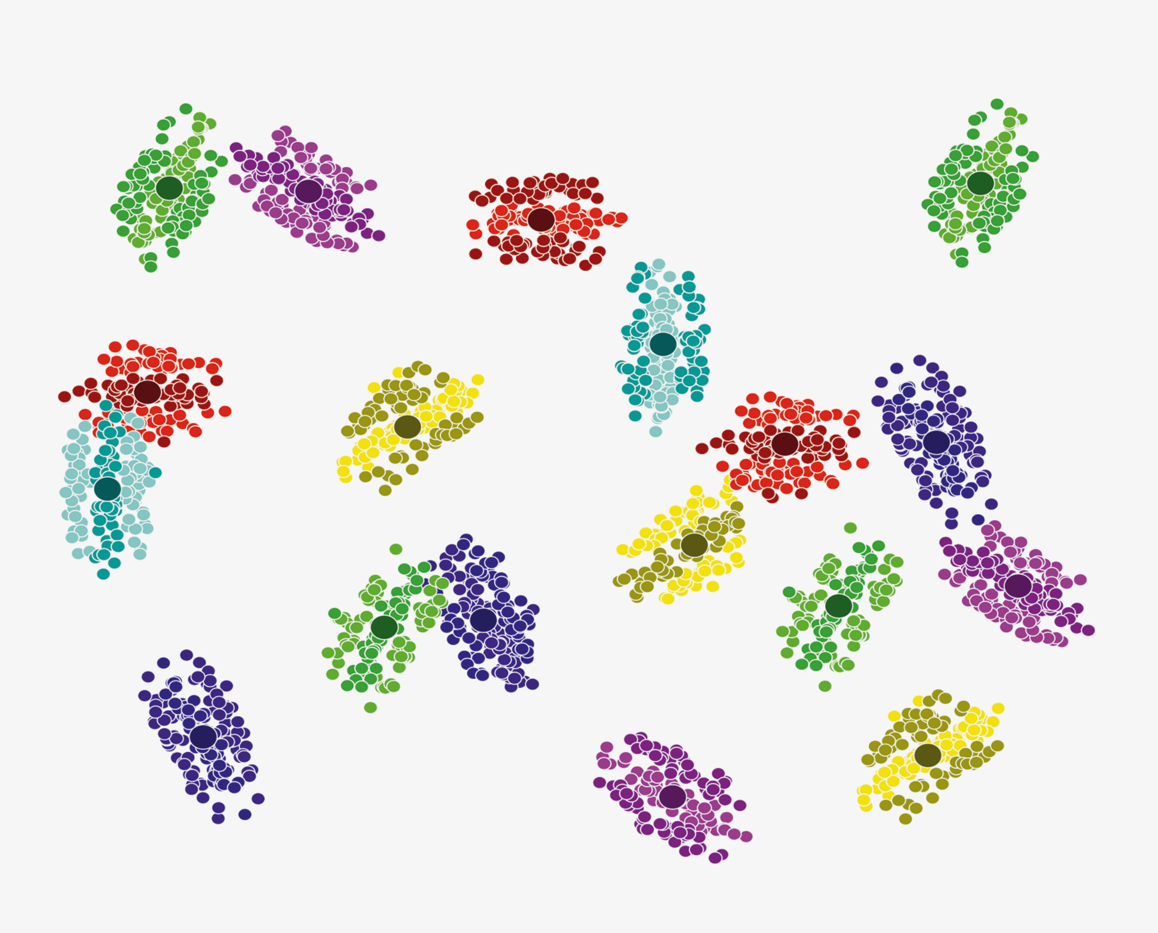


Construction of a Spiking Network Model of Macaque Primary Visual Cortex: Towards Digital Twins

Anno Kurth

Information
Band / Volume 107
ISBN 978-3-95806-800-1

Mitglied der Helmholtz-Gemeinschaft

JÜLICH
Forschungszentrum

Schriften des Forschungszentrums Jülich
Reihe Information / Information Band / Volume 107

Forschungszentrum Jülich GmbH
Institute for Advanced Simulation (IAS)
Computational and Systems Neuroscience (IAS-6)

Construction of a Spiking Network Model of Macaque Primary Visual Cortex: Towards Digital Twins

Anno Kurth

Schriften des Forschungszentrums Jülich
Reihe Information / Information — Band / Volume 107

ISSN 1866-1777 — ISBN 978-3-95806-800-1

Bibliografische Information der Deutschen Nationalbibliothek.
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der
Deutschen Nationalbibliografie; detaillierte Bibliografische Daten
sind im Internet über http://dnb.d-nb.de abrufbar.

Herausgeber und Vertrieb:	Forschungszentrum Jülich GmbH Zentralbibliothek, Verlag 52425 Jülich Tel.: +49 2461 61-5368 Fax: +49 2461 61-6103 zb-publikation@fz-juelich.de www.fz-juelich.de/zb
Umschlaggestaltung:	Grafische Medien, Forschungszentrum Jülich GmbH
Druck:	Grafische Medien, Forschungszentrum Jülich GmbH
Copyright:	Forschungszentrum Jülich 2024

Schriften des Forschungszentrums Jülich
Reihe Information / Information, Band / Volume 107

D 82 (Diss. RWTH Aachen University, 2024)

ISSN 1866-1777
ISBN 978-3-95806-800-1

Vollständig frei verfügbar über das Publikationsportal des Forschungszentrums Jülich (JuSER)
unter www.fz-juelich.de/zb/openaccess.

EIDESSTATTLICHE ERKLÄRUNG

Ich, Anno Christopher Kurth,
erkläre hiermit, dass diese Dissertation und die darin dargelegten Inhalte die eigenen sind und selbstständig, als Ergebnis der eigenen originären Forschung, erarbeitet wurden. Hiermit erkläre ich an Eides statt:

- Diese Arbeit wurde vollständig oder größenteils in der Phase als Doktorand dieser Fakultät und Universität angefertigt;

- Sofern irgendein Bestandteil dieser Dissertation zuvor für einen akademischen Abschluss oder eine andere Qualifikation an dieser oder einer anderen Institution verwendet wurde, wurde dies klar angezeigt;

- Wenn immer andere eigene oder Veröffentlichungen Dritter herangezogen wurden, wurden diese klar benannt;

- Wenn aus anderen eigenen oder Veröffentlichungen Dritter zitiert wurde, wurde stets die Quelle hierfür angegeben. Diese Dissertation ist vollständig meine eigene Arbeit, mit der Ausnahme solcher Zitate;

- Alle wesentlichen Quellen von Unterstützung wurden benannt;

- Wenn immer ein Teil dieser Dissertation auf der Zusammenarbeit mit anderen basiert, wurde von mir klar gekennzeichnet, was von anderen und was von mir selbst erarbeitet wurde.

- Ein Teil oder Teile dieser Arbeit wurden zuvor veröffentlicht und zwar in: *siehe Publications*

Aachen, January 2024

Anno Christopher Kurth

SUMMARY

CONSTRUCTION OF A SPIKING NETWORK MODEL OF MACAQUE PRIMARY VISUAL CORTEX: TOWARDS DIGITAL TWINS The cerebral cortex of the mammalian brain is composed of an unfathomable amount of neurons that are organized in intricate circuits across several spatial scales. If present, cortical activity reflects higher-level information processing in mammals. One approach to study the relationship between the cortex' structure and its activity is to represent the studied physical system by a "digital twin", a computational model in which anatomical and physiological findings can be incorporated. In such digital twins, experiments can be performed and data obtained not feasible using the "physical twin".

This thesis focuses on building a large scale, biologically plausible spiking network model of macaque primary visual cortex. As such, it combines results from the experimental literature and contributes to building ever more sophisticated digital twins of the visual cortex. This quest is embedded into a larger neuroscientific research program aiming at expanding the usage of computer models in Neurosciene. In line with this approach, in this thesis first resting state neural activity recorded from macaque primary visual cortex is analyzed. A separation of neural activity into two clusters that can be related to the monkey's behavior is found that is co-modulated along with top-down signals from V4. To explore whether this co-modulation might be causative for the separation of states, *in silico* experiments of a model of the local cortical circuit are conducted. However, this simple model neglects much of the fine structure of visual cortex. Hence, subsequently a large-scale, biologically plausible digital twin of this area is devised. After unifying and integrating a large body of data across multiple sources, simulations of the model reveal unrealistic activity. This motivates a further investigation of cortical connectivity in light of recent advances of reconstruction of microcircuits in the brain. The findings offer potential resolutions for the encountered problems and highlight stark differences between recent and previous reconstructions of local cortical networks.

