented by a dedicated layer 2/3 (L2/3) readout neuron (Fig. 2.9a, top).

Before evaluating the network's predictive accuracy, it is trained on texture profiles, different from the ones presented later on for evaluation of the predictive accuracy. During training, the network's weights are modified only over the second half of the texture presentation; the network is allowed to settle dynamically over the first half (Fig. 2.10). The network weights are modified during the second half according to the adaptive moment estimation (Adam) method (Kingma and Ba 2014), which is a stochastic gradient descent-like method, with categorical cross entropy

$$H(W, b, W_{\mathrm{L2/3}}) = -\sum_{n}\sum_{c} \delta_{t_n,c} \log y(c|x_n)$$

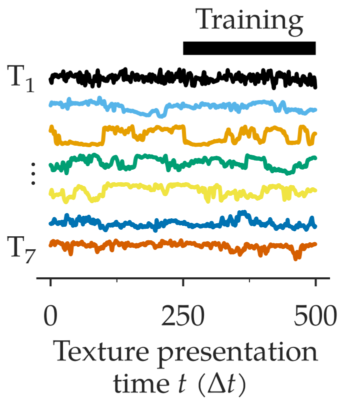

Figure 2.10. Texture profiles and training.
Weight modification occurs only during the second half of the texture presentation. Example profiles of each class are presented for training. These profiles are different from the ones presented during prediction. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

as loss function. Here, t_n denotes the true class of texture profile x_n and $y(c|x_n)$ is the probability assigned to class c by the readout on presentation of texture profile x_n. Only the readout weights $w_{\mathrm{L2/3}}$ and the bias b are trained. During classification, the activity of each of the readout neurons is integrated over the second half of the texture presentation. The class with the maximal integrated activity is interpreted as the network's texture class prediction. The model is implemented as custom layer of the Keras deep learning framework (Chollet et al. 2015). The custom layer is implemented with the Theano library (Team et al. 2016) and trained on GPU.

2.5. ER-ESN: Erdős-Rényi Echo State Network

2.5.1. Key Concepts

The echo state network (ESN), a recurrent network, was introduced by Jaeger (2001b)[3] to perform general computations, for example, for nonlinear systems identification (Jaeger 2002; Jaeger and Haas 2004). Computations are realized implementing dynamics on a graph together with a learning rule. Central to ESNs is that only readout weights are learned, but not the weights within the recurrent network. A certain memory capacity on short time scales is necessary for many computational tasks. Is is therefore not surprising, that the short-term memory capacity (Jaeger 2001a; Pascanu and Jaeger 2011) and long short-term memory capacity (Hochreiter and Schmidhuber 1997; Jaeger 2012) of ESNs

[3] see Jaeger (2010) for a corrected version

were examined. In this context, operation near criticality has been investigated (Buehner and Young 2006; Jaeger 2001b; Jaeger 2007; Jaeger 2010; Manjunath and Jaeger 2013; Yildiz, Jaeger, and Kiebel 2012).

The Erdős-Rényi (ER) random graph (Erdős and Rényi 1959) is classically described by a single pairwise connection probability p. Each possible connection is realized independently of the others with probability p. ESNs were often constructed as sparse ER graphs (Buehner and Young 2006; Jaeger 2001a; Jaeger 2001b; Jaeger 2002; Jaeger and Haas 2004; Pascanu and Jaeger 2011), realizing only 1% of the possible connections and rather rarely as less sparse graphs realizing up to 20% of the possible connections (Buehner and Young 2006; Jaeger 2001b). Beyond the ER topology, also small-world (Watts and Strogatz 1998) and scale-free (Barabási and Albert 1999) topologies, as well as topologies generated by spatial growth (Kaiser and Hilgetag 2004) were tested (Liebald 2004). Moreover, modular decoupled ESNs with lateral inhibition were examined (Xue, Yang, and Haykin 2007) and a hierarchical variant of the ESN was proposed (Jaeger 2007). Problem specific graph optimization was attempted (Jaeger 2012) and recipes for constructing ESNs with good memory capacities have been given (Yildiz, Jaeger, and Kiebel 2012). For a review of ESNs and other reservoir computing techniques see Lukoševičius and Jaeger (2009) and Lukoševičius, Jaeger, and Schrauwen (2012).

While the ER model constitutes a structural null hypothesis, the ESN model proposes a way of performing computations on this structure. The ER-ESN captures and combines both aspects.

2.5.2. Implementation and Adaptation

The ER-ESN model is implemented as a directed ER random graph with two subpopulations: an excitatory one and an inhibitory one (Fig. 2.11a). The probability to realize a graph $G = (V, E)$ with vertex set $V = \{1, \dots, n_\mathrm{e} + n_\mathrm{i}\}$ and edge set $E \subset V \times V$ is

$$ER(G|n_\mathrm{e}, n_\mathrm{i}, p_\mathrm{e}, p_\mathrm{i}) = \prod_{v=1}^{n_\mathrm{e}+n_\mathrm{i}} \left(\prod_{u=1}^{n_\mathrm{e}} p_\mathrm{e}^{\delta_{uv}} (1-p_\mathrm{e})^{1-\delta_{uv}} \right) \left(\prod_{u=n_\mathrm{e}+1}^{n_\mathrm{e}+n_\mathrm{i}} p_\mathrm{i}^{\delta_{uv}} (1-p_\mathrm{i})^{1-\delta_{uv}} \right), \quad (2.10)$$

in which δ denotes the indicator function

$$\delta_{uv} = \begin{cases} 1 & \text{if } (u,v) \in E \\ 0 & \text{otherwise} \end{cases} .$$

The first index u corresponds to the source vertex (presynaptic neuron), the second index v to the target vertex (postsynaptic neuron). The expected value of the excitatory-excitatory reciprocity r_ee can be analytically calculated. Let $1 \le u, v \le n_\mathrm{e}$, then

$$\mathbb{E}[r_\mathrm{ee}] = ER\left((u,v) \in G | (v,u) \in G\right) = ER\left((u,v) \in G\right) = p_\mathrm{e}.$$

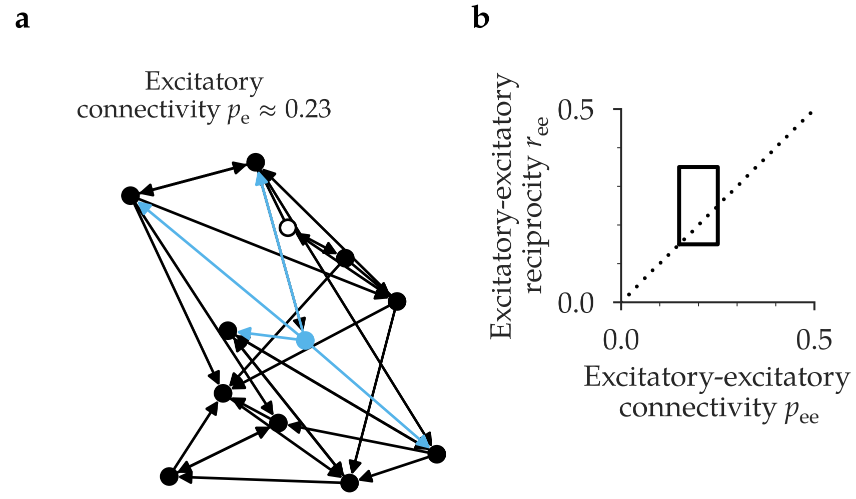

Figure 2.11. ER-ESN model.
(a) Small subgraph of an ER-ESN graph. The probability of realizing an edge in the ER-ESN model depends only on whether the presynaptic (●→) neuron is excitatory (filled circle ●) or inhibitory (open circle ○). It does not depend on the type of the postsynaptic (→●) neuron or any other property such as spatial distance between neurons. The neuron in the center of the plot is highlighted in blue to facilitate distinguishing its connections. (b) Stochastically realized ER-ESN networks. Excitatory-excitatory connectivity (p_{ee}) is plotted on the horizontal axis, excitatory-excitatory reciprocity (r_{ee}) on the vertical axis. The ER-ESN model satisfies the circuit constraints indicated by the black rectangle. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

The second equality holds since the edges (u, v) and (v, u) are independently realized with probability p_e; conditioning on (v, u) has therefore no effect. The ER-ESN model therefore satisfies the circuit constraints in expectation if the connectivity is chosen within the circuit constraints (Fig. 2.11b). Given the size of the networks of about 2000 neurons, i.e. about 4 million possible connections, the deviations from the expected values are small (Fig. 2.11b).

2.5.3. Functional Viability

The ER model is functionally to discriminate textures (Sect. 2.4). The L4 weights are not trained for the ER-ESN model. This follows the ESN paradigm, which is to keep the weights within L4[4] constant (this is different for the LAYERED model, Sect. 2.7.3). Although the readout neurons do not very reliably predict the texture class at individ-

[4] The recurrent network, here referred to as L4, is often called "reservoir".

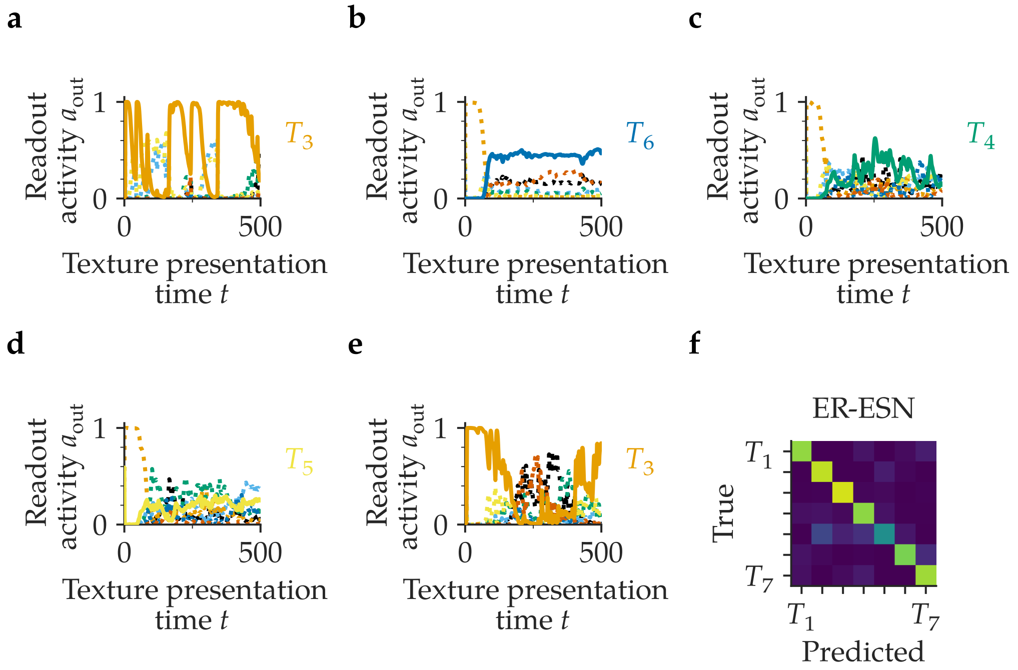

Figure 2.12. ER-ESN texture classification performance.
Although the readout neurons do not very reliably predict the texture class at individual timepoints the overall performance of the ER-ESN model is substantially above chance level. (a), (b), (c), (d), (e) Readouts of randomly chosen texture profiles. The label on the right side of each panel indicates the ground truth class. See, Fig. 2.8a for the corresponding textures. (f) Confusion matrix for the ER-ESN model in the texture classification task. The vertical axis denotes the ground truth class, the horizontal axis the predicted class. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

ual timepoints (Figs. 2.12a to 2.12e), the overall performance of the ER-ESN model is substantially above chance level achieving 80.8% accuracy (Fig. 2.12f, chance level is $1/7 \approx 14.2\%$ accuracy). The model is therefore considered functional.

2.6. EXP-LSM: Exponentially Decaying Liquid State Machine

2.6.1. Key Concepts

The liquid state machine (LSM), was proposed as a model of general cortical computations (Maass, Natschläger, and Markram 2002; Maass, Natschläger, and Markram 2004a). Different computational properties of LSMs, such as summation of spike rates, nonlinear combinations, coincidence detection, speech recognition and various classification tasks, were examined (Maass, Joshi, and Sontag 2006; Maass and Markram 2004; Maass, Natschläger, and Markram 2004b; Schliebs, Fiasché, and Kasabov 2012). Phase space trajectories, in particular with respect to the role of fixed points were examined to gain a better mechanistic understanding of how computations in LSMs are performed (Sussillo and Barak 2013). Compared to the ESN literature the theoretical focus was rather on the input-output mapping properties than on the reservoir properties. They share, however, the idea of training readout weights only. Initially the p-delta learning rule, a variation of the perceptron learning rule, (Auer, Burgsteiner, and Maass 2001) was used for readout training, but alternative learning rules were also proposed (Sussillo and Abbott 2009). The LSM has also been specifically proposed for cortical layer 4 of somatosensory cortex (Probst et al. 2012).

The impact of network topology on LSM robustness has been discussed (Hazan and L. M. Manevitz 2010; Hazan and L. M. Manevitz 2012; L. Manevitz and Hazan 2010). Random topologies have been compared to hub topologies (Hazan and L. M. Manevitz 2012) and small-world topologies (Watts and Strogatz 1998) with focus on network robustness (Hazan and L. M. Manevitz 2012). Neurons were often assumed to be located on the integer points of a three dimensional grid (Maass, Joshi, and Sontag 2006; Maass and Markram 2004; Maass, Natschläger, and Markram 2002). Maass, Joshi, and Sontag (2006), Maass and Markram (2004), and Maass, Natschläger, and Markram (2002) assumed a quadratic exponential decay of connection probability with distance whereas Probst et al. (2012) assumed a linear exponential decay. The recurring theme, however, is the distance dependent connectivity: connections to closeby neurons are assumed to be more likely than to distant neurons. For reviews on LSMs see for instance Buonomano and Maass (2009) and Maass (2010).

2.6.2. Implementation and Adaptation

The exponentially decaying liquid state machine (EXP-LSM) implements distance dependent connectivity. The connection probability decays with increasing inter-soma distance. The soma locations $x_k \in [0,1]^3, k = 1, \dots, n$ uniformly distributed in a barrel. The three spatial dimensions $l = 1, \dots, 3$ of the soma location x_k of neuron k are independent and identically distributed according to $x_{kl} \sim \mathcal{U}(0,1)$. That is, the barrel is modeled as a cube (Fig. 2.13a) and the somata are assumed to be uniformly distributed within this cube. The probability $p((k,l) \in G | x_k, x_l)$ to realize a (directed) edge from vertex k to vertex l decays exponentially with inter-soma distance $\|x_k - x_l\|$ (Fig. 2.13b)

$$p((k,l) \in G | x_k, x_l, \lambda_\mathrm{e}, \lambda_\mathrm{i}) = \exp(-\|x_k - x_l\| / \lambda(k))$$

with

$$\lambda(k) = \begin{cases} \lambda_\mathrm{e} & \text{if } k \leq n_\mathrm{e} \\ \lambda_\mathrm{i} & \text{otherwise} \end{cases},$$

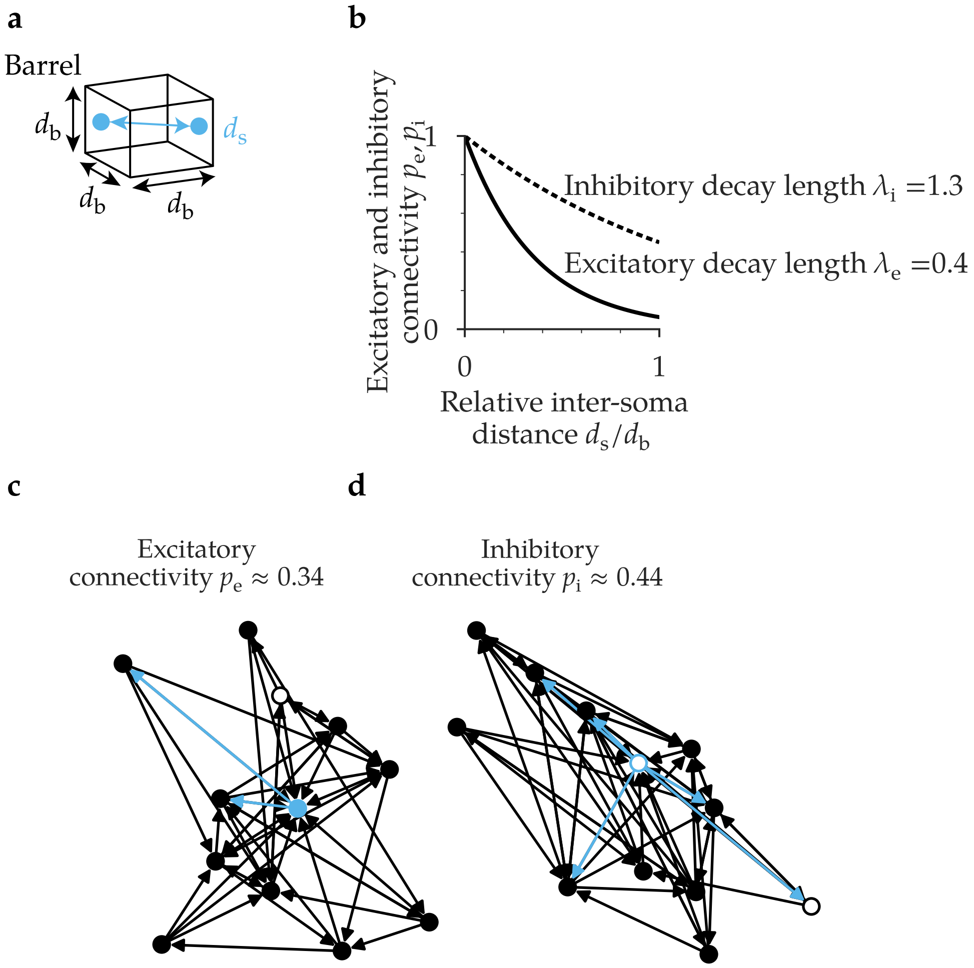
a
Barrel
d_b
d_s
d_b
d_b
b
Excitatory and inhibitory connectivity p_e, p_i
1
0
0
1
Relative inter-soma distance d_s/d_b
Inhibitory decay length λ_i =1.3
Excitatory decay length λ_e =0.4
c
Excitatory connectivity $p_e \approx 0.34$
d
Inhibitory connectivity $p_i \approx 0.44$

Figure 2.13. Exponentially decaying liquid state machine.
(a) A barrel is modeled as a cube of side length d_b. The intersoma distance is denoted by d_s. (b) The connection probability decays exponentially with relative inter-soma distance (d_s/d_b). The excitatory decay length (λ_e) and inhibitory decay length (λ_i) determine how fast connectivity decays with distance. (c) Small subgraph of an EXP-LSM graph. The probability of realizing an edge in the EXP-LSM model depends on whether the presynaptic neuron is of excitatory type (filled circle ●) or inhibitory type (open circle ○) and on the spatial distance between neurons. The neuron in the center of the plot is highlighted in blue to facilitate distinguishing its connections. (d) As in (c), but inhibitory neuron highlighted. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

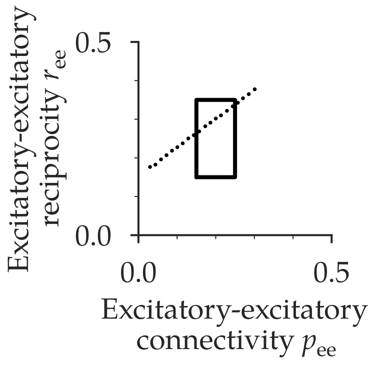

Figure 2.14. Stochastically realized EXP-LSM networks.
Excitatory-excitatory connectivity (p_{ee}) is plotted on the horizontal axis, excitatory-excitatory reciprocity (r_{ee}) on the vertical axis. The EXP-LSM model satisfies the circuit constraints. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

in which n_e denotes the number of excitatory neurons. The speed of the decay is determined for the excitatory and the inhibitory subpopulation by the excitatory decay length (λ_e) and the inhibitory decay length (λ_i). Small subgraphs of stochastically sampled EXP-LSM networks are depicted in Figs. 2.13c and 2.13d. Importantly, if λ_e and λ_i are set such that excitatory connectivity (p_e) and inhibitory connectivity (p_i) lie within the circuit constraints, then the circuit constraints on the excitatory-excitatory reciprocity (r_{ee}) and on the excitatory-excitatory connectivity (p_{ee}) are also satisfied (Fig. 2.14). No further adaptation or modification is therefore required for the EXP-LSM model.

2.6.3. Functional Viability

The functional test for the EXP-LSM model is the same as for the ER-ESN model described in Sect. 2.4. The EXP-LSM model does not reliably predict the correct texture class at individual time points (Figs. 2.15a to 2.15e) but performs above chance level (14.2%) taking the integrated activity as predictor (Fig. 2.15f). The achieved accuracy is 82.1%, which is comparable to the accuracy achieved by the ER model (Sect. 2.5.3, 80.7%). The model is therefore considered functional.

2.7. LAYERED: Layered Network

2.7.1. Key Concepts

Layered network (LAYERED) models represent the hypothesis of processing cortical input via a hierarchy of sequentially stacked layers. The multi-layer perceptron is the

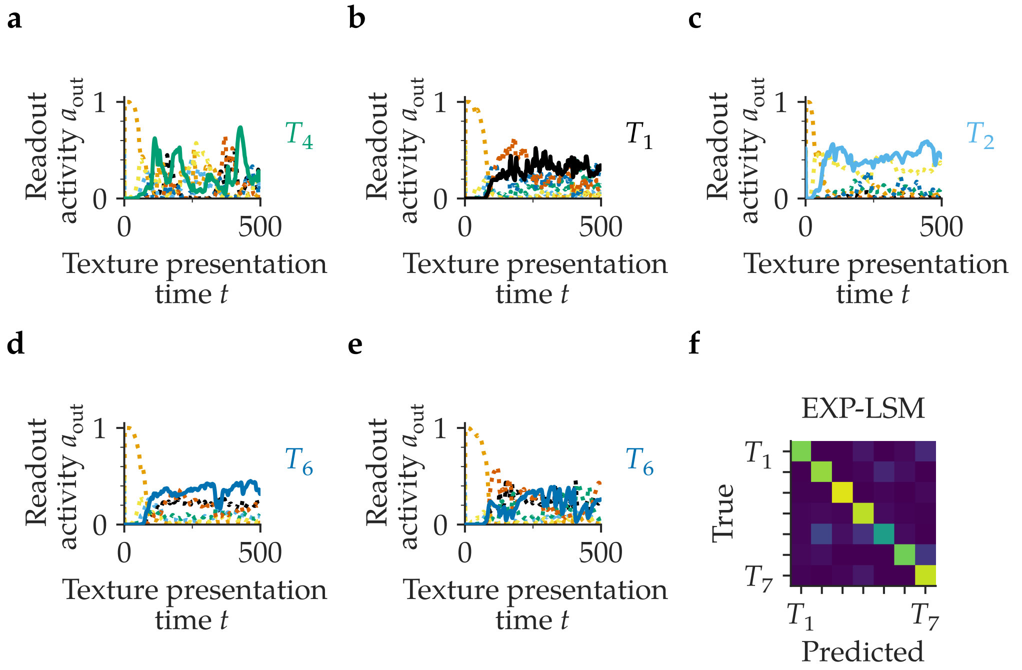

Figure 2.15. EXP-LSM texture classification performance.
(a), (b), (c), (d), (e) Readouts of randomly chosen texture profiles. The label on the right side of each panel indicates the ground truth class. See, Fig. 2.8a for the corresponding textures. (f) Confusion matrix for the EXP-LSM model in the texture classification task. The vertical axis denotes the ground truth class, the horizontal axis the predicted class. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

earliest example of such a model (Rosenblatt 1961). Hierarchical, layered models were often used in studies of early sensory processing, in particular visual processing (Griffith 1963; Hubel and Wiesel 1962). Layered structures have also been used in the context of self-organization for translation invariant geometrical pattern recognition (Fukushima 1979; Fukushima 1980) or as deep neural networks (Ivakhnenko 1971; Ivakhnenko and Lapa 1965; Lecun et al. 1998; LeCun et al. 1989). Beyond the purely feed-forward structures considered in the studies above, also stacked recurrent neural networks were used as a model of hierarchical processing (Pascanu, Gulcehre, et al. 2013; Schmidhuber 1992).

2.7.2. Implementation and Adaptation

The excitatory population is subdivided into n_l disjoint sets of approximately equal size (Fig. 2.16). Denoting the layer boundaries by

$$B_l = \left\lfloor \frac{l}{n_l} n_e \right\rfloor ,$$

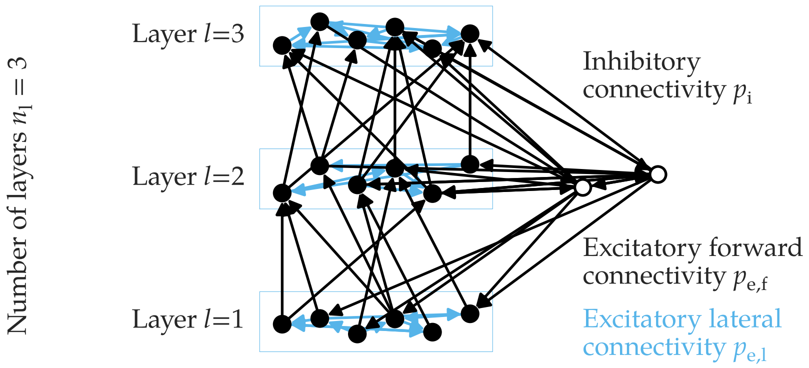
Number of layers $n_l = 3$
Layer l=3
Layer l=2
Layer l=1
Inhibitory connectivity p_i
Excitatory forward connectivity $p_{e,f}$
Excitatory lateral connectivity $p_{e,l}$

Figure 2.16. Exemplary layered network (LAYERED) model.
Illustration of an LAYERED model with number of layers n_l= 3. In the LAYERED model, excitatory neurons are uniquely grouped into sequentially ordered layers. Layer l= 1 is the first (thalamic input receiving) layer, layer l= 3 is the output layer. Inhibitory neurons provide global inhibition. The excitatory lateral connectivity $p_{e,l}$ is the pairwise connetion probability for connections within a layer, the excitatory forward connectivity $p_{e,f}$ is the pairwise connetion probability for connections from one layer to the next layer. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

layer l consists of the neurons $V_l = \{k \in V | B_{l-1} < k \leq B_l\}$, for $l = 1, \dots, n_l$, denoting by $V = \{1, \dots, n\}$ the set of neurons in the network. The pairwise connection probability $p((u,v)|p_{e,f}, p_{e,l}, n_l, p_i, p_e, n_e)$ of realizing an edge (u, v) from vertex u to vertex v is

$$p((u,v)|p_{e,f}, p_{e,l}, n_l, p_i, p_e, n_e) = \begin{cases} p_{e,l} & \text{if } \exists l \in \{1, \dots, n_l\} : u \in V_l \text{ and } v \in V_l \\ p_{e,f} & \text{if } \exists l \in \{1, \dots, n_l - 1\} : u \in V_l \text{ and } v \in V_{l+1} \\ p_e & \text{if } u \leq n_e \text{ and } v > n_e \\ p_i & \text{if } u > n_e \\ 0 & \text{otherwise} \end{cases},$$

in which $p_{e,l}$ denotes the excitatory lateral connectivity, $p_{e,f}$ the excitatory forward connectivity, n_l the number of layers, p_i the inhibitory connectivity, $p_e = p_{ee} = p_{ei}$ the excitatory connectivity, and n_e the number of excitatory neurons. Together, the two parameters $p_{e,l}$ and $p_{e,f}$ determine the overall excitatory-excitatory connectivity (p_{ee}). The expected excitatory-excitatory connectivity is obtained by elementary calculations (Fig. 2.17a)

$$\mathbb{E}[p_{ee}] = \frac{n_l p_{e,l} + (n_l - 1) p_{e,f}}{n_l^2} \tag{2.11}$$

and similarly the expected excitatory-excitatory reciprocity (Fig. 2.17b)

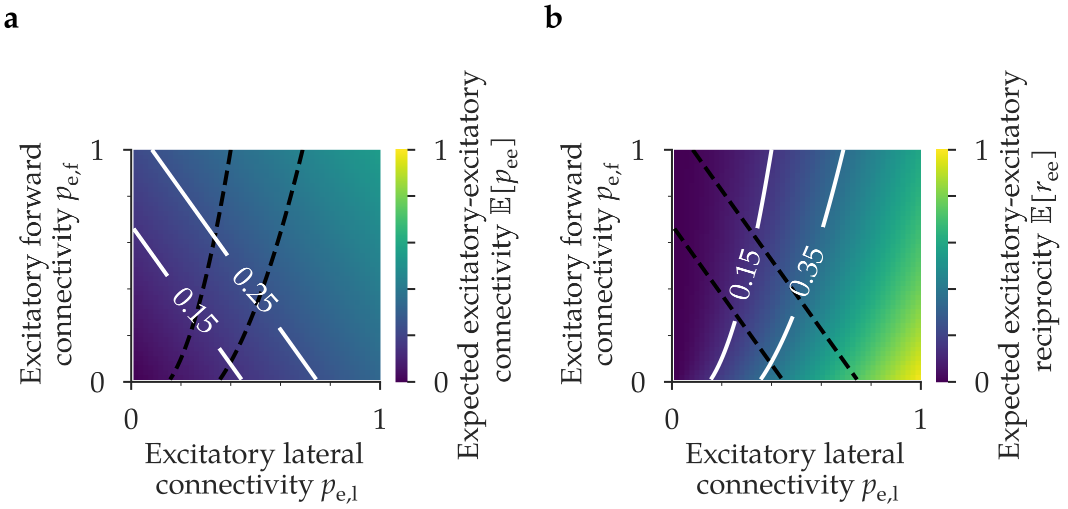

Figure 2.17. Expected excitatory-excitatory connectivity and expected excitatory-excitatory reciprocity as function of the excitatory lateral connectivity and the excitatory forward connectivity for a LAYERED network with three layers.
(a) Expected excitatory-excitatory connectivity $\mathbb{E}[p_{ee}]$ as function of the excitatory lateral connectivity $p_{e,l}$ and the excitatory forward connectivity $p_{e,f}$. The white isolines indicate the connectivity constraints, the black dashed lines are the white isolines from (b). (b) Expected excitatory-excitatory reciprocity $\mathbb{E}[r_{ee}]$ as function of the excitatory lateral connectivity $p_{e,l}$ and the excitatory forward connectivity $p_{e,f}$. The white isolines indicate the reciprocity constraints, the black dashed lines are the white isolines from (a). (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

$$\mathbb{E}[r_{ee}] = \frac{p_{e,l}^2}{n_l \mathbb{E}[p_{ee}]}.$$

A LAYERED network with $n_l = 3$ satisfies the circuit constraints for appropriately chosen $p_{e,l}$ and $p_{e,f}$ (Figs. 2.17a and 2.17b). The LAYERED model is however not able to satisfy the circuit constraints for arbitrary n_l. For instance, the maximally achievable connectivity, setting $p_{e,l} = 1$ and $p_{e,f} = 1$ in Eq. (2.11), decreases as n_l increases (Fig. 2.18). Realizations of the LAYERED model which do not push $p_{e,l}$ or $p_{e,f}$ to their extreme values are therefore only possible for a moderate range of n_l. This determines the prior $n_l \sim \mathcal{UI}(2, 4)$ (Fig. 2.19a). The prior over $p_{e,l}$ and $p_{e,f}$ is a joint prior (Fig. 2.19b) for which the circuit constraints are satisfied.

2.7.3. Functional Viability

The functional test of the LAYERED model is the texture discrimination task (Sect. 2.4). For the LAYERED model, also the forward connections connecting neurons of one layer V_l to the next V_{l+1} are trained. Synaptic normalization is also slightly different compared

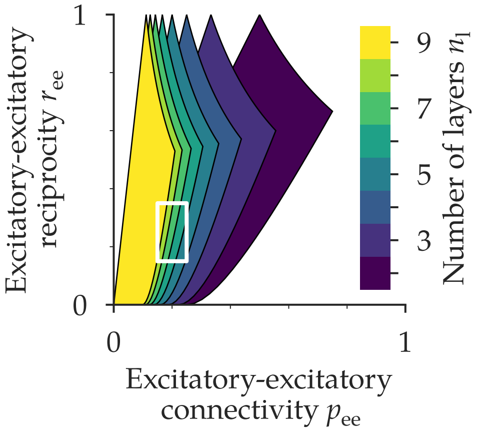

Figure 2.18. Achievable connectivity and reciprocity for the LAYERED model.
The circuit constraints are satisfied for non-extreme excitatory forward connectivity and excitatory lateral connectivity with two to four layers. The colored areas describe the coverable regime. The boundaries of the colored areas correspond to the boundary cases of the excitatory lateral connectivity and the excitatory forward connectivity.
Boundaries:
Bottom: $p_{e,l} = 0, p_{e,f} \in [0, 1]$.
Right: $p_{e,l} \in [0, 1], p_{e,f} = 1$.
Top: $p_{e,l} = 1, p_{e,f} \in [0, 1]$.
Left: $p_{e,l} \in [0, 1], p_{e,f} = 0$.
The white rectangle corresponds to the circuit constraints. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

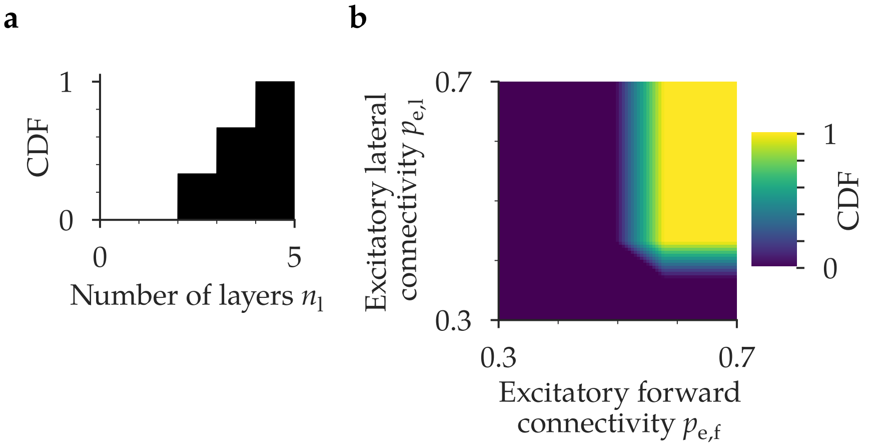

Figure 2.19. Prior of the LAYERED model.
(a) Prior over number of layers (n_l) as cumulative distribution function (CDF). (b) Joint prior of the excitatory lateral connectivity ($p_{e,l}$) and the excitatory forward connectivity ($p_{e,f}$). The CDF is color coded. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

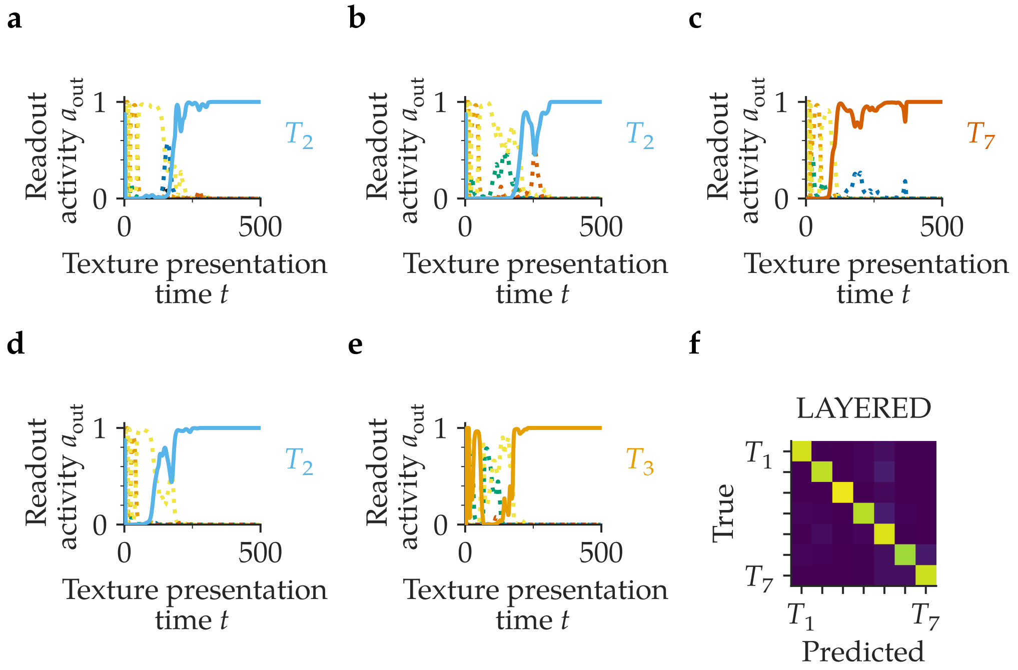

Figure 2.20. LAYERED texture classification performance.
(a) – (e) Readouts of randomly chosen texture profiles. The label on the right side of each panel indicates the ground truth class. See, Fig. 2.8a for the corresponding textures. (f) Confusion matrix for the LAYERED model in the texture classification task. The vertical axis denotes the ground truth class, the horizontal axis the predicted class. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

to th ER-ESN model and the EXP-LSM model For each excitatory neuron, the incoming weights of neurons from the same layer are normalized to 1. The incoming weights from neurons of the predecessor layer are also normalized to 1, separately from the within-layer connections. The LAYERED model predicts, compared to the ER-ESN and EXP-LSM model, the true texture class more reliably at individual time points towards the end of the texture presentation (Figs. 2.20a to 2.20e, compare to Figs. 2.12a to 2.12e and Figs. 2.15a to 2.15e). The overall achieved accuracy is 91.9%. Compared to the ER-ESN model and the EXP-LSM model, the overall accuracy is, however, only moderately higher (Fig. 2.21). The LAYERED model is considered functional.