To employ digital twins as research platforms in Neuroscience, simulation technologies need to be readily available for the research community. Such technologies have to be continuously developed and updated to meet the requirements of the researchers. To contribute to this endeavor, in this thesis the performance of the neural simulation tool `NEST` is assessed and compared with alternative approaches. Additionally, a benchmarking workflow with a view towards neural network simulations is developed that aids the continuous development of spiking neural network simulation technologies.

ZUSAMMENFASSUNG

KONSTRUKTION EINES PULS-GEKOPPELTEN NETZWERKMODELLS DER PRIMÄREN VISUELLEN RINDE IM MAKAKEN: AUF DEM WEG ZU DIGITALEN ZWILLINGEN Der im Gehirn der Säugetiere vorzufindende *Cortex cerebri* besteht aus einer schier unfassbaren Zahl von in komplexen Netzwerken organisierten Neuronen. Die in ihm zu beobachtende Aktivität ist Anzeichen kognitiver Vorgänge. Ein Ansatz zur Untersuchung der Beziehung zwischen kortikaler Architektur und Aktivität besteht nun darin, das zu untersuchende System in einem "digitalen Zwillingen", d.h. einem Computermodell, abzubilden.

Diese Arbeit konzentriert sich vorrangig auf die Konstruktion eines biologisch plausiblen, puls-gekoppelten Netzwerkmodells der primären visuellen Rinde im Makaken. Damit liefert sie einen Beitrag in Richtung stetig realistisch werdender digitaler Zwillinge. Ein solcher Ansatz muss allerdings auch immer schon in ein weiter gefasstes neurowissenschaftliches Forschungsvorhaben eingebettet sein, dessen Ziel die fortwährend bessere Nutzung von Computermodellen in der Neurowissenschaft ist. In Übereinstimmung mit diesem Ansatz wird in dieser Arbeit die neuronale Aktivität der visuellen Rinde im Makaken im Ruhezustand analysiert. Es wird gezeigt, dass sich besagter Zustand aus zwei Unterzuständen zusammensetzt, die unter anderem mit Signalen aus höheren kortikalen Arealen korreliert sind. Um zu untersuchen, inwiefern diese Signale auch kausal für die Zustandsänderung verantwortlich sein können, werden *in silico* Experimente durchgeführt. Allerdings vernachlässigt das dafür genutzte Modell wichtige Aspekte der komplexen Struktur der Netzwerke in der visuellen Rinde. Daher wird im Anschluss ein biologisch plausibler digitaler Zwilling dieses Areals erstellt. Simulationen dieses Netzwerks zeigen jedoch physiologisch unrealistische Aktivität, was —gerade vor dem Hintergrund neuer experimenteller Möglichkeiten —eine tiefer gehende Analyse lokaler kortikaler Netzwerke motiviert. Die hierbei gefundenen Resultate bieten Verbesserungen der Schätzungen kortikaler Konnektivität und weisen einen Weg, die im Modell der visuellen Rinde gefunden Probleme zu beheben.

Um die zuvor beschriebenen digitalen Zwillinge auch tatsächlich in neurowissenschaftlicher Forschung nutzen zu können, bedarf es passender Simulationstechnologien, die stetig weiterentwickelt und verbessert werden müssen. In diesem Kontext wenden wir uns schließlich Messungen der Leistungsfähigkeit des NEST Simulators zu. Im Anschluss wird ein Benchmarking Framework mit besonderem Blick auf Simulationstechnologien für puls-gekoppelte Netzwerke entwickelt, das die kontinuierliche Weiterentwicklung ebenjener unterstützen soll.

PUBLICATIONS

The work presented in this thesis is in part based on the following publications:

Neural manifolds in V1 change with top-down signals from V4 targeting the foveal region
Aitor Morales-Gregorio, Anno C. Kurth, Junji Ito, Alexander Kleinjohann, Frédéric V. Barthélemy, Thomas Brochier, Sonja Grün, and Sacha J. van Albada
Published on bioRxiv: Morales-Gregorio et al. (2023)
Currently under review in Cell Reports.
Results of this publication enter Chapter 4 and Chapter 6

Sub-realtime simulation of a neruonal network of natural density
Anno C. Kurth, Johanna Senk, Dennis Terhorst, Justin Finnerty, and Markus Diesmann
Published in Neuromorphic Computing and Engineering: Kurth et al. (2022)
Results of this publication enter Chapter 10

A modular workflow for performance benchmarking of neuronal network simulations
Jasper Albers, Jari Pronold, Anno C. Kurth, Stine Brekke Vennemo, Kaveh Haghighi Mood, Alexander Patronis, Dennis Terhorst, Jakob Jordan, Susanne Kunkel, Tom Tetzlaff, Markus Diesmann, and Johanna Senk
Published in Frontiers in Neuroinformatics: Albers et al. (2022)
Results of this publication enter Chapter 11

Author contributions are stated at the beginning of each chapter.

ACKNOWLEDGMENTS

I would to like thank my supervisors, Prof. Dr. Sacha van Albada and Prof. Dr. Markus Diesmann, for introducing me to Computational Neuroscience and offering me the opportunity to pursue research in this exciting field. I am very grateful for their as well as Dr. Johanna Senk's constant, ongoing and thorough support in any sense of the word throughout the making of this thesis.

More broadly, I want to thank all group leaders of the IAS-6 at the Forschungszentrum Jülich and especially its directors, Prof. Dr. Sonja Grün and Prof. Dr. Markus Diesmann, for creating an atmosphere in which Science as well as companionship could flourish equally. This immeasurably enriched my time at the IAS-6 and drove my academic as well as personal growth.

I am moreover grateful to ...

... the secretariat of the IAS-6 for the continued effort in oiling the gears of the institute.

... the technical support team at the IAS-6 for the ongoing help with our equipment. Without our high-performance machines large parts of my research would not have been possible.

... the scientific coordination team at the IAS-6 for their perpetual help in matters regarding the more bureaucratic part of the scientific work. Especially, I want to thank Martina Reske for her help in various situations.

... my fellow doctoral researchers as well as the Master and Bachelor Students at the IAS-6 for lively discussion about and beyond Science, and the great help and solidarity among each other which lie at the heart of the experience of pursuing a PhD at this institute.

Especially, I want to thank my current and former colleagues Alexander van Meegen, Aitor Morales Gregorio, Jasper Albers, Jonas Oberste-Frielinghaus, Alessandra Stella, Peter Bouss, Simon Essink, Alexander Kleinjohann, Michael Dick, Kirsten Fischer, Joschua Böttcher and Claudia Merger whom I had the pleasure to either work with or to discuss with and to learn from.

Additionally, I want to thank Sacha van Albada, Aitor Morales Gregorio, Jonas Oberste-Frielinghaus, Michael Dick and Johanna Senk for reading earlier versions of this thesis.

The author thanks the Studienstifung des deutschen Volkes for financial support.

CONTENTS

VI APPENDIX

LIST OF FIGURES

LIST OF TABLES

Part I

INTRODUCTION

1 BACKGROUND

The *cerebral cortex* —or *cortex* for short —is the roughly 1 mm to 3 mm thick outer shell of the human or more general mammalian forebrain (Abeles, 1991). It consists of approximately 100 billion interconnected cells which have an average indegree of about one to ten thousand synapses, where the precise numbers vary across species and single brains (ibid.).
Human neocortex is, among other structures, primarily associated with and responsible for processing sensory information, planning and executing motor commands. It can be subdivided into more than 52 different cortical areas on the basis of differences in cell types as well as their densities (see e.g. Brodmann (1909) for a scheme that is still widely used and refined to this day). Moreover, these anatomically defined areas have been recognised to serve *functionally* different purposes. They can be thought of as specific "modules" oftentimes capable of controlling specific brain functions (Kandel et al., 2013). The interactions of these modules which are mediated by their interconnections, organised into multiple parallel pathways, underlie brain function as a whole in humans and other mammals. Despite tremendous effort and, indeed, progress over the last century, the precise relationship between the structure of the cortex and its function viewed across all scales remains a paramount problem in modern-day Neuroscience.