2.8. API: Anti Phase Inhibition

2.8.1. Key Concepts

The anti phase inhibition (API) model (Kayser and Miller 2002; Troyer et al. 1998) and similar models of opponent inhibition (Hansen and Neumann 2004) were originally

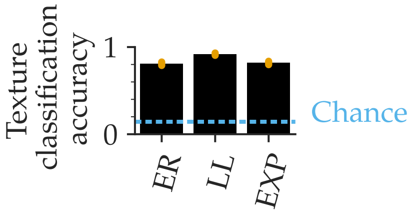

Figure 2.21. Comparison of the texture classification accuracy of the ER-ESN, LAYERED and EXP-LSM model.
The ER model achieves 80.8% accuracy, the EXP-LSM model 82.1% and the LAYERED model 91.9%. Chance level is $1/7 \approx 14.3\%$. The error bars ▮ indicate 95% confidence intervals. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

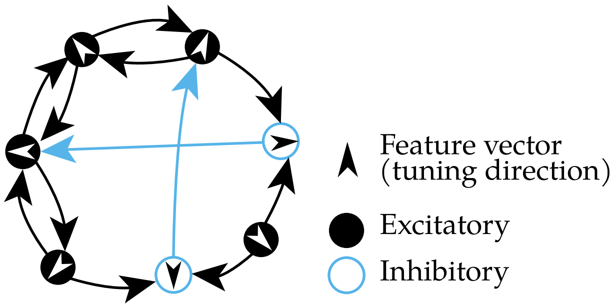

Figure 2.22. The API model.
Excitatory connections are preferentially made onto neurons with similarly tuned feature vectors, inhibitory connections onto neurons with antiphasicly (oppositely) tuned feature vectors.

proposed for visual cortex (Hansen and Neumann 2004; Kayser and Miller 2002; Troyer et al. 1998) but were also suggested in a somatosensory setting to describe the function of multiple adjacent barrels (Miller, Pinto, and Simons 2001). The central concept is that neurons excite other neurons with similar feature vectors (receptive fields) and inhibit neurons with antiphasic (opposite) feature vectors (Fig. 2.22). This mechanism implements stimulus tuning in the sense that the purely excitatory VPM input leads to excitation of cortical neurons, which are similarly tuned to the thalamic input, but to inhibition of cortical neurons, which are dissimilarly tuned to the thalamic input. Due to the positive nature of the thalamic input, this inhibition has to be provided by the cortical L4 circuit itself.

2.8.2. Implementation and Adaptation

In the API model, similar to the FEVER model, a feature vector vector $FV_k \in \mathbb{R}^{d_\text{f}}$ is associated with each neuron k. The connection probability from neuron k to neuron l

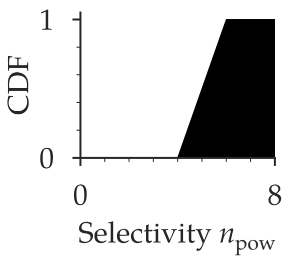

Figure 2.23. Prior over the selectivity n_{pow}.
The selectivity n_{pow} is uniformly distributed $n_{\mathrm{pow}} \sim \mathcal{U}(4,6)$. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

depends on the similarity of these feature vectors, as measured by their cosine similarity $c_{kl} = C_{\mathrm{sim}}(FV_k, FV_l) = \langle FV_k, FV_l \rangle / \|FV_k\| / \|FV_l\|$. In the original formulation (Troyer et al. 1998) feature vectors are assumed to be Gabor like. A connection in the graph G is established with probability

$$\tilde{p}((k,l) \in G) = 1 - \left(1 - \max\{0, c_{kl}s_j\}^{n_{\mathrm{pow}}}\right)^{10}, \tag{2.12}$$

in which $s_k = 1$ if neuron k is excitatory and $s_k = -1$ if neuron k is inhibitory. The selectivity n_{pow} is originally suggested to lie in the range $n_{\mathrm{pow}} \in [4,6]$ (Troyer et al. 1998), yielding here the prior $n_{\mathrm{pow}} \sim \mathcal{U}(4,6)$ (Fig. 2.23). Equation (2.12) describes the probability that a binomially distributed random variable $w_{kl} \sim \mathrm{Bin}(10, \max\{0, c_{kl}s_j\}^{n_{\mathrm{pow}}})$ realizes a non-zero value, $\tilde{p}((k,l) \in G) = p(w_{kl} > 0)$. This can be interpreted in the sense that 10 connection attempts form neuron k to neuron l are made and each attempt is successful with probability $\max\{0, c_{kl}s_j\}^{n_{\mathrm{pow}}}$. The larger n_{pow}, the less likely are excitatory connections to neurons with dissimilar feature vectors; i.e. the neuron's selectivity increases. An analogous statement holds for inhibitory connections. Notably, excitatory connections from neurons k to neurons l are forbidden if $c_{kl} < 0$, and similarly, inhibitory connections are forbidden if $c_{kl} > 0$. The original formulation does not realize networks satisfying the circuit constraints: the excitatory-excitatory connectivity (p_{ee}) is too low while the excitatory-excitatory reciprocity (r_{ee}) is too high (Figs. 2.24a and 2.24b). Variation of n_{pow} and d_{f} have only small effects on p_{ee} and r_{ee} (Figs. 2.24a and 2.24b). Only a minor decrease of r_{ee} with decreasing n_{pow} is observable (Fig. 2.24b); no systematic influence on d_{f} is visible.

Adaptations are applied to generate an implementation that satisfies the circuit constraints. The feature vectors are, similar to the FEVER model, drawn from a unit sphere of dimension d_{f} according to Eq. (2.7). Cosine similarities c_{kl} between the feature vectors of neurons k and l are transformed into connection probabilities $p((k,l) \in G)$ between

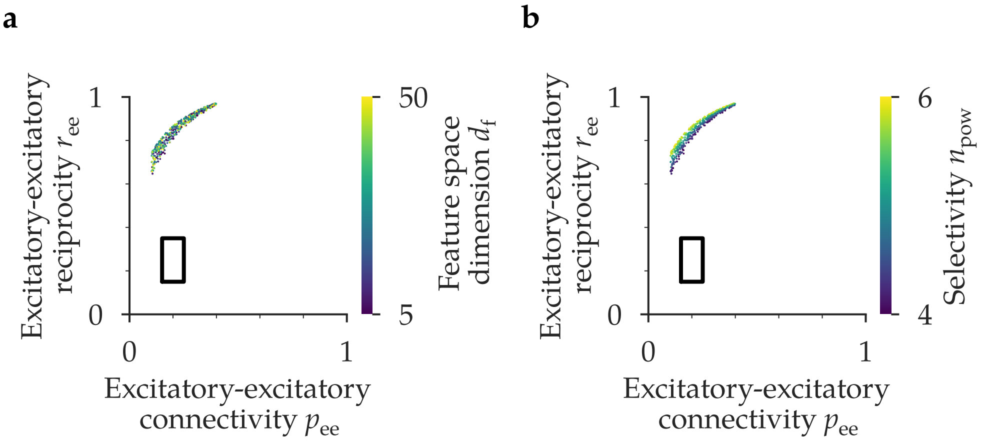

Figure 2.24. Original API model.
Excitatory-excitatory reciprocity r_{ee} over excitatory-excitatory connectivity p_{ee} for the original API model. The black rectangle indicates the circuit constraints. (a) The feature space dimension (d_{f}) is color coded. (b) The selectivity (n_{pow}) is color coded. The original model does not realize networks satisfying the circuit constraints. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

neuron k and l according to

$$p((k,l) \in G) = 1 - \left(1 - \rho_{kl}(n_{\mathrm{pow}})\right)^{n_{s_k}{}^{\mathrm{bin}}}, \quad \rho_{kl}(n_{\mathrm{pow}}) = \left(\frac{c_{kl}s_k + 1}{2}\right)^{n_{\mathrm{pow}}}, \tag{2.13}$$

similar to Eq. (2.12). The connection probabilities can then be expressed in terms of a binomially distributed variable w_{kl}

$$p((k,l) \in G) = p(w_{kl} > 0), \quad w_{kl} \sim \mathrm{Bin}\left(n_{s_k}{}^{\mathrm{bin}}, \rho_{kl}(n_{\mathrm{pow}})\right).$$

The coefficient $n_{s_k}^{\mathrm{bin}}$ is interpreted as the number of connection attempts neuron k makes to any other neuron l with probability $\rho_{kl}(n_{\mathrm{pow}})$. In contrast to the original formulation Eq. (2.12), the relaxed rule Eq. (2.13) also permits connections from excitatory neurons k to neurons l with $c_{kl} < 0$, and similarly, permits connections from inhibitory neurons k onto neurons l with $c_{kl} > 0$ (Fig. 2.25a). The probability to obtain pairs of neurons with close to zero cosine similarity increases as d_{f} increases (Fig. 2.25b). The expected values of p_{e} and p_{i} are upper bounded by $1/2$ in the original formulation, since in expectation half of the neuronal pairs (k,l) have $c_{kl} < 0$ and the other half $c_{kl} > 0$ (see also Fig. 2.25b). The same upper bound does not hold for the adapted model. The coefficients n_x^{binomial} with $x \in \{-1, 1\}$ are fitted to match the specified excitatory and inhibitory connectivity using the `scipy.optimize.fsolve` function.

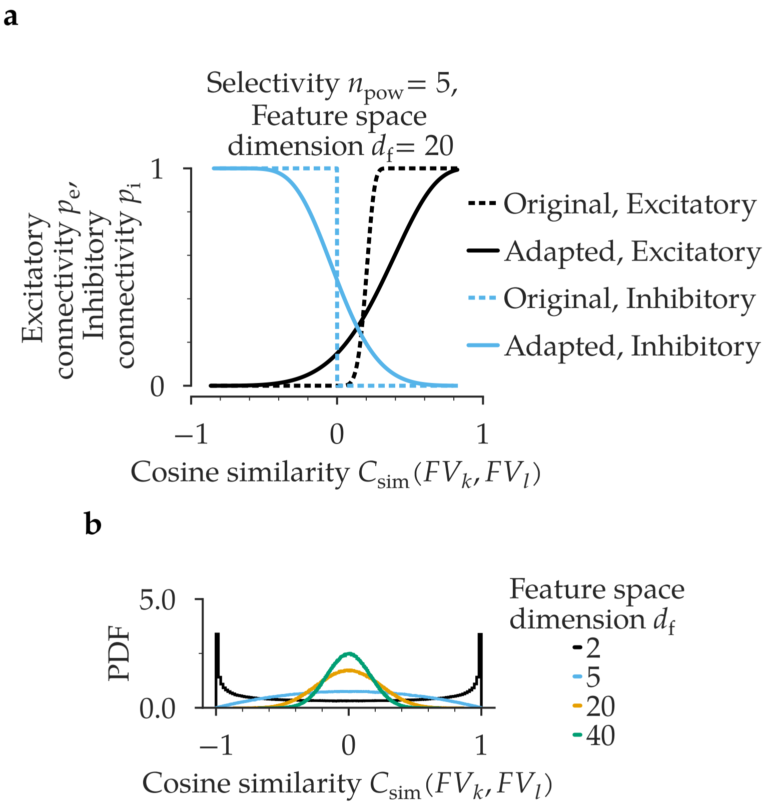

Figure 2.25. Feature vector cosine similarity and pairwise connection probabilities in the original and the adapted API model.
(a) Pairwise excitatory connectivity p_e and inhibitory connectivity p_i as function of the pairwise feature vector cosine similarity C_{sim}. (b) Distribution of the pairwise cosine similarity C_{sim} for varying feature space dimension d_f. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

2.8.3. Functional Viability

The key property of the API model is the excitation of cortical L4 neurons which are similarly tuned to the thalamic input and the inhibition of those which are dissimilarly tuned. The latter inhibition of antiphasicly tuned neurons has to be provided by the cortical circuit, as the thalamic input activity a_{in} is purely excitatory. To evaluate if this antiphase inhibition property is satisfied the circuit is simulated according to the discrete time dynamics

$$a(t+1) = (1 - \alpha\, a(t)) + \alpha\, W \max\{0, a(t) + a_{\text{in}}\},$$

with $\alpha = 1/10$ and max acting elementwise. The input $(a_{\text{in}})_k$ received by neuron k is given by

$$(a_{\text{in}})_k = \max\{0, C_{\text{sim}}(Stim, FV_k)\},$$

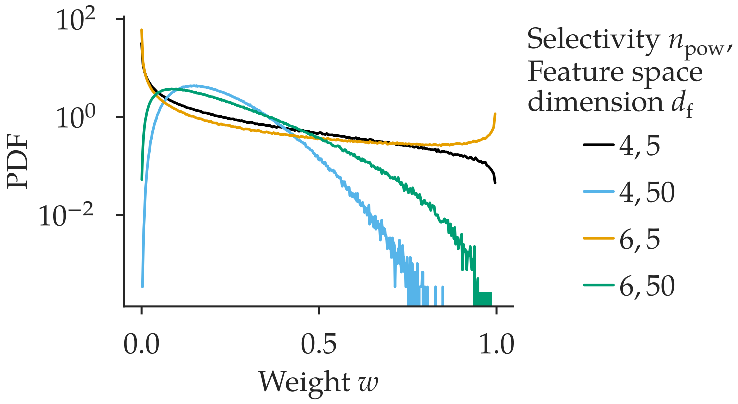

Figure 2.26. Excitatory weight distribution in the anti phase inhibition model. Probability density function (PDF) of the weight distribution. Selectivity (n_{pow}) and feature space dimension (d_{f}) are color coded. The weights are smaller on average for higher feature space dimension

in which $Stim = FV_{k^*}$ is the feature vector of a randomly chosen neuron $k^* \sim \mathcal{UI}(1, n)$. The weights w_{lk}[5] of the non-zero entries of W are proportional to the probability with which connection (k, l) is realized: $w_{lk} = s_k p((k, l) \in G)$ (Fig. 2.26). Notably, all entries of a_{in} are positive. The trial duration $t_{\mathrm{end}} = 100$ time constants. The simulation is implemented in Theano (Team et al. 2016).

Indeed, antiphasically tuned neurons (with negative C_{sim}) are in effect inhibited at t_{end} despite the excitatory input a_{in} they receive (Fig. 2.27). The effectiveness of the antiphase inhibition is summarized by considering the whole population's activity $a(t_{\mathrm{end}})$. The higher the correlation of $\{(a(t_{\mathrm{end}})_k, C_{\mathrm{sim}}(FV_k, Stim))\}_{k=1}^{n}$ the more effective is the antiphase inhibition. Of particular interest are the antiphasicly tuned neurons k with $C_{\mathrm{sim}}(FV_k, Stim) < 0$. Therefore, the correlation $C|_{C_{\mathrm{sim}}<0}$ of the points in the set $\{(a(t_{\mathrm{end}})_k, C_{\mathrm{sim}}(FV_k, Stim) | 1 \leq k \leq n,\ C_{\mathrm{sim}}(FV_k, Stim) < 0)\}$ is considered. The antiphase inhibition property vanishes with increasing d_{f} (Figs. 2.28a and 2.28b, see Fig. 2.28c for an example of the population activity $a(t_{\mathrm{end}})$ with efficient antiphasic inhibition and Fig. 2.28d for less efficient inhibition, i.e. less pronounced correlation). This behavior is explained by an increasing number of neuronal pairs with feature vector cosine similarity close to zero for increasing feature space dimension (Fig. 2.25b). If no dissimilarly tuned (C_{sim} close to -1) neurons exists they cannot be inhibited. From the above analysis is the prior $d_{\mathrm{f}} \sim \mathcal{UI}(5, 50)$ derived (Fig. 2.29), as the anti phase inhibition is still efficient for $d_{\mathrm{f}} \leq 50$. The adapted model is therefore considered functional under the chosen prior. Moreover, the adapted model satisfies the circuit constraints (Figs. 2.30a and 2.30b).

[5] w_{lk} is the weight for a connection (k, l) from k to l. The indices are reversed.

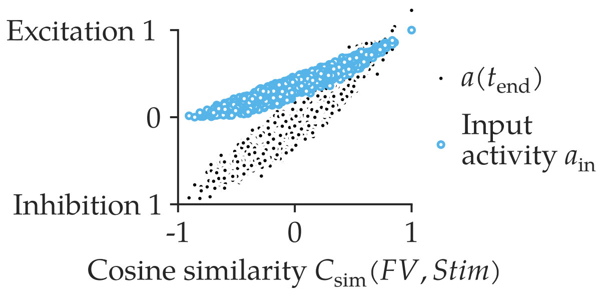

Figure 2.27. Stimulus tuning in the API model.
All neurons receive positive thalamic input. Neurons tuned antiphasicly relative to the thalamic input are inhibited in the API model through cortical projections. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

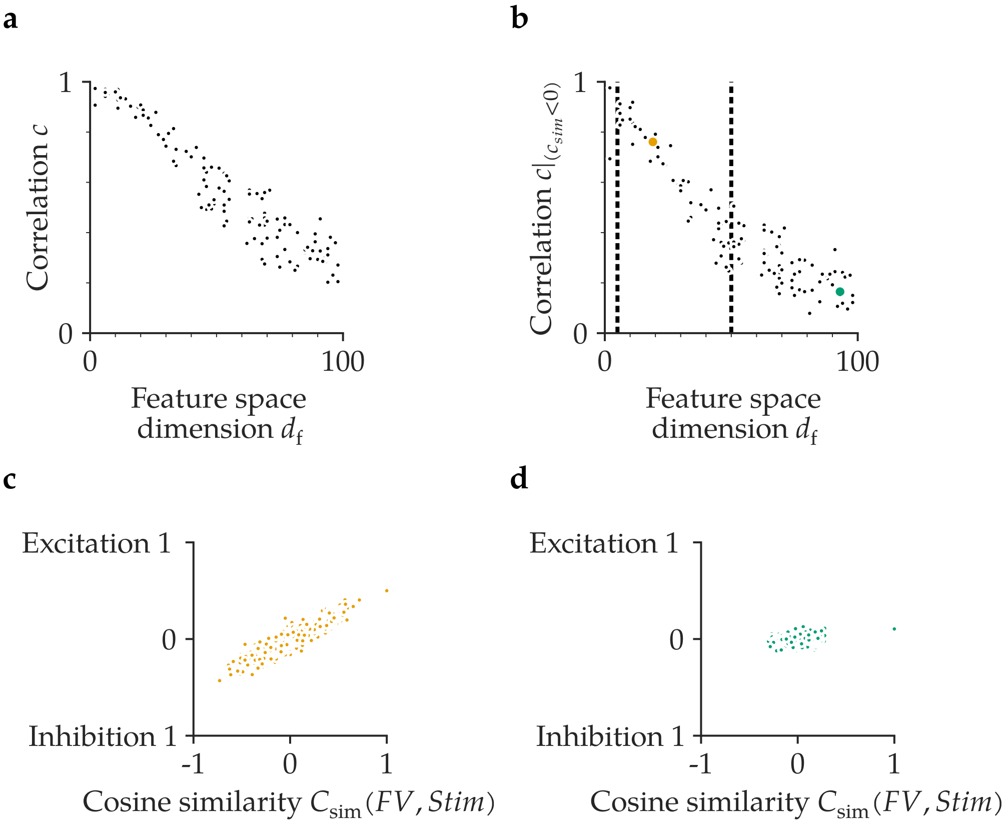

Figure 2.28. Efficiency of the anti phase inhibition as function of the feature space dimension. (a) Correlation of the network's activity over feature space dimension d_f. (b) Correlation of the activity of the subset of neurons with antiphasicly tuned feature vectors (relative to thalamic input) over feature space dimension d_f; the dashed lines indicate $d_f = 5$ and $d_f = 50$ which are the boundaries of the uniform prior over d_f. The two population activities $a(t_{\text{end}})$ corresponding to the highlighted points (● and ● are shown in Fig. 2.28c and Fig. 2.28d). (c) Complete network's activity $a(t_{\text{end}})$ from which the correlation restricted to antiphasicly tuned neurons is calculated in (b) for the highlighted point ●. (d) Complete network's activity $a(t_{\text{end}})$ from which the correlation restricted to antiphasicly tuned neurons is calculated in (b) for the highlighted point ●. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

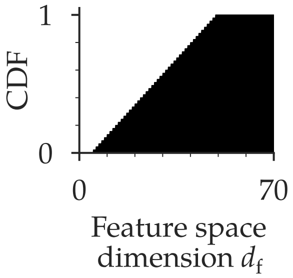

Figure 2.29. Prior over the feature space dimension d_f.
The feature space dimension d_f is uniformly distributed $d_f \sim \mathcal{UI}(5, 50)$. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

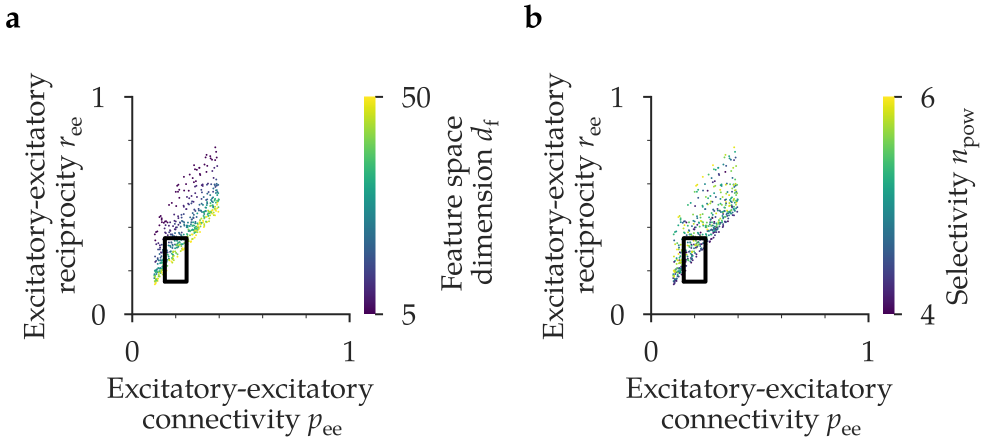

Figure 2.30. Adapted API model.
Excitatory-excitatory reciprocity r_{ee} over excitatory-excitatory connectivity p_{ee} for the adapted API model. The black rectangle indicates the circuit constraints. (a) The feature space dimension (d_f) is color coded. (b) The selectivity (n_{pow}) is color coded. The adapted model realizes networks satisfying the circuit constraints. (Compare to Figs. 2.24a and 2.24b.) (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

2.9. SORN: Self-Organizing Recurrent Neural Network

2.9.1. Key Concepts

With the self-organizing recurrent neural network (SORN) model (Lazar, Pipa, and Triesch 2009; Zheng, Dimitrakakis, and Triesch 2013) the hypothesis that barrel circuitry is shaped over time by local, self-organizing learning rules is investigated (Fig. 2.31a). The SORN was proposed by Lazar, Pipa, and Triesch (2007) and further analyzed by Lazar, Pipa, and Triesch (2009). However, similar ideas have also been investigated earlier (Bienenstock 1995; Hertz and Prügel-Bennett 1996; Levy et al. 2001). Lazar, Pipa, and Triesch (2009) considered several mechanisms, supposedly shaping the network structure: an intrinsic plasticity rule (Desai, Rutherford, and Turrigiano 1999; Zhang and Linden 2003), spike timing dependent plasticity (STDP) (Bi and Poo 1998; Gerstner et al. 1996; Markram et al. 1997) and synaptic normalization (Turrigiano et al. 1998) (Fig. 2.31b). Further plasticity rules, namely inhibitory STDP and structural plasticity (Fig. 2.31b) were added later and their topological effects were investigated (Zheng, Dimitrakakis, and Triesch 2013; Zheng and Triesch 2014). Reward-modulated learning was also examined (Aswolinskiy and Pipa 2015).

2.9.2. Implementation and Adaptation

The original SORN (Zheng, Dimitrakakis, and Triesch 2013) does not satisfy the circuit constraints (Fig. 2.32). The STDP rule prunes almost all reciprocated connections; the network has close to zero reciprocity (Fig. 2.32). Moderate alterations are thus necessary. In the following, the original and the adapted formulation are described and contrasted.

The initial adjacency matrix $\widetilde{W}$ is drawn as directed ER graph according to $\widetilde{W} \sim ER(n_\mathrm{e}, n_\mathrm{i}, p_\mathrm{e}, p_\mathrm{i})$ as defined in Eq. (2.10). Each possible excitatory connection is realized with probability p_e, each possible inhibitory connection with probability p_i. A weight normalized adjacency matrix W is generated from $\widetilde{W}$. The weights w_{uv} of W are obtained by normalizing the incoming excitatory weights of each neuron to 1 and the incoming inhibitory weights of each neuron to -1,

$$w_{uv} = \begin{cases} \frac{\tilde{w}_{uv}}{\sum_{v'=1}^{n_\mathrm{e}} \tilde{w}_{uv'}} & \text{if } v \leq n_\mathrm{e} \\ -\frac{\tilde{w}_{uv}}{\sum_{v'=n_\mathrm{e}+1}^{n} \tilde{w}_{uv'}} & \text{otherwise} \end{cases} .$$

No self-connections are allowed. The network W is then modified by dynamical learning rules over a temporal evolution. Neurons are modeled as binary threshold units with noisy input

$$x(t + \Delta t) = \Theta(Wx(t) + \xi - T),$$

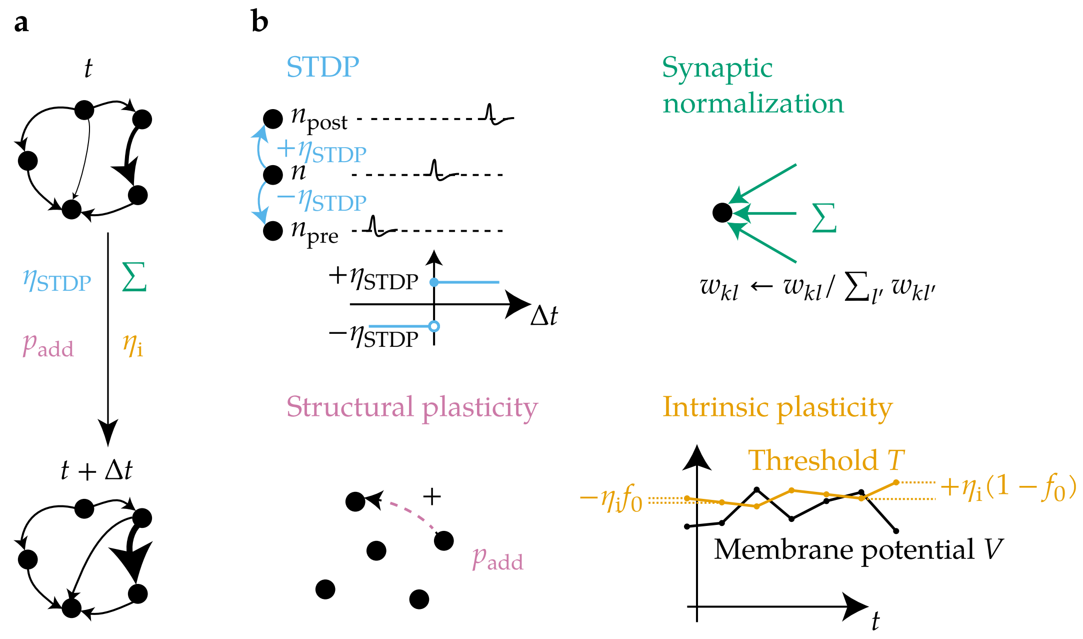

Figure 2.31. Learning rules in the SORN model.
(a) The network evolves from time t to $t + \Delta t$ applying four learning rules. (b) Spike timing dependent plasticity (STDP): a weight from neuron n to neuron n_{post} is increased by η_{STDP} if n fires an action potential at time Δt before neuron n_{post} or at the same time. A weight from neuron n to neuron n_{pre} is decreased by η_{STDP} if n_{pre} fires an action potential at time Δt before neuron n. Synaptic connections are pruned if the weight decrease would render a connection negative. Synaptic normalization: the incoming excitatory weights of each neuron are normalized to 1. Structural plasticity: new connections are randomly created with probability p_{add}. Intrinsic plasticity: the firing threshold T is increased by $\eta_{\text{i}}(1 - f_0)$ if the neuron fires and decreased by $\eta_{\text{i}} f_0$ if it does not fire. The target firing rate is f_0.

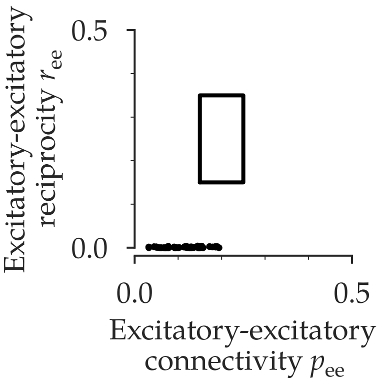

Figure 2.32. Excitatory-excitatory connectivity and excitatory-excitatory reciprocity for the original SORN model.
The circuit constraints (indicated by the black rectangle) are not satisfied by the original SORN model. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

in which $x \in \{0,1\}^n$ is the activity of the network, ξ denotes stochastic white noise input, $\xi_k \sim \mathcal{N}(0, \sigma^2)$ iid., $\sigma = 0.05$ and Θ is the Heaviside step function

$$\Theta(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

acting element-wise, and $T \in \mathbb{R}^n$ is the vector of firing thresholds. In this model, a neuron k fires if the sum of its synaptic inputs is greater than its firing threshold T_k. Six consecutive functional transformations are applied to evolve the network form one time step t to the next time step $t + \Delta t$, altering network properties such as the number of synapses (Figs. 2.33a to 2.33c). The neuronal activity $x \in \{0,1\}^n$ is initialized $x_k = 0 \forall k$ and the past activity $x^- \in \{0,1\}^n$ as well, $x_k^- = 0 \forall k$. The firing thresholds are initialized $T_k = 1 \forall k$.

1. **Propagation** The neuronal state vector x is updated $x \leftarrow \Theta(Wx + \xi - T)$.
2. **Intrinsic plasticity** The firing thresholds are updated $T \leftarrow T + \eta_\mathrm{i}(x - f_0)$, in which $f_0 = 1/10$ denotes the target firing rate and η_i the intrinsic learning rate (Fig. 2.31b).
3. **Synaptic normalization** The excitatory incoming weights of each neuron are normalized to 1 (Fig. 2.31b).
4. **STDP** The excitatory-excitatory weights are updated according to the STDP rule $w_{kl} \leftarrow w_{kl} + \eta_\mathrm{STDP} \left(x_k x_l^- + x_k x_l - x_k^- x_l \right)$ for $k \neq l$. The past state x^- is afterwards updated $x^- \leftarrow x$ (Fig. 2.31b).
5. **Pruning** Negative weights w_{kl} generated by the STDP rule are pruned: $w_{kl} \leftarrow 0 \forall k, l : w_{kl} < 0, l \leq n_\mathrm{e}$ (Figs. 2.33a and 2.33b).
6. **Structural plasticity** Random synapse creation attempts are made. It is attempted to add $n_\mathrm{add} = \frac{n_\mathrm{e}^2 p_\mathrm{e} - n_\mathrm{s}}{1 - p_\mathrm{e}}$ synapses between randomly chosen neuronal pairs (k, l), $k, l \sim \mathcal{UI}(0, n_\mathrm{e})$, denoting by $n_\mathrm{s} = \sum_{k,l: w_{kl} > 0} 1$ the number of excitatory synapses in the network (Figs. 2.31b and 2.33b). The synapse creation attempt fails if $k = l$ or if a connection from neuron k to neuron l already exists. A failed attempt is not substituted by a new attempt. The weight of a newly created synapse is $w_\mathrm{add} = 1/n$.

The model is implemented in Cython and parallelized with OpenMP for efficiency. Is is taken care to reduce memory access as far as possible due to its impact on the computational speed.

Compared to the original formulation (Zheng, Dimitrakakis, and Triesch 2013), moderate alterations are introduced. The STDP rule did originally not affect neuronal weights between neurons firing at the same time. The structural plasticity rule prescribed originally a constant number of synapse creation attempts per time step (Fig. 2.33c). Here, a homeostatic plasticity rule is introduced through which the number of addition attempts is adaptively adjusted (Fig. 2.33c). The overall number of synapses in the network stays

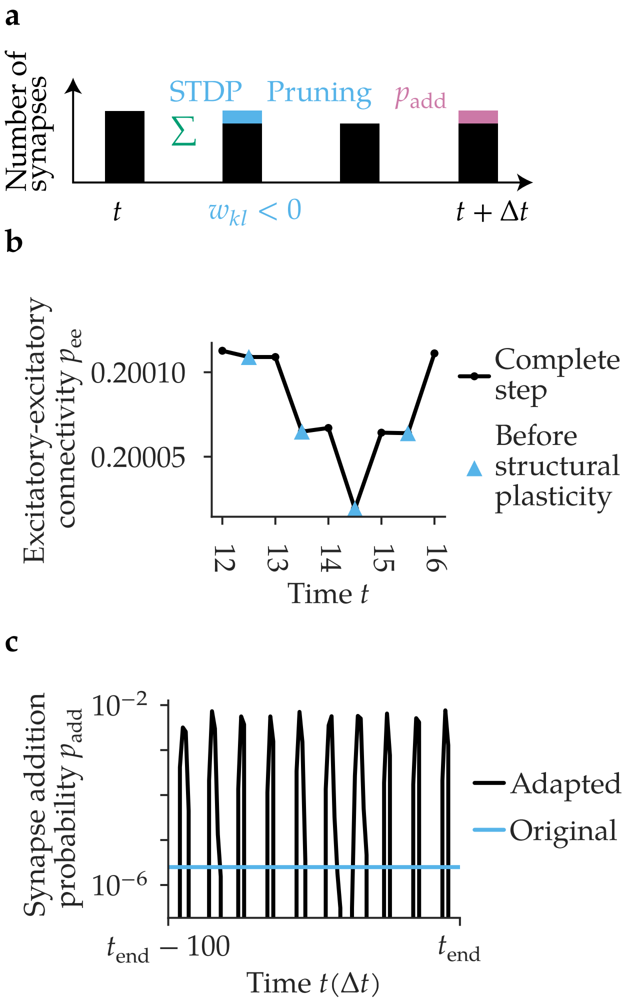

Figure 2.33. Effect of the learning rates on the number of synapses.
(a) The SORN learning rules affect the number of synapses present in the network on a sub-time step scale. Pruned synapses are approximately replaced through newly added synapses. (b) Synapses are pruned by the STDP rule and added by the structural plasticity rule resulting in a decrease followed by an increase of p_{ee}. (c) Originally, a small but constant number of synapse addition attempts is made each time step. In the adapted formulation, this is replaced by a homeostatic structural plasticity rule. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

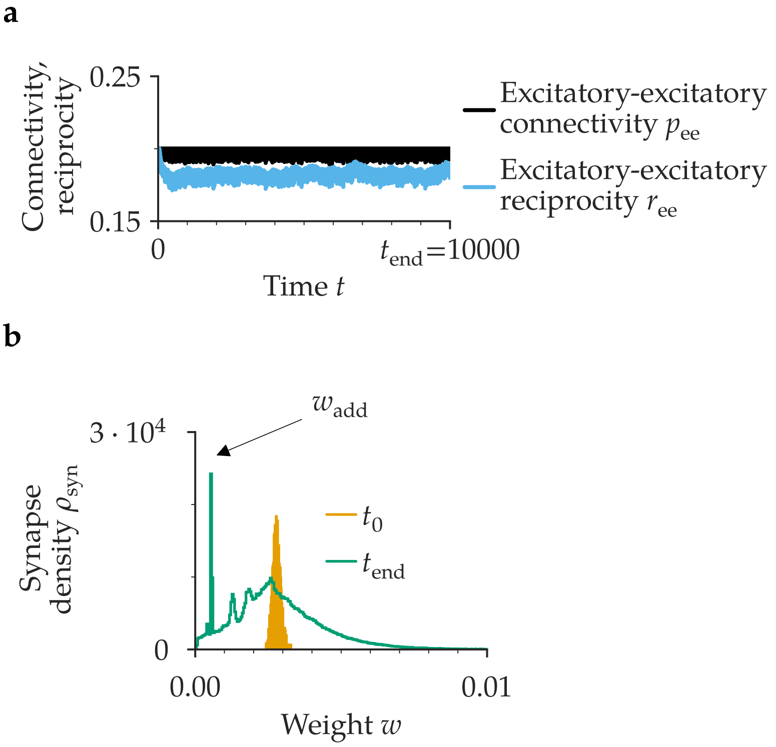

Figure 2.34. Connectivity, reciprocity and weight distribution of the adapted SORN model over time. (a) The excitatory-excitatory connectivity p_{ee} stays approximately constant; so does the excitatory-excitatory reciprocity r_{ee} after an initial decay phase. (b) Synapse density ρ_{syn} over synaptic weights w at the beginning $t_0 = 1$ and at the end $t_{\mathrm{end}} = 10000$. New synapses are created through the structural plasticity rule with weight w_{add}. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

approximately constant (Fig. 2.34a). The distribution of synaptic weights gets smeared out over time (Fig. 2.34b). Initially, at t_0, the distribution is narrower than at the end t_{end}.

2.9.3. Functional Viability

The SORN model was tested by Lazar, Pipa, and Triesch (2009) for healthy activity structure as measured by the correlation c between neuronal spiking – the closer to zero the better – and the spike source entropy (SSE) – the higher the better. The correlation $c_{k,l}$ of the neuronal pair (k, l) is calculated as

$$c_{k,l} = \frac{\sum_{t=t_0}^{t_{\mathrm{end}}} (x_k(t) - \overline{x_k})(x_l(t) - \overline{x_l})}{\sqrt{\sum_{t=t_0}^{t_{\mathrm{end}}} (x_k(t) - \overline{x_k})^2 \sum_{t=t_0}^{t_{\mathrm{end}}} (x_l(t) - \overline{x_l})^2}}$$

in which

$$\overline{x_k} = \frac{1}{t_{\text{end}}} \sum_{t=t_0}^{t_{\text{end}}} x_k(t)$$

denotes the mean firing rate and $t_0 = 1$. The obtained coefficients $c_{k,l}$ of the neuronal pairs are summarized as the median correlation $\text{med}(c_{\text{N,N}})$. This quantifies how synchronized the network activity is. Overly synchronized activity is regarded pathological and therefore implausible. The spike source entropy (SSE) is defined by

$$\text{SSE} = \frac{\sum_{k=1}^{n} \overline{x_k} \log \overline{x_k}}{-\log n},$$

which is the normalized entropy of the spike rates. Thus $0 \leq \text{SSE} \leq 1$. The maximal value $\text{SSE} = 1$ is attained if all the spike rates $\overline{x_k} = \overline{x} \forall k$ are identical and $0 < \overline{x} < 1$. The minimal value $\text{SSE} = 0$ is attained if, for example $\overline{x_k} = 1 \forall k$, $\overline{x_k} = 0 \forall k$ or $\overline{x_k} = 0$ for some neurons $k \in Z$ and $\overline{x_k} = 1$ for some other neurons $\overline{x_k} \in O$, $Z \cap O = \emptyset$. The SSE quantifies if the whole network spikes equally or if only a subpopulation spikes. The latter behavior is regarded pathological.

The median correlation $\text{med}(c_{\text{N,N}})$ increases with increasing η_{STDP} (Fig. 2.35a) but depends also on η_{i}. For appropriately chosen η_{STDP} and η_{i}, $\text{med}(c_{\text{N,N}})$ is very low, the network's activity decorrelated (Figs. 2.35a and 2.35c). For too large η_{STDP}, the network is very synchronized and its activity pathological (Fig. 2.35d). The spike source entropy (SSE) is comparatively unaffected by changes in η_{STDP} or η_{i}, always very close to 1 (Fig. 2.35b). Within the bounds of the priors over $0.05 \leq \eta_{\text{i}} < 0.1$ and $0.0006 \leq \eta_{\text{STDP}} < 0.0014$ (Figs. 2.35e and 2.35f) the network's activity correlation remains below 0.0422 and the SSE is greater than 0.99, similar to Lazar, Pipa, and Triesch (2009). The SORN model is hence regarded functional under the chosen prior. The adapted SORN model satisifes the circuit constraints (Fig. 2.36).