Since the days of the Spanish Neuroscientist Ramon y Cajal, the classical approach of neuroscientific research is closely tied to his famous paradigm, the *neuron doctrine* (Ramón y Cajal, 1888) (see Yuste (2015) for a history of the neuron doctrine with a view towards neural networks). It asserts that the nervous system is composed of an unfathomable number of discrete, individual, but highly interconnected functional units, the neurons. Cajal's paradigm broke the soil for various approaches to Neuroscience: anatomical questions regarding the cellular composition and organisation of the cortex could be addressed by e.g. histological or other means (e.g. Braitenberg and Schüz (1991), Brodmann (1909), and Markov et al. (2014a)). Hypotheses regarding electrochemical properties and the inner workings of single neural cells could be formulated. Investigations following this line of thought led to, e.g., the discovery of the *action potential* (Bernstein, 1868; Hodgkin and Huxley, 1952) —a large, stereotyped deviation in the membrane potential of a single neuron relative to its *resting potential*, propagating a signal in a self-sustained way from the cell body along its *axon* to its *synapses*. There, connections to other neurons are established. The short temporal extent of the action potential in comparison to the intrinsic time scale of the dynamics of the membrane potential motivates to conceptually abstract from the biophysical mechanism of its generation. Instead, one might focus on its timing as a binary event. As such, it is referred to as a *spike*. The binary signal defined as the collection of these events as a whole is called a *spike train*.

More involved, but still classical questions addressed within the neuron doctrine include response properties of single neurons assessed by the generated spike trains to a wide class of different stimuli, ranging from artificial step currents injected directly into the cell to naturalistic stimuli perceived by test animals. These investigations ultimately led to the concept of the *receptive field* of a neuron, the domain in sensory space that needs to be covered by a matching (with respect to the domain) stimulus to elicit a response in a neuron. This notion was initially conceived by Sherrington (1906) to describe the part of skin that needs to be irritated to elicit a certain reflex, and afterwards used by Hartline (1941) for studying non-cortical neurons involved in vision. Later, receptive fields were also discovered in cortical neurons in the visual system by Hubel and Wiesel (1959, 1962). Subsequently, receptive fields have become a central concept of sensory Neuroscience.

Although highly successful and foundational for the entire field, this ansatz suffered from the complication that the discharge pattern of a single neuron varies from trial to trial, even when identical stimuli are presented. This phenomenon is known as *neuronal variability* (Festa et al., 2021). Groundbreaking conclusions about the statistical nature of neural activity were drawn, for example regarding the encoding of sensory information therein. *Rate coding*, based on the *firing rate* of single neurons (the number of spikes in a given unit of time) over prolonged periods of time needs to be mentioned here as a prominent coding scheme. Again, it was first discovered in extra-cortical neurons (Adrian, 1926).

On the other hand, *temporal neural codes* —the notion that some stimuli are not only encoded in the strength of the response of neurons but also in the temporal relationship between their spikes—emerged as an alternative to the more classical view of rate coding: neurons can produce different responses based on the temporal pattern of incoming signals (see e.g. Segundo et al. (1963)). This suggested that not necessarily all functions performed by the nervous system might be simply reduced to a rate-based input-output relation of the constituting neurons. It became apparent that one should also take into account the *interactions* of the neurons on a larger scale to gain a deeper understanding of brain functions. One needs to view the connections not as mere cables (as for instance Cajal did) but as providing the structure for and thus taking part in and shaping the cortical computation itself. The concept of *neural circuits*, which can be traced back to de No

(1933), offers an appealing view on this issue: neural functions are emergent properties arising from complex interactions of neurons. Viewing neurons as atoms of neural computations, function is here not merely determined by the behavior of individual neurons; more importantly, it is shaped by their interconnections, which underlie their interaction.

In this spirit, Hebb (1949) introduced the notion of cell or *neuronal assemblies*. He suggested that cortical neurons organise dynamically into functional units being capable of performing certain computations, e.g. distributed encoding of information. In this theory, the same neuron can be part of several assemblies. Membership of an assembly might on the one hand be determined by a neuron's anatomical connections, which are formed and enforced by *synaptic plasticity*. This refers to the ability of neural circuits to change over time in order to adapt to new situations. For this, Hebb (1949) suggested a mechanism that is mediated by the correlations of the neural activity and that can be summarized in the statement "neurons wire together if they fire together" (Löwel and Singer, 1992). Additionally, membership could also depend on the current dynamical state of the neural network, due to a change in the networks *effective connectivity* (Aertsen and Preißl, 1990). For the first time, Hebb's theory shed light on the immense computational power inherent to neural circuits stemming from neurons interacting in a complex network.

Technological progress in the following decades also improved neuroscientific equipment, leading to the micro-electrode arrays[1] or laminar probes[2] of these days. They allow for the parallel recording of multiple cortical neurons in behaving animals. Additionally, advancing anatomical techniques enabled the study of neural connections more effectively and systematically (from axonal tracing with horseradish peroxidase (Kristensson and Olsson, 1971) to the usage of electron microscopy (Kubota et al., 2018)) revealing statistical patterns in the connectivity of different cortical areas (e.g. Braitenberg and Schüz (1991)). Insights stemming from the improved technologies supported the idea of distributed computations carried out by neural circuits in different brain areas. They, however, also highlighted the apparent randomness of specific neuron-to-neuron connections.

To effectively deal with the increasing amount and complexity of data, another ansatz allowing for a different, mathematicallly more rigorous, and quantitative study of the nervous system was needed that could support and extend the already existing ones. Computational Neuroscience offers the desired ansatz and a complementary approach to experimental Neuroscience. Due to its methods taken mainly from Mathematics, Computer Science, and Physics, it allows for quantita-

[1] For example the Utah Array by Blackrock Neurotech https://blackrockneurotech.com/products/utah-array

[2] For example Neuropixels by IMEC https://www.neuropixels.org

tive as well as rigorous qualitative theoretical studies of the nervous system.
Examples of the research conducted in this field include detailed mathematical models of single-neuron dynamics (starting from Hodgkin and Huxley (1952)), advanced methods for analysing the growing amount of neural data recorded in paralllel (e.g. for finding low-dimensional structures in neural data (Yu et al., 2009), identifying context-dependent, precise spike synchronization (Riehle et al., 1997) or identifying spatio-temporal patterns in spiking activity (Quaglio et al., 2017)), models of associative memory (e.g. Hopfield networks first studied by Hopfield (1982) and their beautiful analysis using techniques from Statistical Mechanics by Amit et al. (1985)) as well as explanations of certain dynamical features of cortical activity (e.g. the asynchronous irregular nature of the resting-state in inhibition-dominated recurrent spiking neural networks Brunel (2000)).
Thus, Computational Neuroscience breaks the soil for the application and extension of methods from a wide class of different scientific fields ranging from Dynamical Systems Theory (Izhikevich, 2007) and Statistical Field Theory (Helias and Dahmen, 2020) to Information Theory (Rieke et al., 1997) and modern Statistics (Grün and Rotter, 2010).

On a parallel strand of Computational Neuroscience, computer simulations enabled a different approach to neuroscientific research: On the one hand, simulations of models of neural populations collapsing entire neural circuits to a small number of equations shed light on e.g. macroscopic signal propagation in the visual cortex (Mejias et al., 2016) or the mesoscopic relationship between structure and function in the saccade generator in the reticular formation (Gancarz and Grossberg, 1998). These models, however, neglect the possibly rich and intricate dynamics on the single-neuron level. On the other hand, growing computational power and advances in simulation technology (e.g. Gewaltig and Diesmann (2007) and Goodman and Brette (2013)) allow for simulations of biologically realistic network with microscopic dynamics of ever increasing size. Here, the granularity of the nervous system can be taken into account at different levels of detail. For example, Markram et al. (2015) payed attention to the detailed morphology of single neurons. In Billeh et al. (2020), Izhikevich and Edelman (2008), and Traub et al. (2005) the authors employed abstracted morphological features of neurons instead. Additionally, hybrid approaches modeling some neuron types in great detail while reducing others to more abstract representations have been used (Dura-Bernal et al., 2023). Finally, some approaches refrain from trying to incorporate the spatial extent of neurons altogether and use *point neuron models*, allowing to shift the focus to larger networks (Antolík et al., 2018; Billeh et al., 2020; Potjans and Diesmann, 2014; Schmidt et al., 2018a; Senk et al., 2023).

This large-scale modelling approach to Neuroscience has, however, been criticised (see e.g. Eliasmith and Trujillo (2014) and Frégnac (2017)). The main points raised by its critics often concern the issue that the right level of description of the nervous system is to this day not known. Hence, the biologically realistic, bottom-up approach to modelling the brain might miss essential features, and overemphasize unimportant ones. Moreover, the statistical nature of the parameters used to construct these models does not allow for meaningful cortical computations to take place that are ultimately reflected in some meaningful behavior. This behavior, however, determines the success of an organism and thus evolution optimized brains to perform computations leading to advantageous behavior. Therefore, Eliasmith and Trujillo (2014) advocate for a top-down approach —that is, starting from the functional (meaning here: important for behavior) results of the cortical computation —, taking into account the vast body of knowledge from Behavioral Neuroscience. A possible execution of this is exemplified by Eliasmith et al. (2012). This work and the paths leading to it (e.g Eliasmith and Anderson (2004)) are clearly interesting and insightful. But their arguments do not refute the relevance of the bottom-up approach.

In bottom-up modelling, the question of complex neural functions and how they emerge is postponed, and the quantities of interest rather concern the dynamics of neural circuits. Focusing on the cerebral cortex again, clearly, meaningful computation, whatever this precisely means and how to identify it, is not expected to appear out of thin air. Yet, eventually the dynamical activity of neural tissue mediates the function of the brain. Since not only the function but also even more fundamentally the *cortical dynamics* is far from being understood, the bottom-up approach might reveal new insights: it might help to shed light on the structure-function relationship via the structure-dynamics relationship, eventually embedded into the structure-dynamics-function triad.

The relevance of bottom-up modelling is further supported by the possibility of incorporating parts of the immense amount of knowledge accumulated by Neuroscientists over the last one and a half centuries. Ever expanding insights regarding the anatomy and electrophysiology of the cerebral cortex in combination with novel techniques like predictive connectomics or electron microscopy may be used to refine the connectomes used in the generation of large-scale models (e.g. Binzegger et al. (2004), Schmidt et al. (2018b), and Shapson-Coe et al. (2021)). Electrophysiological results may constrain parameters of a chosen neuron model (as by Billeh et al. (2020)) and hypotheses from molecular biology on the effects of certain neurotransmitters can be systematically explored in network simulations. Additionally, the realistic number of neurons and synapses used in these models are necessary to faithfully represent the correlation structure of neural activity in network simulations (van Albada et al., 2015a).

Finally, experimental data can be used to not only gauge large-scale models of the brain, but also validate them. This ultimately may lead to new theories about cortical dynamics and function that can be generated via these models and tested experimentally, closing the loop between theory and experiment.

2 SCOPE AND STRUCTURE OF THIS THESIS

The topic of this thesis is the construction and simulation of a biologically plausible, spatially organized spiking neural network model of the macaque primary visual cortex. As such, it extends previous approaches by e.g. Potjans and Diesmann (2014) and Senk et al. (2023). In the former the spatial structure of cortical tissue was neglected and the focus was put onto inter-laminar connections in local microcircuits. In contrast to the generic cortical circuit studied there, the primary visual cortex exhibits a rich spatial structure across different length-scales. The second study generalized the model of Potjans and Diesmann (2014) to a spatially structured network and investigated spatial propagation of and extracellular potentials derived from the spiking activity. Here, we extend the work of Senk et al. (2023) by deriving a more detailed model for spatial decay of connection probabilities and additionally endowing the network with spatially organized structural features that are found in the intra-area connectivity of the macaque primary visual cortex and are believed to underlie cortical computation in that area.

Thus, the presented work continues the path of designing ever more realistic *digital twins* of cortical tissue. "Digital twin" here refers to an *in silico* representation of a physical system (also referred to as *physical model* or *physical twin*), aiming at incorporating the static and dynamic structure of said system into a computational model. Such a model allows for investigating properties of the original system in ways and under conditions not possible in the physical one. The term digital twin hence is used here in a general sense, in contrast to the more stricter usage in for example Clinical Neuroscience where the term usualy refers to personalized representations[1].

These models can be viewed as platforms for Neuroscience, in which prior and ever expanding scientific knowledge regarding anatomy and physiology is integrated into coherent frameworks unifying these results in a quantitative manner (Billeh et al., 2020; Markram et al., 2015; Potjans and Diesmann, 2014; Schmidt et al., 2018b; Senk et al., 2023). Digital twins can thus be regarded as a tool for modern day Neuroscience: they force us to make assumptions regarding the structure and working of the nervous system explicit as well as quantifiable, and ultimately provide a possibility for studying the dynamics and function of neural system in complementary ways to classical approaches in the field. These possibilities range from recording all observables to

[1] see e.g. https://www.neurotwin.eu/ where personalized "Neurotwins" offer optimization of brain stimulations in patients

constructing counterfactual scenarios. Moreover, this line of thought suggests a more holistic view on Neuroscience that should be embedded into a broader *neuroscientific research program*. For such a research program, we here suggest that the following three perspectives need to be taken into account:
First, identifying hypotheses about brain dynamics or empirical neural phenomena that lack explanations by an underlying mechanism. A by now classical example is the question of how signals, encoded in the simultaneous activity of a large number of neurons, propagate in the brain. Griffith (1963) suggested a mechanism that can be roughly described as a feed-forward network of excitatory neurons, later named *synfire chains* (Abeles, 1982). Note that to this day, the evidence for synfire chains is weak. This is due to the technical limitations in recording the activity of single-neurons, leading to severe subsampling of the cortical activity.
Secondly, appropriate models of neural tissue —in this thesis we will focus on spiking neural networks —that allow for an investigation of the identified effects need to be constructed. Here, it is of high relevance to take into account the constraints set by the anatomy and physiology of the nervous system. Turning again to the example of synfire chains, numerical simulations showed that stable propagation of synchronous activity in spiking neural networks is indeed possible under plausible conditions and for a wide range of the relevant parameters (Diesmann et al., 1999). Thus, the simulation of appropriately constructed, biologically plausible spiking neural networks provided a testbed for the hypothesis, and indeed gave credibility to synfire chains as a mechanism employed by the brain.
Finally, providing reliable simulation technology to instantiate and simulate the neuronal network models (Einevoll et al., 2019). Clearly, it is not sustainable if every researcher or research group develops their own research software: not only is the process labor intensive and prone to errors; it also requires expertise that is not part of the standardized education of most researchers. This expertise includes for example systematic testing of software and the assessment of performance in ever changing hardware and software ecosystems. The `NEST` simulator (Gewaltig and Diesmann, 2007) is an example for such a simulation technology. Started in 1994 under the name of `SYNOD`, the `NEST` simulator has been continuously developed ever since and is able to efficiently use *High-Performance Computing* (HPC) facilities (Jordan et al., 2018; Morrison et al., 2003). This allows for the simulation of biologically plausible spiking neural network models in a reliable, efficient and scalable way.
All of the three aforementioned perspectives are integrated into this thesis. In the remainder of this chapter, we briefly outline the subsequent content and their relation to these perspectives.
In Part ii, we first turn to the analysis of extracellular multi-electrode

recordings of macaque primary visual cortex (V1), see Chapter 4. We focus on the *resting-state*, i.e. the non input-driven state of the cortical activity. In experimental studies of the visual cortex, the resting-state is often neglected and the attention is focused on stimulus driven or task evoked neural activity, allowing for investigations of visual processing. However, the resting-state in itself already can have a rich dynamical structure, cf. Dąbrowska et al. (2021). We here report for the first time two distinguishable "sub-states" in the V1 resting-state activity of macaque that are closely correlated with eye-closure. Our analysis moreover reveals a change in the top-down modulation from V4 —a cortical area higher in the visual hierarchy —to V1 coinciding with the change of the sub-states. This suggests the observed top-down modulation to be a potential mechanism underlying the change in sub-state.