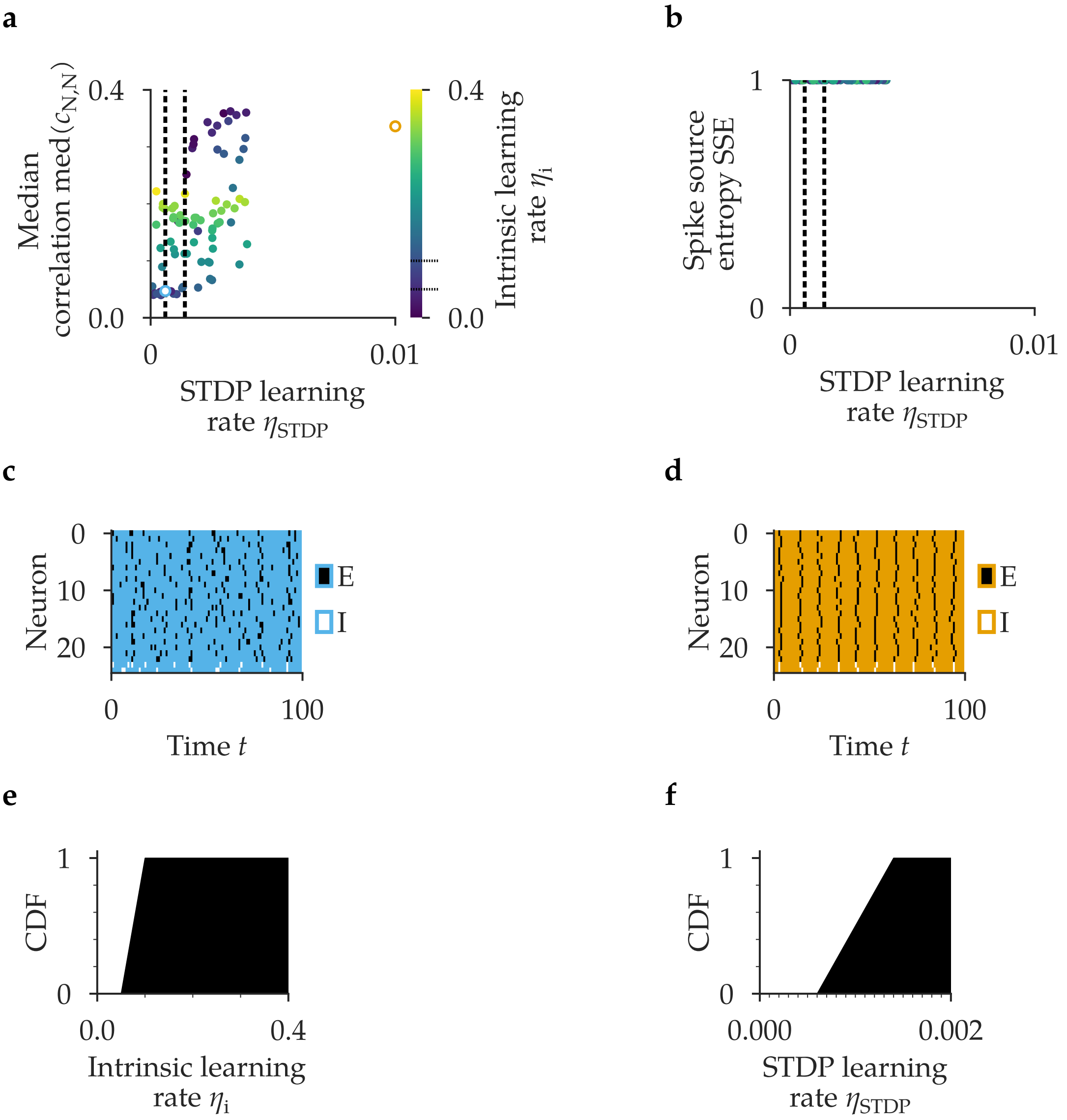

Figure 2.35. Prior and functional viability for the SORN model.
(a) Median correlation (med($c_{N,N}$)) over STDP learning rate (η_{STDP}); the intrinsic learning rate (η_i) is color coded. The activity at the highlighted parameters ○ and ○ is depicted in (c) and (d) respectively. The dashed lines indicate the boundaries of the support of the priors as depicted in (e) and (f). (b) Spike source entropy (SSE) over STDP learning rate (η_{STDP}); the intrinsic learning rate (η_i) is color coded. (c) Spiking activity of a SORN network with parameters corresponding to ○ in (a). Only a subset of the complete population is shown. Excitatory spikes ("E") are depicted in black, inhibitory spikes ("I") are depicted in white. (d) Spiking activity of a SORN network with parameters corresponding to ○ in (a). (e) Prior over the intrinsic learning rate (η_i) as cumulative distribution function (CDF). (f) Prior over the STDP learning rate (η_{STDP}) as cumulative distribution function (CDF). (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

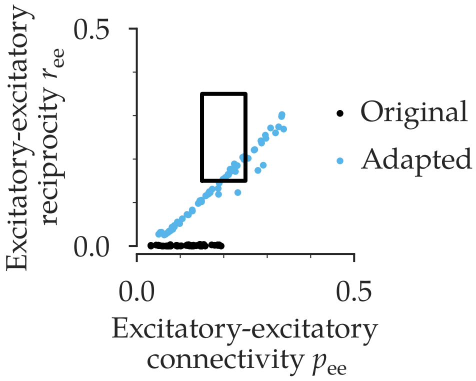

Figure 2.36. Excitatory-excitatory connectivity and excitatory-excitatory reciprocity for the adapted and the original SORN model.
The circuit constraints are not satisfied by the original model but are satisfied by the adapted model. ("Original" depicts the same data as in Fig. 2.32) (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

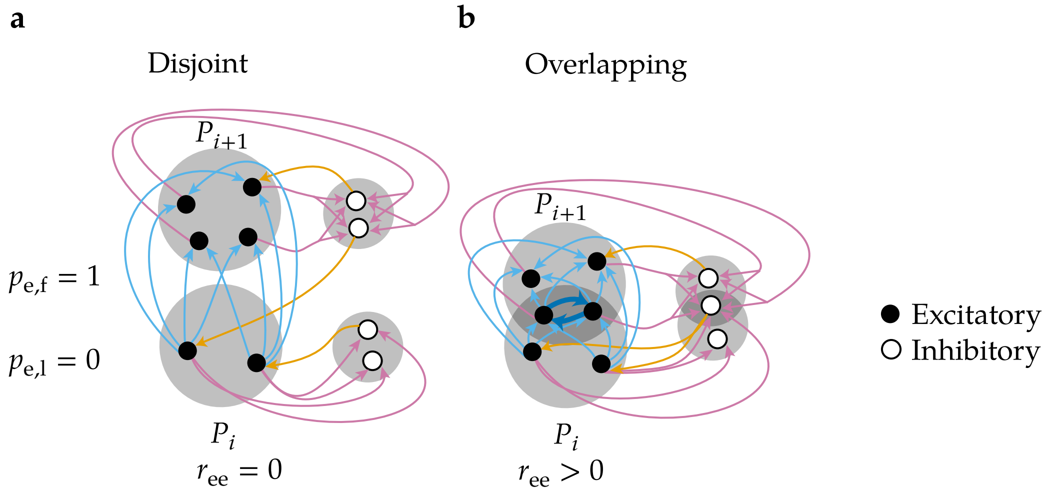

Figure 2.37. The synfire chain.
(a) Disjoint pools. Forward connections ↷ from pool P_i to pool P_{i+1} are realized all-to-all, i.e. with pairwise connection probability $p_{\mathrm{e,f}} = 1$. No within-pool connections are realized, $p_{\mathrm{e,l}} = 0$ and therefore $r_{\mathrm{ee}} = 0$. The inhibitory population is also organized in a pool structure, parallel to the excitatory pool structure. Excitatory-inhibitory connections ↷ are realized between corresponding pool pairs. Inhibitory-excitatory connections ↷ are realized onto each excitatory neuron with independently with probability p_{i}. (b) Overlapping pools. Reciprocated connections ↷ emerge between neurons participating in more than one pool, consequently $r_{\mathrm{ee}} > 0$. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

2.10. SYNFIRE: Synfire Chain

2.10.1. Key Concepts

The synfire chain (SYNFIRE) model was proposed by Abeles (1982) as explanation for cortical synchrony (Abeles 1991; Abeles 1982). In its simplest formulation, a SYNFIRE is a sequence of neuronal pools[6] with connections form one pool to the next pool (Fig. 2.37a). It has been proposed that SYNFIRE patterns were the fundamental computational units in cortex (Bienenstock 1995) and that composed or hierarchical mental representations were implemented by interacting SYNFIREs (Abeles, Hayon, and Lehmann 2004; Bienenstock 1995). Storage capacity, embedding and computational properties of SYNFIREs were examined (Aviel, Horn, and Abeles 2005; Aviel, Mehring, et al. 2003; Bienenstock 1995; Gewaltig, Diesmann, and Aertsen 2001; Goedeke and Diesmann 2008; Herrmann, Hertz, and Prügel-Bennett 1995; Hertz 1997; Kumar, Rotter, and Aertsen 2008; Mehring et al. 2003; Trengove, Leeuwen, and Diesmann 2012) but also the question of how to find evidence for the implementation of SYNFIREs has been addressed (Schrader et al. 2008). SYNFIREs were used for translation-invariant pattern recognition in pictures (Arnoldi, Englmeier,

[6] A neuronal pool is a set of neurons

and Brauer 1999), to parse and classify French sentences (Jacquemin 1994), for auditory scene analysis (Wrigley and Brown 1999) and in compositional systems to address the binding problem, i.e. the binding of many simple components to a meaningful mental representation of a composite object (Aertsen and Braitenberg 1996; Hayon, Abeles, and Lehmann 2005).

The simplest implementations were networks with connections only between subsequent pools (Fig. 2.37a) and a single chain only (Gewaltig, Diesmann, and Aertsen 2001; Griffith 1963). Other studies allowed multiple chains in a SYNFIRE network (Abeles 1982; Abeles, Hayon, and Lehmann 2004; Aviel, Mehring, et al. 2003; Bienenstock 1995; Herrmann, Hertz, and Prügel-Bennett 1995; Schrader et al. 2008). Pools were also allowed to overlap (Abeles 1991; Abeles 1982; Aviel, Horn, and Abeles 2005; Aviel, Mehring, et al. 2003; Bienenstock 1995; Herrmann, Hertz, and Prügel-Bennett 1995; Schrader et al. 2008; Trengove, Leeuwen, and Diesmann 2012) thereby forming a recurrent network which still operated, similar to multilayer perceptrons (Rosenblatt 1961), in a feed forward fashion (Abeles 1991) (Fig. 2.37b). Neurons have been restricted to participate in each chain only once while still being allowed to participate in several chains (Schrader et al. 2008). The emergence of SYNFIREs by plasticity mechanisms was suggested (Bienenstock 1995; Hertz and Prügel-Bennett 1996; Levy et al. 2001; Zheng and Triesch 2014). A tendency towards short cycles formed by 4 to 5 pools and under certain input conditions up to 20 pools was observed (Hertz and Prügel-Bennett 1996; Levy et al. 2001; Zheng and Triesch 2014). SYNFIREs embedded in other networks have been examined (Abeles, Hayon, and Lehmann 2004; Aviel, Horn, and Abeles 2005; Aviel, Mehring, et al. 2003; Kumar, Rotter, and Aertsen 2008; Mehring et al. 2003). SYNFIREs were either assumed to be complete, i.e. featuring all-to-all connectivity between pools (Abeles 1991; Abeles 1982; Bienenstock 1995; Gewaltig, Diesmann, and Aertsen 2001; Mehring et al. 2003) or were allowed to be incomplete (Abeles 1991; Abeles 1982; Abeles, Hayon, and Lehmann 2004; Bienenstock 1995). It was also proposed hat connections could omit layers by means of non-constant conduction delays (Bienenstock 1995). Furthermore, continuous synfire braids (Bienenstock 1995), relying on non-uniform conduction delays, and other related models were proposed (Izhikevich 2006; Izhikevich, Gally, and Edelman 2004; Paugam-Moisy, Martinez, and Bengio 2008). For a review on SYNFIREs see for example Abeles (2009).

2.10.2. Implementation and Adaptation

The SYNFIRE formulation employed in this work follows Trengove, Leeuwen, and Diesmann (2012) implementing a single chain with overlapping pools. This implementation is chosen since disjoint pools would yield no reciprocated connections (Fig. 2.37a) and could therefore not satisfy the circuit constraints. To construct the chain, neuronal pools are consecutively randomly drawn and then connected (Fig. 2.37b) as described in Alg. 3: (1) an initial excitatory source pool $P_{\mathrm{E}}^{\text{source}}$ of pool size s_{pool} is uniformly drawn from the excitatory population. (2) Next, an excitatory target pool $P_{\mathrm{E}}^{\text{target}}$ of size s_{pool} is uniformly drawn from the excitatory population and an inhibitory target pool $P_{\mathrm{I}}^{\text{target}}$ of size $s_{\text{pool,i}}$ is uniformly drawn from the inhibitory population. The excitatory source and target pools are allowed to share neurons, i.e. the pools are allowed to overlap. (3) The excitatory source pool is connected all-to-all to the excitatory target pool and is also connected all-to-all to the inhibitory target pool. (4) The excitatory target pool becomes the excitatory source pool in the next iteration. Steps (1) to (4) are repeated n_{pool} times, so that the connectivity p_{e} is attained in expectation. The number of pools n_{pool} is obtained by

Algorithm 3: Synfire chain construction

```
Input: $V, p_{\mathrm{e}}, p_{\mathrm{i}}, s_{\text{pool}}, n_{\mathrm{e}}$
Output: $G$
$E \leftarrow \emptyset$
$n_{\text{pool}} \leftarrow \left\lfloor \frac{\log(1-p_{\mathrm{e}})}{\log\left(1-\frac{s_{\text{pool}}^2}{n_{\mathrm{e}}^2}\right)} \right\rfloor$
$V_{\mathrm{E}} \leftarrow \{v \in V | v \leq n_{\mathrm{e}}\}$
$V_{\mathrm{I}} \leftarrow V \setminus V_{\mathrm{E}}$
$n_{\mathrm{i}} \leftarrow |V| - n_{\mathrm{e}}$
$s_{\text{pool,i}} \leftarrow \left\lfloor \frac{n_{\mathrm{i}}}{n_{\mathrm{e}}} s_{\text{pool}} \right\rfloor$
$P_{\mathrm{E}}^{\text{source}} \sim \mathcal{U}(\{V' | V' \subset V_{\mathrm{E}}, |V| = s_{\text{pool}}\})$
for $k = 1$ to $n_{\text{pools}}$ do
    $P_{\mathrm{E}}^{\text{target}} \sim \mathcal{U}(\{V' | V' \subset V_{\mathrm{E}}, |V'| = s_{\text{pool}}\})$
    $P_{\mathrm{I}}^{\text{target}} \sim \mathcal{U}(\{V' | V' \subset V_{\mathrm{I}}, |V'| = s_{\text{pool,i}}\})$
    $E \leftarrow E \cup P_{\mathrm{E}}^{\text{source}} \times \left(P_{\mathrm{E}}^{\text{target}} \cup P_{\mathrm{I}}^{\text{target}}\right)$
    $P_{\mathrm{E}}^{\text{source}} \leftarrow P_{\mathrm{E}}^{\text{target}}$
end
$G \leftarrow (V, E)$

foreach $(u, v) \in V_{\mathrm{I}} \times V_{\mathrm{E}}$ do
    $b \sim \mathrm{Bin}(p_{\mathrm{i}})$
    if $b = 1$ then
        $E \leftarrow E \cup \{(u, v)\}$
    end
end
```

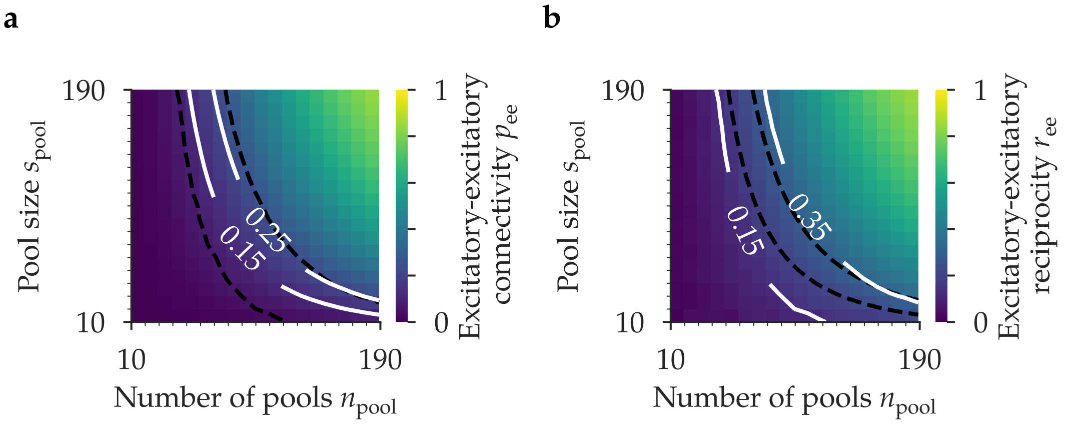

Figure 2.38. Connectivity and reciprocity of the synfire chain.
(a) Pool size (s_{pool}) over number of pools (n_{pool}); the excitatory-excitatory connectivity (p_{ee}) is color coded. The white isolines indicate the circuit constraints. The black-dashed lines are the white isolines from (b). (b) Pool size (s_{pool}) over number of pools (n_{pool}); the excitatory-excitatory reciprocity (r_{ee}) is color coded. The white isolines indicate the circuit constraints. The black-dashed lines are the white isolines from (a). (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

solving Eq. (2.14) for n_{pool}

$$p_{\text{ee}} = 1 - \left(1 - \frac{s_{\text{pool}}^2}{n_{\text{e}}^2}\right)^{n_{\text{pool}}} . \tag{2.14}$$

In Eq. (2.14), $\frac{s_{\text{pool}}^2}{n_{\text{e}}^2}$ is the fraction of connections attempted to be added in each iteration, i.e. the probability with which an excitatory-excitatory connection is realized, which yields the formula for n_{pool} in Alg. 3. Similarly, setting $s_{\text{pool,i}} \leftarrow \left\lfloor \frac{n_{\text{i}}}{n_{\text{e}}} s_{\text{pool}} \right\rfloor$ in Alg. 3 ensures that $p_{\text{ei}} = p_{\text{e}}$. Inhibitory-excitatory connections are independently realized with pairwise connection probability p_{i}.

The SYNFIRE model satisfies the circuit constraints for appropriate choices of s_{pool} and n_{pool} (Figs. 2.38a and 2.38b). The excitatory-excitatory reciprocity r_{ee} increases with increasing s_{pool} (Fig. 2.39). However, a lower bound on s_{pool} is expected from a functional consideration since stable propagation of activity seems unlikely for very small pool sizes.

2.10.3. Functional Viability

The SYNFIRE model aims for synchronous propagation of spike groups along the chain (Abeles 1982; Trengove, Leeuwen, and Diesmann 2012). The SYNFIRE is therefore functionally tested to achieve such propagation stably (Fig. 2.40a). For this, the first pool is

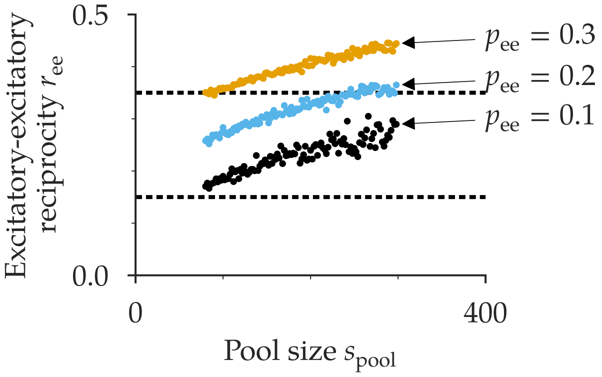

Figure 2.39. Excitatory-excitatory reciprocity of the synfire chain as function of the pool size. The excitatory-excitatory reciprocity r_{ee} increases with increasing pool size s_{pool} for constant excitatory-excitatory connectivity p_{ee}. The dashed lines indicate the circuit constraints. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

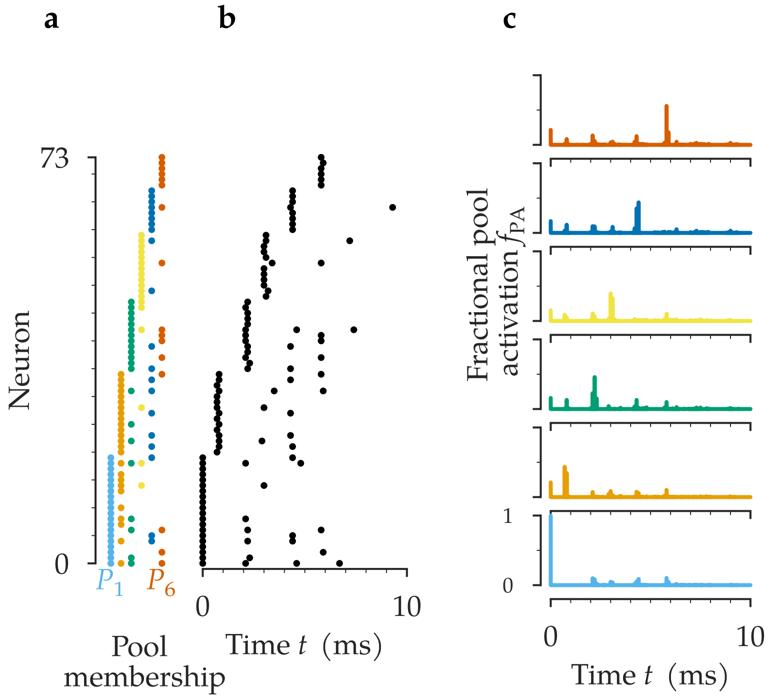

Figure 2.40. Pool activation in the synfire chain model.
(a) Neurons in the SYNFIRE model are grouped into pools $P_1, \ldots, P_{n_{pool}}$ (here $n_{pool} = 6$ and only a fraction of the neurons of the network is depicted). Each neuron is allowed to be member of more than a single pool. (b) Pool P_1 is initially activated. The remaining pools are sequentially activated ny their predecessor pool as time t progresses. The number of activated pools determines the quality of the SYNFIRE implementation. (c) Fractional pool activation (f_{PA}) over time (t). (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

Table 2.2. Parameters of the dynamical SYNFIRE model.
The index i refers to pools, the indices k and l to neurons. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

Parameter	Symbol	Value
neuronal time constant	τ_{p}	20 ms
excitatory reversal potential	$v_{\text{reversal,exc}}$	0 mV
inhibitory reversal potential	$v_{\text{reversal,inh}}$	-80 mV
spiking threshold	$v_{\text{threshold}}$	-55 mV
resting potential	v_{rest}	-70 mV
excitatory refractory period	$\tau_{\text{ref,exc}}$	2 ms
inhibitory refractory period	$\tau_{\text{ref,inh}}$	1 ms
inter pool delay	d_i^{pool}	$\mathcal{U}(0.5, 2)$, iid.
excitatory inter pool jitter	$d_{kl}^{\text{jitter,exc}}$	$\mathcal{U}(0, 0.3)$, iid.
inhibitory inter pool jitter	$d_{kl}^{\text{jitter,inh}}$	$\mathcal{U}(0.3, 0.9)$, iid.
efficacy jitter	g_{jitter}	0.7

initially fully activated. The remaining neurons are initially inactive and are subsequently activated by their respective predecessor pools (Fig. 2.40b). Pools have to be activated in the correct sequence; no pool should be skipped or prematurely activated. A pool is considered activated if at least half of its constituting neurons are activated (Fig. 2.40c).

To dynamically simulate the SYNFIRE model, a conductance based, spiking neuronal model

$$\dot{v} = (v_{\text{rest}} - v)/\tau_{\mathrm{p}}$$

is employed, in which $\tau_{\mathrm{p}} = 20\,\mathrm{ms}$ denotes the neuronal time constant and v the membrane potential. The membrane potential v_l of neuron l is increased by $g_{kl}(v_{\text{reversal},k} - v_l)$ when neuron k spikes and $(k, l) \in E$; it is not affected otherwise. The efficacy g_{kl} is a function of the pool size (Fig. 2.41) and the neuronal type

$$g_{kl} \sim \begin{cases} g_e(s_{\text{pool}})(1+u) \text{ if } k \leq n_{\mathrm{e}} \\ g_i(s_{\text{pool}})(1+u) \text{ otherwise} \end{cases}, \quad u \sim \mathcal{U}(-g_{\text{jitter}}/2, g_{\text{jitter}}/2).$$

The efficacy decreases with increasing pool size (Fig. 2.41). This is compensated by the larger number of neurons providing input to a single postsynaptic neuron. The membrane potential of neuron k is set to the reversal potential v_{rest} after a spike is emitted by neuron k. A neuron spikes if its membrane potential surpasses $v_{\text{threshold}}$. An excitatory neuron is deactivated after a spike over the excitatory refractory period $\tau_{\text{ref,exc}}$, an inhibitory neuron over the inhibitory refractory period $\tau_{\text{ref,inh}}$. The synaptic delay, i.e. the time between

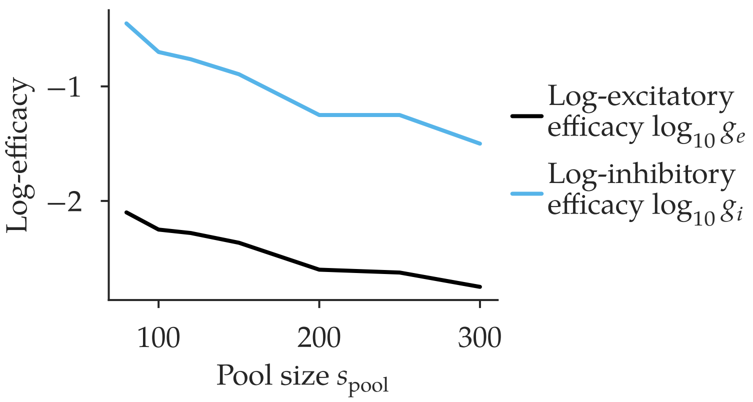

Figure 2.41. Synaptic efficacies as function of the pool size.
Log-synaptic efficacy over pool size (s_{pool}). The functional form of g_e and g_i are obtained by optimizing the fractional chain activation. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

spiking of neuron k and arrival of the spike at neuron l, between consecutive pools P_i and P_{i+1} is d_i^{pool}. The delay varies additionally per connection (k,l) between these pools with $d_{kl}^{\text{jitter,exc}}$. Several connection attempts are possibly made between each pair (k,l). In such a case, the delay between k and l is the one of the last connection attempt. The overall delay d_{kl} is hence

$$d_{kl} = d_{\hat{i}}^{\text{pool}} + d_{kl}^{\text{jitter,exc}}, \quad \hat{i} = \max\{i : (k,l) \in P_i \times P_{i+1}\}.$$

The delay is $d_{kl}^{\text{jitter,inh}}$ for inhibitory synapses (k,l). The parameters' values are listed in Table 2.2. The model is implemented in the "Brian 2" simulation environment (Goodman and Brette 2008; Goodman and Brette 2009).

The fractional chain activation f_{CA} is calulcated as follows: let $n_i(t)$ denote the number of active neurons in pool P_i at time t. Define the pool activity indicator

$$\delta_i(t) = \begin{cases} 1 & \text{if } n_i(t) > \frac{s_{\text{pool}}}{2}, n_j(t) < \hat{n}_i(t) \forall j \neq i \\ 0 & \text{otherwise} \end{cases}.$$

That is, a pool P_i is considered active at time t, $\delta_i(t) = 1$, if at least half of its neurons are active, and it is the unique pool with the maximum number of active neurons at time t (all other pools are less active). Denote the cumulative activity of pool P_i up to time t, i.e. the number of spikes emitted by the pool during its activity

$$c_i(t) = \sum_{t':\leq t} n_i(t')\delta_i(t').$$

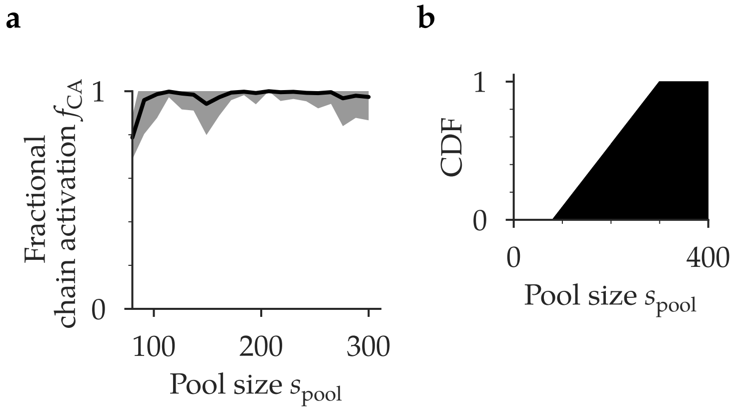

Figure 2.42. Functional viability of the synfire chain under the prior parameter distribution. (a) Fractional chain activation (f_{CA}) over pool size (s_{pool}). (b) The pool size (s_{pool}) is uniformly distributed $s_{\mathrm{pool}} \sim \mathcal{UI}(80, 300)$. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

The end time of a trial is $t_{\mathrm{end}} = \max\{t|c_i(t) < \alpha\, s_{\mathrm{pool}} \forall i\}, \alpha = 1.2$. That is, if at least one pool has more than αs_{pool} cumulative spikes during its periods of activity, the trial is considered terminated. The number of activated pools N is the number of pools activated before the end time $N = |\{i|\exists t < t_{\mathrm{end}}$ and $\delta_i(t) = 1\}|$. The fractional activation $f_{\mathrm{CA}} = N/n_{\mathrm{pool}}$ is the number of activated pools normalized by the total number of pools n_{pool}.

The average fractional activation of the chain is robustly greater than 95% for $s_{\mathrm{pool}} \geq 100$ and still achieves about 80% for $s_{\mathrm{pool}} = 80$. The prior is therefore chosen uniform on the integers in $[80, 300]$ (Fig. 2.42b). The circuit constraints are satisfied under the prior (Figs. 2.38a, 2.38b and 2.39) and the SYNFIRE model is considered functional.

3. Distributed and Adaptive Approximate Bayesian Computation

In this chapter, an explorative connectomic analysis approach is first examined and its shortcomings are demonstrated (Sect. 3.1). A more systematic approach to model selection is then introduced (Sect. 3.2). To cope with the high computational demands associated with this approach, pyABC (Sect. 3.3), a distributed inference software framework used here for model selection, and a scheme which adaptively selects one of the main options of this framework, the population size (Sect. 3.4), are developed.

3.1. The Insufficiency of Explorative Connectome Analysis and the Need for a Systematic Approach

Section is similar to Klinger, Marr, Theis, Helmstaedter (under review)

It is necessary to falsify or to find evidence for theories of cortical computation to improve the understanding of these (Denk, Briggman, and Helmstaedter 2012). Here, having established a range of competing hypotheses (Chap. 2) it is now necessary to establish a method which enables to select the models which explain a given local cortical module best. So far, it is largely unclear how this goal can be achieved. In the past it was tried to find the best fitting model out of a range of competing models on the basis of functional data (Barak et al. 2013). But alternatively, structural (connectomic) data might be used as well. These data have the advantage of being independent of behavioral paradigms or anesthesia conditions. The data (once available) allows to extract a directed graph, with neurons represented as its nodes and connections as its edges.

However, it is so far largely unclear whether structural data is informative about local computational cortical models or not (Bargmann and Marder 2013; Denk, Briggman, and Helmstaedter 2012; Jonas and Kording 2017; Morgan and Lichtman 2013). The positions are quite extreme: on the one hand it is argued, that structural data might be hard to interpret (Bargmann 2012; Jonas and Kording 2017), on the other hand it is argued, that structural data might be highly relevant to understand functional properties (Denk, Briggman, and Helmstaedter 2012). It is here explored to which extent structural data aids to discriminate models of cortical computational models. The focus lies on a concrete circuit module: a layer 4 “barrel”.

To which extent can exploratory approaches yield insights about the likely implemented models? For simplicity and practical applicability only unweighted graphs are considered in the following and the problem of relating morphological properties such as synapse sizes and numbers to connection strengths is not treated. Networks are commonly depicted in adjacency matrix representation. A naive, exploratory approach to cortical model selection is to ask whether inspection of the adjacency matrix might reveal interesting structures. Amongst the models considered here (Chap. 2), the LAYERED model is one of the models with the most clearly defined and best understandable structure. Inspecting the adjacency matrix, however, not even the structure of a LAYERED model is visually discernible (Fig. 3.1, left column, first row from top). Of course, the structure perceived by the human eye in such a matrix depends highly on the sorting of the rows and columns. Since the neurons do not carry labels, all joint permutations of the rows and columns describe the same network structure, but transmit a visually quite different experience. One possible route to find a meaningful permutation is fitting a stochastic block model (SBM) (H. C. White, Boorman, and Breiger 1976). The SBM attempts to detect hidden (latent) groups in the network. The network is organized in a block structure if such groups exist. Therefore, SBMs are fitted to connectomes sampled from the different models using the graph-tool software (`https://graph-tool.skewed.de`) for Python with the default settings of the `graph_tool.inference.minimize.minimize_blockmodel_dl` function.

Examination of connectome samples obtained from the investigated model classes shows, that the SBM indeed reveals the layered structure of the LAYERED model (Fig. 3.1, left column, second row from top, same network as directly above). This is expected, since the LAYERED model is explicitly organized into blocks. The ordering of the blocks $b_1, \dots b_4$ is $b_1 \rightarrow b_3 \rightarrow b_4 \rightarrow b_2$. Also the EXP-LSM, SYNFIRE and API model exhibit some structure. However, for the API and EXP-LSM model not much beyond the existence of some notion of similarity of neurons and organization of the network by such a principle can be said. In the SYNFIRE case, the interpretation of the SBM fit is less obvious. Even worse, the SORN, ER-ESN and FEVER model seem not to possess any structure at all.

The SBM approach raises at least two questions: First, does the failure of the SBM approach to model selection imply that connectomes sampled from theses model classes are not structurally discriminable or is the SBM just not suitable to discriminate the underlying models? Second, and even more importantly, how to decide from the SBM fit which model is the most likely implemented one? The explorative SBM approach seems to be insufficient for the model selection task – and so is any other exploratory approach, since the second question will remain unanswered in any case. A more systematic approach is required.

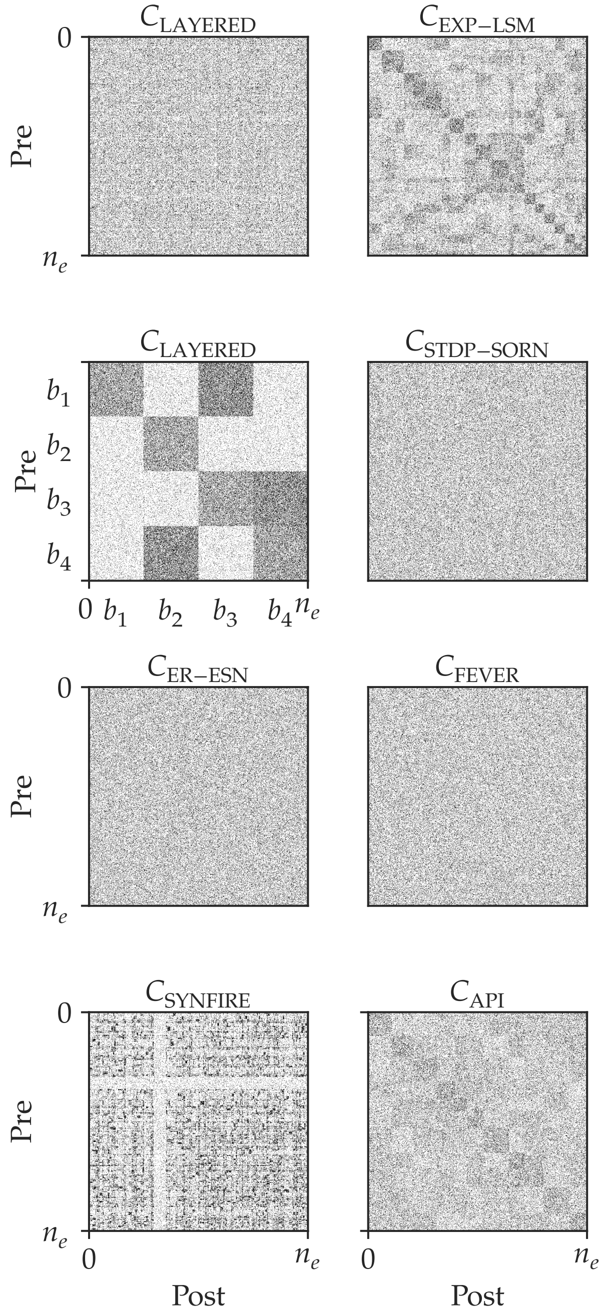

Figure 3.1. Stochastic block model fits do not discriminate the underlying models.
Upper left panel: layered network with random neuron permutation. Remaining panels: neuron sorting according to a stochastic block model fit. Although similarities between, e.g., EXP-LSM and API and dissimilarities between, e.g. SORN and LAYERED are visible, stochastic block model fits do not allow for systematic model selection. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

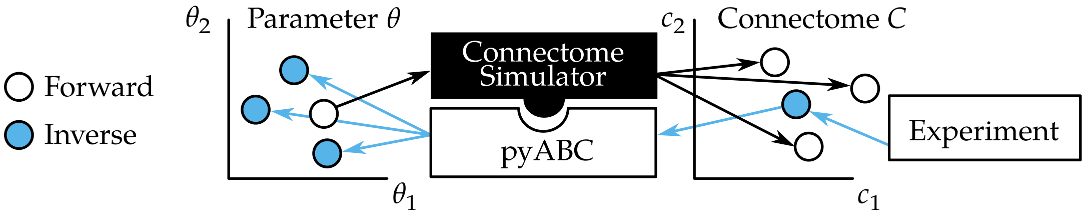

Figure 3.2. Parameter inference for a simulator model.
Forward direction: model simulations of a parameter θ are realized by a stochastic simulator. This generates artificial connectomes C.
Inverse direction: a connectome C is experimentally obtained. The Approximate Bayesian Computation (ABC) approach, implemented in pyABC, uses the simulator model to infer the parameters θ which likely explain the observed connectome C.

3.2. A Systematic Approach to Likelihood-free Model Selection and Parameter Inference: Approximate Bayesian Computation - Sequential Monte Carlo

Ideally, model selection would be performed systematically in a fully Bayesian way, instead of by mere explorative inspection. Ideally, the posterior probability $p(m|C)$ of a model class m given a connectome C could be calculated. However, many of the local cortical candidate models (Chap. 2) are formulated as simulator models only. Lacking efficiently computable expressions of these models' likelihood functions, the fully Bayesian approach is severely complicated.