Next, Part iii focuses on the construction and simulation of biologically plausible spiking neural network models. First, in Chapter 6, the suggested mechanism inducing a change in the resting-state activity of V1 is investigated in a minimal spiking model of cortical networks. We find that our hypothesis is consistent with numerical simulations. However, it needs to be noted that this minimal model neglects prominent structural features of cortical anatomy. We hence turn to the construction of a more detailed model —a digital twin in our usage of the term —of macaque primary visual cortex in Chapter 7. To this end, we derive a new network model taking into account the layered as well as spatial structure of V1 in a data-driven way. Specific structural features added to the network model include patchy as well as push-pull connectivity. Numerical simulations reveal that the obtained connectivity does not allow for suitable spiking activity in the network. An analysis of the network connectivity suggests that this can be traced back to one of the connectomic studies used in our derivation. In the next step, we thus try to mitigate this problem by constructing microcircuit models of cortical circuits based on more recent anatomical data obtained with electron microscopy (EM), see Chapter 8. Indeed, there are stark differences in the architecture of the derived models of local cortical circuits. We find connectivity structures which suggest that the issues observed in the simulations of the detailed model of macaque V1 in Chapter 7 might be overcome.

Finally, in Part iv, we turn to the simulation technology on which the aforementioned *in-silico* experiments rely. To investigate neuronal networks as well as phenomena of ever-growing complexity —may it be networks comprising more neurons and synapses or networks including complicated properties like *synaptic plasticity* —simulations need to get faster. New technologies as Neuromorphic Computing offer a speedup, but must be validated in their functionality and usability against more classical approaches. In Chapter 10 we conduct performance benchmarks of a model microcircuit for NEST and show that on state-of-the-art computer hardware, classical HPC can compete with more specialized approaches. Moreover, we point to bottlenecks in current systems that might pave the way for future improvements. For keeping up with alternative technologies and satisfying the needs of the research community, NEST needs to be continuously developed. Performance benchmarks are an integral part of this process. In order to reproducibly perform these benchmarks across machines, we propose a benchmarking workflow for HPC applications in Chapter 11. Here, the objective is to first derive a conceptual workflow where the necessary components are identified and specified. After this, a reference implementation with a view towards the simulation of spiking neural network is provided and we demonstrate how to use such a workflow in the development process of NEST.

Part II

ANALYSIS OF RESTING STATE ACTIVITY

3 PREFACE

In the last decades, the analysis of resting-state activity has become a standard paradigm in Neuroscience, mainly conducted with brain-wide data, e.g. from functional magnetic resonance imaging (fMRI) (Snyder and Raichle, 2012). The resting-state is defined as a dynamical state in which the activity of the brain is not driven by external stimuli or task-related events (ibid.). It has been suggested that resting-state activity thus may reflect the underlying structural architecture of the brain (Deco et al., 2011).

Additionally, the resting-state has also been explored at the level of single brain areas, e.g. in multi-electrode recordings (Dąbrowska et al., 2021). Such analyses are of great importance for the research program sketched in Chapter 2, since they allow for gauging and validating digital twins of cortical circuits.

Here, we set out to characterize resting-state activity of the primary visual cortex (area V1) in macaque. Our analysis presented in Chapter 4 reveals a previously unknown change of the cortical activity in the resting-state in extracellular recordings of V1. Further analyses regarding the interaction between the primary visual cortex and higher visual areas suggest that top-down modulations might play a role in this change. We thus describe a dynamical phenomenon that can be subsequently investigated *in silico* with numerical simulations of cortical networks.

4

NEURAL MANIFOLDS IN V1 CHANGE WITH TOP-DOWN SIGNALS FROM V4

The chapter is based on the following publication:

Morales-Gregorio, A., Kurth, A. C., Ito, J., Kleinjohann, A., Barthélemy , F. V., Brochier, T., Grün, S., & van Albada, S. J. (2023). Neural manifolds in V1 change with top-down signals from V4 targeting the foveal region. bioRxiv, doi: 10.1101/2023.06.14.544966

Author contributions:
Under the supervision of SvA, SG, and JI, the author and AMG conceptualized the work. AMG curated and processed the data from macaques L and A. AKJ curated and processed the data from macaque Y, recorded by TB and FB. The author and AMG performed the dimensionality analysis, statistical testing, and Granger causality analysis. These were refined through discussions between the author, AMG, JI, SG, and SvA. All contributed to the preprint.

4.1 INTRODUCTION

Massively parallel recordings allow interpreting neural activity as a trajectory in a high-dimensional space. In this interpretation, each electrode recording the extracellular potential or each identified spiking neuron represents one dimension. The time-resolved brain activity thus can be described as a high-dimensional dynamical system. In this view, the representation and processing of lower-dimensional variables can be investigated by studying the time evolution of the population dynamics in the high-dimensional space.

The high-dimensional space in which the recorded activity unfolds is called the state space. Usually, not all possible states in this space are attained. Rather, the dynamics is confined to —often times geometrically lower-dimensional —subsets, also referred to as neural manifolds (Chaudhuri et al., 2019; Feulner and Clopath, 2021; Gallego et al., 2017, 2018; Gao et al., 2017). In recent years, these manifolds have gained substantial attention across different brain areas and animals: Mante et al. (2013) showed that they encode decision making in the pre-frontal cortex in macaque, Churchland et al. (2012) and Gallego et al. (2017, 2018) related hand movement trajectory in the macaque motor cortex to neural manifolds, Chaudhuri et al. (2019) found that they encode head direction in the anterodorsal thalamic nucleus of mice, Gardner et al. (2022) similarly showed that they encode spatial position in the hippocampus of mice. Other studies investigated neural manifolds in visual cortex in mice (Froudarakis et al., 2020; Stringer et al., 2019a) and macaque (Singh et al., 2008).

A complementary, more global view on neural dynamics highlights cortical states occurring at different times, also referred to as dynamical states (Harris and Thiele, 2011). These are periods that be can be characterized and distinguished by their fluctuations in the spontaneous population activity. This view on brain dynamics asserts that different states not only reflect sensory input, but also endogenous interactions within local circuits or brain areas, and can be related to higher-level effects like selective attention (ibid.). The dynamical state of the activity in the primary visual cortex is influenced by the hierarchical organization of the visual system. Information travels from lower to higher areas (bottom-up) and vice versa (top-down), often within specific frequency bands (Bastos et al. (2015), Klink et al. (2017), and van Kerkoerle et al. (2014), see also Section 7.1) Top-down signals from V4 to V1 have been shown to direct visual attention for figure-ground segregation and mediate contour integration in macaque (Liang et al., 2017; Poort et al., 2012; Self et al., 2013; van Kerkoerle et al., 2017).

Theoretical studies showed that top-down signals in principle can modulate neural manifolds and their intrinsic properties (Dahmen et al., 2020; Naumann et al., 2022). As effective connectivity is input-dependent (Aertsen and Preißl, 1990), a change in top-down input may therefore also affect the dimensionality of neural activity (Avitan and Stringer, 2022; Cunningham and Byron, 2014; Gao and Ganguli, 2015; Gao et al., 2017; Mazzucato et al., 2016; Pang et al., 2016; Stringer et al., 2019a; Williamson et al., 2016). However, whether top-down signals modulate the neural manifold geometry and their dimensionality *in vivo* has yet to be shown.

Here, we study the state space of the primary visual cortex of the macaque (N=3) during the resting-state and its relation to the top-down signals from higher visual areas (V4, DP). To this end, we first employ tools from computational topology (c.f. Chaudhuri et al. (2019), Ghrist (2008), and Singh et al. (2008)) and find that the population activity of macaque V1 is organized into two distinct high-dimensional neural manifolds. These are correlated with the behaviour (eye closure) of the macaques, but not related to any external visual stimuli. Additionally, we investigate the dimensionality of the neural activity. In the observed manifolds, the dimensionality differs significantly, with higher dimensionality found during the eyes-open periods. Moreover, we estimate top-down interaction from V4 via spectral Granger causality. We find that these interactions are significantly stronger during eyes-open periods. Thus, our data analysis suggests that top-down signals could actively modulate the V1 population activity, leading to two distinct neural manifolds in macaque visual cortical resting-state activity.

4.2 METHODS

To explore the activity in the visual cortex, the intracortical electrical potential from the visual cortex of three rhesus macaques (*Macaca mulatta*) was recorded. The experimental setup and data processing steps are illustrated in Figure 4.1. The experiments simultaneously recorded the activity from V1 and V4 (macaques L & A, Chen et al. (2022)) and from V1 and DP (macaque Y, de Haan et al. (2018), see Figure 4.1b). The activity was recorded in the resting-state: the macaques sat head-fixed in a dark room, no instruction to perform a particular task was provided. In this condition the macaques often showed signs of somnolence and kept their eyes closed for periods of variable duration. The right eye—contralateral to the site of neural recording—was tracked using an infrared camera, allowing the identification of periods of open or closed eyes.

4.2.1 *Electrophysiological data from macaques*

Table 4.1 provides an overview of sessions, duration and number of electrodes per subject used in the different monkeys and experiments. For monkeys L and A (Figure 4.1b left), we use publicly available

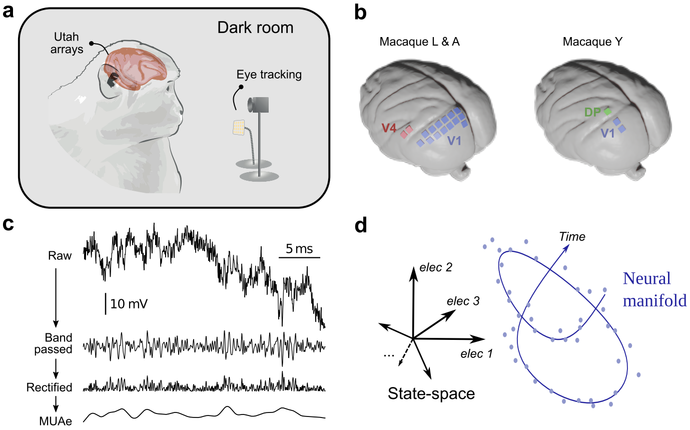


Figure 4.1: Overview of the experiment and neural manifold. **a** Illustration of the experimental setup. **b** Approximate locations of array implants in both experiments. Exact placement of the arrays differs slightly between subjects L and A. **c** Steps for obtaining the multi-unit activity envelope (MUAe) (Supèr and Roelfsema, 2005) used in this chapter. Band-pass filtering: between 500 Hz and 9 kHz. Rectified signal is low-passed at 200 Hz resulting in the MUAe. **d** Schematic representation of state space and a neural manifold. Note that time is implicit within the neural manifold.

(Chen et al., 2022) neural activity recorded from the neocortex of rhesus macaques (N=2) during rest and a visual task. The macaques were implanted with 16 Utah arrays (Blackrock microsystems), two of them in visual area V4 and the rest in the primary visual cortex (V1), with a total of 1024 electrodes. The recording system recorded the electric potential at each electrode with a sampling rate of 30 kHz. The electrodes had a length of 1.5 mm. This implies that the electric potential is recorded in the deeper layers, likely layer 5. A full description of the experimental setup, the data collection and preprocessing has already been published (Chen et al., 2022); here we only provide the details relevant to this chapter.

Three resting-state (RS) sessions were recorded per macaque. During the recordings, the macaque's head was held in position with a custom-made headstage. Pupil position and diameter data were collected to determine the direction of gaze and eye closure of the macaques. A visual response task was performed on the same day as the RS recording sessions. The recorded response data were used to calculate the signal-to-noise ratio (SNR) of each electrode. All electrodes with an SNR lower than 2 were excluded from further analysis. Additionally, up to 100 electrodes that contributed to high-frequency cross-talk in each session were removed. For more details we refer to the original data publication.

The raw neural data were processed to obtain the multi-unit activ-

ity envelope (MUAe) signal as well as local field potential (LFP). To calculate the MUAe, the raw data were high-pass filtered at 500 kHz, rectified, low-pass filtered at 200 kHz, and downsampled to 1 kHz. Finally, the 50, 100, and 150 kHz components were removed with a band-stop filter to remove the European electric grid noise and its main harmonics. To obtain the LFP data, the raw data was low-pass filtered at 250 Hz, downsampled to 500 Hz and a band-stop filter was applied to remove the European electric grid noise 50, 100, and 150 Hz.
The raw data from one session (L_RS_250717) were spike-sorted using a semi-automatic workflow with `Spyking Circus`[1] (Yger et al., 2018). Roughly, Spyking Circus first applied a band-pass filter to the raw signals between 250 Hz and 5 kHz. Next, the median signal across all 128 channels that shared the same reference (2 Utah arrays) was calculated and subtracted, in order to reduce cross-talk and movement artefacts. The spike threshold was set conservatively, at eight times the standard deviation of each signal. After filtering and thresholding, the resulting multi-unit spike trains were whitened—removing the covariance from periods without spikes to reduce noise and spurious spatio-temporal correlations. After whitening, a subsample of all spike waveforms is selected, reduced to the first five principal components, and clustered into different groups with the k-medians method. Finally, all spikes in each electrode are assigned to one of the waveform clusters based on a template fitting algorithm, which can also resolve overlapping waveforms.

After the automatic sorting, the waveform clusters were manually merged and labelled as single-unit activity, multi-unit activity, or noise. Only single-unit activity (SUA) spike trains were included in this chapter. The waveform signal-to-noise ratio (wfSNR) was calculated for all SUA, and those with a wfSNR < 2 or electrode SNR < 2 (from the visual response task) were excluded from the analysis.

In addition to the published data from macaques L & A, we also use an unpublished data set from one additional rhesus macaque (N=1) (Y, see Figure 4.1b right). Neural activity was recorded during rest and during a visuomotor integration task. The experimental and recording setup is described elsewhere (de Haan et al., 2018). The macaque was implanted with five Utah arrays (Blackrock microsystems), two of them in the primary visual cortex (V1), one in dorsal prelunate cortex (area DP), one in area 7A and one in the motor cortex (M1/PMd). In this chapter we only included the 6x6 electrode arrays from V1 (two arrays) and DP (one array), for a total of 108 electrodes. The recording system recorded the electric potential at each electrode with a sampling rate of 30 kHz. The electrodes had a length of 1 mm, thus recording the extracellular potential from the central layers, likely layer 4.

Two resting-state (RS) sessions were recorded. In these sessions (as

[1] spyking-circus.readthedocs.io

in the ones described above), the macaque did not have to perform any particular task and sat in a quiet dark room. Pupil position and diameter data were collected using an infrared camera in order to determine the gaze direction and eye closure of the macaque.
MUAe and LFP signals were computed using the same procedure as for the other data sets.

Table 4.1: Summary of subjects and recordings included here.

Subject	Session	Duration (s)	Areas	Electrodes	Clean Electrodes
L	L_RS_250717	1363	V1	896	765
			V4	128	116
L	L_RS_090817	1321	V1	896	761
			V4	128	116
L	L_RS_100817	1298	V1	896	774
			V4	128	118
A	A_RS_150819	2278	V1	896	402
			V4	128	11
A	A_RS_160819	2441	V1	896	369
			V4	128	9
Y	Y_RS_180122	906	V1	72	42
			DP	36	25
Y	Y_RS_180201	699	V1	72	44
			DP	36	24

4.2.2 *Neural manifolds, outlier removal and topological data analysis*

For the data analysis, the MUAe data is downsampled to 1 Hz and consists of between 50 and 900 recording electrodes per session.
For visualizing the data we use principal component analysis (PCA) as a method for dimensionality reduction. The first three principal components are retained, the remaining ones discarded. This reduces the high-dimensional neural activity to a 3D representation. We use a two-component Gaussian mixture model on the 3D projection to assign each data point to one of two clusters observed in the projection These clusters are in the following identified with the neural manifold. From the clustering using the mixture model, the log odds are calculated: $\log \frac{p_0}{p_1}$. Here, p_0 is the predicted probability of a data point in the projection to belong to the one, p_1 the predicted probability to

belong to the other cluster. The log odds are thus a measure for the likelihood of a given neural data point to belong to one or the other cluster, and capture the clustered structure of the manifold in a single function of time.

In the performed analysis, the neural manifolds are a collection of time points in the state space. In the recorded neural data, some time points have a large euclidean distance from all other points. These points are attributed to noise, and it is therefore desirable to remove them for further analyses. To determine the outliers we use a procedure similar to the one suggested by Chaudhuri et al. (2019). First, the distance between all points is computed. Then the 1st percentile value from the distance distribution, D_1, is determined. Finally, the number of neighbors that each point had within D_1 distance is estimated and the 20 percent of points with the fewest neighbors are discarded.

On the thus obtained data, we employ persistent homology (Edelsbrunner and Harer, 2022) to confirm that the lower-dimensional structures found in the 3D projection are actual topological features of the neural manifolds and not just an artifact of the dimensionality reduction. Before computing the persistence barcodes, we project the data into a 10D subspace using the isomap technique (Tenenbaum et al., 2000). This method is designed to preserve the geodesic distance (i.e. shortest path on the manifold) between the data points and is thus especially suited for reducing the dimensionality of data when performing a topological data analysis.