The problem encountered here is a widely encountered one. By the means of numerical simulations, modeling efforts (such as here the modeling of a local cortical module) can be extended into regimes which are not analytically amenable anymore. These models are equipped with a set of parameters which have to be inferred from experimentally observed data to enable predictions from the model or to select the most likely parameter (Fig. 3.2). However, it is more difficult to select the model parameters which likely explain an experimental finding, than simulating the finding artificially given the correct parameters (Beaumont 2010).

No tailored approaches exist for the connectomic model selection problem, although tailored approaches exist for some other model classes, which exploit certain characteristics of these model classes. For example, specialized toolboxes exist for the parameterization of ordinary differential equations (Raue et al. 2015). Similarly, methods such as "accelerated maximum likelihood" (Daigle et al. 2012) or the "generalized method of moments" (Lück and Wolf 2016) are specialized to parameterize chemical reaction networks. However, similarly specific methods do not exist for other model classes, such as multi-scale models or the models considered here (Chap. 2). These models have to be

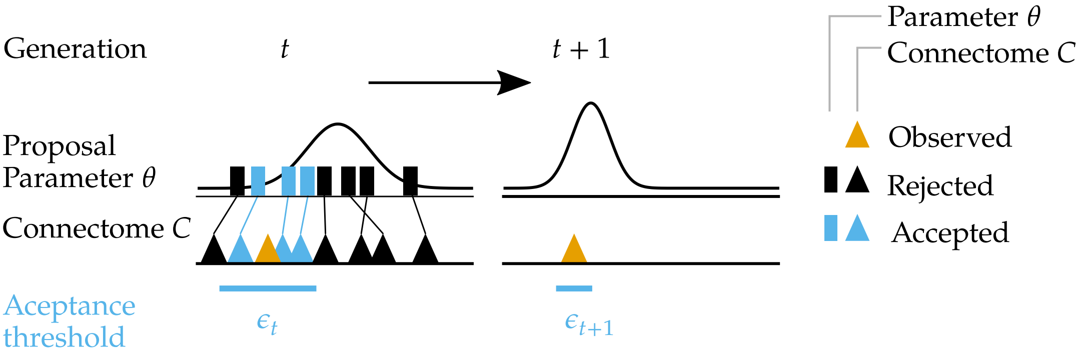

Figure 3.3. Approximate Bayesian Computation - Sequential Monte Carlo scheme. Proposal parameters (and models) are sampled based on the population of generation t. The model is simulated for these parameters. If the corresponding simulated connectome C is close to the observed connectome, the parameter is accepted for the next population; it is rejected otherwise. For the next population $t+1$, the approximation quality of the posterior is increased by decreasing the acceptance threshold ϵ, i.e., the maximally allowed distance an artificially generated connectome may have such that its parameter is still accepted for the subsequent generation.

considered as black-boxes, i.e. they can be simulated but their internal structure cannot be exploited for inference.

The main problem is, that to perform analytical Bayesian model selection it is necessary to compute the likelihood function. In the aforementioned cases, of multi-scale models and many of the models introduced in Chap. 2 this is unfortunately not possible anymore as only algorithmic descriptions of how to simulate data from these models are available. To enable parameter inference – and also model selection – in cases where the likelihood is not tractable but the model can still be simulated, likelihood-free approaches such as ABC have been developed (Beaumont, Zhang, and Balding 2002; Marjoram et al. 2003). One particularly popular and computationally efficient class of ABC schemes are the ABC-SMC schemes (Sisson, Fan, and Tanaka 2007; Toni and Stumpf 2010; Toni, Welch, et al. 2009).

In ABC-SMC, the posterior distribution of a parameter, given some (observed) data, is approximated through a sample from the posterior distribution or an approximate posterior distribution. This sample is commonly referred to as (particle) population. The populations are iteratively refined over the course of several consecutive generations of the ABC-SMC scheme to approximate the posterior increasingly well (Fig. 3.3). The model is repeatedly numerically evaluated to obtain those samples.

ABC-SMC methods are actively researched. Contributions were made on the question of how to generate proposal parameters for the subsequent population from the previous generation to maximize the fraction of accepted particles (Filippi et al. 2013; Koutroumpas et al. 2016), how to adjust the acceptance threshold from generation to generation (Silk, Filippi, and Stumpf 2013), and how to choose the summary statistics to describe and compare the observed data and artificially sampled data (Marin et al. 2014; Nunes and Balding 2010). Beyond these questions, two key questions which are relevant for the practical application of ABC-SMC in the context of connectmic model selection are addressed here:

1. Parallelization of the model simulations on distributed hardware.
2. Selection of the population sizes.

Efficient and scalable parallelization is essential to infer parameters for even the most computationally demanding models or in regimes of very low acceptance rates (Jagiella et al. 2017). Here, the simulation runtimes are very heterogeneous. While some models simulate within less than a second on modern hardware (ER-ESN, EXP-LSM), others are more demading (FEVER, SORN). The problem of choosing the population size adequately has been noted in the literature before: "It appears unfortunately difficult to give useful general guidelines how to select [the population size] as it is highly case dependent" (Moral, Doucet, and Jasra 2012). If the population size is chosen too small, the approximation error might be large, if it is chosen too large, the ABC-SMC scheme might become computationally inefficient. These two key questions are addressed by

1. implementing pyABC, a general-purpose distributed ABC-SMC framework for Python 3 (Sect. 3.3, `http://pyabc.readthedocs.io`), and
2. the development of a scheme for the adaptive selection of population sizes in ABC-SMC (Sect. 3.4).

The pyABC framework was used for connectomic model selection (Chap. 4). The scheme for automated selection of population sizes was added later and was not applied to connectomic model simulation, but instead analyzed with the help of a range of example applications.

3.3. pyABC: Distributed, Likelihood-free Inference

3.3.1. Usage

Section is similar to Klinger, Rickert, Hasenauer (under review)

Some ABC-SMC frameworks for Python exist already. However, these support Python 2 only (Ishida et al. 2015; Jennings and Madigan 2017; Kangasrääsiö et al. 2016; Liepe et al. 2010) and have a range of other shortcomings. They lack flexible customization options such as (adaptive) acceptance threshold schedules or transition kernels (Kangasrääsiö et al. 2016), only implement parallelization strategies which do not scale well (Jennings and Madigan 2017), can only parallelize in a multi-core setting (Ishida et al. 2015), or do only use GPUs for parallelization but cannot exploit distributed infrastructure (Liepe et al. 2010). The combination of features present in pyABC is unique:

- single-core, multi-core and distributed execution
- distributed model selection
- less scalable computation-minimizing and more scalable runtime-minimizing parallelization strategies, for both multi-core and distributed execution
- adaptive, local transition kernels, adaptive acceptance threshold schedules, adaptive population size selection
- web interface for the visualization of posterior parameter distributions, acceptance threshold, configuration options and other relevant information
- early stopping of model simulation
- easy interruption and continuation of ABC-SMC runs
- interface to the R language to support execution of models defined in R
- pluggable, extensible architecture facilitating experimentation with new ABC-SMC schemes without alteration of pyABC's source code.

pyABC is applicable to model selection and parameter inference. It is user-friendly, in particular to new users, as model definitions can be as simple as defining a function taking the model's parameters as input and returning the simulated data or the summary statistics of them:

```
import scipy as sp
def model(parameter):
    return {"trajectory": sp.cumsum(sp.randn(100) + parameter.delta)}
```

It is a conscious decision as part of pyABC's design not to enforce usage of model definition languages such as the Systems Biology Markup Language (SBML). If necessary, SBML simulators can be called internally. A distance function d can be defined as function with two parameters x and y returning their distance, for example

```
import scipy as sp
def distance(x , y):
    return sp.absolute(x.trajectory - y.trajectory).sum()
```

Priors can be conveniently defined with the help of the `scipy.stats` package. For example, a uniformly distributed random variable $x \sim \mathcal{U}(0, 1)$ can be defined by

```
from pyabc import RV
x = RV("uniform", 0, 1)
```

Post-processing of ABC-SMC runs is facilitated through pyABC's API for data querying, providing the user with Pandas data frames for further analysis. Beyond the provided functionality, custom query routines are easily implemented on top of the relational database used by pyABC for logging and storage of the ABC-SMC runs. pyABC is extensively documented, featuring an API documentation, a user's guide, a developer's guide, tutorial style examples and how-to guides (`http://pyabc.readthedocs.io`).

3.3.2. Distributed Computation

pyABC is designed to be particularly suitable for distributed computing in cloud or cluster environments. Distributed computation is highly important for ABC inference since the often very computational demanding models have to be simulated repeatedly. Only efficient parallelization schemes render ABC model selection possible at all. The pyABC framework therefore supports a variety of distributed execution modes, including ad hoc clusters, such as the Dask distributed cluster or the IPython cluster. It runs on bare grid submission systems such as SGE or UGE. Moreover, it ships with a low-latency distributed execution engine which leverages Redis as a message broker. pyABC supports two different parallelization strategies: static scheduling (STAT) and dynamic scheduling (DYN). In the following, let n denote the population size.

Static Scheduling (STAT) In the STAT strategy, n parallel tasks (jobs, processes or threads) run independently of each other. Each tasks samples new parameter proposals from the proposal distribution of the current generation and evaluates the model until exactly one parameter is found whose corresponding simulated data is accepted (Fig. 3.4a, left). Once the tasks obtains one accepted parameter it terminates. It is waited until all n tasks are finished. Even if the number of available cores is larger than n, this strategy only uses n cores. If the number of available cores is smaller than n, the tasks are queued (e.g. by the job submission system such as SGE).

Dynamic Scheduling (DYN) In the DYN strategy all parallel tasks sample proposal parameters and simulate the model continuously on all available hardware until n particles are accepted (Fig. 3.4a, right). Once this is the case, it is waited for each task to finish its current model simulation (which might be accepted or not), yielding potentially $m \geq n$ accepted particles. The n accepted particles which were started first (not necessarily accepted first due to non-constant model simulation times) are then included in the next generation. The remaining $m - n$ particles are discarded. This is necessary to prevent a bias towards parameters with shorter model simulation times.

The STAT and the DYN strategy are optimized for different purposes and have different advantages and drawbacks. For the STAT strategy, the degree of parallelism is always

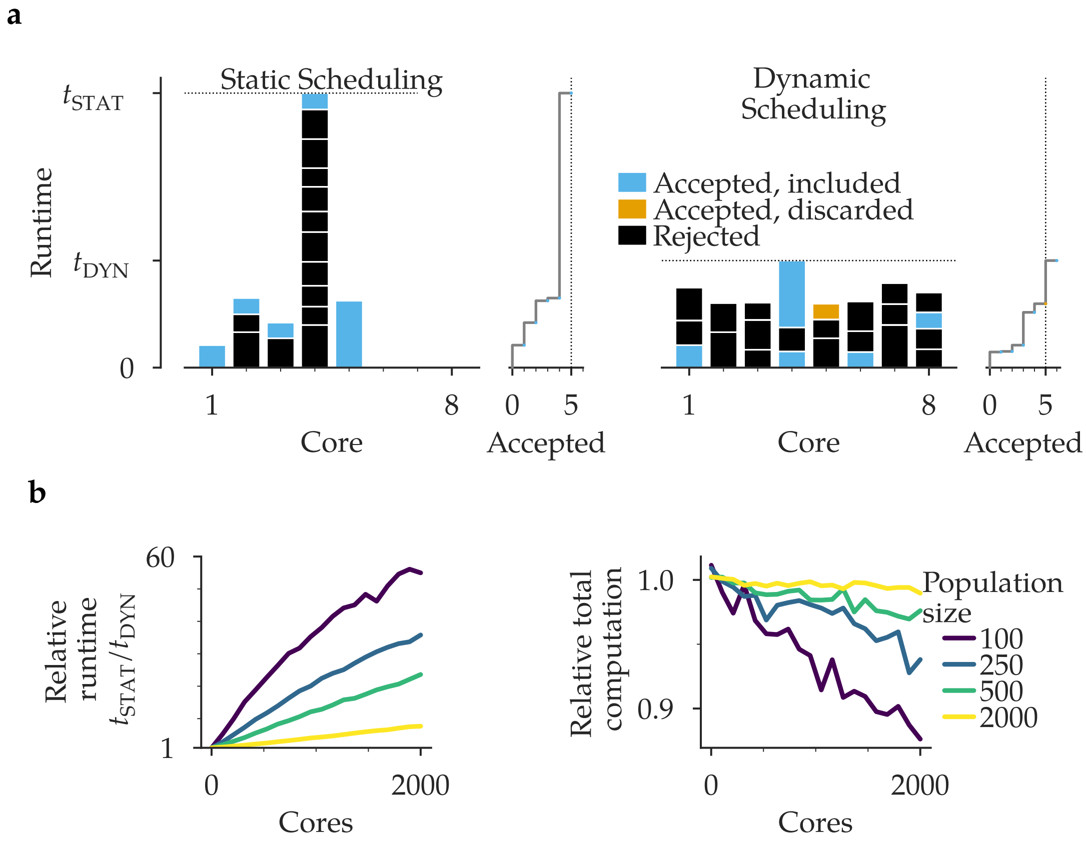

Figure 3.4. Comparison of the sampling strategies implemented in pyABC.
(a) Static Scheduling (STAT) and Dynamic Scheduling (DYN) illustrated for 5 particles and 8 cores. The model simulation times are exponentially distributed and are the same for STAT and DYN. The coloring indicates whether a particle is accepted and included in the next generation, accepted but discarded, or rejected. The cumulative number of accepted particles over time is also depicted. For DYN, the number of accepted particles surpasses the target population size of 5 particles in this example. (b) Average of relative runtime ratios and relative amount of computation of STAT relative to DYN as function of the population size and the number of available cores. The DYN strategy is faster than the STAT strategy (left), but the STAT strategy performs less computation overall (right) and is therefore less resource demanding. (Figure similar to Klinger, Rickert, Hasenauer (under review))

limited to the population size n. It decreases even over the course of a population, as particles get accepted (Fig. 3.4a, left). The DYN strategy, in contrast, uses all available hardware until n particles are accepted. The degree of parallelism decreases therefore only for a comparatively short period of time (on the order of a single model evaluation) towards the very end of a population. Even this decrease can be eliminated if model simulations are interrupted, which might be safe if no clean-up, such as e.g. deleting temporary files generated during model simulation, has to be done. Importantly, the DYN strategy is more scalable, scaling further even if the number of available cores is larger than the population size n (Fig. 3.4b, left). However, it also performs more computation overall (Fig. 3.4b, right), which can be problematic if the computational resources are scarce. Additionally, the DYN strategy requires worker-server communication after each accepted particle. If the model simulation times are very short (much less than a second) and the network is comparatively slow, the communication overhead might be notable. The STAT strategy requires only two worker-server interactions per accepted particle. For typical acceptance rates as low as 10^{-3} or 10^{-5}, as sometimes encountered in the very last populations, this implies several orders more worker-server interactions.

3.3.3. Configuration, Extension and Development

pyABC follows established design patterns (Gamma et al. 1994) and best software-engineering practices to achieve a highly configurable and extensible ABC framework. From the five “SOLID” principles (Martin 2013), in particular the “open/closed principle” is respected. As a result, many ABC options, such as transition kernels, acceptance threshold schedules, distance functions, summary statistics, execution engines, population size selection strategies, database back ends and more can be configured and customized through user provided implementations without alterations of pyABC’s source code. Therefore, pyABC is not only ideally suited for the applied user, wishing to perform ABC-SMC inference on a particular problem, but also well suited for the advanced and research oriented users experimenting with and developing new ABC-SMC schemes. One example of pyABC’s advanced usage pattern is “early stopping” in model simulation. For applications such as Markov Jump Processes or stochastic differential equations, the distance function is often a cumulative sum over the time points in the simulation and can be calculated during model simulation instead of at the end (see for example Sect. 3.4.6). Once this sum surpasses the allowed acceptance threshold, the simulation cannot be accepted anymore. In such cases, performance improvement is gained by stopping the simulation early. pyABC is well tested, shipping with a large set of unit and integration tests. Continuous integration is performed with the help of the Travis continuous integration system.

Algorithm 4: ABC-SMC scheme

```
Input: $t_{\max}$, $\epsilon_{\min}$, $n_0$, $p_0$, $s_{\text{obs}}$, KDE, $d$
Output: $P$
$t \leftarrow 0$
$K \leftarrow p_0$
$\epsilon \leftarrow \infty$
$n \leftarrow n_0$
while $t < t_{\max}$ and $\epsilon > \epsilon_{\min}$ do
    $(P, D) \leftarrow$ sample_population$(K, p_0, \epsilon, n, s_{\text{obs}}, d)$
    $K \leftarrow \text{KDE}(P)$
    $n \leftarrow$adapt_population_size$(P, \text{KDE})$
    $\epsilon \leftarrow$ adapt_threshold$(D)$
    $t \leftarrow t + 1$
end
```

3.4. Adaptive Population Size Selection in Approximate Bayesian Computation - Sequential Monte Carlo

3.4.1. ABC-SMC Scheme

Section is similar to Klinger, Hasenauer (2017)

In ABC-SMC, the posterior distribution of the models of interest and their parameters is approximated through a particle population $P = \{(w_i, \theta_i)\}_{i=1}^{n}$ of n parameter samples $\theta_i \in \mathbb{R}^{d_{\text{par}}}$ of parameter dimension d_{par} with associated weights $w_i > 0$ and $\sum_i w_i = 1$. The curly braces {} denote here a sequence – not a set – since parameter weight combinations, albeit unlikely, might occur multiple times in a single populations. The ABC-SMC scheme considered here, and provided in Alg. 4, is based on the scheme proposed by Toni, Welch, et al. (2009). The particle population is sequentially refined, yielding a sequence of populations approximating the posterior distribution increasingly well. For this, proposal parameters are drawn from a proposal density K and the model is stochastically simulated yielding the simulated data $s \in \mathcal{S}$ (Fig. 3.3). The proposal density K is obtained from a kernel density estimator KDE : $P \mapsto K$ mapping the population P to a density estimate $K : \mathbb{R}^{d_{\text{par}}} \to \mathbb{R}^+$, $\int K = 1$. Toni, Welch, et al. (2009) formulated the ABC-SMC scheme in terms of a perturbation kernel, instead of a kernel density estimate. However, application of a perturbation kernel is a special case of sampling from a kernel density estimate. After each generation, this density is updated with an estimate from the current population. The newly obtained estimate K serves then as proposal for the subsequent population. Initially, the proposal density $K = p_0$ is the prior distribution p_0. The observed data $s_{\text{obs}} \in \mathcal{S}$ is compared to the simulated data s with the help of a distance function $d : \mathcal{S} \times \mathcal{S} \to \mathbb{R}^+$ (Fig. 3.3). A parameter θ is accepted if the distance $\delta = d(s, s_{\text{obs}})$ of its generated data s to the observed data s_{obs} is smaller than the current acceptance threshold ϵ: $\delta < \epsilon$. Thereby a sequence $D = \{(w_i, \delta_i)\}_{i=1}^n$ of weighted distances is constructed for the population P. The acceptance threshold ϵ is decreased from generation to generation to improve the approximation quality sequentially. Throughout this work, the acceptance threshold is adapted through the function `adapt_threshold` $: D \mapsto \epsilon$ which sets as new threshold the (weighted) median of the weighted distances D of the previous generation. The population size is adapted from generation to generation via `adapt_population_size` (see Sect. 3.4.3 for a detailed treatment of this adaptation and Alg. 6 for pseudo code). The adaptation scheme is detailed in Sect. 3.4.3. Initially, the population size is set to $n_0 \in \mathbb{N}$. The sampling procedure is stopped when either the acceptance threshold is small enough, $\epsilon < \epsilon_{\text{min}}$ for a predefined $\epsilon_{\text{min}} > 0$, or the maximally allowed number of generations t_{max} is reached.

Sampling of a new population is implemented in the function `sample_population`, explained in Alg. 5. The population P and the corresponding weighted distances D are initialized as empty sequences $\{\}$. Then, a single tentative parameter θ is sampled from the proposal density K. Since the proposal density might yield parameters outside the support of the prior distribution p_0, this process is repeated until a parameter θ with $p_0(\theta) > 0$ is obtained. Stochastic model simulation is implemented in the function `simulate` $: \theta \mapsto s$, which simulates the model for parameter θ and returns stochastically realized data s. The parameter θ, together with its corresponding weight w is appended to the population P only if its distance $\delta = d(s, s_{\text{obs}})$ to the observed data s_{obs} is less than ϵ. In this case, the weighted distance (w, δ) is also appended to the sequence of weighted distances D. Here, concatenation of sequences is denoted by the $+$ operator.

In the model selection case, the parameter θ, the prior p_0 and the proposal density K decompose into a component over the model classes and into a component over the model specific parameters. Let M denote the total number of models. For model selection, the parameter θ is a sequence of model specific parameters $(\theta_m)_{m=1}^M$ with $\theta_m \in \mathbb{R}^{d_{\text{par,m}}}$ in which $d_{\text{par,m}}$ denotes the dimension of the parameter space of model m. The prior factorizes similarly into a component over the models and one over the parameters according to $p_0(\theta) = p_0(\theta_m|m)p(m)$; the proposal density analogously $K(\theta) = K(\theta_m|m)K(m)$. The output of the ABC-SMC scheme from Alg. 4 is an approximate sample from the posterior distribution. The algorithm is implemented in the pyABC framework (`http://pyabc.readthedocs.io`).

Algorithm 5: `sample_population`

```
Input: $K, p_0, \epsilon, n, s_{\text{obs}}, d$
Output: $(P, D)$

$P \leftarrow \{\}$
$D \leftarrow \{\}$
$Z \leftarrow 0$

while $|P| < n$ do
    repeat
        repeat
            $\theta \leftarrow$ sample_parameter($K$)
        until $p_0(\theta) > 0$;
        $s \leftarrow$ simulate($\theta$)
        $\delta \leftarrow d(s, s_{\text{obs}})$
    until $\delta < \epsilon$;
    $w \leftarrow p_0(\theta)/K(\theta)$
    $Z \leftarrow Z + w$
    $P \leftarrow P + \{(w, \theta)\}$
    $D \leftarrow D + \{(w, \delta)\}$
end
$P \leftarrow \{(w/Z, \theta) | (w, \theta) \in P\}$
$D \leftarrow \{(w/Z, \delta) | (w, \delta) \in D\}$
```

3.4.2. Kernel Density Estimation

To refine the populations from generation to generation, new parameter proposals are sampled based on the particles accepted in the previous population. This is often (Toni, Welch, et al. 2009) achieved by selecting an accepted parameter with probability according to its weight and perturbing it with a perturbation kernel. This process is a special case of sampling from a non-parametric distribution approximation – a kernel density estimate. Here, all employed kernel density estimators share the same common structure.

Structure of the Density Estimators

The density estimate $K = \text{KDE}(P)$ is a superposition of normal kernels

$$K(\theta') = \sum_{(w,\theta)\in P} w\, \mathcal{N}(\theta' | \theta, \Sigma(P, \theta)),$$

in which P denotes the particle population and $\mathcal{N}(\theta' | \theta, \Sigma(P, \theta))$ is a normal density with mean θ and covariance $\Sigma(P, \theta)$, evaluated at θ'. The covariance matrix is in the following referred to as bandwidth. Notably, the bandwidth $\Sigma(P, \theta)$ is, in general, a function of the complete population P and the location θ around which the kernel is centered. Three different bandwidths selection strategies are employed: global bandwidth selection (Silverman 1986), local bandwidth selection (Filippi et al. 2013) and cross-validated bandwidth selection (Bowman 1984; Rudemo 1982; M. Stone 1974).

Global Bandwidth Selection

In the global bandwidth selection strategy, the bandwidth does not depend on the (local) parameter θ around which the kernel is centered, that is $\Sigma(P, \theta) = \Sigma$. It is therefore called "global". The global bandwidth is proportional to the covariance $\operatorname{cov} P$ of the entire population

$$\operatorname{cov} P = \sum_{(w,\theta)\in P} w(\theta - \overline{\theta})(\theta - \overline{\theta})^t, \quad \overline{\theta} = \sum_{(w,\theta)\in P} w\,\theta.$$

The factor of proportionality b_{Silv} is determined according to Silverman's rule of thumb (Silverman 1986)

$$b_{\text{Silv}} = \left(\frac{4}{n_{\text{eff}}(d_{\text{par}} + 2)} \right)^{\frac{1}{d_{\text{par}}+4}},$$

in which d_{par} denotes the parameter dimension, n_{eff} the effective population size

$$n_{\text{eff}} = \frac{1}{\sum_{w\in\{w\}} w^2}$$

and $\{w\}$ is the sequence of weights of P. The kernel bandwidth Σ is finally obtained as $\Sigma = b_{\text{Silv}}^2 \operatorname{cov} P$.

Local Bandwidth Selection

The global bandwidth selection strategy can yield poor approximations of the underlying distribution (Salgado-Ugarte and Perez-Hernandez 2003; Silverman 1986). It does not take into account possible local distribution properties such as, multimodality. Therefore, local bandwidths $\Sigma_{k,\text{nn}}(P, \theta)$ are considered which are constructed individually for each parameter θ form its k nearest neighbors (measured in Euclidean distance). Hence, let $\Sigma_{k,\text{nn}}(P, \theta)$ denote twice the covariance of the k nearest neighbors of θ. The local density K is defined as

$$K(\theta') = \sum_{(w,\theta)\in P} w\,\mathcal{N}(\theta' | \theta, \Sigma_{k,\text{nn}}(P, \theta)).$$

This form of density estimation has been analyzed with respect to acceptance rates in ABC-SMC before and was shown to perform well (Filippi et al. 2013).

Cross-Validated Bandwidth Selection

The cross-validated bandwidth selection strategy is also a global bandwidth selection strategy since the same bandwidth is used for all kernels. Silverman's rule of thumb is known to yield too large bandwidths, i.e. over smoothing density estimates (Silverman 1986). It is therefore considered to replace Silverman's rule of thumb by cross-validated selection of the scaling factor of the population covariance matrix cov P. A grid consisting of five different scaling factors is probed. The largest scaling factor $b_0 = b_{\text{Silv}}$ is the one obtained according to Silverman's rule of thumb, the other factors $b_1 > b_2 > b_3 > b_4$ are obtained by down-scaling on an exponential scale

$$b_c = 2^{-\frac{c}{2C}} b_{\text{Silv}},$$

with $c \in \{0, \dots, C\}$ and $C = 4$. The quality of a scaling factor b_c is determined according to the score function

$$S(K_c, P') = \sum_{(w,\theta) \in P'} w \log K_c(\theta),$$

in which $P' \subset P$ is a sub-population of P and K_c the density estimated on its complement $\overline{P'}$ with bandwidth b_c (the score function is always evaluated on the sub-population which was not used for density estimation). Five fold cross-validation is applied and the scaling factor b_{c^*} yielding the highest cumulative score is selected. The density estimation is then repeated on the complete population P with the scaling factor b_{c^*} yielding the final density estimate K.

3.4.3. Population Size Adaptation

In ABC-SMC schemes a trade-off between efficiency (computational cost) and accuracy has to be made. Besides the strategy applied to generate proposal parameters for the subsequent population, an important parameter is the population size. Increasing the population size improves the accuracy but also increases the total amount of computation necessary, and thereby decreases the ABC-SMC scheme's efficiency. Not only the size of the last population, but also the sizes of all the intermediate populations have material influence. If intermediate populations are chosen too large, more computation than necessary is performed, if they are chosen too small, information about the posterior might get lost which cannot be efficiently recovered in the last population, rendering the ABC-SMC scheme inaccurate.

For instance, assume that the true posterior has two modes, but the second mode has a comparatively small mass. Small intermediate populations might lack samples in the second mode. It is unlikely, that this mode is recovered in the last population, unless the last population is chosen very large. However, this would render ABC-SMC

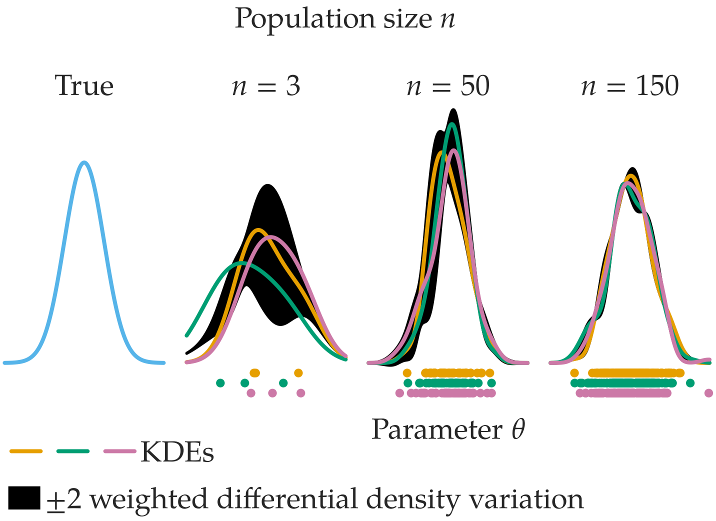

Figure 3.5. Effect of the population size on the density variation.
True density (density estimate on the current population) and kernel density estimates (KDEs) on populations of varying size n. The populations are sampled from the true distribution, the samples are depicted as points below the densities. The weighted differential density variation is obtained by calculating the pointwise variation at each parameter θ and weighting it by the true density at this parameter. The density variation is the integral over the weighted differential density variation. It decreases with increasing population size. (Figure similar to Klinger, Hasenauer (2017))

inefficient again, essentially equivalent to rejection sampling and maybe even worse since the proposal distribution is now centered at the first mode, having potentially less mass at the second mode than the prior distribution. Similar scenarios can be thought of in the model selection case. If one of the models has only a small posterior mass, it is likely to become extinct in one of the intermediate populations. Hence, it is crucial to guarantee a consistent approximation quality across all the generations of an ABC-SMC scheme to obtain an accurate and efficient scheme.

Therefore, an ABC-SMC scheme in which a target accuracy is pre-specified is presented. The population sizes, across all generations, are then automatically adapted to match the specified target accuracy. The target accuracy is expressed in terms of the variation of kernel density estimates on the populations. A lower variation corresponds to a larger accuracy. To predict the appropriate population size for the next population, the effect of increasing or decreasing the population size is estimated based on the current population (Fig. 3.5). If the current variation is too large, the population size is increased for the next populations; if the current variation is too small, the population size is decreased for the next population. To this end, bootstrapped populations of varying sizes are drawn. Their variation is calculated and the population size of the next generation is selected by interpolating to smaller population sizes or extrapolating to larger population sizes with the help of a parametric approximation of the population size dependent variation.

To present the proposed scheme for adaptive population size selection, let E_{CV} denote the desired target density variation. The initial population size of generation $t = 0$ is n_0 and is pre-specified. Given the current population $P_t = \{(w_i, \theta_i)\}_{i=1}^{n_t}$, $t \geq 0$, of size n_t, tentative population sizes $n^*_{t,q}$, evenly spaced between $\lfloor n_t/3 \rfloor$ and $2\,n_t$ are probed at step size $\lfloor n_t/10 \rfloor$. Hence,

$$n^*_{t,q} = \lfloor n_t/3 \rfloor + (q-1)\lfloor n_t/10 \rfloor,\ q \in \{1, \dots, Q\},\ Q = \max\{q | n^*_{t,q} \leq 2 n_t\}. \tag{3.1}$$

The variation is estimated for each tentative population size $n^*_{t,q}$. To do so, $B = 5$ bootstrapped populations $P_{t,q,b}$, $b = 1, \dots, B$, each of size $n^*_{t,q}$, are drawn from the current density estimate $K = \mathrm{KDE}(P_t)$. Usually, $B \approx 5$ bootstrapped populations are employed. For each of these bootstrapped populations, a density estimate $K_{t,q,b} = \mathrm{KDE}(P_{t,q,b})$ is calculated. The estimated density variation $E_{CV}(n^*_{t,q})$ is calculated for each tentative population size $n^*_{t,q}$ as

$$E_{CV}(n^*_{t,q}) = \sum_{(w,\theta) \in P_t} w \,\mathrm{CV}\left(\{K_{t,q,b}(\theta)\}_{b=1}^{B}\right), \tag{3.2}$$

with the coefficient of variation CV defined by

$$\mathrm{CV}\left(\{k_b\}_{b=1}^{B}\right) = \frac{\mathrm{std}\left(\{k_b\}_{b=1}^{B}\right)}{\mathrm{mean}\left(\{k_b\}_{b=1}^{B}\right)}$$

and mean and (biased) standard deviation given by

$$\mathrm{mean}\left(\{k_b\}_{b=1}^{B}\right) = \frac{1}{B}\sum_{b=1}^{B} k_b$$

and

$$\mathrm{std}\left(\{k_b\}_{b=1}^{B}\right) = \left(\frac{1}{B}\sum_{b=1}^{B}\left(k_b - \mathrm{mean}\left(\{k_b\}_{b=1}^{B}\right)\right)^2\right)^{1/2}.$$

A parametrized functional approximation f is used to facilitate interpolation and extrapolation of E_{CV}. The functional form of f is motivated by the scaling of the kernel density mean squared error as function of the population size n (Bowman and Azzalini 1997). Silverman (1986) showed that the kernel density mean squared error decreases with $\alpha\, n^{-b}$, depending on the employed kernel density estimator and distribution properties. Here, E_{CV} is approximated by this functional form. Non-linear least squares and the Levenberg-Marquardt algorithm (Levenberg 1944; Marquardt 1963) are employed to fit the open parameters α and β of $f(n; \alpha, \beta)$ to the points $\{(n^*_{t,q}, E_{CV}(n^*_{t,q}))\}_{q=1}^{Q}$ (Fig. 3.6, "bootstrap"). The population size $n_{t+1} = \mathrm{round}(f^{-1}(E_{CV}; \alpha, \beta))$ of the subsequent popu-

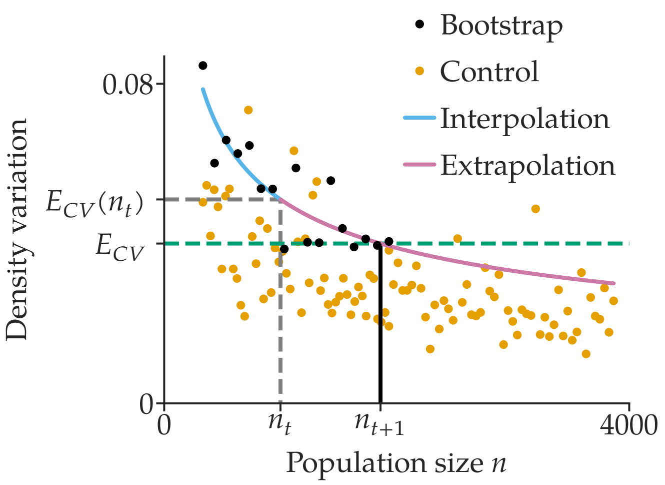

Figure 3.6. Estimation of the subsequent population size.
The density variation E_{CV} is parametrized as $E_{CV}(n; \alpha, \beta)$. It is fitted to the variations obtained from the bootstrapped populations (black points, "Bootstrap"). The bootstrapped populations are drawn from the kernel density estimate on the current population t. The tentative population sizes $n^*_{t,q}$ of these points are calculated according to Eq. (3.1), the corresponding variations according to Eq. (3.2). The population size n_{t+1} of the subsequent population $t+1$ is obtained from the functional approximation (interpolation and extrapolation) as to match the specified target variation E_{CV} (green dashed line). To evaluate the quality of the approximation, control estimates ("Control") were directly obtained from the underlying density. The bandwidth was selected according to the global bandwidth selection strategy (Sect. 3.4.2), following Silverman's rule. (Figure similar to Klinger, Hasenauer (2017))

lation is determined by inverting the functional approximation f (Fig. 3.6). For model selection, this scheme is applied to the full joint parameter space.

The algorithm is outlined in Alg. 6. There, a list of tentative population sizes N^* is created. Then, the variation associated with each population size is estimated with `estimate_cv` (Alg. 7). The functional approximation f is obtained from fitting with the Levenberg-Marquardt algorithm, implemented in `fit`. The next population size is calculated by inverting f. In Alg. 7, `sample`(K, n^*) denotes drawing n^* samples from the density K.

Notably, the population size is selected before the sampling of a population starts. Alternatively, one might think of checking if the population size is sufficient after acceptance of each particle in a continuous fashion during sampling. However, this has at least two drawbacks. First, kernel density estimation can be costly for some estimators, thus increasing the computational cost. Second, and more importantly, this might introduce a bias towards distributions which yield lower variations for the same population size. This might be the case for distributions of "simpler" shapes such as unimodal compared to multimodal.

Algorithm 6: `adapt_population_size`

```
Input: P, KDE
Output: $n$
$N^* \leftarrow [n^*_1, \dots, n^*_Q]$
$C^* \leftarrow$ [estimate_cv($n^*$, KDE, $P$)$|n^* \in N^*$]
$f \leftarrow$ fit($N^*, C^*$)
$n \leftarrow \text{round}(f^{-1}(E_{CV}))$
```

Algorithm 7: `estimate_cv`

```
Input: $n^*$, KDE, $P$
Output: cv
$K \leftarrow \text{KDE}(P)$
$K^* \leftarrow [\text{KDE}($sample($K, n^*$)$)|b \in \{1, \dots, B\}]$
$\text{cv} \leftarrow \sum_{(w,\theta)\in P} w \; \text{CV}([K'(\theta)|K' \in K^*])$
```

3.4.4. Evaluation of the Adaptation Scheme

To evaluate the proposed adaptation scheme, it is applied to analytical and practically relevant problems (Table 3.1). First, it is assessed if the proposed parameterized functional approximation appropriately predicts the variation at different population sizes. Then, the method is applied to an example with multiple posterior modes and its stability is investigated (Sect. 3.4.5). This model is a symmetrized multivariate Gaussian model such that the posterior features several modes. Next, it is applied to model selection for Markov Jump Processes (Sect. 3.4.6). The Markov Jump Process describes a chemical reaction and is simulated with the Gillespie algorithm (Gillespie 1977). Finally, a computationally demanding multi-scale tumor growth model (Jagiella et al. 2017) is investigated (Sect. 3.4.7). This model is a hybrid, discrete-continuum model, employing an agent-based model for the individual cells and a PDE-based model for the extracellular matrix. Markov chains and decision rules model the intracellular mechanisms of cell-division. The model is in the initial (temporal) phase highly stochastic, since the cell-numbers are then still low. Averaging effects occur in the later (temporal) phases due to higher cell numbers and the model is then less stochastic.

First, the quality of the functional approximation $E_{CV}(n) = f(n; \alpha, \beta)$ is assessed. This approximation does not have to be perfect. It is only required that the approximation is good enough such that the population size evolves towards the population size yielding the desired target variation. Fluctuations are tolerable. A simple unimodal model shows, that the chosen functional approximation is appropriate (Fig. 3.6, "Extrapola-

Table 3.1. Models evaluated for population size adaptation.
All three models are stochastic but employ different simulation algorithms. Model selection is also evaluated. The models have between two and seven parameters which are inferred.