To calculate the persistence barcodes for the first three homology groups H_0, H_1 and H_2 we use the open-source implementation `Ripser` [2]. Briefly, the methods first constructs the Vietoris-Rips complex by successively inflating balls with radius r around each data point. If k points have a pairwise distance smaller than r (that is, for all pairs of points both points are contained in the ball with radius r of any other point), they form a $(k-1)$ dimensional simplex. Thus, the neural manifold gives rise to a simplicial complex (a collection of simplices of potentially different dimension). Topological features of this simplicial complex represent the topological structure of the neural manifold. These features are extracted by computational means. As r is grows, many short-lived features appear by chance. These do not capture actual topological properties of the manifold. True topological properties appear and persist for a wide range of radii r. The long-lasting bars in the n-th persistence barcode correspond to the number of independent generators β_n of the respective homology group H_n, also called Betti numbers. For low dimensionalities, they can be interpreted intuitively: β_0 is the number of connected components, β_1 the number of 1D holes, β_2 the number of enclosed 2D voids. Throughout all plots of this chapter we display the top 1% longest-lasting barcodes for each homology group.

[2] https://pypi.org/project/ripser/

4.2.3 *Dimensionality, coherence, and Granger causality analysis*

Two different approaches to study the dimensionality of the neural data are employed: first, we compute the time-varying participation ratio (PR) from the covariance matrix defined as

$$\mathrm{PR} = \frac{\left(\sum_i \lambda_i\right)^2}{\sum_i \lambda_i^2} \tag{4.1}$$

where λ_i are the eigenvalues of the covariance matrix (Mazzucato et al., 2016). Dahmen et al. (2020) showed that this is equivalent to

$$\mathrm{PR} = \frac{N}{1 + v^2 + (N-1)(m^2 + s^2)}. \tag{4.2}$$

Here, N is the number of electrodes, and v, m, as well as s are the ratios between the standard deviation of auto-covariances (δa), average cross-covariances ($\bar{c}$), and the standard deviation of cross-covariances (δc) with respect to the average auto-covariances ($\bar{a}$), respectively:

$$v = \frac{\delta a}{\bar{a}}, \quad m = \frac{\bar{c}}{\bar{a}}, \quad s = \frac{\delta c}{\bar{a}} \tag{4.3}$$

We take a 30 s sliding window with a 1 s offset over the MUAe data and compute the PR for each window separately (resulting in each window having a 29 s overlap with the adjacent windows). Since higher activity leads to a higher variance and thus a bias in the PR we normalize the data within each window via z-scoring to minimize this effect.

Secondly, we determine the distribution of the eigenvalues of the neural data within each cluster. Again the data is normalized the data after sampling each cluster. We use a linear regression in log-log space to fit a power law the distribution. Here, the slope of the linear fit corresponds to the exponent α of the power law.

To assess the communication between cortical areas we rely on the coherence and Granger causality of the LFP. Coherence quantifies linear correlations in the frequency domain:

$$C_{xy}(f) = \frac{|S_{xy}(f)|^2}{S_{xx}(f)S_{yy}(f)} \tag{4.4}$$

Here x and y denotes the signals, $S_{xy}(f)$ is the cross-spectral density, and $S_{xx}(f)$ and $S_{yy}(f)$ are the auto-spectral densities (power spectrum densities).

In order to assess the directionality of frequency dependent interactions between the areas, we perform a spectral Granger causality analysis again on the LFP signal (Dhamala et al., 2008). For this, first the cross-spectral matrix is computed,

$$S(f) = \begin{pmatrix} S_{xx}(f) & S_{xy}(f) \\ S_{yx}(f) & S_{yy}(f) \end{pmatrix}, \tag{4.5}$$

using the multitaper method (Thomson, 1982). To this end, the signals of pairs of electrodes are subdivided into 10 s long segments with an overlap of 50%. Each segment is processed individually with 3 Slepian tapers and subsequently averaged, yielding the cross-spectrum. Next, the cross-spectrum is decomposed into the covariance matrix Σ and the transfer function $H(f)$ with using Wilson spectral matrix factorization algorithm (Wilson, 1972):

$$S(f) = H(f)\Sigma H^{\dagger}(f) \tag{4.6}$$

As a covariance matrix, Σ is a symmetric positive-definite matrix, † denotes the conjugate transposition, and $H(f) = \psi(e^{if}) = \sum_{k=0}^{\infty} A_k e^{ikf}$ is a matrix valued function with $A_0 = 1$ (identity matrix), ψ is a function initially defined for $z \in \mathbb{C}$ satisfying $|z| = 1$ that can be extended to a holomorphic function on the disk $\{z|z \in \mathbb{C} \text{ s.t. } |z| < 1\}$ (that is, the power series $\sum_{k=0}^{\infty} A_k z^k$ converges on that domain) (Dhamala et al., 2008). With Σ and H one is able to obtain a version of directional functional connectivity in the frequency domain between the first and second signal with

$$\mathrm{GC}_{x\to y}(f) = \frac{S_{xx}(f)}{\tilde{H}_{xx}(f)\Sigma_{xx}\tilde{H}^{\dagger}_{xx}(f)} \tag{4.7}$$

where $\tilde{H}_{xx}(f) = H_{xx}(f) + \Sigma_{xy}/\Sigma_{xx}H_{xy}(f)$ and mutatis mutandis for the influence of the second onto the first signal (ibid.). The analysis is performed for all pairs of channels between the areas that exhibited a peak in the coherence in the β-band $12\,\mathrm{Hz} < f < 30\,\mathrm{Hz}$ (see below). The β-band Granger causality strength is quantified as

$$\mathcal{B} = \sum_{f=12\,\mathrm{Hz}}^{30\,\mathrm{Hz}} \mathrm{GC}_{x\leftarrow y}(f) - \mathrm{GC}_{x\to y}(f). \tag{4.8}$$

Similarly, we also analyse the time-varying spectral Granger causality. For this we use 10 s windows and moved them across the data with 1 s steps, for a final time resolution of 1 Hz. The spectral Granger causality is calculated for each window separately. The initial and final 5 s are discarded to avoid disruptions at the boundaries. Thus, time-varying causality spectrogram reads as

$$\mathrm{GC}_{x\to y}(t,f) = \begin{cases} \mathrm{GC}_{x\to y}(f)\Big|_{[t_0,t_1]} & \text{if } t \in [t_0,t_1], \\ \ldots & \\ \mathrm{GC}_{x\to y}(f)\Big|_{[t_{n-1},t_n]} & \text{if } t \in [t_{n-1},t_n] \end{cases} \tag{4.9}$$

and mutatis mutandis for the $y \to x$ direction where each interval has a length of 1 s.

Finally, we define the time-varying Granger causality difference, $\Delta\text{GC}(t,f) = \text{GC}_{x\leftarrow y}(t,f) - \text{GC}_{x\rightarrow y}(t,f)$, and again sum over the β-band (c.f. Equation 4.8):

$$\mathcal{B}(t) = \sum_{f=12\,\text{Hz}}^{30\,\text{Hz}} \Delta\text{GC}(t,f) \tag{4.10}$$

Both the coherence and spectral Granger causality are implemented in the Electrophysiology Analysis Toolkit (Elephant, Denker et al. (2022)) For the aforementioned peak detection in the coherence and the Granger causality analysis (used to identify admissible channel pairs for our analysis) we employ a standard peak detection algorithm for time series using wavelet transforms. We compute the continuous wavelet transform (cwt) for wavelets with widths from 10 Hz to 100 Hz (at 0.1 Hz steps), using a Ricker wavelet. Next, we search for ridge lines in the cwt —peaks across different wavelet lengths —following standard criteria suggested in the literature (Du et al., 2006). Finally, the ridge lines are filtered based on their total length, gaps, and signal-to-noise ratio (snr). The resulting ridge lines (if any) are considered as peaks in the coherence.
The detected peaks tend to be broad, since our parameter choice intentionally rejected narrow peaks. This configuration was chosen in favor of robustness and to minimize false positives. Nonetheless, peaks are detected for a majority of electrode pairs.

Table 4.2: Peak detection algorithm parameters.