Name	Stochastic	Model Selection	Parameters	Simulation
Symmetrized Gaussian	Yes	No	2	Mersenne Twister
Markov Jump Process	Yes	Yes, 2 models	1 per model and mixture probability	Gillespie
Multi-scale Model	Yes	No	7	Hybrid, discrete-continuum, agent-based, PDE, Markov Chain, decision rules

tion"). Although it seems to overestimate E_{CV} slightly on average in the extrapolated regime (Fig. 3.6), it captures the overall relation between E_{CV} and the population size n reasonably well.

3.4.5. Stability for a Symmetrized Gaussian Model

The symmetrized Gaussian features 1, 2 or 4 posterior modes, similar to the example treated by Koutroumpas et al. (2016). With this example, the stability of the adaptive population size selection scheme over the course of the generations, the influence of the kernel density estimator, and the effect of the posterior modes are investigated. The density variation is estimated with $B = 10$ bootstrapped populations. The simulated data $s \in \mathbb{R}^2$ are obtained from the model $s \sim \mathcal{N}(\mathrm{sq}(\theta, n_{\mathrm{modes}}), \sigma^2 I)$, denoting by I the identity matrix in $\mathbb{R}^2$ and $\sigma^2 > 0$. The number of posterior modes is determined by n_{modes} through the function $\mathrm{sq} : \{1, 2, 4\} \times \mathbb{R}^2 \to \mathbb{R}^2$ which performs element-wise squaring on a subset of the dimensions. Let $\theta = (\theta_1, \theta_2)$. Define

$$\mathrm{sq}(\theta, n_{\mathrm{modes}}) = \begin{cases} (\theta_1, \theta_2) & \text{if } n_{\mathrm{modes}} = 1 \\ (\theta_1^2, \theta_2) & \text{if } n_{\mathrm{modes}} = 2 \\ (\theta_1^2, \theta_2^2) & \text{if } n_{\mathrm{modes}} = 4 \end{cases} .$$

The function sq ensures that the number of posterior modes equals n_{modes}. For $n_{\mathrm{modes}} = 2$, $\mathrm{sq}((\theta_1, \theta_2), 2) = \mathrm{sq}((-\theta_1, \theta_2), 2)$. This symmetry generates a second mode. Similarly,

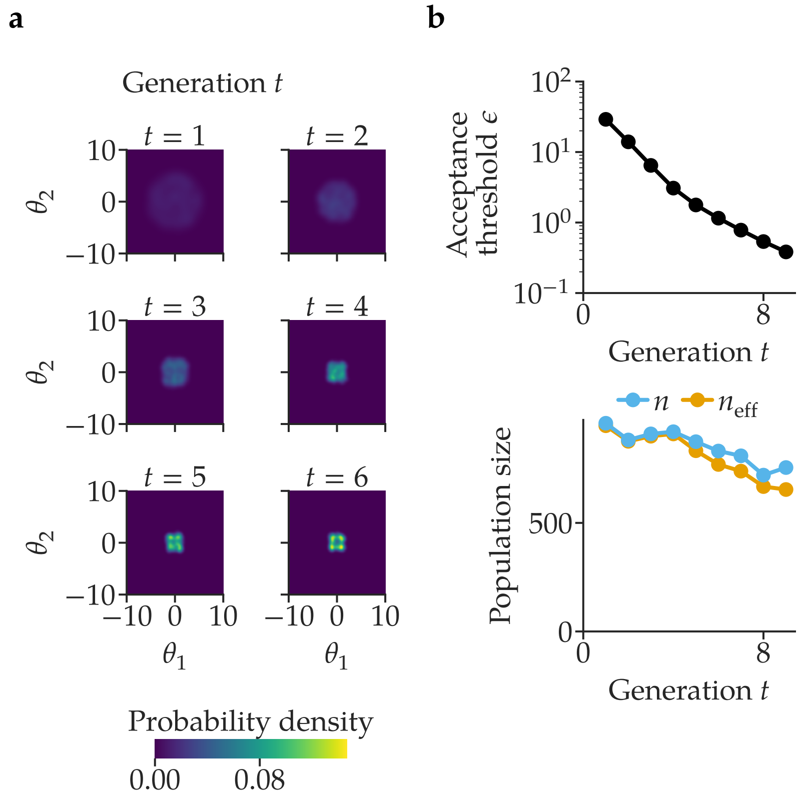

Figure 3.7. ABC-SMC inference for a model with a multimodel posterior.
(a) First six generations of an ABC-SMC run with $E_{CV} = 0.1$, initial population size $n_0 = 500$, observed data (set to) $s_{\text{obs}} = (1, 1)$. The model parameters are in this example $\sigma^2 = 0.5$ and $n_{\text{modes}} = 4$. Global bandwidth selection is performed (Sect. 3.4.2). The probability density is color coded. The four posterior modes are captured after four generations. (b) Acceptance threshold ϵ, population size n and effective population size n_{eff} for the ABC-SMC run from Fig. 3.7a. The population size remains approximately constant. (Figure similar to Klinger, Hasenauer (2017))

for $n_{\text{modes}} = 4$, $\text{sq}((\theta_1, \theta_2), 4) = \text{sq}((-\theta_1, \theta_2), 4) = \text{sq}((\theta_1, -\theta_2), 4) = \text{sq}((-\theta_1, -\theta_2), 4)$ generating four posterior modes. The parameter $\theta \in [-10, 10]^2$ is subject to posterior inference. The prior over θ is uniform $\theta \sim \mathcal{U}([-10, 10]^2)$. Distance is measured in the ℓ_1 norm: $d(s, s_{\text{obs}}) = \|s - s_{\text{obs}}\|_1$.

ABC-SMC runs with $n_{\text{modes}} \in \{1, 2, 4\}$, $s_{\text{obs}} = (1, 2)$, $\sigma^2 = 2$ and $E_{CV} = 0.1$ reveal that in all scenarios the the posterior modes are correctly captured after a few generations (Fig. 3.7a, for $n_{\text{modes}} = 4$). The acceptance threshold ϵ decreases substantially over the course of the generations (Fig. 3.7b, top). Surprisingly, the population size and effective population size decay slightly (Fig. 3.7b, bottom).

To evaluate the effect of the initial population size n_0 on the population sizes of the subsequent populations, ABC-SMC runs with varying initial population sizes $n_0 \in$

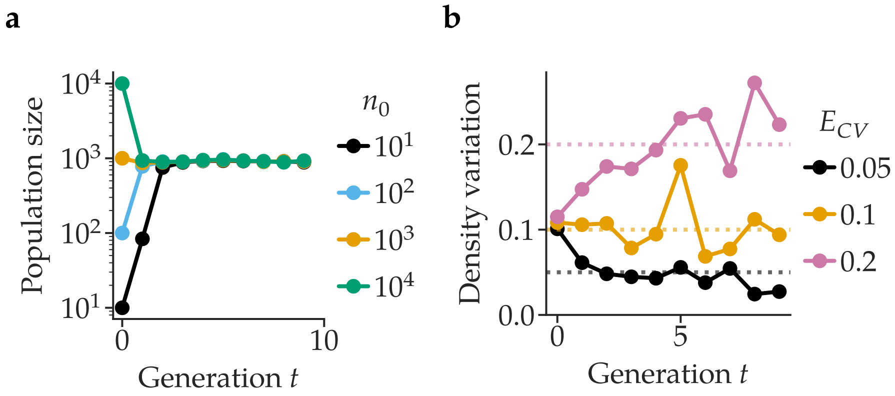

Figure 3.8. Effect of the initial population size and matching the target variation. (a) For each initial population size $n_0 \in \{10^1, 10^2, 10^3, 10^4\}$ ten independent ABC-SMC runs are performed and their population sizes are averaged. The model parameters are $n_{\text{modes}} = 4$, $\sigma^2 = 2$, $s_{\text{obs}} = (1, 1)$ and $E_{CV} = 0.05$. The population sizes converge after a few generation and are not further influence by the initial population size. (b) Target variation (dashed) and actual variation (solid) for varying target variation E_{CV}. The parameters are $\sigma^2 = 0.5$, $s_{\text{obs}} = (1, 1)$, $n_{\text{modes}} = 4$ and initial population size $n_0 = 500$ On average, the specified target variation is matched well. (Figure similar to Klinger, Hasenauer (2017))

$\{10^1, 10^2, 10^3, 10^4\}$ are performed. For each initial population size 10 ABC-SMC runs are performed and the obtained population sizes are averaged. The population size converges after a few generations (Fig. 3.8a). The attained value is independent of the initially chosen population size n_0 (Fig. 3.8a).

Varying the target variation $E_{CV} \in \{0.05, 0.1, 0.2\}$, it is examined if the specified target variation is actually attained. Here, ABC-SMC runs with $\sigma^2 = 0.5$, $s_{\text{obs}} = (1, 1)$, $n_{\text{modes}} = 4$ and initial population size $n_0 = 500$ are performed. On average, the achieved variation matches the target variation well (Fig. 3.8b). The lower the target variation is set, the less fluctuation in the achieved variation is observed (Fig. 3.8b).

To examine the effect of the target variation E_{CV} on the population size, ABC-SMC runs with $s_{\text{obs}} = (1, 1)$, $\sigma^2 = 0.5$, $n_{\text{modes}} = 4$ for varying E_{CV} are performed. The population sizes remain approximately constant for each target E_{CV} (Fig. 3.9, left). The median population size over each complete ABC-SMC run is then calculated. The median population size decrease with increasing target variation (Fig. 3.9, right).

To investigate whether the number of posterior modes influences the selected population size, ABC-SMC runs with one, two and four posterior modes are performed. Since the employed density estimator is expected to influence the results, global, local and cross-validated bandwidth selection strategies are examined (Sect. 3.4.2). Surprisingly, there is no visible or systematic dependency of the population size on the number of posterior modes (Fig. 3.10). However, the employed density estimator influences the se-

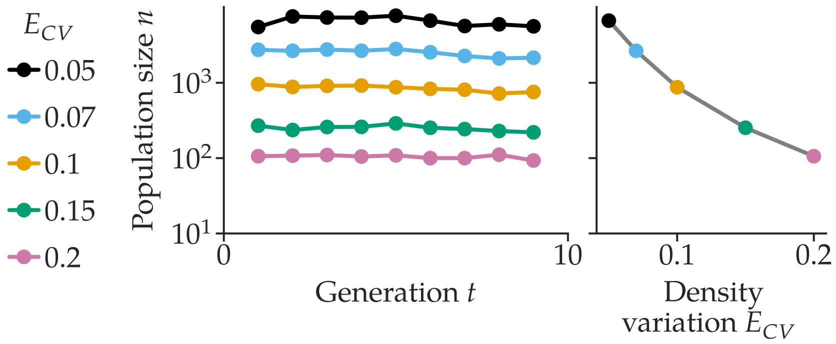

Figure 3.9. Dependence of the population size on the specified target variation.
Left: selected population sizes for varying E_{CV} over the course of the generations. The parameters are $\sigma^2 = 2$, $s_{\text{obs}} = (1,1)$, $\sigma^2 = 2$, $n_{\text{modes}} = 4$. The population sizes remain approximately constant. Right: median population size over target density variation. The median population size decreases with increasing target variation. (Figure similar to Klinger, Hasenauer (2017))

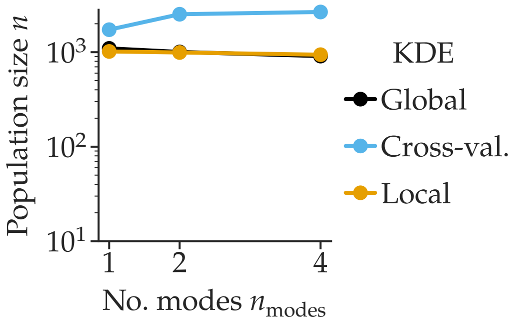

Figure 3.10. Influence of the bandwidth selection strategy on the population sizes.
Population size n depicted as af function of the number of posterior modes and the employed bandwidth selection strategy (color coded). Global, local and cross-validated bandwidth selection strategies are probed (Sect. 3.4.2). The target variation is $E_{CV} = 0.1$, the model parameters are $\sigma^2 = 2$ and $s_{\text{obs}} = (1,1)$. Although the number of posterior modes does not influence the selected population size, the employed density estimator does so. (Figure similar to Klinger, Hasenauer (2017))

lected population sizes. The cross-validated estimator yields the largest population sizes. (Sect. 3.4.9). This is reasonable since it is the estimator featuring the smallest bandwidth.

3.4.6. Model Selection for a Markov Jump Process

Markov jump processes constitute a class of models which is of high practical relevance. For example, chemical reaction equations belong to this class. Conveniently, Markov jump processes can be simulated exactly with the Gillespie algorithm (Gillespie 1977). An important application in this context is model selection between competing models (Souza et al. 2013; Westerhuis et al. 2004). It is therefore next investigated whether the adaptive population size scheme is applicable to model selection for Markov Jump Processes at the example of chemical reaction kinetics models.

The two chemical reaction kinetics models m_1 and m_2 for conversion of chemical species X into species Y,

$$m_1 : X + Y \xrightarrow{k_1} 2\,Y$$

and

$$m_2 : X \xrightarrow{k_2} Y$$

with reaction rates k_1 and k_2, are studied. This model has been considered before (Eigen 1996; Prusiner 1982) and was used to analyze ABC-SMC schemes (Toni, Welch, et al. 2009). The Gillespie algorithm (Gillespie 1977) is used to simulate these models from $t = 0$ to $t = T = 0.1$. The initial concentrations are always $X(t) = 40$ and $Y(0) = 3$. Representative realizations are depicted in Fig. 3.11. For ABC-SMC the distance between a trajectory $s = (X, Y)$ and $s' = (X', Y')$ is defined as cumulative sum over the absolute differences of the concentrations at individual times evenly spread over the interval $[0, T]$

$$d(s, s') = \sum_{n=1}^{n} |X(t_n) - X'(t_n)|, \quad t_n = \frac{n}{N}T, \quad N = 20.$$

The observed data is generated from model m_1 with reaction rate $k_1 = 2.1$ ($\log_{10} k_1 \approx 0.32$). The prior over the model is uniform $p_0(m = m_1) = p_0(m = m_2) = 1/2$. The priors over the reaction rates are uniform on a logarithmic scale, $\log_{10} k_1 \sim \mathcal{U}(-2, 2)$ and $\log_{10} k_2 \sim \mathcal{U}(-2, 2)$ for both models. The proposal densities for the log-reaction rates are obtained by kernel density estimation. The probability $K_t(m)$ of proposing model m in generation t is

$$K_t(m) = \begin{cases} p^{\text{stay}} p_t(m) + (1 - p^{\text{stay}}) p_t(\overline{m}) & \text{if } p_t(m) p_t(\overline{m}) > 0 \\ p_t(m) & \text{otherwise} \end{cases}$$

in which $\overline{m}$ denotes the other model $\overline{m_1} = m_2$ and $\overline{m_2} = m_1$ and $p^{\text{stay}} = 0.7$.

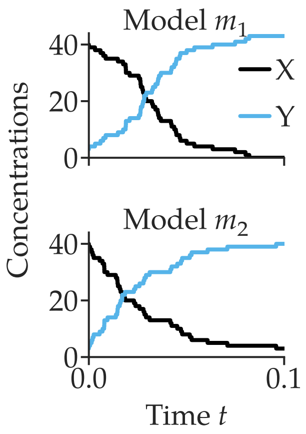

Figure 3.11. Example simulations from two competing chemical reaction kinetics models. Species concentrations X and Y over simulation time t. The reaction rates are $k_1 = 2.1$ for model m_1 and $k_2 = 30$ for model m_2. (Figure similar to Klinger, Hasenauer (2017))

To examine the adaptive population size scheme for model selection, an ABC-SMC run with $E_{CV} = 0.05$ is performed, taking as observed data a trajectory which is artificially generated from model m_1 with $k_1 = 2.1$. In the first generations, the posterior probability at model m_1 is still comparable to the posterior probability at model m_2 (Fig. 3.12, top left). However, as the generations progress, model m_1 gains more probability and model m_2 extinguishes slowly (Fig. 3.12, top left). At the end of the run, the true model is hence selected. Also the log-reaction rates become narrower as the generations progress (Fig. 3.12, right). The true rates is included in the posterior distribution of the last generation. Note, however, that since the observed data was generated stochastically, the true rate does not necessarily coincide with the maximum a posteriori estimate, not even for an exact posterior. Interestingly, the population size and effective population size decay during the later generations systematically (Fig. 3.12, bottom left).

To examine the quality of the posterior distribution approximation and its dependency on the target density variation E_{CV}, the adaptive population size posteriors are compared to a reference posterior. In case of the present model, posterior inference is not analytically tractable. Therefore, the reference posterior is generated from ABC-SMC, runs with large and constant population sizes. Four ABC-SMC runs are repeated, each run with 60000 particles, and the results are averaged to generate the reference posterior.

The reference posterior assigns non-zero mass to both models m_1 and m_2 (Fig. 3.13a). However, the mass assigned to model m_2 is very low, (Fig. 3.13a, $p(m = m_2 | s_{\text{obs}}) \approx 0.0038$). No mass is assigned to model m_2 for adaptive population size ABC-SMC runs with $E_{CV} = 0.2, 0.1$. (Fig. 3.13a). But for the smaller $E_{CV} = 0.05$ model m_2 has posterior

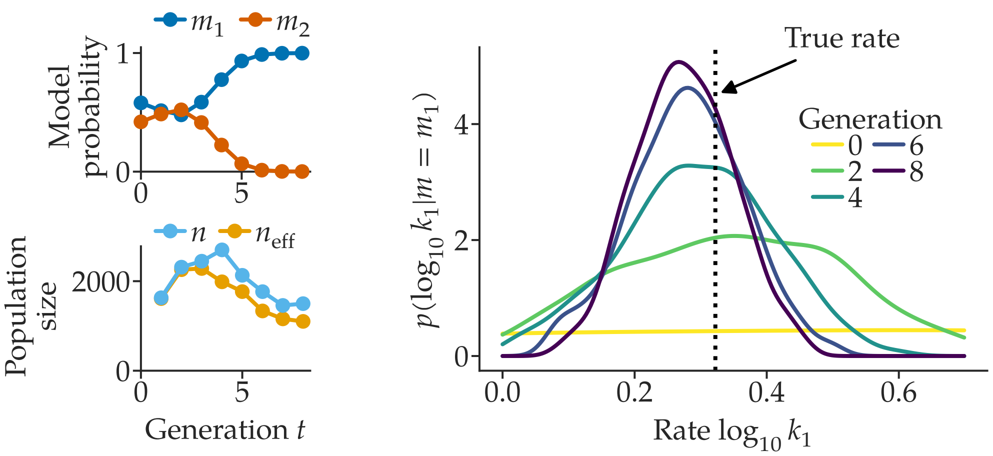

Figure 3.12. Single ABC-SMC run for two competing chemical reaction kinetics models. The observe data is generated from model m_1 with reaction rate $k_1 = 2.1$ ($\log_{10} k_1 \approx 0.32$) and $E_{CV} = 0.05$. Top left: posterior distributions over the models during the generations of the ABC-SMC run. Bottom left: population size n and effective population size n_{eff} depicted over the course of the generations t. Right: posterior distribution over the log-reaction rate of of model m_1 for different generations together with the true reaction rated (dashed line) used to simulate the observed data. The posterior gains more mass close to the true rate as the populations progress. (Figure similar to Klinger, Hasenauer (2017))

mass again. The error in the log-rate posteriors is quantified in terms of the Kolmogorov-Smirnov (KS) distance between the adaptive population size runs and the reference posterior (Fig. 3.13b). First, the distances of the individual large, constant population size runs to the reference posterior are examined. Since the reference posterior is their average, this comparison reveals whether the constant population size is chosen large enough. Indeed, the KS distance is very small compared to the KS distances of the adaptive population size runs (Fig. 3.13b), confirming the validity of the reference posterior. As expected, the KS distance increases with increasing E_{CV}. However, also the selected population size decreases with increasing E_{CV} (Fig. 3.13c), resulting in a lower computational burden.

3.4.7. A Computationally Challenging Problem: Multi-scale Model of Tumor Growth

The computational complexity of the models examined in the previous two sections is comparatively low. Typical simulation run times are less than a second and the models posses only up to two unknown parameters. To investigate whether the proposed method is also suitable for higher-dimensional parameter estimation problems and for models with longer simulation times, a multi-scale model of tumor growth on a two-dimensional

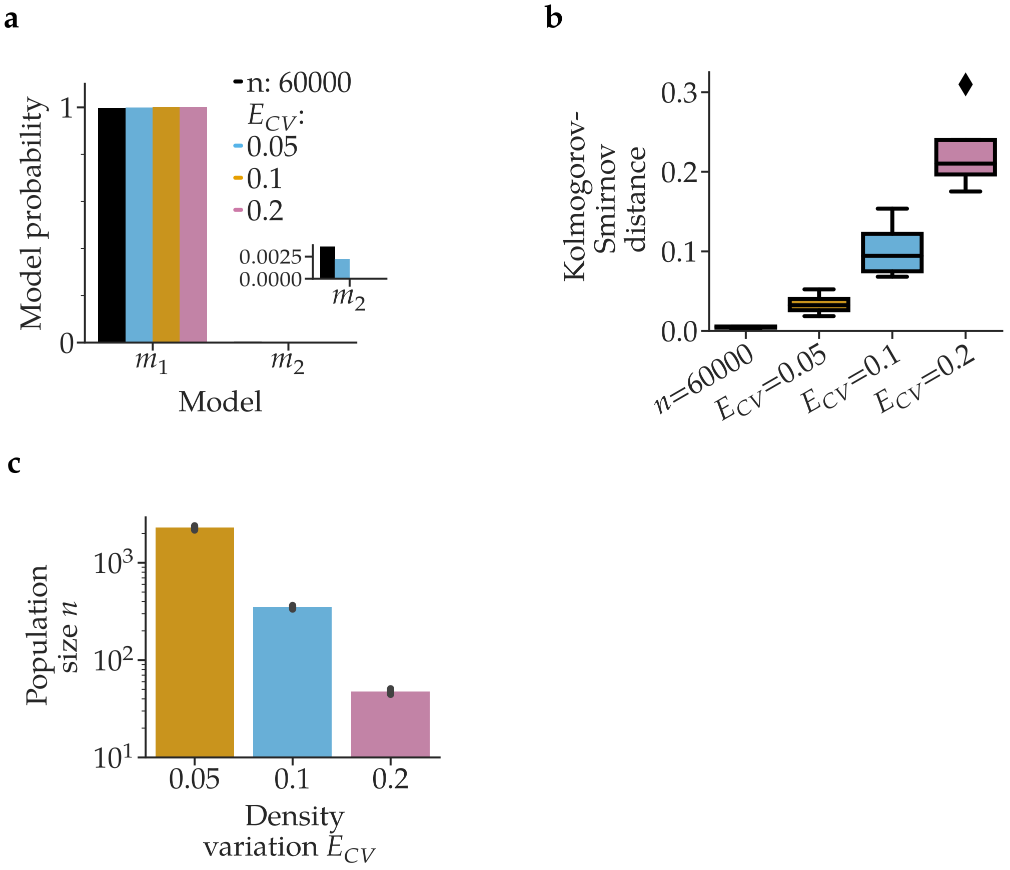

Figure 3.13. Model posterior error, parameter error and population size dependency on the target variation.

(a) Model posterior distribution for large, constant population size and adaptive population size selection. The inset shows the mass at model m_2 only. For every run, sampling of new generations is stopped as soon as the acceptance threshold falls below 1.5. The accuracy decreases with increasing E_{CV}. (b) Kolmogorov-Smirnoff (KS) distances relative to the reference posterior. The reference posterior is obtained from the large, constant population size ABC-SMC runs. The KS distance of the large, constant population size run is small compared to the adaptive population size runs. (c) Population size and target density variation for model selection. The population size n is averaged over all generations and decreases with increasing target variation E_{CV}. (Figure similar to Klinger, Hasenauer (2017))

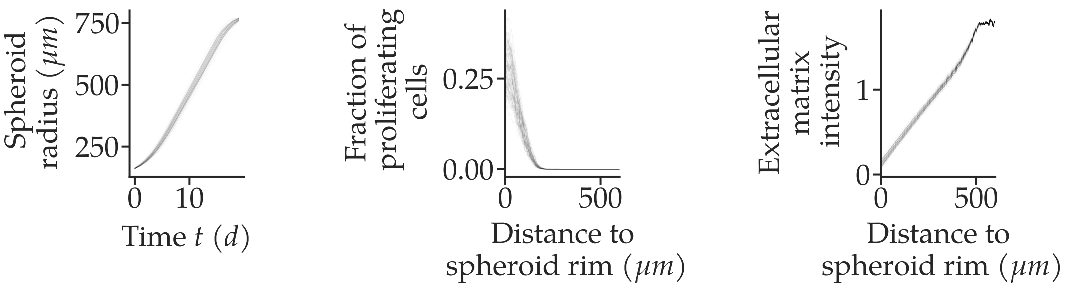

Figure 3.14. Multi-scale model simulations of tumor growth data.
100 independent samples obtained from the reference parameters as defined by Jagiella et al. (2017). The data are: the spheroid radius over time, the fraction of proliferating cells over distance to the spheroid rim, and the extracellular matrix intensity over distance to the spheroid rim. (Figure similar to Klinger, Hasenauer (2017))

plane, as described by Jagiella et al. (2017), is considered. This model possesses seven unknown parameters and the average simulation time is approximately 22.3 s. While ABC-SMC inference for the previous two models is still feasible in a multi-core setting on a single machine, the multi-scale model considered here requires distributed computation to achieve reasonable run times of the ABC-SMC scheme.

The model's output is described in terms of three summary statistics: spheroid radius, fraction of proliferating cells and extracellular matrix intensity (Fig. 3.14). The ground truth (observed) data s_{obs} is generated by averaging 100 independent samples drawn from the reference parameters (Fig. 3.15, cross, "True") described by Jagiella et al. (2017). Inference is performed in the $\log_{10}$ domain. To do so, a prior on each parameter which is uniform in the log domain is imposed. The support of these priors is the same as specified by Jagiella et al. (2017). Also, the employed distance function is the same as the one used by Jagiella et al. (2017). To perform ABC-SMC runs the local bandwidth selection strategy (Sect. 3.4.2) is used. That is, only the 20% nearest neighbors of each particle are considered for calculation of the covariance matrix of the kernel centered at that particle.

The populations contract slowly close to the reference parameters (Fig. 3.15). The posterior is, except for the parameter $k_{\text{e}}^{\text{pro}}$, still widely spread out at generation $t = 13$ and only starts to cluster slowly (Fig. 3.15). At the later generation $t = 40$, all the parameters, except for e^{div}, feature very narrow posteriors (Fig. 3.15). To examine if the ABC-SMC scheme is converged, the distances relative to the ground truth data of samples drawn from the MAP estimate at generation $t = 40$ are considered. These distances are compared to the distances of the same 100 samples which generate the ground truth data relative to the ground truth data. Indeed, the distributions of distances are comparable (Fig. 3.16a).

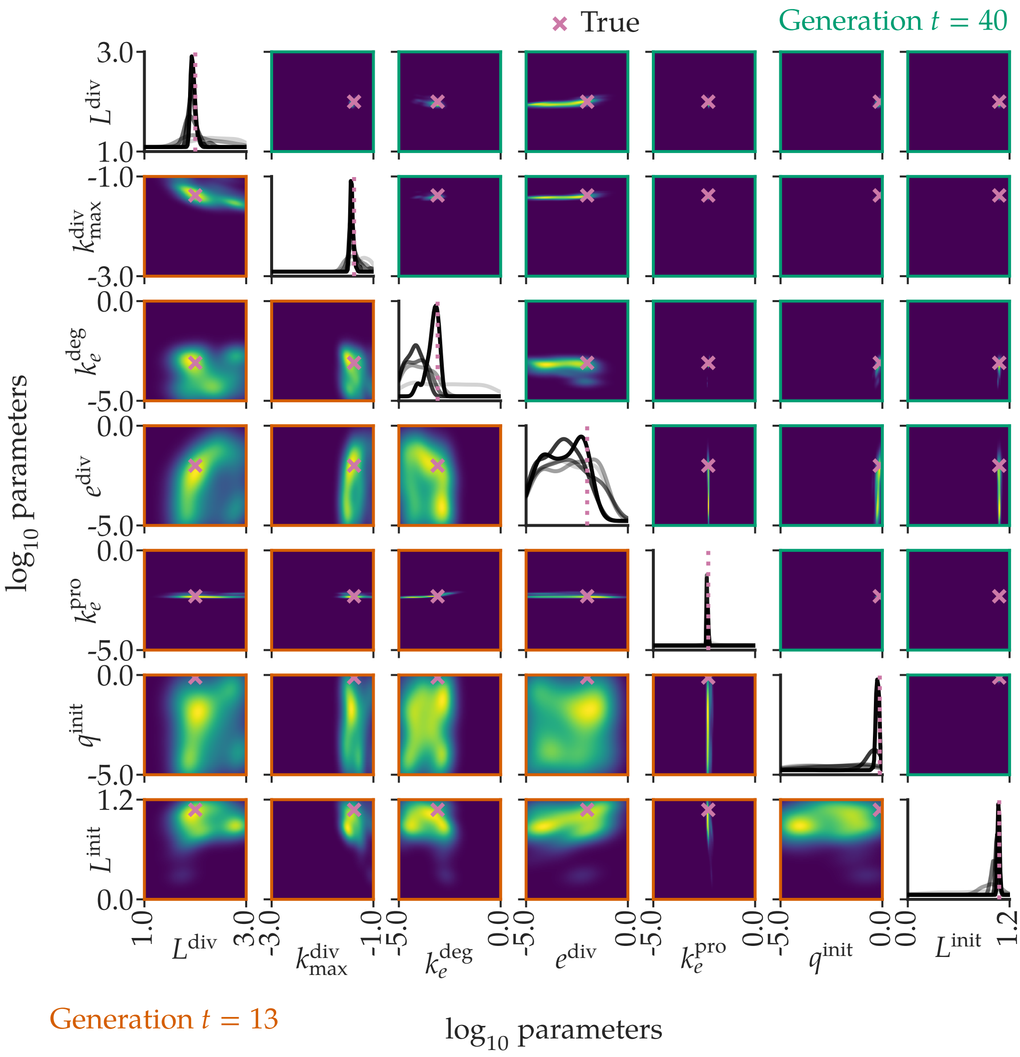

Figure 3.15. Evolution of the generations of the multi-scale tumor growth model. The posterior distribution is obtained from adaptive population size ABC-SMC inference with $E_{CV} = 0.25$. The axis limits correspond to the support of the prior distribution. The prior is uniform in the log domain. The crosses and dashed lines (True) indicate the reference parameter used for generation of the ground truth (observed) data s_{obs}. Lower triangle: distribution of generation $t = 13$. Upper triangle: distribution of generation $t = 40$. Diagonal: distributions over the course of the generations from earlier generations (lighter color) to later generations (darker color). (Figure similar to Klinger, Hasenauer (2017))

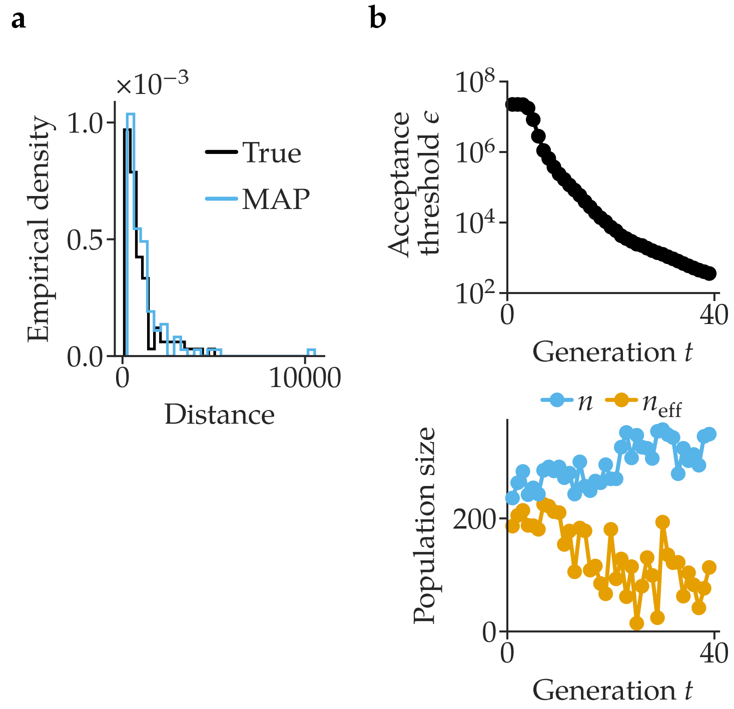

Figure 3.16. Evolution of the populations and approximation quality of the multi-scale tumor growth ABC-SMC run.
(a) Distance distribution of maximum a posteriori samples and ground truth samples. True: distances of the same 100 samples which are used to generate the ground truth data via averaging, relative to the ground truth data. MAP: distances of 100 maximum a posteriori samples, relative to the ground truth data. Both distance distributions are comparable. (b) Evolution of the population sizes and the acceptance threshold for the multi-scale model. Acceptance threshold, population size and effective population size over the generations t. The effective population size decreases although the population size increases. (Figure similar to Klinger, Hasenauer (2017))

The ground truth samples are only slightly closer to the (average) ground truth data than the MAP samples (Fig. 3.16a). This result indicates that the proposed population size adaptation scheme is successfully combined with the local density estimator.

Next, the evolution of the population sizes over the course of the generations is examined. The acceptance threshold decreases as expected from the acceptance threshold adaptation scheme (Fig. 3.16b). Interestingly, the effective population size decays despite the increasing population size.

3.4.8. Complexity and Computational Cost of the Adaptation Scheme

Since ABC-SMC is already a computationally challenging undertaking, the proposed adaptation scheme should preferentially not add substantially to the already high computational cost. The complexity of the proposed scheme, with respect to the population size n_t and the parameter dimension d_{par}, is therefore analyzed in this section and its overhead is benchmarked for the practically relevant multi-scale example (Sect. 3.4.7).

The function `adapt_population_size` (Alg. 6) is called once per generation (Alg. 4). The number of tentative populations Q is constant (Alg. 6). Therefore, the generation of N^*, and `fit`(N^*, C^*) in Alg. 6 are both $\mathcal{O}(1)$; the final round operation in Alg. 6 is also $\mathcal{O}(1)$. The function `estimate_cv` is called Q times. In summary, since the complexity of no operation in `adapt_population_size` — with exception of `estimate_cv` — depends on n_t or d_{par}, the overall complexity per iteration is Q times the complexity of `estimate_cv`.

In `estimate_cv` (Alg. 7), a KDE fit is performed once on the entire population (which can be cached), and then B times on populations of sizes n^*, with $\lfloor n_t/3 \rfloor \leq n^* \leq 2n_t$. Sampling of a population of size n^* from a fitted KDE is also performed B times. The probability density functions K' are evaluated Bn_t times to calculate the variation cv (last line of Alg. 7). The overall complexity of the adaptation scheme per iteration emerges therefore from thee factors: (FIT) fitting the KDE $Q(B+1)$ times on populations of size $\mathcal{O}(n_t)$ each, (RVS) drawing samples from a fitted KDE $\mathcal{O}(QBn_t)$ times, and (PDF) evaluating the probability density function of a fitted KDE QBn_t times. However, Q and B are constants with respect to n_t and d_{par} and can thus be ignored in the following.

For rather simplistic implementations of the global and local bandwidth selection, the complexity is bounded as follows: (FIT) For global bandwidth selection (Sect. 3.4.2), fitting a KDE can be performed by calculating a single covariance matrix, which is of $\mathcal{O}(n_t d_{\text{par}}^2)$, as well as calculating the inverse covariance matrix and its determinant, which can be performed in $\mathcal{O}(d_{\text{par}}^3)$ or better. For local bandwidth selection (Sect. 3.4.2), covariance calculation, matrix inversion and determinant calculation are performed n_t times, the complexity is therefore $\mathcal{O}(n_t^2 d_{\text{par}}^2 + n_t d_{\text{par}}^3)$. (RVS) Drawing n_t samples from a local or global density can be performed by one vector–matrix multiplication of $\mathcal{O}(d_{\text{par}}^2)$ per sample and is therefore of $\mathcal{O}(n_t d_{\text{par}}^2)$ in both cases. (PDF) The probability density function is evaluated $\mathcal{O}(n_t)$ times, each evaluation is of $\mathcal{O}(n_t d_{\text{par}}^2)$. The complexity is hence $\mathcal{O}(n_t^2 d_{\text{par}}^2)$ in both, the local and global case.

In summary, the adaptation scheme is for global bandwidth selection bounded by $\mathcal{O}(n_t^2 d_{\text{par}}^2 + d_{\text{par}}^3)$ and for local bandwidth selection by $\mathcal{O}(n_t^2 d_{\text{par}}^2 + n_t d_{\text{par}}^3)$. The complexity can differ for other KDE methods or different implementations. For example, matrix multiplication and inversion can be performed in $\mathcal{O}(d_{\text{par}}^{2.376})$ with the Coppersmith-Winograd algorithm (Coppersmith and Winograd 1987). Independent of the implementation details, in the standard ABC-SMC scheme, (FIT) KDE fitting is already performed once in Alg. 4, (RVS) sampling from a fitted density is already performed $\mathcal{O}(n_t)$ times in Alg. 5, and (PDF) evaluating the probability density function is already performed n_t

times in Alg. 5. The overall complexity of the entire ABC-SMC scheme is therefore never increased by the proposed adaptation scheme.

Practically, the highest computational cost emerges often from model simulations, although only $O(n_t)$ such simulations are performed per generation. But often, simulation runtimes are very long, outweighing all other steps in an ABC-SMC scheme. However, the adaptation schemes does not perform additional model simulation. In the multi-scale example considered in Sect. 3.4.7, the adaptation took only about 0.00208% of the total model simulation time. The computational overhead caused by the adaptation scheme is therefore practically negligible.

3.4.9. Discussion of the Population Size Adaptation Scheme

The parameterization of complex computational models is a task many model developers face in systems biology and neuroscience. In principle, ABC-SMC schemes are well suited for this task, since they require neither analytical expressions of the likelihood functions nor do they make assumptions on the model structure. In practice, however, the need to manually tune population sizes limits the applicability of ABC-SMC schemes. So far, tuning population sizes has been a task which required substantial experience (or luck). The method presented here aims to simplify population size selection. It complements existing methods for the selection of perturbation kernels (Filippi et al. 2013), acceptance thresholds (Silk, Filippi, and Stumpf 2013), and summary statistics (Marin et al. 2014; Nunes and Balding 2010). As illustrated, the method is applicable to model selection and parameter inference. It is compatible with a range of density estimation methods and is scalable to computationally demanding, higher-dimensional parameter inference problems, such as posed by multi-scale models.

In the proposed method, the desired target approximation quality is expressed in terms of the target density variation E_{CV}, which has to be chosen adequately. However, the presented method does not merely replace manual tuning of population sizes by manual tuning of yet another parameter (the density variation E_{CV}). The parameter E_{CV} is better accessible because it is easier to interpret. Moreover, the examples with decreasing, increasing or approximately constant population size show that the proposed method is not a simple re-parameterization. Empirically, $0.1 \leq E_{CV} \leq 0.2$ works in many examined examples. It might therefore be possible to make a default choice, such as $E_{CV} = 0.15$, and alleviate the model developer completely from selecting population size related parameters. Such a default choice is not directly possible for the population size.

The proposed method can in principle be applied with arbitrary density estimators. The selection of the population size, however, depends on the choice of the estimator.

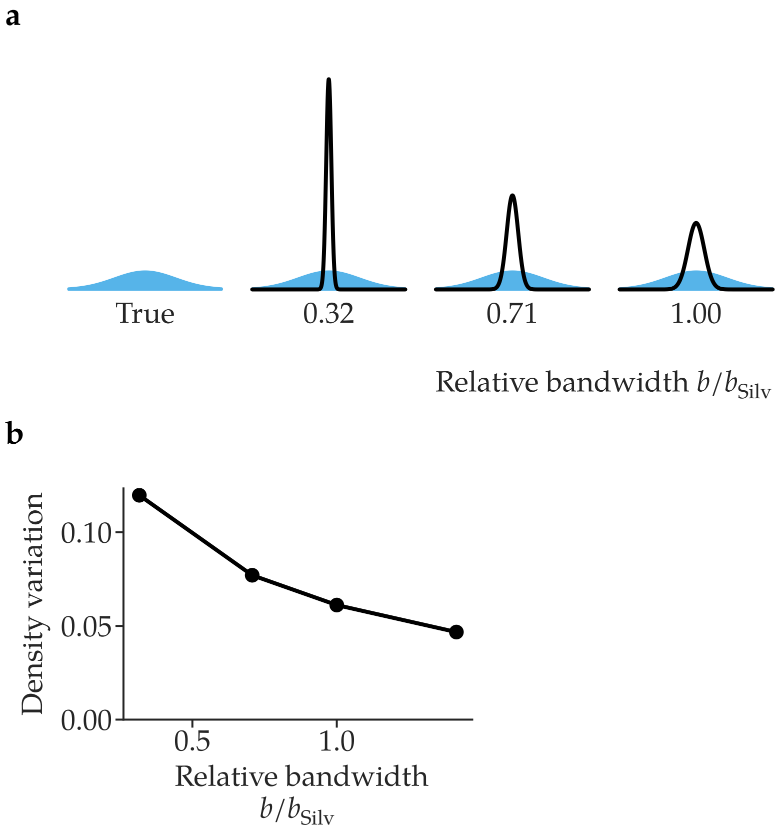

Figure 3.17. Influence of the bandwidth on the population sizes.
(a) Illustration of bandwidth variation. True, unimodal normal distribution and bandwidths b relative to the Silverman bandwidth b_{silv}. The population size is $n = 10^3$ in this illustration. (b) Density variation for kernels with bandwidth b and population size $n = 10^3$. The density variation is estimated from drawing 250 bootstrapped populations for each bandwidth. (Figure similar to Klinger, Hasenauer (2017))

Over-smoothing estimators require lower population sizes than estimators with narrower bandwidths. This is consistent with the lower density variation associated with larger bandwidths (Figs. 3.17a and 3.17b).

Alterations of the proposed scheme are of course possible. For example, it is possible to bootstrap directly from the particle population instead of from the density estimate. This comes, however, with the disadvantage of likely drawing the same particles repeatedly since the bootstrapping has to be performed with replacement. This possibility was indeed investigated, but no obvious differences were found and it was therefore decided to rather bootstrap from the density. Also, the functional relationship assumed between the population size and the density variation, which is motivated by the Silverman rule (Silverman 1986) might be further improved. However, for the examples considered here, this relationship predicts the required population sizes sufficiently well.

The difficulty of choosing population sizes has been noted in the literature before (Moral, Doucet, and Jasra 2012). To the best of my knowledge this work here is the first attempt towards automated and adaptive population size selection. The results suggest that probing population sizes over an order of magnitude (Jagiella et al. 2017) can be avoided. In the future, the interplay between density estimators, population sizes and acceptance rates could be examined further. The effect of density estimators on acceptance rates has been already investigated (Filippi et al. 2013; Koutroumpas et al. 2016) but has not yet been related to population sizes and the overall number of samples required for an ABC-SMC inference run. Additional information, such as the effective population size, could also be incorporated. For instance, it might be aimed to keep the effective population size constant or in a certain range. It might be interesting to compare ABC-SMC approaches to these methods as well. In summary, the compatibility with virtually any existing ABC-SMC scheme, paired with the generality of ABC-SMC schemes lead to expect that the proposed method will be applied to a wide range of model selection and parameter inference tasks.

4. Approximate Bayesian Connectomic Model Selection

Section is similar to Klinger, Marr, Theis, Helmstaedter (under review)

Since explorative approaches to local cortical connectomic model selection are insufficient (Sect. 3.1) the more systematic approach of approximate Bayesian model selection is pursued here (Chap. 3). First, summary statistics for ABC-SMC based on the description of graphs via permutation invariant functions mapping these graphs to real numbers are introduced (Sect. 4.1). The (experimental) conditions under which the proposed approach works reliably are examined (Sect. 4.2), thereby giving guidelines for a concrete connectomic reconstruction experiment.

4.1. Circuit Motifs as Summary Statistics

Neuronal networks were in the past characterized by the frequencies with which certain network motifs occur (Kretschmar 2013; Milo et al. 2002; Perin, T. K. Berger, and Markram 2011; Song et al. 2005). Similar approaches were also widely applied in the social network sciences (Newman 2010) and for macroscopic connectomics, where graphs consist of few nodes, each node representing a whole brain area (Rubinov and Sporns 2010; Shi et al. 2013). The question hence arises whether such network motifs are suitable summary statistics for ABC-SMC (Chap. 3).

Different types of motifs as summary statistics for ABC-SMC are probed. Denoting by n the number of neurons, by n_e the number of excitatory neurons and by n_i the number of inhibitory neurons, the excitatory-excitatory connectivity (p_ee) is estimated from the adjacency matrix W according to

$$p_\mathrm{ee} = \frac{|\{(k,l) | 1 \leq k,l \leq n_\mathrm{e}, w_{kl} \neq 0\}|}{(n_\mathrm{e}-1)^2}$$

and similarly the excitatory-inhibitory connectivity

$$p_\mathrm{ei} = \frac{|\{(k,l) | 1 \leq l \leq n_\mathrm{e}, n_\mathrm{e} < k \leq n, w_{kl} \neq 0\}|}{n_\mathrm{e}\, n_\mathrm{i}}.$$

The reciprocities are defined by

$$r_\mathrm{ee} = \frac{|\{(k,l) | 1 \leq k,l \leq n_\mathrm{e}, w_{kl}\, w_{lk} \neq 0\}|}{(n_\mathrm{e}-1)^2}$$

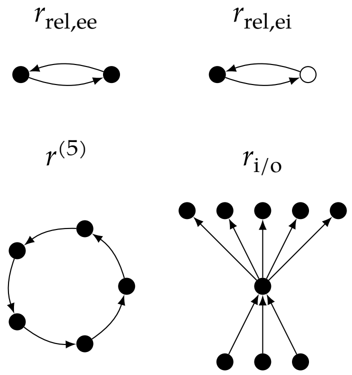

Figure 4.1. Network summary statistics used for connectome description.
The summary statistics are relative excitatory-excitatory reciprocity $r_{\text{rel,ee}}$, relative excitatory-inhibitory reciprocity $r_{\text{rel,ei}}$, relative number of cycles of length 5 $r^{(5)}$ and in-out-degree correlation $r_{\text{i/o}}$. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

and

$$r_{\text{ei}} = \frac{|\{(k,l)|1 \leq k,l \leq n_{\text{e}}, w_{kl}\, w_{lk} \neq 0\}|}{n_{\text{e}}\, n_{\text{i}}}.$$

Based on the previous analysis and adaptation of the models to the circuit constraints, the investigated connectome samples are expected to differ in their relative excitatory-excitatory reciprocity $r_{\text{rel,ee}}$ and in their relative excitatory-inhibitory reciprocity $r_{\text{rel,ei}}$ (Fig. 4.1). These are defined by

$$r_{\text{rel,ee}} = \frac{r_{\text{ee}}}{p_{\text{ee}}}$$

and

$$r_{\text{rel,ei}} = \frac{r_{\text{ei}}}{p_{\text{ei}}}.$$

The recurrency of a network is quantified in terms of its relative number of cycles $r^{(\ell)}$ of length ℓ. This statistic describes how much of the information processed within the network flows back to the network itself. It is defined by

$$r^{(\ell)} = \frac{\operatorname{tr} W_{\text{E}}^{\ell}}{(n_{\text{e}} p_{\text{ee}})^{\ell}}.$$

Here, $\ell = 5$ is used as trade-off between descriptiveness and computational efficiency. Moreover, the correlation $r_{\text{i/o}}$ between in-degree and out-degree of the network is quantified (Fig. 4.1). This statistic indicates whether an organization in input neurons and output neurons is present (negative correlation), or, whether the network is organized into more central and less central neurons (positive correlation). The in-out-degree correlation ($r_{\text{i/o}}$) is the Pearson correlation coefficient of the in- and out-degrees of the neurons

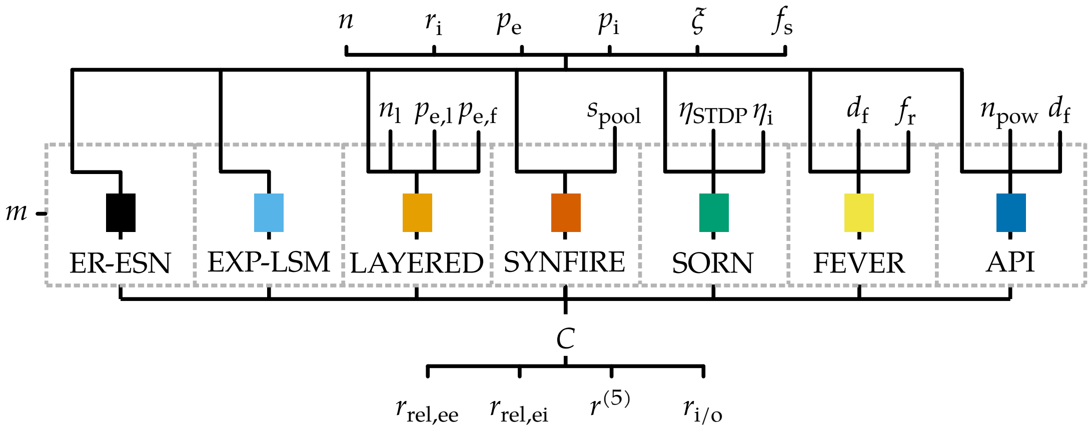

Figure 4.2. Complete graphical model.
Every model has the parameters number of neurons (n), fraction of inhibitory neurons (r_i), excitatory connectivity (p_e), p_i, noise (ξ) and reconstruction fraction (f_s). The latter two are only important in the case of noisy, incompletely reconstructed connectomes. The LAYERED model has additionally the parameters number of layers (n_l), excitatory lateral connectivity ($p_\mathrm{e,l}$) and excitatory forward connectivity ($p_\mathrm{e,f}$); the SYNFIRE model pool size (s_pool); the FEVER model feature space dimension (d_f) and feverization (f_r); the API model selectivity (n_pow) and feature space dimension (d_f). Every model generates a connectome C. The connectome is then described by the four summary statistics relative excitatory-excitatory reciprocity ($r_\mathrm{rel,ee}$), relative excitatory-inhibitory reciprocity ($r_\mathrm{rel,ei}$), relative number of cycles of length 5 ($r^{(5)}$) and in-out-degree correlation ($r_\mathrm{i/o}$). (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

of the excitatory subpopulation (Fig. 4.1). Other, stochastic, network statistics (App. A.2) were also briefly probed, but not further evaluated due to their computational cost.

To assess the performance and discriminative power of the proposed summary statistics, these are evaluated on samples drawn from the full (noise free) models (Fig. 4.2). Normal distributions are fitted to the summary statistics obtained from 50 samples from the models (Figs. 4.3a to 4.3d). As expected, some of the models are distinct in terms of their relative excitatory-excitatory reciprocity ($r_\mathrm{rel,ee}$) (Fig. 4.3a). However, this summary statistic alone is not sufficient to discriminate all the models (Fig. 4.3a). Similarly, the distributions of the remaining summary statistics are not completely overlapping indicating that the selected summary statistics do possess discriminative power. But again, none of the summary statistics alone can discriminate between all the model classes (Figs. 4.3b to 4.3d). An alternative view on the data is provided by a t-SNE embedding (Maaten and Hinton 2008). The t-SNE embedding shows that the samples from the different model classes lie in distinct regions of the embedding manifold (Fig. 4.4). The discriminative power of the selected circuit motifs motivates to further pursue an Approximate Bayesian model selection approach mediated by these motifs as summary statistics.

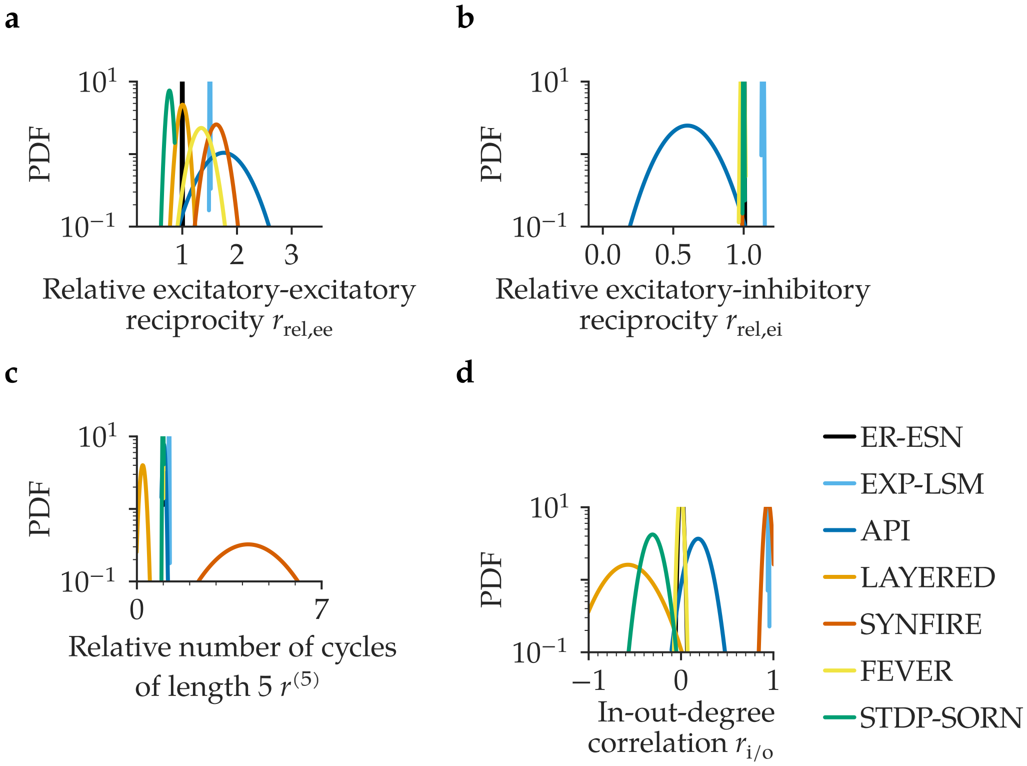

Figure 4.3. Connectomic network summary statistics are discriminative, but no summary statistic alone suffices.
Probability density functions of Gaussian fits of 50 samples drawn form each of the network models: (a): relative excitatory-excitatory reciprocity ($r_{\text{rel,ee}}$). (b): relative excitatory-inhibitory reciprocity ($r_{\text{rel,ei}}$). (c): relative number of cycles of length 5 ($r^{(5)}$). (d): in-out-degree correlation ($r_{\text{i/o}}$). (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

4.2. Evaluation of the Approximate Bayesian Connectomic Model Selection Scheme

Approximate Bayesian connectomic model selection is evaluated for several scenarios using the ABC-SMC scheme presented in Chap. 3, with constant population size 2000, implemented in the pyABC framework (Sect. 3.3). The presented circuit motifs (Sect. 4.1) are used as summary statistics. Frist, perfect, noise-free experimental conditions are considered (Sect. 4.2.1). Then, the more realistic setting of noisy, i.e. erroneous connectomes (Sect. 4.2.1) and the case of fractional (incomplete) connectomic reconstruction (Sect. 4.2.3) are examined. It is systematically explored how much and with which accuracy it is necessary to reconstruct connectomes for successful model selection (Sect. 4.2.4). Last, the model selection approach is applied to connectome samples which

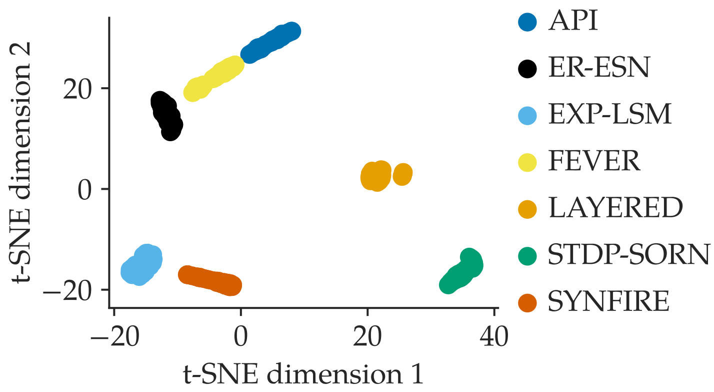

Figure 4.4. t-SNE embedding of connectome samples mediated via summary statistics separates the model classes.
t-SNE embedding performed with original data being the four network summary statistics relative excitatory-excitatory reciprocity $r_{\mathrm{rel,ee}}$, relative excitatory-inhibitory reciprocity $r_{\mathrm{rel,ei}}$, relative number of cycles of length 5 $r^{(5)}$ and in-out-degree correlation $r_{\mathrm{i/o}}$.

are not generated by the candidate models, but by modifications of these (Sect. 4.2.5).

4.2.1. Noise-free Connectomic Reconstruction

The method is first evaluated for the scenario of perfect, noise-free, and complete connectomic reconstruction. A single connectome sample is drawn from each of the seven models and model selection is performed on this sample. A uniform prior over the models is imposed. The posterior has in all cases mass one at the original model (Fig. 4.5). The connectome samples are thus perfectly discriminated. Although the model selection works well so far, the assumption of perfect experimental conditions made is rather unrealistic.

4.2.2. Noisy Connectomic Reconstruction

Under real conditions, noise from several sources is expected. One possible source of noise are reconstruction errors. Today, connectomic reconstruction is still relying on human annotation. These annotations however contain errors (Boergens et al. 2017; Helmstaedter, Briggman, and Denk 2011). For example, annotators, when tracing dense nervous tissue might erroneously "jump" from one wire to another one. Moreover, the quality with which a connectome is reconstructed depends also on the accuracy of synapse classifiers, which are also not perfect (Becker et al. 2012; Kreshuk, Funke, et al. 2015; Kreshuk, Koethe, et al. 2014; Kreshuk, Straehle, et al. 2011). A more subtle source of noise might be biological implementation errors. Assuming that one of these models is implemented by biology, it

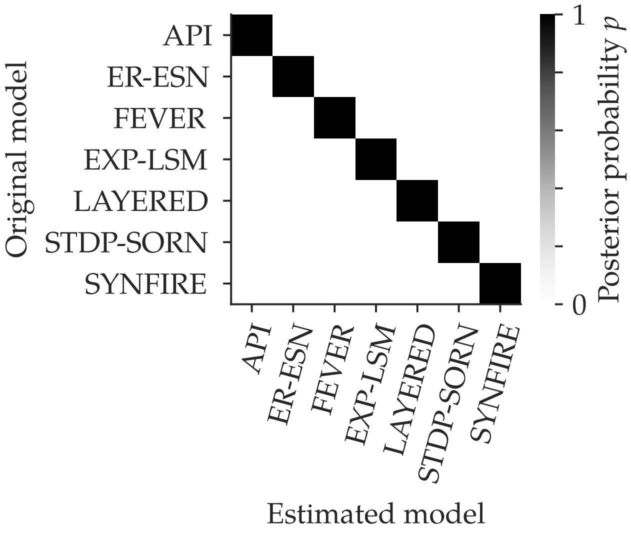

Figure 4.5. Noise-free connectomic model selection.
A connectome is drawn from each model and then subjected to the proposed ABC-SMC model selection method (see Chap. 3). Here, perfect, noise-free and complete reconstruction is assumed. The samples are perfectly discriminated. Each row is the result of a single ABC-SMC run. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

might still be erroneously implemented or the implementation might become perturbed due to external influences (for example chemicals, mechanical forces). Therefore, noisy (erroneous) connectomic reconstruction is modeled. Let $\xi \in [0, 1]$ denote the noise level to be applied on the binary network $G = (V, E)$. Then, $n_\Delta = \lfloor \xi |E| \rfloor$ edges of E are randomly removed from E, regardless of their excitatory or inhibitory type. After that, n_Δ new edges are randomly re-inserted into E, irrespective of their excitatory or inhibitory type. This procedure does not preserve the number of excitatory or inhibitory connections or the degree distribution. Degree-preserving connectome perturbations were also implemented (App. A.1) but not explored in detail because they are less severe and rely on more assumptions.

First, the case of $\xi = 0.15$ reconstruction noise is inspected (Figs. 4.6a and 4.6b). The posterior is substantially less sharply peaked at the original model (Fig. 4.6a). The posterior of the LAYERED connectome sample has now 68% mass at the incorrect FEVER model and only 32% mass at the correct LAYERED model. The maximum a posteriori (MAP) estimate is wrong in this case. Similarly, the sample drawn from the SORN model has 51% posterior mass at the incorrect ER-ESN model and only 49% mass at the correct SORN model. Although the estimate is better than the one of the LAYERED model, the MAP estimate is still wrong.

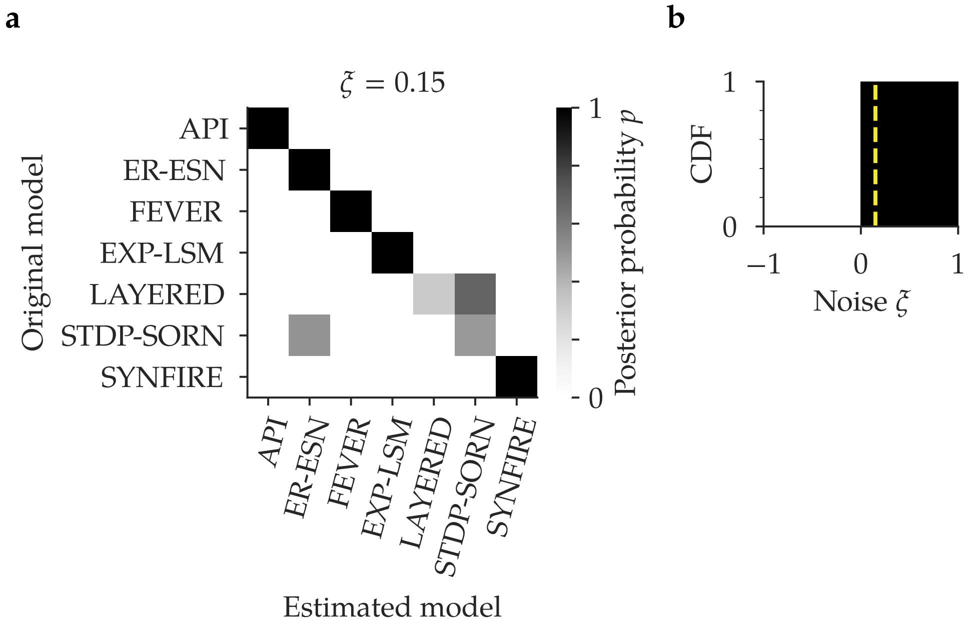

Figure 4.6. Noisy connectomic model selection.
(a) A connectome is drawn from each model, perturbed by random edge rewiring and then subjected to the proposed model selection method. Each row is the result of a single ABC-SMC run. Discrimination is not perfect anymore. For example the LAYERED sample has posterior mass at both the LAYERED model and the SORN model. (b) Cumulative distribution function (CDF) over the noise (ξ), which is not modeled in the prior. The dashed line indicates the applied noise, $\xi = 0.15$. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

The proposed model selection method is only applicable under realistic settings if it can be applied under noisy conditions. Noisy reconstruction was so far not modeled in the prior distribution (Fig. 4.6b). Instead, $\xi = 0$ (Fig. 4.6b) has been assumed. However, a noise model can be explicitly included in the full generative model considered here (Fig. 4.2). The corresponding error rates can be obtained from the evaluation of automated reconstruction methods (Dorkenwald et al. 2017; Kreshuk, Straehle, et al. 2011) and analysis of human tracing accuracy (Boergens et al. 2017; Helmstaedter, Briggman, Turaga, et al. 2013; Takemura et al. 2013; Wanner et al. 2016). The prior $\xi \sim \mathrm{Beta}(2, 10)$ is imposed on the reconstruction noise (Fig. 4.7a). The proposed model selection method is again evaluated on seven connectome samples, one sample drawn from each model and perturbed by noise $\xi = 0.15$. The explicit noise model is beneficial and model discrimination is perfect again (Fig. 4.7b).

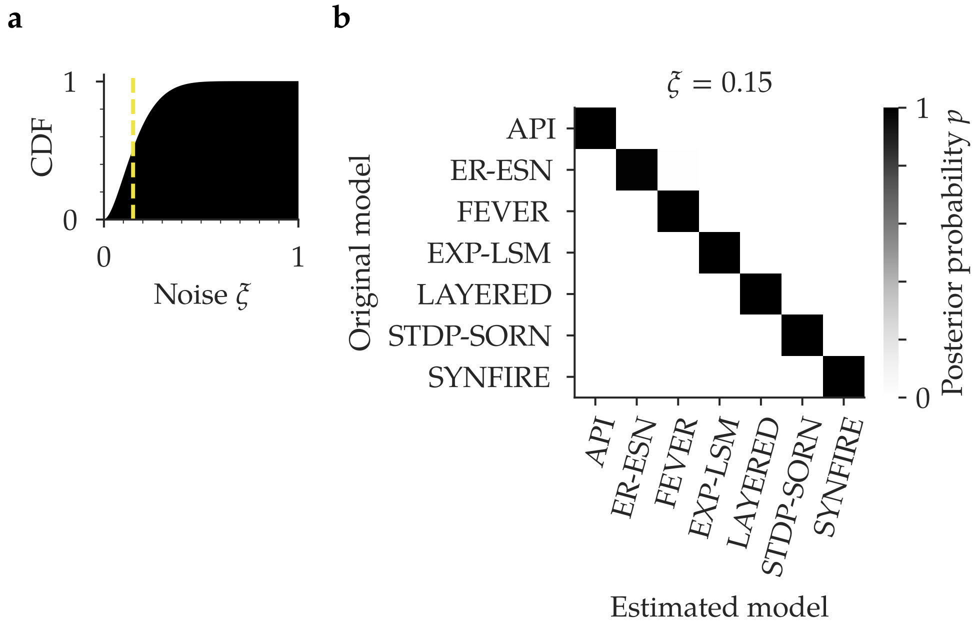

Figure 4.7. Noisy connectomic model selection with noise model.
(b) Cumulative distribution function (CDF) over the noise (ξ) prior, which is modeled as Beta(2, 10) distribution. The dashed line indicates the applied noise, $\xi = 0.15$. (a) A connectome is drawn from each model, perturbed by random edge rewiring and then subjected to the proposed model selection method. Here, an explicit noise model is included in the full generative model. Each row is the result of a single ABC-SMC run. Discrimination is perfect again. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

4.2.3. Fractional Connectomic Reconstruction

Beside the challenges associated with reconstruction noise, connectomic measurements are severely limited by the many human annotation hours currently necessary for reconstruction. Assuming a barrel volume of $(300\,\mu\text{m})^3$, a reconstruction speed of 1.5 mm/h (Boergens et al. 2017) and 5 km to 10 km path length per cubic millimeter cortical tissue, a full reconstruction required between 90 000 and 180 000 human annotator hours. Clearly, the reconstruction of connectomes for the purpose of model selection would be rendered much more realistic if it was sufficient to reconstruct only a fraction of the connectome (Fig. 4.8a). It is hence investigated if reconstruction of only a fraction f_s of the entire connectome is sufficient for model selection purposes. The proposed method is thus applied to model selection of connectomes which are only reconstructed with reconstruction fraction $f_s = 0.3$ under noise-free conditions. Perfect model discrimination is obtained (Fig. 4.8b).

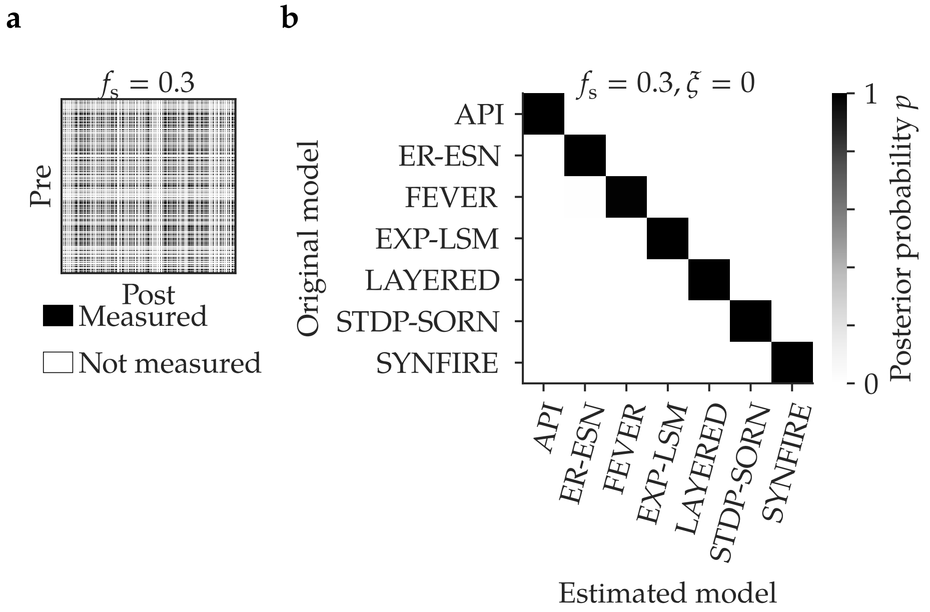

Figure 4.8. Fractional connectomic reconstruction.
(a) Illustration of fractional (incomplete) connectomic reconstruction with reconstruction fraction $f_s = 0.3$ (only 30% of the neurons are reconstructed). (b) A connectome is drawn from each model and incomplete reconstruction is simulated. It is then subjected to the proposed model selection method. Each row is the result of a single ABC-SMC run. Discrimination is perfect. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

4.2.4. How Much and How Accurate to Reconstruct?

Having validated, that the proposed method is applicable under noisy conditions and for fractionally reconstructed connectomes, it is next evaluated which magnitudes of noise can be tolerated and which fraction of the connectome is required to recover the underlying model with a given probability. From these results, it is expected to design a concrete connectomic reconstruction experiment, defining the requirements for synapse classifier accuracy and human annotator hours. The noise prior $\xi \sim \mathrm{Beta}(2, 10)$ is kept fixed and the applied noise (ξ) and reconstruction fraction (f_s) are varied. For each combination of ξ and f_s one connectome sample is drawn from each model and the model discrimination performance is evaluated in two different ways. The average probability assigned to the original (correct) model and the average MAP accuracy are evaluated. The average probability at the original model decays slowly with increasing noise and with decreasing reconstruction fraction (Fig. 4.9a). Surprisingly, the MAP estimate is highly reliable up to $f_s = 0.1$ and $\xi = 0.3$ (Fig. 4.9b). This implies, that only 18 000 instead of 180 000 work hours have to be invested to perform connectomic model selection. The

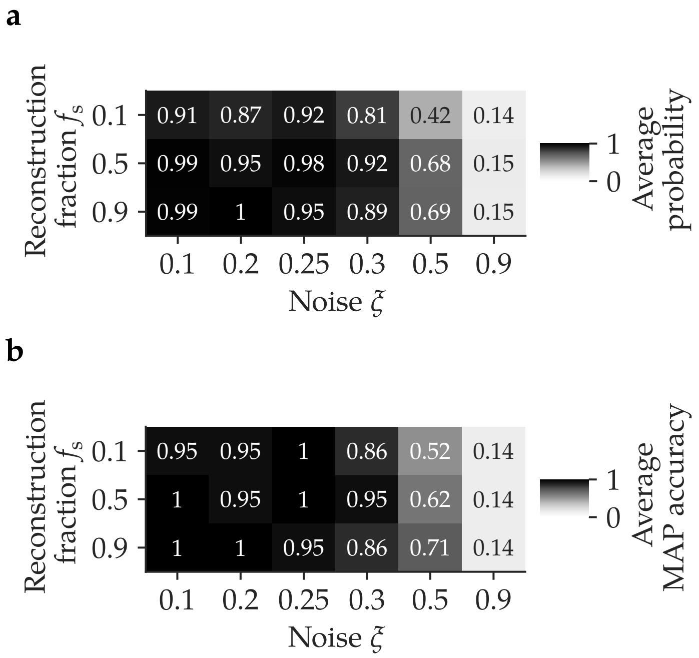

Figure 4.9. Discrimination performance for varying noise and reconstruction fraction. For each combination of noise (ξ) and reconstruction fraction (f_s), 3 connectome sample are drawn from each of the 7 models, such that each entry is assembled from 21 ABC-SMC inference runs. The discrimination performance is quantified in two ways (a) average probability at the original (correct) model, and (b) average MAP accuracy. The latter is calculated as the fraction of correct MAP estimates. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

so far largest undertaken connectomic reconstruction efforts invested between 14 000 and 25 000 work hours (Helmstaedter, Briggman, Turaga, et al. 2013; Takemura et al. 2013; Wanner et al. 2016). A rather challenging reconstruction is thereby rendered much more realistic.

Next, it is examined how far the requirements on noise and reconstruction fraction can be relaxed, assuming precise knowledge on the applied noise. The noise prior is now varied as well and is specified as uniform prior with width 0.08, centered at the applied ξ. Although this might not be realistic in an experimental setting, it hints on how much noise the approach might tolerate in principle. As expected, knowledge of the noise (ξ) increases reconstruction accuracy (Figs. 4.10a and 4.10b) compared to the case without precise knowledge of the noise (Figs. 4.9a and 4.9b). However, the classification performance decays with f_s in a way comparable to the scenario with fixed noise prior. In summary, this indicates, that precise estimations on the reconstruction increase model selection performance.

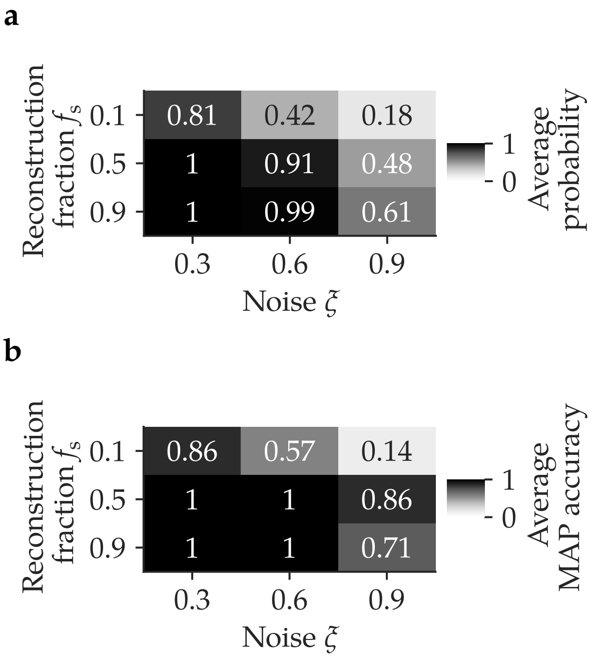

Figure 4.10. Discrimination performance for varying noise, reconstruction fraction and noise prior. For each combination of noise (ξ) and reconstruction fraction (f_s), a connectome sample is drawn from each model. The prior on the noise ξ is set to the applied noise. The discrimination performance is quantified in two ways (a) average probability at the original (correct) model, and (b) average MAP accuracy. Precise knowledge of the present noise allows for less accurate reconstruction.

4.2.5. Incorrect Assumptions on the Model Priors

A weakness – and at the same time a strength – of the Bayesian model selection approach is that the investigated hypotheses have to be precisely defined. It is impossible to obtain posterior mass where no prior mass was defined. The posterior support is always included in the prior support. However, what if none of the investigated models is actually implemented in a barrel? What if none of the investigated models is the "right" model? Can meaningful information still be deduced from the proposed model selection method?

By including a wide range of possible candidate models in the hypothesis space (the prior), the chance to miss structurally distinct models is minimized. However, the data observed in terms of the selected summary statistics might still be generated by another model or by some kind of mixture of the investigated models. To examine the behavior of the proposed method under such circumstances, a family of models which interpolates between the ER-ESN and EXP-LSM model is introduced. The EXP-LSM model turns into the ER-ESN model for very large decay lengths $\lambda_e \to \infty$ and $\lambda_i \to \infty$ (Fig. 4.11a). The

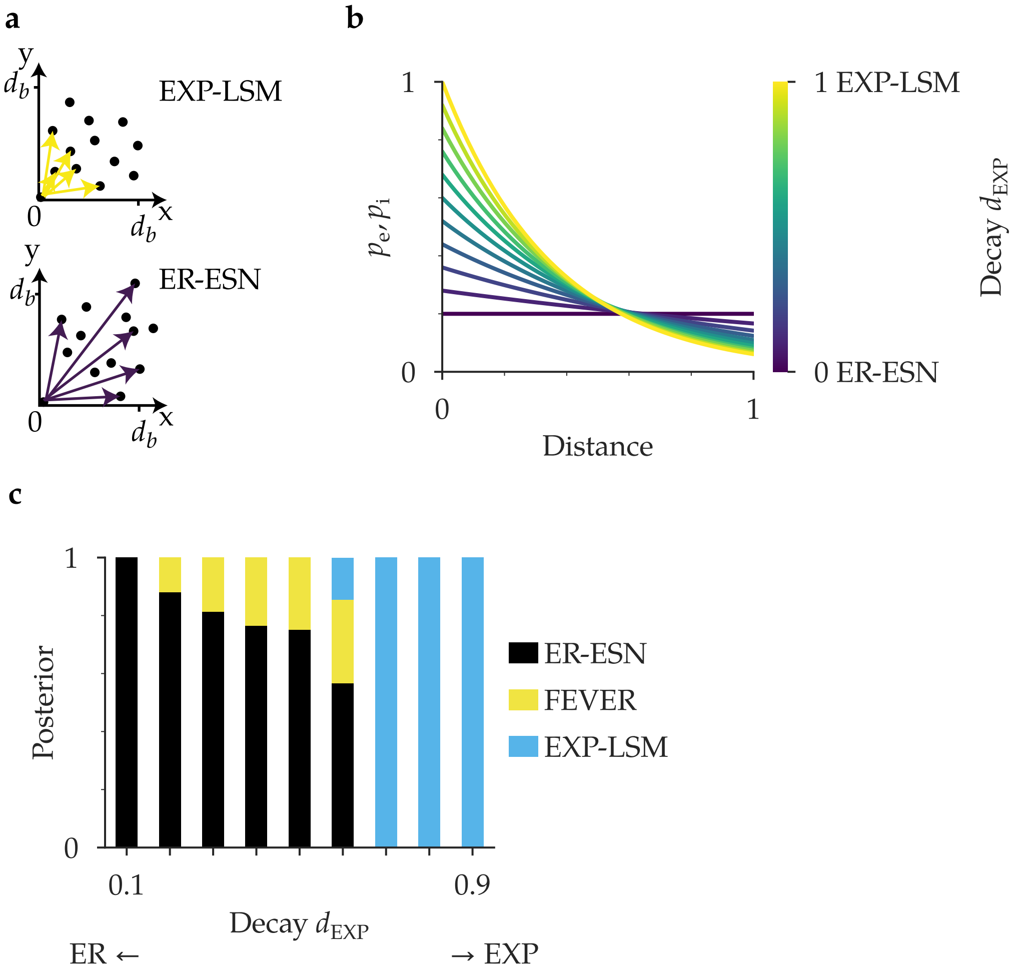

Figure 4.11. Model interpolation.
(a) In the EXP-LSM model, connections to closeby neurons are preferred. In the ER-ESN model, the connection probability does not depend on distance (b) The ER-ESN model is obtained for $d_{\mathrm{EXP}} = 0$, the EXP-LSM model for $d_{\mathrm{EXP}} = 1$. (c) In the ER limit ($d_{\mathrm{EXP}} = 0.1$), the ER model is recovered, in the EXP-LSM limit ($d_{\mathrm{EXP}} = 0.9$) the EXP-LSM model is recovered. In between it is interpolated between these models with a non-dominant contribution of the FEVER model. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

newly introduced parameter d_{EXP} governs the connection probability p_0 for zero distance according to $p_0 = p + (1 - p)d_{\mathrm{EXP}}$, denoting by p the overall connectivity. Hence, for $d_{\mathrm{EXP}} = 1$, the EXP-LSM model is obtained, for $d_{\mathrm{EXP}} = 0$ the ER-ESN model (Fig. 4.11b). As expected for small d_{EXP}, connectome samples from this model are classified as ER-ESN samples (Fig. 4.11c). Similarly, for larger d_{EXP} the samples are classified as EXP-LSM samples (Fig. 4.11c). Surprisingly, in the intermediate regime, the posterior probability is an interpolation between the ER-ESN and the EXP-LSM model, with mass at both models with a non dominant contribution fo the FEVER model (Fig. 4.11c). This indicates, that the method is in fact stable against model mixing and that the posterior encodes model similarity in an interpretable way.

In a second experiment it is investigated to which extent the input statistics influence the network structure shaped by the STDP rule (Sect. 2.9). A family of SORN models, which differ in the number of input patterns n_{p} they receive as input is introduced. The n_{p} distinct input patterns are repeated after n_{p} time steps (Fig. 4.12a). For high enough n_{p}, connectome samples from this model are identified as SORN samples (Fig. 4.12b). However, for $n_{\mathrm{p}} \leq 500$, the LAYERED model gains posterior mass (Fig. 4.12b) and the median correlation increases in this regime (Fig. 4.12c). It has been noted before, that the SORN model produces LAYERED like structures for specific input statistics (Zheng and Triesch 2014). The interpolation performed by the posterior is therefore in accordance with the literature. The method is sufficiently robust to variation in the model formulation and interpolates model classes in a meaningful way.

4.3. Conclusion

In conclusion, it is shown that binary connectomic data (i.e. connectomic data which ignores the strength of connections) contains already sufficient information to discriminate cortical, computational models in local modules of primary somatosensory cortex. The proposed method guides the necessary reconstruction experiments with respect to the required annotation time and reconstruction accuracy. Importantly, it suffices to reconstruct only 10% of a connectome as long as the noise (ζ) stays below 30%. This requires about 18 000 hours reconstruction time, which is realistic already today and in the range of or even less than the reconstruction times of former experiments (Helmstaedter, Briggman, Turaga, et al. 2013; Takemura et al. 2013; Wanner et al. 2016). Connectomic measurements are rendered highly informative through the method proposed here.

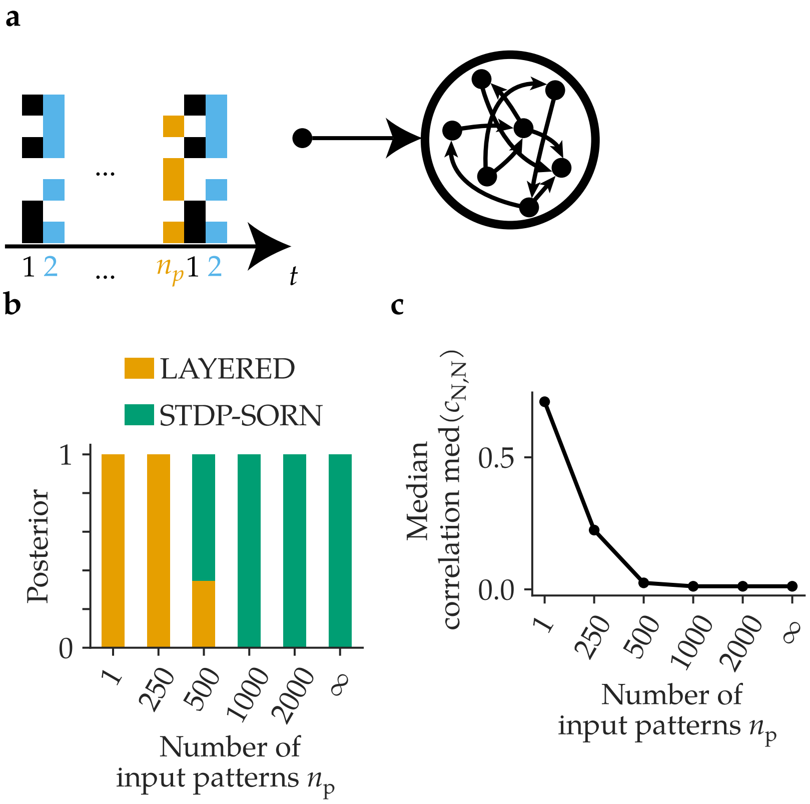

Figure 4.12. Input statistics of the SORN model.
(a) Illustration of the number of input patterns n_p of the SORN model. The input to the network is repeated after n_p patterns. (b) Posterior probability of connectome samples generated with n_p input patterns. (c) Median correlation med$(c_{\mathrm{N,N}})$ of the SORN network activity with n_p input patterns. (Figure similar to Klinger, Marr, Theis, Helmstaedter (under review))

5. Discussion and Conclusion

5.1. Summary

A major challenge in neuroscience today is to close the falsification gap it is experiencing (Denk, Briggman, and Helmstaedter 2012). In this thesis I therefore developed a method that falsifies or supports theories of cortical computation, narrowing the falsification gap. Specifically, it was demonstrated that local connectomes (circuits) serve as arbiters of local computational models in the cerebral cortex and falsify or support these. Seven competing hypotheses of local cortical computational models were selected and implemented as concrete circuit models adapting them to known constraints (Chap. 2). It was demonstrated that these models were functionally meaningful after their adaptation (Chap. 2). It was then examined how to select the models which describe a particular (experimentally determined) local cortical circuit best, given only the structural (network) data. To computationally tackle this problem, a general-purpose framework for distributed Approximate Bayesian Computation - Sequential Monte Carlo inference was developed (Chap. 3). This framework scales to thousands of CPU cores and can be widely applied throughout systems biology and beyond. This framework was extended with a scheme for the automated selection of population sizes (Chap. 3), which is a main practical issue in the application of Approximate Bayesian Computation (Moral, Doucet, and Jasra 2012). The specific Approximate Bayesian Computation - Sequential Monte Carlo scheme proposed for cortical model selection was demonstrated to be robust to noise and applicable to model selection in realistic experimental settings (Chap. 4). Moreover, this thesis work guides the design of concrete reconstruction experiments giving concrete requirements to the percentage of neurons in a circuit to be reconstructed and the tolerable reconstruction errors. For model selection purposes the reconstruction effort can be cut down by a factor of 10 through the proposed method (compared to full, dense reconstruction), making such an undertaking much more realistic. Beyond model selection, a framework for the interpretation of network models is provided, relating them to the seven concrete models examined here.

5.2. Robustness, Generality and Scalability

A strength of the proposed method lies in its probabilistic nature. The method does not only classify connectomes but treats the model selection problem in a fully Bayesian way. In this thesis a flat prior over the model classes was assumed. Evidently, the posterior probabilities would be multiplied by the prior probabilities if the prior was chosen non-uniform. In a concrete experiment preconceptions on the expected computational models could be implemented as non-uniform prior. The probabilistic treatment ensures the usefulness of the proposed method even if a measured connectome is strongly perturbed by noise or is a superposition of different models. At the same time, the method detects reliably the original model for moderate noise levels. Importantly, since any connectome reconstructed in the near future will be a partial one, the demonstrated robustness to fractional reconstruction makes the method applicable to real experiments. Using easily interpretable network summary statistics also facilitates the interpretation of measured connectomes and provides concrete constraints to theoretical modeling efforts once cortical connectomes become available. The proposed method is computationally and conceptually feasible since it does not require analytical access to likelihoods, which are often expensive to compute, if at all possible. The method is therefore expected to be applicable to new network models as well. Moreover, the method works for rather large connectomes of thousands of neurons and even if the precise neuron number is unknown. More efficient implementations of the employed algorithms and scaling up computational power will likely allow to examine connectomes much larger than the ones used in this study. The method was also proven to be robust to details of model implementations. The method is not sensitive to precise parameter choices, or the exact neuronal numbers. It was taken care to verify the correctness of many algorithms using an extensive set of unit tests.

5.3. Related Structural Network Analysis

Structural description and comparison of networks has a long history in the social network sciences (Newman 2010). The focus was often the description of social networks in terms of specific network properties such as, e.g., to which extent networks are clustered or high-degree nodes connect preferentially to other high-degree nodes (Newman 2010). Other studies spanned across different categories of networks. For example, general sparse networks were categorized and grouped in a non-probabilistic fashion (Onnela et al. 2012). The examined networks were quite diverse, ranging from neuronal data to political data (Onnela et al. 2012). Onnela et al. (2012) showed that, in terms of dendrograms, similar network categories were often grouped together by their method.

Not surprisingly, methods from the social network sciences were also applied to neuronal networks (Rubinov and Sporns 2010). Such methods were often used together with small networks in which — in contrast to the present thesis — the nodes did not represent individual cells but whole brain regions consisting of millions of cells each (Rubinov and Sporns 2010). Those networks consisted typically of less than 100 nodes (in contrast to the 2000 node networks considered here) and were obtained from diffusion magnetic resonance imaging or functional magnetic resonance imaging. In addition, the node identity (label) was well-defined in these networks as these corresponded to specific, known brain regions. Here however, the permutation invariance resulting from the absence of node labels resulted in an additional difficulty. In small whole-brain networks, specific structural network properties were related to diseases such as schizophrenia (Heuvel et al. 2013). The gender of patients has been classified based on such networks (Vogelstein and Priebe 2015) and it was quantified to which extent loosing the label information degraded classification performance. It was also attempted to approximately solve the graph matching problem for graph classification (Vogelstein, Conroy, et al. 2015; Vogelstein and Priebe 2015). However, graph matching based approaches are currently not computationally feasible for larger unlabeled networks such as these considered in this study due to the emerging combinatorial complexity (Vogelstein, Conroy, et al. 2015).

5.4. Functional Testing as Alternative and Complement to Structural Testing

Functional neuronal model testing is an alternative to structural model testing which has been pursued using temporally resolved recordings (functional data) of small neuronal population of tens or hundreds of neurons (Barak et al. 2013). Barak et al. (2013) compared three models of delayed discrimination. In that study, dynamic properties of the examined models were compared with experimental data to select the best-fitting model. However, the criteria defining similarity and ultimately leading to the model selection were not well defined. The best fitting model was selected on a (partially subjective) discussion of different properties, but not as result of a mathematical model selection procedure. This is not a limitation of functional data, but rather of the employed method. Functional testing can also be performed through, e.g., an ABC-SMC method, in a more rigorous and well defined fashion. However, a functional approach has other drawbacks: it relies necessarily on a certain behavioral paradigm and anesthesia conditions which influence the neuronal network dynamics. The structural approach in comparison, is independent of the momentary behavior of the animal or anesthesia conditions but depends most likely on the complete history of the animal. Functional model testing can be complementary to structural model testing. Ideally, both approaches are combined.

5.5. Design Decisions and Limitations

The proposed method (such as many sampling based methods) is computationally limited by the efficiency with which the (network) models can be numerically simulated. These simulations are performed repeatedly during the ABC-SMC procedure. It is also necessary to find computationally efficient and at the same time descriptive (network) summary statistics. If the computation of the summary statistics takes very long, then the same problem is faced as with long running model simulations. The descriptiveness of the summary statistics is important as well: ABC methods might yield biased results if the summary statistics are poorly chosen (Fay et al. 2015; Robert et al. 2011). The proposed summary statistics (relative excitatory-excitatory reciprocity $r_{\mathrm{rel,ee}}$, relative excitatory-inhibitory reciprocity $r_{\mathrm{rel,ei}}$, relative number of cycles of length 5 $r^{(5)}$ and in-out-degree correlation $r_{\mathrm{i/o}}$) work most reliably for networks of the order of approximately 100 neurons or more. While the method can also be applied to smaller networks, e.g., 10 or 20 neurons, it is in this scenario less robust to noise and fractional connectomic reconstruction. However, such small networks were not the scope of this study. Different summary statistics, such as, e.g., spectral statistics or motif counts are an alternative to the summary statistics employed here. In fact, approximate network motif counts as summary statistics were evaluated here as well (App. A.2). However, these motif counts did not improve model selection performance. Instead, since these motif counts rely on Monte Carlo sampling, they introduced additional stochasticity and increased the computational cost. Therefore, computationally efficient network statistics which did not require Monte Carlo sampling were used. All the statistics used here were calculated on an unweighted (binary) graph representation, but the proposed method can also be extended to take synaptic weights into account. For example, only the sub-connectomes consisting of the strongest (largest) or weakest (smallest) connections could be considered. Alternatively, the connection weights or the number of synapses of which a connection consists could be directly incorporated into the summary statistics. But taking the connection weights into account complicates data pre-processing. The connection strength would have to be inferred, for example exploiting the correlation between synapse strength and morphological properties such as synapse size and shape (Branco, Marra, and Staras 2010; Harris and Stevens 1989; Murthy et al. 2001). In comparison, it is much easier to only determine whether a synapse between two neurons is present at all or not, irrespective of its strength.

5.6. The Importance of Systematic Pre Hoc Analysis

Connectomic data of whole barrels (as local circuit modules) is not available yet. Why design the analysis before the connectome is reconstructed? Would it not be easier to explore the connectome and "look" for discriminative features once it is there (R. Gütig, personal communication)? The pre hoc design of the analysis is beneficial in several ways: First, the statistical power of tests with pre hoc defined endpoints is higher than the power of tests with post hoc defined endpoints (Pocock and G. W. Stone 2016; Wilson, Karakasis, and Oza 2015). Definition of pre hoc endpoints is therefore standard in the design of clinical studies (Pocock and G. W. Stone 2016). This is especially important in the field of connectomics as most studies rely so far on a single connectome (Helmstaedter, Briggman, Turaga, et al. 2013; Kasthuri et al. 2015; Varshney et al. 2011; J. G. White et al. 1986). Second, in terms of experimental design, the pre hoc analysis has the advantage to specify with which accuracy it has to be reconstructed and how much has to reconstructed. It has been noted that one of the problems in connectomics is, that "you never know when you are done" (L. Scheffer, connectomics conference 2017, Berlin). Here, it is specifed how much is needed to be reconstructed and with which precision. Third, exploratory analysis is subjective and not systematic. It is rather unclear how to discriminate models by exploratory analysis (Sect. 3.1).

5.7. Future Directions

Several extensions are possible for future studies: For example, a focus could be set on the precise inference of the individual models' parameters. The proposed method was not optimized for inferring individual model parameters as precisely as possible. In fact to save computational resources, ABC-SMC sampling stopped as soon as one model had probability 1 since the interested lied in the models' probabilities only. To obtain more accurate posteriors on the models' parameters more intensive ABC-SMC sampling would likely be necessary, maybe even accompanied by additional summary statistics. Other types of network reconstruction noise and their effects on connectome classification could be examined in a systematic fashion. Types of noise that preserve all degrees were implemented and examined (App. A.1). However, such perturbation is less severe. Therefore the more drastic approach of adding and removing edges randomly was pursued. Apart from model selection on individual animals, comparative studies could be performed through the proposed network summary statistics. The structure of the same barrel across animals and across animal development, the structure of different barrels within an animal or even mouse and rat barrels could be compared for differences beyond size. The proposed method is also general enough to be extended to other brain areas, such as visual cortex.

5.8. Conclusion

The present thesis is one of the very first attempts to make connectomic data usable for analysis and interpretation in a systematic fashion. It was shown that a connectome can be used to select the most likely implemented model out of a range of candidate models under realistic experimental conditions. In addition the necessary tools to perform such inference on distributed hardware are provided. Concrete connectomic data is leveraged to constrain cortical computational models, providing a step towards a mechanistic understanding of computations performed in local cortical modules.

A. Appendix

A.1. Degree-Preserving Noise

Let $p(u|v)$ denote the transition matrix of the Markov chain with transitions defined by the canonical moves, the square swap and the triangle swap (Roberts and Coolen 2012), on a graph (connectome) (Fig. A.1). In this Markov chain the states are graphs (connectomes) themselves and a transition from one state u to a second state v exists if u can be transformed into v by applying one of the canonical moves on a subset of the 4 nodes (square swap) or 3 nodes (triangle swap) of the state (which is a graph) u. For simplicity, focus first on the case of excitatory neurons. On the associated Markov graph $G_\mathrm{M} = (V_\mathrm{M}, E_\mathrm{M})$ states (connectomes) are represented by nodes $v \in V_\mathrm{M}$ and edges $e \in E_\mathrm{M}$ exist between states with strictly positive transition probability (Fig. A.2). The transition probability from a graph (connectome) u to a graph (connectome) v is positive if u can be transformed into v by application of one of the canonical moves. Let $n(v) = |\{e \in E_\mathrm{M} | \exists u \in G_\mathrm{M} : e = (v, u)\}|$ denote the number of neighbors of state v. By finiteness of the graph there exists $\rho > 0$ with $\max\{\rho n(v) | v \in V_\mathrm{M}\} < 1$. The transition matrix is chosen such that the transition from a given state to each neighboring state occurs with constant probability ρ for all neighboring states. The probability of staying

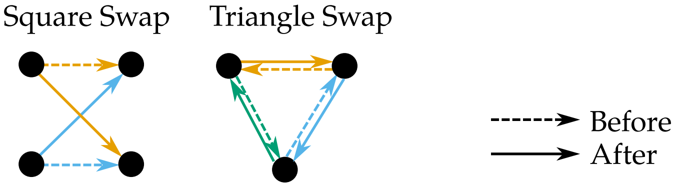

Figure A.1. Canonical network moves preserve the degree distribution.
The square swap and the triangle swap constitute the canonical moves on a directed network. Both actions are degree distribution preserving. Any two networks with the same degree distribution can be transformed into each other by these moves.

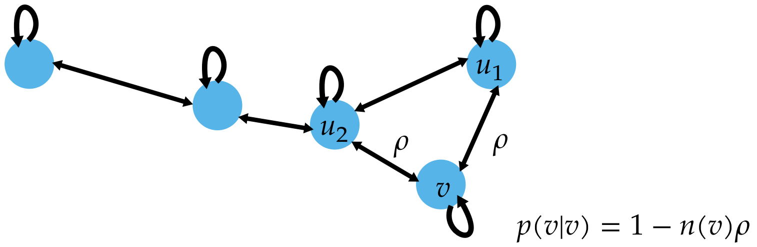

Figure A.2. Uniform degree preserving shuffling.
Transitions from node to node are realized with probability ρ. The probability $p(v|v)$ to stay at a node v depends on its degree $n(v)$.

in a state is therefore $1 - \rho n(v)$ and the transition matrix

$$p(u|v) = \begin{cases} \rho & u \sim v \\ 1 - \rho n(v) & u = v \\ 0 & \text{otherwise} \end{cases} .$$

The so defined Markov chain is ergodic. Aperiodicity holds as long as the Markov graph has at least one self-loop, which is given in all practical cases. Irreducibility holds by connectedness of the Markov graph. The Markov graph is connected since the canonical moves allow to transition from any graph with a given degree distribution to any other graph with the same degree distribution. A stationary state exists by symmetry of the transition matrix $p(u|v) = p(v|u)$. Thus, the uniform distribution $\pi_v = c$ for $c = \frac{1}{|V_\mathrm{M}|}$ is a stationary distribution. By aperiodicity and irreducibility it is also the unique stationary distribution. The same argument holds within the inhibitory population only and can be similarly applied to the excitatory-to-inhibitory submatrix and the inhibitory-to-excitatory submatrix.

A.2. Stochastic Network Measures

A stochastic alternative to the recurrency measure introduced in Sect. 4.1 is the stochastic sampling of cycle network motifs employing path sampling (Kretschmar 2013). Here, an efficient version of the algorithm presented by Kretschmar (2013), achieving a performance improvement by a factor of 900 is implemented. A factor of approximately 30 is gained by using more efficient data structures and another factor of 30 by an implementation in Cython as opposed to Matlab. Since only the cycle counts and path counts are of interest, only these are returned but not the nodes of which these consist.

A.2.1. Strong Path Sampling

For strong path sampling, i.e. sampling of strong directed paths, Alg. 8 is employed. There, paths up to length N_{max} are sampled from a graph G. Lists c_{path} and c_{cycle} are initialized, which hold the counts of paths and cycles of length n at position n. For each node in the vertex set V of G, a path extension attempt is made with probability p_{extend}. The extension proceeds recursively according to Alg. 9. There, the path counter is increased first. If the path is a cycle (i.e. the last node in P is connected by a directed edge to the first node in P), then also the cycle counter is increased. If the maximum path

Algorithm 8: Strong Path Sampling

```
Input: $G, N_{\text{max}}, p_{\text{extend}}$
Output: $c_{\text{path}}, c_{\text{cycle}}$
$(V, E) \leftarrow G$
$c_{\text{path}} \leftarrow$ zeros($N_{\text{max}}$)
$c_{\text{cycle}} \leftarrow$ zeros($N_{\text{max}}$)
foreach $v \in V$ do
    if $\mathcal{U}(0,1) < p_{\text{extend}}$ then
        extend_strong([$v$], $c_{\text{path}}, c_{\text{cycle}} N_{\text{max}}, G, p_{\text{extend}}$)
    end
end
```

Algorithm 9: `extend_strong`

```
Input: $P, c_{\text{path}}, c_{\text{cycle}}, N_{\text{max}}, G, p_{\text{extend}}$
$c_{\text{path}}[|P|] \mathrel{+}= 1$
if $P$ is a strong cycle then
    $c_{\text{cycle}}[|P|] \mathrel{+}= 1$
end
if $|P| > N_{\text{max}}$ then
    return
end
foreach $v \in$ target nodes of the last node of $P$ do
    if $v \notin P$ and $\mathcal{U}(0,1) < p_{\text{extend}}$ then
        extend_strong($P$ + [$v$], $c_{\text{path}}, c_{\text{cycle}}, N_{\text{max}}, G, p_{\text{extend}}$)
    end
end
```

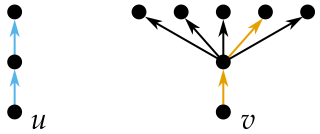

Figure A.3. Strong path sampling.
The strong path sampling Alg. 8 visits each path of length n with probability p^n_{extend}. A random walk would visit the blue path with higher probability than the orange path.

length $N_{\max}$ is reached, the algorithm returns; if not, then each of the target nodes v of the last node in the current path P is probed for extension. Only if v is not already in the path, the path is extended with probability $N_{\max}$ and Alg. 9 is called recursively.

The obtained counts c_{path} and c_{cycle} are compared with counts obtained from perturbations of the graph G, e.g. by the degree-preserving randomization procedure described in App. A.1, to assess whether directed cycles occurred with higher-than-expected frequency. Also, the fractions of paths which are cycles can be assessed for varying lengths.

Algorithm 8 visits each path of length n with probability p^n_{extend}. This is in contrast to a random walk on the graph. Consider the example in Fig. A.3. A random walk would choose the start nodes u and v with the same probability. If u is chosen, the only possible path of length 3 is the blue path. In contrast, if v is chosen, there are 5 possible paths of length 3 to choose from; the orange path is only one of them. Therefore, the probability of visiting the orange path is one fifth of the probability of visiting the blue path in a random walk on this example graph. However, Alg. 8 visits both paths with equal probability but the total number of visited paths is not fixed. In case v is chosen, the number of visited paths is likely higher than in the case of choosing u as start node.

A.2.2. Weak Path Enumeration Sampling

Weak path enumeration sampling ignores the direction of the edges. A weak path or cycle in a directed graph is one that is a path or cycle irrespective of the edge direction. Sampling weak paths or cycles hence amounts to sampling paths or cycles in an undirected graph. To ensure uniform sampling, an enumeration technique is applied (Kretschmar 2013; Wernicke 2006). A strict total order $\succ$ is introduced on the edges E of a graph $G = (V, E)$ which exists by finiteness of E. Each edge e of the graph is probed as starting edge e_{start} of a weak path P and is attempted to be extended (Alg. 10). The extension proceeds according to Alg. 11. It is first checked whether the weak path P is an open path, defines a balanced cycle or an unbalanced cycle (Alg. 11, Fig. A.4). A weak cycle is called balanced if a traversal of the cycle traverses the same number of edges in-direction as in opposite direction. A balanced cycle consists always of an even number of edges. A weak cycle that is not balanced is called unbalanced. It is attempted to expand the weak path P at its

Algorithm 10: Weak Path Enumeration Sampling

```
Input: $G, N_{\max}, p_{\text{extend}}$
Output: $c_{\text{open}}, c_{\text{balanced}}, c_{\text{unbalanced}}$
$(V, E) \leftarrow G$
$c_{\text{open}} \leftarrow$ zeros($N_{\max}$)
$c_{\text{balanced}} \leftarrow$ zeros($N_{\max}$)
$c_{\text{unbalanced}} \leftarrow$ zeros($N_{\max}$)
foreach $e \in E$ do
    if $\mathcal{U}(0,1) < p_{\text{extend}}$ then
        extend_weak($e, G, True, c, p_{\text{extend}}$)
    end
end
```

last node v_{end} (Alg. 11). A set $E_{\text{candidate}}$ of candidate edges for extension is assembled from all edges adjacent to last node v_{end} of the path but not to any other node of the path. The path is extended only if a candidate edge e is a successor of the first edge e_{start} in the weak path P (Fig. A.4). To reduce the number of visited paths, this extension is performed with probability p_{extend}. The start of the weak path P is extended similarly to the end of the path P. However, an extension of the start does not trigger an extension of the end in the next recursive call; the argument *extend_both* is set to *False*. The number of open paths, balanced cycles and uncounted cycles visited over the recursive calls is tracked through counts stored in c (Alg. 11, Fig. A.4).

Algorithm 10 visits all weak paths of length n with probability p^n_{extend} (Wernicke 2006). Consider first the simple case of $p_{\text{extend}} = 1$. Each path P of length n is visited exactly once. Let $e_{\min}$ denote the unique edge in P with $e' \succ e_{\min} \forall e' \in P \setminus \{e\}$. This edge was chosen as the first edge added to the path, which is by definition of the algorithm the minimum edge of P. If the path consists of n additional edges appended to the path and m edges prepended to the path, then the path is visited by extending n times at the end ($extend\_both = True$), followed by m extensions at the start ($extend\_both = False$). No other calling sequence yields this path, so every path is visited exactly once. The probability p^n_{extend} follows from executing the recursive calls with probability p_{extend}.

The counts of open paths, balanced and unbalanced cycles obtained from a graph G can be compared to the counts obtained from reshuffled version of G to assess over- or underrepresentation of these motifs.

Algorithm 11: `extend_weak`

```
Input: $P, G, extend\_both, N_{\max}, c, p_{\text{extend}}$
if $P$ is an open path then
    $c_{\text{open}}[|P|] += 1$
end
if $P$ is a balanced cycle then
    $c_{\text{balanced}}[|P|] += 1$
end
if $P$ is an unbalanced cycle then
    $c_{\text{unbalanced}}[|P|] += 1$
end
if $|P| > N_{\max}$ then
    return
end
$(V, E) \leftarrow G$
$e_{\text{start}} \leftarrow (P_1, P_2)$
if extend_both then
    # Extend the end of $P$
    $v_{\text{end}} \leftarrow P_{|P|}$
    $E_{\text{candidate}} \leftarrow \{e \in E | e \cap P = \{v_{\text{end}}\}\}$
    foreach $e \in E_{\text{candidate}}$ do
        if $e \succ e_{\text{start}}$ and $\mathcal{U}(0,1) < p_{\text{extend}}$ then
            $\{v_{ext}\} \leftarrow e \setminus \{v_{\text{end}}\}$
            extend_weak($P + [v_{ext}], G, True, N_{\max}, c, p_{\text{extend}}$)
        end
    end
end
# Extend the start of $P$
$v_{\text{start}} \leftarrow P_1$
$E_{\text{candidate}} \leftarrow \{e \in E | e \cap P = \{v_{\text{start}}\}\}$
foreach $e \in E_{\text{candidate}}$ do
    if $e \succ e_{\text{start}}$ and $\mathcal{U}(0,1) < p_{\text{extend}}$ then
        $\{v_{ext}\} \leftarrow e \setminus \{v_{\text{start}}\}$
        extend_weak($[v_{ext}] + P, G, False, N_{\max}, c, p_{\text{extend}}$)
    end
end
```

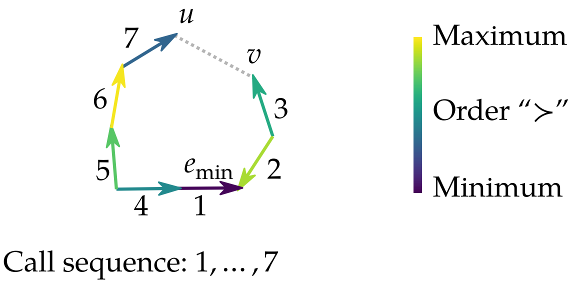

Figure A.4. Weak path enumeration sampling.
The edges are ordered according to "≻". The numbers indicate the sequence in which edges are added to the path through recursive calls of Alg. 11. The minimal edge $e_{\min}$ is by construction the first edge in the path. Subsequently added edges e have to satisfy $e \succ e_{\min}$. The path is a weak path, edges in the path are not necessarily aligned. The presence or absence and possible direction of an edge from u to v (gray dashed line) determines whether the path is open, defines a balanced cycle or an unbalanced cycle. In this example, the path is open if no edge from node u to node v exists, the path defines a balanced cycle if an edge with direction from v to u exists and an unbalanced cycle if an edge with direction from u to v exists.

Glossary

η_{i} intrinsic learning rate. 47, 48, 51, 52, 99

η_{STDP} STDP learning rate. 47, 48, 51, 52, 99

a_{in} input activity. 24, 25, 41, 42

a_{L4} L4 activity. 24, 25

a_{out} readout activity. 25

ABC Approximate Bayesian Computation. 66, 67, 72, 111, 114

ABC-SMC Approximate Bayesian Computation - Sequential Monte Carlo. 8, 67–70, 72–74, 76–78, 83, 84, 86–94, 96, 97, 100, 102–106, 111, 113–115, 130, 131

Adam adaptive moment estimation. 25

API anti phase inhibition. 12, 37, 38, 40–43, 45, 64, 65, 99

c correlation. 44, 50

C_{sim} cosine similarity. 22, 23, 39–42

CDF cumulative distribution function. 17, 23, 36, 52, 103, 104

CERN European Organization for Nuclear Research. 1

connectomics the field within neuroscience dedicated to the reconstruction of the connections between individual neural entities, such as single neurons. 6

cytoarchitecture the cellular composition of tissue. From Greek “κτος” = “cell” and “αρχιτεκτονικ” = “architecture”. 2

d_{EXP} decay. 108, 109

d_{f} feature space dimension. 15, 17, 18, 20, 22, 23, 38–42, 44, 45, 99, 129

d_{par} parameter dimension. 73, 76, 93

$d_{\mathrm{s}}/d_{\mathrm{b}}$ relative inter-soma distance. 31

$\mathbb{E}[p_{\mathrm{ee}}]$ expected excitatory-excitatory connectivity. 34, 35, 129

$\mathbb{E}[r_{\mathrm{ee}}]$ expected excitatory-excitatory reciprocity. 34, 35, 129

E_{CV} target density variation. 79–85, 87–89, 91, 94

EM electron microscopy. 5, 6

ER Erdős-Rényi. 21, 27, 28, 32, 38, 46, 108

ER-ESN Erdős-Rényi echo state network. 11, 22, 27–29, 32, 37, 38, 64, 68, 99, 102, 107–109, 129

ESN echo state network. 26–28, 30

EXP-LSM exponentially decaying liquid state machine. 11, 22, 30–33, 37, 38, 64, 65, 68, 99, 107–109, 129

f_{CA} fractional chain activation. 60, 61

f_{PA} fractional pool activation. 58

f_{r} feverization. 20–23, 99

f_{s} reconstruction fraction. 99, 104–107

FEVER feature vector recombination. 11, 15–23, 38, 39, 64, 68, 99, 102, 108, 109, 129

KDE kernel density estimator. 73, 75, 79, 81, 93

l layer. 33–35

L2/3 layer 2/3. 25, 26

L4 layer 4. 7, 8, 11–13, 17, 24, 25, 28, 38, 41

λ_{e} excitatory decay length. 30–32, 107

λ_{i} inhibitory decay length. 30–32, 107

LASSO least absolute shrinkage and selection operator. 16, 19, 22

LAYERED layered network. 11, 22, 28, 32, 34–38, 64, 65, 99, 102, 103, 109, 129

LIGO Laser Interferometer Gravitational-Wave Observatory. 1

LSM liquid state machine. 29, 30

m model. 99

MAP maximum a posteriori. 102, 105–107

$\mathrm{med}(c_{\mathrm{N,N}})$ median correlation. 51, 52, 110

n number of neurons. 12, 15, 18, 20, 24, 42, 46, 48, 51, 97, 99

n_{e} number of excitatory neurons. 12, 20, 27, 30, 32–34, 46, 48, 56, 57, 59, 97, 98

n_{eff} effective population size. 76, 83, 88

n_{i} number of inhibitory neurons. 12, 27, 46, 56, 57, 97, 98

n_{l} number of layers. 33–36, 99

n_{modes} number of posterior modes. 82–85

n_{p} number of input patterns. 109, 110

n_{pool} number of pools. 56–58, 61

n_{pow} selectivity. 39, 40, 42, 45, 99

ODE ordinary differential equation. 21

$p_{0,\mathrm{e}}$ initial excitatory connectivity. 18–20

$p_{0,\mathrm{i}}$ initial inhibitory connectivity. 18–20

p_{add} synapse addition probability. 47

p_{e} excitatory connectivity. 20, 27, 28, 32, 34, 40, 41, 46, 48, 56, 57, 99

$p_{\mathrm{e,f}}$ excitatory forward connectivity. 34–36, 54, 99, 129

$p_{\mathrm{e,l}}$ excitatory lateral connectivity. 34–36, 54, 99, 129

p_{ee} excitatory-excitatory connectivity. 13, 18, 23, 28, 32, 34, 39, 40, 45, 49, 50, 57, 58, 97, 98

p_{ei} excitatory-inhibitory connectivity. 13, 34, 57, 97, 98

p_{i} inhibitory connectivity. 20, 27, 32, 34, 40, 41, 46, 54, 56, 57, 99

p_{ie} inhibitory-excitatory connectivity. 13

p_{ii} inhibitory-inhibitory connectivity. 13

PDF probability density function. 42

postsynaptic the neuron with the incoming connection →●. 28

presynaptic the neuron with the outgoing connection ●→. 28

$r^{(5)}$ relative number of cycles of length 5. 98–101, 114

r_{ee} excitatory-excitatory reciprocity. 13, 18, 23, 27, 28, 32, 39, 40, 45, 50, 54, 57, 58, 97, 98

r_{ei} excitatory-inhibitory reciprocity. 98

r_{i} fraction of inhibitory neurons. 99

$r_{\mathrm{i/o}}$ in-out-degree correlation. 98–101, 114

$r_{\mathrm{rel,ee}}$ relative excitatory-excitatory reciprocity. 98–101, 114

$r_{\mathrm{rel,ei}}$ relative excitatory-inhibitory reciprocity. 98–101, 114

ρ_{syn} synapse density. 50

rich-club informally, a graph is said to obey the rich-club property if high-degree nodes tend to connect to other high-degree nodes. 4

$\mathcal{S}$ data space. 73, 74

s simulated data. 73, 74, 82, 83

S1 primary somatosensory cortex. 7, 8, 11, 17

s_{obs} observed data. 73–75, 83–85, 87, 90, 91

s_{pool} pool size. 56–61, 99

$s_{\mathrm{pool,i}}$ inhibitory pool size. 56, 57

SBEM serial section block face electron microscopy. 5

SBM stochastic block model. 64

SEM scanning electron microscopy. 5, 6

small-world informally, a graph is said to obey the small-world property if the typical distance between two nodes grows logarithmically with the number of nodes in the network. That is, although most nodes are not directly connected to each other, the typical distance between nodes is still short. 4

SORN self-organizing recurrent neural network. 12, 46, 50–52, 64, 65, 68, 99, 102, 103, 109, 110, 131

SSE spike source entropy. 50–52

STDP spike timing dependent plasticity. 46–49, 109

SYNFIRE synfire chain. 12, 54, 55, 57–59, 61, 64, 99

t time. 58, 60

uniform distribution $\mathcal{U}(a, b)$ is the uniform distribution on the interval $[a, b)$. 22, 24, 30, 39, 59, 69, 83, 86

uniform integer distribution $\mathcal{U}\mathcal{I}(a, b)$ is the uniform distribution on the integers in the interval $[a, b]$. 17, 35, 42, 45, 48, 61

VPM ventral posteromedial nucleus. 13, 17, 24, 25, 38

W_{E} excitatory submatrix. 12, 98

W_{I} inhibitory submatrix. 12

ξ noise. 99, 102–107, 109

List of Figures

List of Tables

Bibliography

Abeles, M. (1991). *Corticonics: Neural Circuits of the Cerebral Cortex*. Cambridge University Press. 298 pp.

Abeles, M. (1982). *Local Cortical Circuits*. Red. by V. Braitenberg et al. Vol. 6. Studies of Brain Function. Berlin, Heidelberg: Springer Berlin Heidelberg.

— (2009). "Synfire chains". *Scholarpedia* 4.7, 1441.

Abeles, M., G. Hayon, and D. Lehmann (2004). "Modeling Compositionality by Dynamic Binding of Synfire Chains". *Journal of Computational Neuroscience* 17.2, 179–201.

Aertsen, A. and V. Braitenberg (1996). *Brain theory*. Elsevier Science.

Aharon, M., M. Elad, and A. Bruckstein (2006). "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation". *IEEE Transactions on Signal Processing* 54.11, 4311–4322.

Arnoldi, H.-M. R., K.-H. Englmeier, and W. Brauer (1999). "Translation-invariant pattern recognition based on Synfire chains". *Biological Cybernetics* 80.6, 433–447.

Aswolinskiy, W. and G. Pipa (2015). "RM-SORN: a reward-modulated self-organizing recurrent neural network". *Frontiers in Computational Neuroscience* 9.

Auer, P., H. Burgsteiner, and W. Maass (2001). "The p-delta learning rule for parallel perceptrons". *submitted for publication*.

Averbeck, B. B., P. E. Latham, and A. Pouget (2006). "Neural correlations, population coding and computation". *Nature Reviews Neuroscience* 7.5, 358–366.

Aviel, Y., D. Horn, and M. Abeles (2005). "Memory Capacity of Balanced Networks". *Neural Computation* 17.3, 691–713.

Aviel, Y., C. Mehring, et al. (2003). "On Embedding Synfire Chains in a Balanced Network". *Neural Computation* 15.6, 1321–1340.

Azevedo, F. A. et al. (2009). "Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain". *The Journal of Comparative Neurology* 513.5, 532–541.

Bale, M. R. et al. (2013). "Low-Dimensional Sensory Feature Representation by Trigeminal Primary Afferents". *Journal of Neuroscience* 33.29, 12003–12012.

Barabási, A.-L. and R. Albert (1999). "Emergence of Scaling in Random Networks". *Science* 286.5439, 509–512.

Barak, O. et al. (2013). “From fixed points to chaos: Three models of delayed discrimination”. *Progress in Neurobiology*. Conversion of Sensory Signals into Perceptions, Memories and Decisions 103, 214–222.

Bargmann, C. I. (2012). “Beyond the connectome: How neuromodulators shape neural circuits”. *BioEssays* 34.6, 458–465.

Bargmann, C. I. and E. Marder (2013). “From the connectome to brain function”. *Nature Methods* 10.6, 483–490.

Beaumont, M. A. (2010). “Approximate Bayesian Computation in Evolution and Ecology”. *Annual Review of Ecology, Evolution, and Systematics* 41.1, 379–406.

Beaumont, M. A., W. Zhang, and D. J. Balding (2002). “Approximate Bayesian Computation in Population Genetics”. *Genetics* 162.4, 2025–2035.

Becker, C. et al. (2012). “Learning Context Cues for Synapse Segmentation in EM Volumes”. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2012*. Ed. by N. Ayache et al. Lecture Notes in Computer Science 7510. Springer Berlin Heidelberg, 585–592.

Beierlein, M., J. R. Gibson, and B. W. Connors (2003). “Two Dynamically Distinct Inhibitory Networks in Layer 4 of the Neocortex”. *Journal of Neurophysiology* 90.5, 2987–3000.

Berger, H. (1929). “Über das Elektrenkephalogramm des Menschen”. *Archiv für Psychiatrie und Nervenkrankheiten* 87.1, 527–570.

Bi, G.-q. and M.-m. Poo (1998). “Synaptic Modifications in Cultured Hippocampal Neurons: Dependence on Spike Timing, Synaptic Strength, and Postsynaptic Cell Type”. *The Journal of Neuroscience* 18.24, 10464–10472.

Bienenstock, E. (1995). “A model of neocortex”. *Network: Computation in Neural Systems* 6.2, 179–224.

Blanche, T. J. et al. (2005). “Polytrodes: High-Density Silicon Electrode Arrays for Large-Scale Multiunit Recording”. *Journal of Neurophysiology* 93.5, 2987–3000.

Bock, D. D. et al. (2011). “Network anatomy and in vivo physiology of visual cortical neurons”. *Nature* 471.7337, 177–182.

Boergens, K. M. et al. (2017). “webKnossos: efficient online 3D data annotation for connectomics”. *Nature Methods* 14.7, 691–694.

Bowman, A. W. (1984). “An alternative method of cross-validation for the smoothing of density estimates”. *Biometrika* 71.2, 353–360.

Bowman, A. W. and A. Azzalini (1997). *Applied Smoothing Techniques for Data Analysis: The Kernel Approach with S-Plus Illustrations*. Oxford University Press. 210 pp.

Branco, T., V. Marra, and K. Staras (2010). “Examining size-strength relationships at hippocampal synapses using an ultrastructural measurement of synaptic release probability”. *Journal of Structural Biology* 172.2, 203–210.

Briggman, K. L., M. Helmstaedter, and W. Denk (2011). "Wiring specificity in the direction-selectivity circuit of the retina". *Nature* 471.7337, 183–188.

Brodmann, K. (1909). *Vergleichende Lokalisationslehre der Großhirnrinde*. Leipzig: Barth.

Buehner, M. and P. Young (2006). "A tighter bound for the echo state property". *IEEE Transactions on Neural Networks* 17.3, 820–824.

Bullmore, E. and O. Sporns (2009). "Complex brain networks: graph theoretical analysis of structural and functional systems". *Nature Reviews Neuroscience* 10.3, 186–198.

Buonomano, D. V. and W. Maass (2009). "State-dependent computations: spatiotemporal processing in cortical networks". *Nature Reviews Neuroscience* 10.2, 113–125.

Cajal, S. R. y (1888). *Estructura de los centros nerviosos de las aves*.

Chatrchyan, S. et al. (2012). "Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC". *Physics Letters B* 716.1, 30–61.

Chollet, F. et al. (2015). "Keras".

Collin, G. et al. (2014). "Structural and Functional Aspects Relating to Cost and Benefit of Rich Club Organization in the Human Cerebral Cortex". *Cerebral Cortex* 24.9, 2258–2267.

Coppersmith, D. and S. Winograd (1987). "Matrix multiplication via arithmetic progressions". *Proceedings of the nineteenth annual ACM symposium on Theory of computing*. ACM, 1–6.

Daigle, B. J. et al. (2012). "Accelerated Maximum Likelihood Parameter Estimation for Stochastic Biochemical Systems". *BMC Bioinformatics* 13, 68.

Denk, W., K. L. Briggman, and M. Helmstaedter (2012). "Structural neurobiology: missing link to a mechanistic understanding of neural computation". *Nature Reviews Neuroscience* 13.5, 351–358.

Denk, W. and H. Horstmann (2004). "Serial Block-Face Scanning Electron Microscopy to Reconstruct Three-Dimensional Tissue Nanostructure". *PLOS Biol* 2.11, e329.

Desai, N. S., L. C. Rutherford, and G. G. Turrigiano (1999). "Plasticity in the intrinsic excitability of cortical pyramidal neurons". *Nature Neuroscience* 2.6. 515–520.

Diamond, M. E. et al. (2008). "'Where' and 'what' in the whisker sensorimotor system". *Nature Reviews Neuroscience* 9.8, 601–612.

Dorkenwald, S. et al. (2017). "Automated synaptic connectivity inference for volume electron microscopy". *Nature Methods* 14.4, 435–442.

Druckmann, S. and D. B. Chklovskii (2012). "Neuronal Circuits Underlying Persistent Representations Despite Time Varying Activity". *Current Biology* 22.22, 2095–2103.

Edwards, F. A. et al. (1989). "A thin slice preparation for patch clamp recordings from neurones of the mammalian central nervous system". *Pflügers Archiv* 414.5, 600–612.

Eigen, M. (1996). "Prionics or The Kinetic Basis of Prion Diseases". *Biophysical Chemistry* 63.1, A1–A18.

Einstein, A. (1916). "Näherungsweise Integration der Feldgleichungen der Gravitation".

Erdős, P. and A. Rényi (1959). "On random graphs". *Publicationes Mathematicae Debrecen* 6, 290–297.

Fay, D. et al. (2015). "Graph metrics as summary statistics for Approximate Bayesian Computation with application to network model parameter estimation". *Journal of Complex Networks* 3.1, 52–83.

Feldmeyer, D. (2012). "Excitatory neuronal connectivity in the barrel cortex". *Frontiers in Neuroanatomy* 6, 24.

Feldmeyer, D., M. Brecht, et al. (2013). "Barrel cortex function". *Progress in Neurobiology*. Conversion of Sensory Signals into Perceptions, Memories and Decisions 103, 3–27.

Feldmeyer, D., V. Egger, et al. (1999). "Reliable synaptic connections between pairs of excitatory layer 4 neurones within a single 'barrel' of developing rat somatosensory cortex". *The Journal of Physiology* 521.1, 169–190.

Filippi, S. et al. (2013). "On Optimality of Kernels for Approximate Bayesian Computation Using Sequential Monte Carlo". *Statistical Applications in Genetics and Molecular Biology* 12.1, 87–107.

Fukushima, K. (1979). "Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position- Neocognitron". *ELECTRON. & COMMUN. JAPAN* 62.10, 11–18.

— (1980). "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". *Biological Cybernetics* 36.4, 193–202.

Gal, E. et al. (2017). "Rich cell-type-specific network topology in neocortical microcircuitry". *Nature Neuroscience* 20.7, 1004–1013.

Gamma, E. et al. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. 1 edition. Boston: Addison-Wesley Professional. 395 pp.

Gerstner, W. et al. (1996). "A neuronal learning rule for sub-millisecond temporal coding". *Nature* 383.6595, 76–78.

Gewaltig, M.-O., M. Diesmann, and A. Aertsen (2001). "Propagation of cortical synfire activity: survival probability in single trials and stability in the mean". *Neural Networks* 14.6, 657–673.

Gibson, J. R., M. Beierlein, and B. W. Connors (1999). "Two networks of electrically coupled inhibitory neurons in neocortex". *Nature* 402.6757, 75–79.

— (2005). "Functional Properties of Electrical Synapses Between Inhibitory Interneurons of Neocortical Layer 4". *Journal of Neurophysiology* 93.1, 467–480.

Gillespie, D. T. (1977). "Exact Stochastic Simulation of Coupled Chemical Reactions". *The Journal of Physical Chemistry* 81.25, 2340–2361.

Goedeke, S. and M. Diesmann (2008). “The mechanism of synchronization in feed-forward neuronal networks”. *New Journal of Physics* 10.1, 015007.

Golgi, C. (1873). “Sulla struttura della sostanza grigia del cervello”. *Gazzetta Medica Italiana. Lombardia* 33, 244–246.

Goñi, J. et al. (2014). “Resting-brain functional connectivity predicted by analytic measures of network communication”. *Proceedings of the National Academy of Sciences* 111.2, 833–838.

Goodman, D. F. M. and R. Brette (2008). “Brian: a simulator for spiking neural networks in Python”. *Frontiers in Neuroinformatics* 2.

— (2009). “The Brian simulator”. *Frontiers in Neuroscience* 3.

Gray, E. G. (1959). “Axo-somatic and axo-dendritic synapses of the cerebral cortex”. *Journal of Anatomy* 93 (Pt 4), 420–433.

Griffith, J. S. (1963). “On the Stability of Brain-Like Structures”. *Biophysical Journal* 3.4, 299–308.

Hamill, O. P. et al. (1981). “Improved patch-clamp techniques for high-resolution current recording from cells and cell-free membrane patches”. *Pflügers Archiv* 391.2, 85–100.

Hansen, T. and H. Neumann (2004). “A simple cell model with dominating opponent inhibition for robust image processing”. *Neural Networks*. Vision and Brain 17.5, 647–662.

Harris, K. M. and J. K. Stevens (1989). “Dendritic spines of CA 1 pyramidal cells in the rat hippocampus: serial electron microscopy with reference to their biophysical characteristics”. *Journal of Neuroscience* 9.8, 2982–2997.

Hayon, G., M. Abeles, and D. Lehmann (2005). “A Model for Representing the Dynamics of a System of Synfire Chains”. *Journal of Computational Neuroscience* 18.1, 41–53.

Hazan, H. and L. M. Manevitz (2010). “The Liquid State Machine is not Robust to Problems in Its Components but Topological Constraints Can Restore Robustness”. *ICFC-ICNC 2010 - Proceedings of the International Conference on Fuzzy Computation and International Conference on Neural Computation, [parts of the International Joint Conference on Computational Intelligence IJCCI 2010], Valencia, Spain, October 24-26, 2010*. Ed. by J. Filipe and J. Kacprzyk. SciTePress, 258–264.

— (2012). “Topological constraints and robustness in liquid state machines”. *Expert Systems with Applications* 39.2, 1597–1606.

Helmstaedter, M. (2013). “Cellular-resolution connectomics: challenges of dense neural circuit reconstruction”. *Nature Methods* 10.6, 501–507.

Helmstaedter, M., K. L. Briggman, and W. Denk (2011). “High-accuracy neurite reconstruction for high-throughput neuroanatomy”. *Nature Neuroscience* 14.8, 1081–1088.

Helmstaedter, M., K. L. Briggman, S. C. Turaga, et al. (2013). “Connectomic reconstruction of the inner plexiform layer in the mouse retina”. *Nature* 500.7461, 168–174.

Helmstaedter, M., C. P. J. de Kock, et al. (2007). "Reconstruction of an average cortical column in silico". *Brain Research Reviews*. A Century of Neuroscience Discovery: Reflecting on the Nobel Prize to Golgi and Cajal in 1906 55.2, 193–203.

Herculano-Houzel, S. (2012). "The remarkable, yet not extraordinary, human brain as a scaled-up primate brain and its associated cost". *Proceedings of the National Academy of Sciences* 109 (Supplement 1), 10661–10668.

Herrmann, M., J. Hertz, and A. Prügel-Bennett (1995). "Analysis of synfire chains". *Network: Computation in Neural Systems* 6.3, 403–414.

Hertz, J. (1997). *Modelling Synfire Processing*.

Hertz, J. and A. Prügel-Bennett (1996). "Learning synfire chains: turning noise into signal". *International Journal of Neural Systems* 07.4, 445–450.

Heuvel, M. P. van den et al. (2013). "Abnormal rich club organization and functional brain dynamics in schizophrenia". *JAMA psychiatry* 70.8, 783–792.

Higgs, P. W. (1964). "Broken Symmetries and the Masses of Gauge Bosons". *Physical Review Letters* 13.16, 508–509.

Hochreiter, S. and J. Schmidhuber (1997). "Long Short-Term Memory". *Neural Computation* 9.8, 1735–1780.

Hodgkin, A. L. and A. F. Huxley (1952). "A quantitative description of membrane current and its application to conduction and excitation in nerve". *The Journal of Physiology* 117.4, 500–544.

Hogan, N. and T. Flash (1987). "Moving gracefully: quantitative theories of motor coordination". *Trends in Neurosciences* 10.4, 170–174.

Hua, Y., P. Laserstein, and M. Helmstaedter (2015). "Large-volume en-bloc staining for electron microscopy-based connectomics". *Nature Communications* 6, 7923.

Hubel, D. H. and T. N. Wiesel (1962). "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex". *The Journal of Physiology* 160.1, 106–154.

Ishida, E. E. O. et al. (2015). "cosmoabc: Likelihood-Free Inference via Population Monte Carlo Approximate Bayesian Computation". *Astronomy and Computing* 13, 1–11.

Ivakhnenko, A. G. (1971). "Polynomial Theory of Complex Systems". *IEEE Transactions on Systems, Man and Cybernetics* SMC-1.4, 364–378.

Ivakhnenko, A. G. and V. G. Lapa (1965). *Cybernetic predicting devices.* Washington, D.C.: Joint Publications Research Service [available from the Clearinghouse for Federal Scientific and Technical Information].

Izhikevich, E. M. (2006). "Polychronization: Computation with Spikes". *Neural Computation* 18.2, 245–282.

Izhikevich, E. M., J. A. Gally, and G. M. Edelman (2004). "Spike-timing Dynamics of Neuronal Groups". *Cerebral Cortex* 14.8, 933–944.

Jacquemin, C. (1994). "A Temporal Connectionist Approach to Natural Language". *SIGART Bull.* 5.3, 12–22.

Jaeger, H. (2001a). *Short term memory in echo state networks*. GMD-Forschungszentrum Informationstechnik.

— (2001b). *The "echo state" approach to analysing and training recurrent neural networks*.

— (2002). "Adaptive nonlinear system identification with echo state networks". *Advances in neural information processing systems*, 593–600.

— (2007). "Discovering multiscale dynamical features with hierarchical echo state networks". *Jacobs University Bremen, Tech. Rep.*

— (2010). "The "echo state" approach to analysing and training recurrent neural networks-with an erratum note". *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report* 148, 34.

— (2012). *Long short-term memory in echo state networks: Details of a simulation study*. Technical report, Jacobs University Bremen.

Jaeger, H. and H. Haas (2004). "Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication". *Science* 304.5667, 78–80.

Jagiella, N. et al. (2017). "Parallelization and High-Performance Computing Enables Automated Statistical Inference of Multi-Scale Models". *Cell Systems* 4.2, 194–206.e9.

Jennings, E. and M. Madigan (2017). "astroABC : An Approximate Bayesian Computation Sequential Monte Carlo Sampler for Cosmological Parameter Estimation". *Astronomy and Computing* 19, 16–22.

Jonas, E. and K. P. Kording (2017). "Could a Neuroscientist Understand a Microprocessor?" *PLOS Computational Biology* 13.1, e1005268.

Kaiser, M. and C. C. Hilgetag (2004). "Spatial growth of real-world networks". *Physical Review E* 69.3, 036103.

Kangasrääsiö, A. et al. (2016). "ELFI: Engine for Likelihood-Free Inference". *NIPS 2016 Workshop on Advances in Approximate Bayesian Inference*.

Kasthuri, N. et al. (2015). "Saturated Reconstruction of a Volume of Neocortex". *Cell* 162.3, 648–661.

Kayser, A. S. and K. D. Miller (2002). "Opponent Inhibition: A Developmental Model of Layer 4 of the Neocortical Circuit". *Neuron* 33.1, 131–142.

Kim, J. S. et al. (2014). "Space-time wiring specificity supports direction selectivity in the retina". *Nature* 509.7500, 331–336.

Kingma, D. and J. Ba (2014). "Adam: A Method for Stochastic Optimization". *arXiv:1412.6980 [cs]*. arXiv: 1412.6980.

Klinger, E. and J. Hasenauer (2017). "A Scheme for Adaptive Selection of Population Sizes in Approximate Bayesian Computation - Sequential Monte Carlo". *Computational*

Methods in Systems Biology. International Conference on Computational Methods in Systems Biology. Lecture Notes in Computer Science. Springer, Cham, 128–144.

Klinger, E., D. Rickert, and J. Hasenauer (2017). "pyABC: distributed, likelihood-free inference". *bioRxiv*, 162552.

Koelbl, C. et al. (2015). "A Barrel-Related Interneuron in Layer 4 of Rat Somatosensory Cortex with a High Intrabarrel Connectivity". *Cerebral Cortex* 25.3, 713–725.

Koutroumpas, K. et al. (2016). "Bayesian Parameter Estimation for the Wnt Pathway: an Infinite Mixture Models Approach". *Bioinformatics* 32.17, i781–i789.

Kreshuk, A., J. Funke, et al. (2015). "Who Is Talking to Whom: Synaptic Partner Detection in Anisotropic Volumes of Insect Brain". *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by N. Navab et al. Lecture Notes in Computer Science 9349. Springer International Publishing, 661–668.

Kreshuk, A., U. Koethe, et al. (2014). "Automated Detection of Synapses in Serial Section Transmission Electron Microscopy Image Stacks". *PLoS ONE* 9.2, e87351.

Kreshuk, A., C. N. Straehle, et al. (2011). "Automated Detection and Segmentation of Synaptic Contacts in Nearly Isotropic Serial Electron Microscopy Images". *PLoS ONE* 6.10, e24899.

Kretschmar, T. (2013). "Structural Analysis of the Neural Networks in Mouse Retina and Caenorhabditis Elegans". Master's Thesis.

Kumar, A., S. Rotter, and A. Aertsen (2008). "Conditions for Propagating Synchronous Spiking and Asynchronous Firing Rates in a Cortical Network Model". *The Journal of Neuroscience* 28.20, 5268–5280.

Lazar, A., G. Pipa, and J. Triesch (2007). "Fading memory and times series prediction in recurrent networks with different forms of plasticity". *Neural Networks*, 312–322.

— (2009). "SORN: a self-organizing recurrent neural network". *Frontiers in Computational Neuroscience* 3, 23.

Lecun, Y. et al. (1998). "Gradient-based learning applied to document recognition". *Proceedings of the IEEE* 86.11, 2278–2324.

LeCun, Y. et al. (1989). "Backpropagation Applied to Handwritten Zip Code Recognition". *Neural Computation* 1.4, 541–551.

Lee, A. K., J. Epsztein, and M. Brecht (2009). "Head-anchored whole-cell recordings in freely moving rats". *Nature Protocols* 4.3, 385–392.

Levenberg, K. (1944). "A Method for the Solution of Certain Non-linear Problems in Least Squares". *Quarterly of Applied Mathematics* 2.2, 164–168.

Levy, N. et al. (2001). "Distributed synchrony in a cell assembly of spiking neurons". *Neural Networks* 14.6, 815–824.

Liebald, B. (2004). "Exploration of effects of different network topologies on the ESN signal crosscorrelation matrix spectrum". Bachelor's thesis. Jacobs University Bremen.

Liepe, J. et al. (2010). "ABC-SysBio - Approximate Bayesian Computation in Python with GPU Support". *Bioinformatics* 26.14, 1797–1799.

LIGO Scientific Collaboration and Virgo Collaboration et al. (2016). "Observation of Gravitational Waves from a Binary Black Hole Merger". *Physical Review Letters* 116.6, 061102.

Lisman, J. (2015). "The challenge of understanding the brain: where we stand in 2015". *Neuron* 86.4, 864–882.

Lübke, J. et al. (2000). "Columnar Organization of Dendrites and Axons of Single and Synaptically Coupled Excitatory Spiny Neurons in Layer 4 of the Rat Barrel Cortex". *Journal of Neuroscience* 20.14, 5300–5311.

Lück, A. and V. Wolf (2016). "Generalized Method of Moments for Estimating Parameters of Stochastic Reaction Networks". *BMC Systems Biology* 10, 98.

Lukoševičius, M. and H. Jaeger (2009). "Reservoir computing approaches to recurrent neural network training". *Computer Science Review* 3.3, 127–149.

Lukoševičius, M., H. Jaeger, and B. Schrauwen (2012). "Reservoir Computing Trends". *KI - Künstliche Intelligenz* 26.4, 365–371.

Maass, W. (2010). "Liquid state machines: motivation, theory, and applications". *Computability in Context: Computation and Logic in the Real World*, 275–296.

Maass, W., P. Joshi, and E. D. Sontag (2006). "Principles of real-time computing with feedback applied to cortical microcircuit models". *Advances in neural information processing systems* 18, 835.

Maass, W. and H. Markram (2004). "On the computational power of circuits of spiking neurons". *Journal of Computer and System Sciences* 69.4, 593–616.

Maass, W., T. Natschläger, and H. Markram (2002). "Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations". *Neural Computation* 14.11, 2531–2560.

— (2004a). "Computational models for generic cortical microcircuits". *Computational neuroscience: A comprehensive approach* 18, 575.

— (2004b). "Fading memory and kernel properties of generic cortical microcircuit models". *Journal of Physiology-Paris*. Decoding and interfacing the brain: from neuronal assemblies to cyborgs 98.4, 315–330.

Maaten, L. v. d. and G. Hinton (2008). "Visualizing Data using t-SNE". *Journal of Machine Learning Research* 9 (Nov), 2579–2605.

Mairal, J. et al. (2009). "Online Dictionary Learning for Sparse Coding". *Proceedings of the 26th Annual International Conference on Machine Learning*. ICML '09. New York, NY, USA: ACM, 689–696.

Manevitz, L. and H. Hazan (2010). "Stability and Topology in Reservoir Computing". *Advances in Soft Computing*. Ed. by G. Sidorov, A. H. Aguirre, and C. A. R. García. Lecture Notes in Computer Science 6438. Springer Berlin Heidelberg, 245–256.

Manjunath, G. and H. Jaeger (2013). "Echo state property linked to an input: Exploring a fundamental characteristic of recurrent neural networks". *Neural computation* 25.3, 671–696.

Maravall, M. and M. E. Diamond (2014). "Algorithms of whisker-mediated touch perception". *Current Opinion in Neurobiology*. Theoretical and computational neuroscience 25, 176–186.

Margrie, T. W. et al. (2003). "Targeted Whole-Cell Recordings in the Mammalian Brain In Vivo". *Neuron* 39.6, 911–918.

Marin, J.-M. et al. (2014). "Relevant Statistics for Bayesian Model Choice". *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 76.5, 833–859.

Marjoram, P. et al. (2003). "Markov Chain Monte Carlo Without Likelihoods". *Proceedings of the National Academy of Sciences* 100.26, 15324–15328.

Markram, H. (2012). "The Human Brain Project". *Scientific American* 306.6, 50–55.

Markram, H. et al. (1997). "Regulation of Synaptic Efficacy by Coincidence of Postsynaptic APs and EPSPs". *Science* 275.5297, 213–215.

Marquardt, D. W. (1963). "An Algorithm for Least-Squares Estimation of Nonlinear Parameters". *Journal of the Society for Industrial and Applied Mathematics* 11.2, 431–441.

Martín-Araguz, A. et al. (2002). "Neuroscience in ancient Egypt and in the school of Alexandria". *Revista De Neurologia* 34.12, 1183–1194.

Martin, R. C. (2013). *Agile Software Development, Principles, Patterns, and Practices*. International ed edition. Harlow: Pearson Education Limited.

Mehring, C. et al. (2003). "Activity dynamics and propagation of synchronous spiking in locally connected random networks". *Biological Cybernetics* 88.5, 395–408.

Meyer, H. S., R. Egger, et al. (2013). "Cellular organization of cortical barrel columns is whisker-specific". *Proceedings of the National Academy of Sciences* 110.47, 19113–19118.

Meyer, H. S., D. Schwarz, et al. (2011). "Inhibitory interneurons in a cortical column form hot zones of inhibition in layers 2 and 5A". *Proceedings of the National Academy of Sciences* 108.40, 16807–16812.

Miller, K. D., D. J. Pinto, and D. J. Simons (2001). "Processing in layer 4 of the neocortical circuit: new insights from visual and somatosensory cortex". *Current Opinion in Neurobiology* 11.4, 488–497.

Milo, R. et al. (2002). "Network Motifs: Simple Building Blocks of Complex Networks". *Science* 298.5594, 824–827.

Mittmann, W. et al. (2011). "Two-photon calcium imaging of evoked activity from L5 somatosensory neurons in vivo". *Nature Neuroscience* 14.8, 1089–1093.

Moral, P. D., A. Doucet, and A. Jasra (2012). "An Adaptive Sequential Monte Carlo Method for Approximate Bayesian Computation". *Statistics and Computing* 22.5, 1009–1020.

Morgan, J. L. and J. W. Lichtman (2013). "Why not connectomics?" *Nature Methods* 10.6, 494–500.

Morita, T. et al. (2011). "Psychometric Curve and Behavioral Strategies for Whisker-Based Texture Discrimination in Rats". *PLOS ONE* 6.6, e20437.

Murthy, V. N. et al. (2001). "Inactivity Produces Increases in Neurotransmitter Release and Synapse Size". *Neuron* 32.4, 673–682.

Newman, M. (2010). *Networks: An Introduction*. Oxford: Oxford University Press. 720 pp.

Nunes, M. A. and D. J. Balding (2010). "On optimal selection of summary statistics for approximate Bayesian computation". *Statistical Applications in Genetics and Molecular Biology* 9, Article34.

Ogawa, S. et al. (1990). "Brain magnetic resonance imaging with contrast dependent on blood oxygenation". *Proceedings of the National Academy of Sciences* 87.24, 9868–9872.

Onnela, J.-P. et al. (2012). "Taxonomies of networks from community structure". *Physical Review E* 86.3, 036104.

Pascanu, R., C. Gulcehre, et al. (2013). "How to Construct Deep Recurrent Neural Networks". *arXiv:1312.6026 [cs, stat]*. arXiv: `1312.6026`.

Pascanu, R. and H. Jaeger (2011). "A neurodynamical model for working memory". *Neural Networks* 24.2, 199–207.

Paugam-Moisy, H., R. Martinez, and S. Bengio (2008). "Delay learning and polychronization for reservoir computing". *Neurocomputing*. Progress in Modeling, Theory, and Application of Computational Intelligenc: 15th European Symposium on Artificial Neural Networks 2007 71.7, 1143–1158.

Perin, R., T. K. Berger, and H. Markram (2011). "A synaptic organizing principle for cortical neuronal groups". *Proceedings of the National Academy of Sciences of the United States of America* 108.13, 5419–5424.

Pocock, S. J. and G. W. Stone (2016). "The Primary Outcome Fails — What Next?" *New England Journal of Medicine* 375.9, 861–870.

Popper, K. R. (1934). *Logik der Forschung*. Tübingen: Mohr Siebeck.

Probst, D. et al. (2012). "Liquid Computing in a Simplified Model of Cortical Layer IV: Learning to Balance a Ball". *Artificial Neural Networks and Machine Learning – ICANN 2012*. Ed. by A. E. P. Villa et al. Lecture Notes in Computer Science 7552. Springer Berlin Heidelberg, 209–216.

Prusiner, S. B. (1982). "Novel Proteinaceous Infectious Particles Cause Scrapie". *Science* 216.4542, 136–144.

Raue, A. et al. (2015). “Data2Dynamics: a Modeling Environment Tailored to Parameter Estimation in Dynamical Systems”. *Bioinformatics* 31.21, 3558–3560.

Robert, C. P. et al. (2011). “Lack of confidence in approximate Bayesian computation model choice”. *Proceedings of the National Academy of Sciences* 108.37, 15112–15117.

Roberts, E. S. and A. C. C. Coolen (2012). “Unbiased degree-preserving randomization of directed binary networks”. *Physical Review E* 85.4, 046103.

Rosenblatt, F. (1961). *Principles of neurodynamics. perceptrons and the theory of brain mechanisms*. DTIC Document.

Rubinov, M. and O. Sporns (2010). “Complex network measures of brain connectivity: Uses and interpretations”. *NeuroImage*. Computational Models of the Brain 52.3, 1059–1069.

Rudemo, M. (1982). “Empirical choice of histograms and kernel density estimators”. *Scandinavian Journal of Statistics*, 65–78.

Sakmann, B. and E. Neher (1984). “Patch clamp techniques for studying ionic channels in excitable membranes”. *Annual review of physiology* 46.1, 455–472.

Salgado-Ugarte, I. H. and M. A. Perez-Hernandez (2003). “Exploring the Use of Variable Bandwidth Kernel Density Estimators”. *Stata Journal* 3.2, 133–147.

Schliebs, S., M. Fiasché, and N. Kasabov (2012). “Constructing Robust Liquid State Machines to Process Highly Variable Data Streams”. *Artificial Neural Networks and Machine Learning – ICANN 2012*. Ed. by A. E. P. Villa et al. Lecture Notes in Computer Science 7552. Springer Berlin Heidelberg, 604–611.

Schmidhuber, J. (1992). “Learning Complex, Extended Sequences Using the Principle of History Compression”. *Neural Computation* 4.2, 234–242.

Schrader, S. et al. (2008). “Detecting Synfire Chain Activity Using Massively Parallel Spike Train Recording”. *Journal of Neurophysiology* 100.4, 2165–2176.

Shi, F. et al. (2013). “Altered Modular Organization of Structural Cortical Networks in Children with Autism”. *PLOS ONE* 8.5, e63131.

Silk, D., S. Filippi, and M. P. H. Stumpf (2013). “Optimizing Threshold-Schedules for Sequential Approximate Bayesian Computation: Applications to Molecular Systems”. *Statistical Applications in Genetics and Molecular Biology* 12.5, 603–618.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. Vol. 26. CRC press.

Singer, W. (2013). “Cortical dynamics revisited”. *Trends in Cognitive Sciences*. Special Issue: The Connectome 17.12, 616–626.

Sisson, S. A., Y. Fan, and M. M. Tanaka (2007). “Sequential Monte Carlo Without Likelihoods”. *Proceedings of the National Academy of Sciences* 104.6, 1760–1765.

Song, S. et al. (2005). “Highly Nonrandom Features of Synaptic Connectivity in Local Cortical Circuits”. *PLOS Biology* 3.3, e68.

Souza, L. G. M. de et al. (2013). "Model Selection and Parameter Estimation for Chemical Reactions Using Global Model Structure". *Computers & Chemical Engineering* 58, 269–277.

Sporns, O. (2006). "Small-world connectivity, motif composition, and complexity of fractal neuronal connections". *Biosystems* 85.1, 55–64.

— (2013). "Making sense of brain network data". *Nature Methods* 10.6, 491–493.

Sporns, O., G. Tononi, and G. M. Edelman (2000). "Theoretical Neuroanatomy: Relating Anatomical and Functional Connectivity in Graphs and Cortical Connection Matrices". *Cerebral Cortex* 10.2, 127–141.

Sporns, O., G. Tononi, and R. Kötter (2005). "The Human Connectome: A Structural Description of the Human Brain". *PLOS Comput Biol* 1.4, e42.

Stone, M. (1974). "Cross-validation and multinomial prediction". *Biometrika* 61.3, 509–515.

Stosiek, C. et al. (2003). "In vivo two-photon calcium imaging of neuronal networks". *Proceedings of the National Academy of Sciences* 100.12, 7319–7324.

Sussillo, D. and O. Barak (2013). "Opening the Black Box: Low-Dimensional Dynamics in High-Dimensional Recurrent Neural Networks". *Neural Computation* 25.3, 626–649.

Sussillo, D. and L. F. Abbott (2009). "Generating Coherent Patterns of Activity from Chaotic Neural Networks". *Neuron* 63.4, 544–557.

Takemura, S.-y. et al. (2013). "A visual motion detection circuit suggested by Drosophila connectomics". *Nature* 500.7461, 175–181.

Team, T. T. D. et al. (2016). "Theano: A Python framework for fast computation of mathematical expressions". *arXiv:1605.02688 [cs]*. arXiv: 1605.02688.

Tibshirani, R. (1996). "Regression Shrinkage and Selection via the Lasso". *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1, 267–288.

Toni, T. and M. P. H. Stumpf (2010). "Simulation-Based Model Selection for Dynamical Systems in Systems and Population Biology". *Bioinformatics* 26.1, 104–110.

Toni, T., D. Welch, et al. (2009). "Approximate Bayesian Computation Scheme for Parameter Inference and Model Selection in Dynamical Systems". *Journal of The Royal Society Interface* 6.31, 187–202.

Trengove, C., C. v. Leeuwen, and M. Diesmann (2012). "High-capacity embedding of synfire chains in a cortical network model". *Journal of Computational Neuroscience* 34.2, 185–209.

Troyer, T. W. et al. (1998). "Contrast-Invariant Orientation Tuning in Cat Visual Cortex: Thalamocortical Input Tuning and Correlation-Based Intracortical Connectivity". *The Journal of Neuroscience* 18.15, 5908–5927.

Turrigiano, G. G. et al. (1998). "Activity-dependent scaling of quantal amplitude in neocortical neurons". *Nature* 391.6670. 892–896.

Varshney, L. R. et al. (2011). "Structural Properties of the Caenorhabditis elegans Neuronal Network". *PLoS Comput Biol* 7.2, e1001066.

Vogelstein, J. T., J. M. Conroy, et al. (2015). "Fast Approximate Quadratic Programming for Graph Matching". *PLOS ONE* 10.4, e0121002.

Vogelstein, J. T. and C. E. Priebe (2015). "Shuffled Graph Classification: Theory and Connectome Applications". *Journal of Classification* 32.1, 3–20.

Vreeswijk, C. v. and H. Sompolinsky (1996). "Chaos in Neuronal Networks with Balanced Excitatory and Inhibitory Activity". *Science* 274.5293, 1724–1726.

Wanner, A. A. et al. (2016). "Dense EM-based reconstruction of the interglomerular projectome in the zebrafish olfactory bulb". *Nature Neuroscience* 19.6, 816–825.

Watts, D. J. and S. H. Strogatz (1998). "Collective dynamics of 'small-world' networks". *Nature* 393.6684, 440–442.

Wernicke, S. (2006). "Efficient Detection of Network Motifs". *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 3.4, 347–359.

Westerhuis, J. A. et al. (2004). "Model Selection and Optimal Sampling in High-Throughput Experimentation". *Analytical Chemistry* 76.11, 3171–3178.

White, H. C., S. A. Boorman, and R. L. Breiger (1976). "Social Structure from Multiple Networks. I. Blockmodels of Roles and Positions". *American Journal of Sociology* 81.4, 730–780.

White, J. G. et al. (1986). "The structure of the nervous system of the nematode Caenorhabditis elegans". *Philos Trans R Soc Lond B Biol Sci* 314.1165, 1–340.

Wilson, M. K., K. Karakasis, and A. M. Oza (2015). "Outcomes and endpoints in trials of cancer treatment: the past, present, and future". *The Lancet Oncology* 16.1, e32–e42.

Wrigley, S. N. and G. J. Brown (1999). *Synfire chains as a neural mechanism for auditory grouping*. Tech. Report CS-99-11, U. Sheffield.

Xue, Y., L. Yang, and S. Haykin (2007). "Decoupled echo state networks with lateral inhibition". *Neural Networks*. Echo State Networks and Liquid State Machines 20.3, 365–376.

Yeung, A. W. K., T. K. Goto, and W. K. Leung (2017). "The Changing Landscape of Neuroscience Research, 2006–2015: A Bibliometric Study". *Frontiers in Neuroscience* 11.

Yildiz, I. B., H. Jaeger, and S. J. Kiebel (2012). "Re-visiting the echo state property". *Neural Networks* 35, 1–9.

Zhang, W. and D. J. Linden (2003). "The other side of the engram: experience-driven changes in neuronal intrinsic excitability". *Nature Reviews Neuroscience* 4.11, 885–900.

Zheng, P., C. Dimitrakakis, and J. Triesch (2013). "Network Self-Organization Explains the Statistics and Dynamics of Synaptic Connection Strengths in Cortex". *PLoS Comput Biol* 9.1, e1002848.

Zheng, P. and J. Triesch (2014). “Robust development of synfire chains from multiple plasticity mechanisms”. *Frontiers in Computational Neuroscience* 8, 66.