CWT peak detection parameters

widths	100 Hz to 500 Hz	Width range for CWT matrix.
Width$_{s}tep$	0.1 Hz	Step between widths.
Wavelet	Ricker	Wavelet used for convolution.
Max_distances	widths / 4	Criterion to consider ridge lines connected.
Gap_thresh	10 Hz	Ridge lines farther apart will not be connected.
Min_length	225	Minimum length of ridge lines.
Min_snr	1	Minimum snr of ridge lines.
Noise_perc	10	Percentile of ridge considered noise for snr calculation.

4.3 RESULTS

4.3.1 *Two distinct neural manifolds in V1 correlated with eye closure*

To explore the activity of the visual cortex, we analyse the high-dimensional population multi-unit activity envelope (MUAe) (between 40 and 800 electrodes) for each area and macaque (Supèr and Roelfsema, 2005), Figure 4.2a. We projected the population activity into a 3D space using PCA (see Section 4.2.2), Figure 4.2b-d.

In V1, at least two distinct clusters are apparent in the 3D projection space (sample session in Figure 4.2b-d, see Figure S1-S6 for all other sessions and subjects). In the following, we will refer to the clusters also as neural manifolds. The manifolds are labeled according to the sign of the log odds of a two-component Gaussian mixture model (see Section 4.2.2).

To exclude that two manifolds visible in the lower-dimensional projection are an artifact of the dimensionality reduction, we computed the Betti numbers of higher-dimensional population activity using persistent homology (see Section 4.2.2). We find at least two independent generators of the H_0 homology groups higher-dimensional population activity. This corresponds to at least two connected components (Figure 4.3), i.e. two distinct neural manifolds.

Additionally, we tested whether the observed manifolds could be an induced by the signal processing needed to obtain the MUAe signal. One session (L_RS_250717) was spike sorted with a semi-automatic spike sorter (see Section 4.2.1) and the population activity resulting from the single-neuron firing rates analyzed (Figure 4.4). The spiking activity also displays two manifolds, in agreement with the findings from the MUAe signals.

While the activity of visual cortex is mainly driven by visual input (Talluri et al., 2022), whether and to what extent it is separately modulated by eye closure is unclear. Marking data points on the V1 manifolds with the eye closure signal (Figure 4.2b) reveals that one manifold strongly relates to the eyes-open periods, whereas the other manifold strongly relates to the eyes-closed periods.

To confirm the correlation between eye closure and manifolds, we tested the differences between the eyes-open and eyes-closed periods with a logistic regression between the eye closure signal and the log odds, revealing a significantly higher than chance correlation in all sessions (Figure 4.2e). Additionally, the distribution of the log odds during the eyes-open and eyes-closed periods show a clear correspondence between the eye closure and the sign of the log odds in most cases (Figure 4.2e). This demonstrates that membership of a point in state space in one of the two V1 manifolds is closely related to eye closure. Given this, we will refer to the manifolds as the eyes-open manifold or the eyes-closed manifold.

Note that the existence of two separate manifolds could be trivially explained if the MUAe activity levels were significantly higher in the eyes-open periods when compared with the eyes-closed periods. If this was true, the manifolds would simply reflect the population activity level. To discard this possibility, we tested whether higher-activity periods correspond uniquely to one of the manifolds. The data distribution —based on the z-scored MUAe shown in Figure 4.2a —show that there is no clear separation into two manifolds (Figure 4.2f). Additionally, we determined the joint distributions of z-scored MUAe and log odds (Figure S7, represented with a density heatmap). Both the data distribution of the MUAe and the joint distribution of the MUAe and the log odds suggest that the activity level (and thus their difference between eyes-open and -close periods) alone does not fully explain the presence of the two neural manifolds.

Finally, we also analyzed the population activity from V4 and DP (Figure S8, Figure S9). In contrast to V1, the population activity in areas V4 and DP does not appear to contain two distinct neural manifolds (Figure S10). Thus, we conclude that the observed manifolds are restricted to V1 and are not present in V4 or DP.

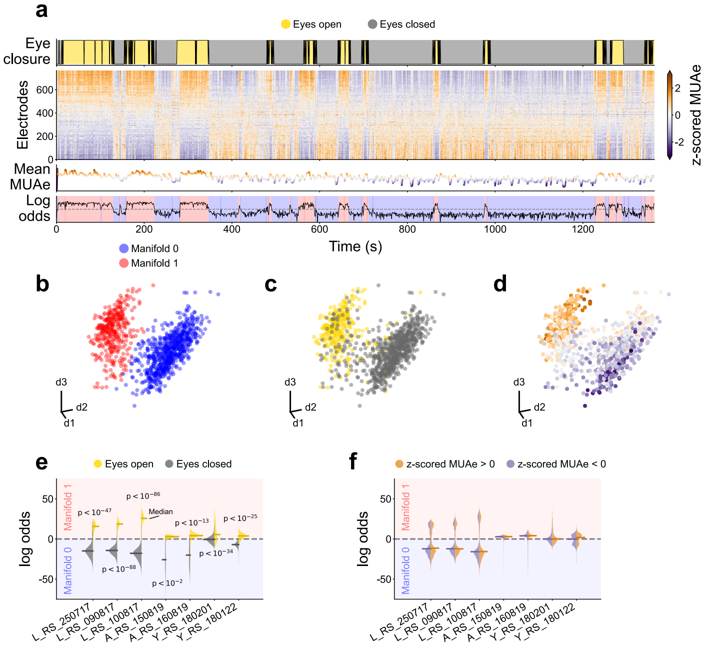


Figure 4.2: Overview of the behavioral as well as neural activity data in V1. The two distinct neural manifolds are correlated with eye closure. **a** Overview of the experimental data from session L_RS_250717. From top to bottom: Time evolution of the eye signal; the z-scored MUAe signal for each electrode (electrodes ordered by their correlation with the eye signal); the mean z-scored MUAe at each time point; and the log odds overlaid with the most likely manifold (two clusters, Gaussian mixture model). **b, c, d** First three principal components of the MUAe population activity. Colours indicate the manifold identified via the log odds of a Gaussian mixture (**b**), the eye closure (**c**) and the mean z-scored MUAe (**d**). Each dot represents a different point in time. Outliers were excluded from the neural manifolds shown in **b**–**d**, see Section 4.2.2. **e, f** Violin plots of the distribution of the log odds across epochs, respectively distinguished according to eye closure (**e**, result of a logistic regression test shown) and z-scored MUAe (**f**). Horizontal bars indicate medians of the distributions.

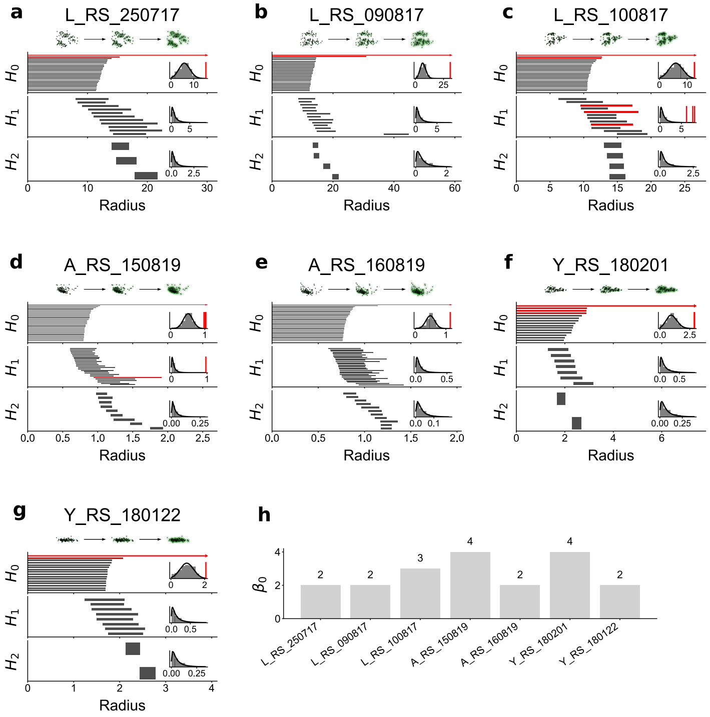


Figure 4.3: Persistent homology of the high-dimensional neural activity show the presence of at least two clusters. **a** - **g** Each panel shows data for one session. (Top) Sample clouds with a green radius used for persistent homology computation. (Main plots) Persistence barcodes of the 10D neural manifolds, for all sessions. (Inset plots) Distribution of barcode length with a fitted lognormal distribution, long barcodes coloured red. **h** Number of clusters found in each session.

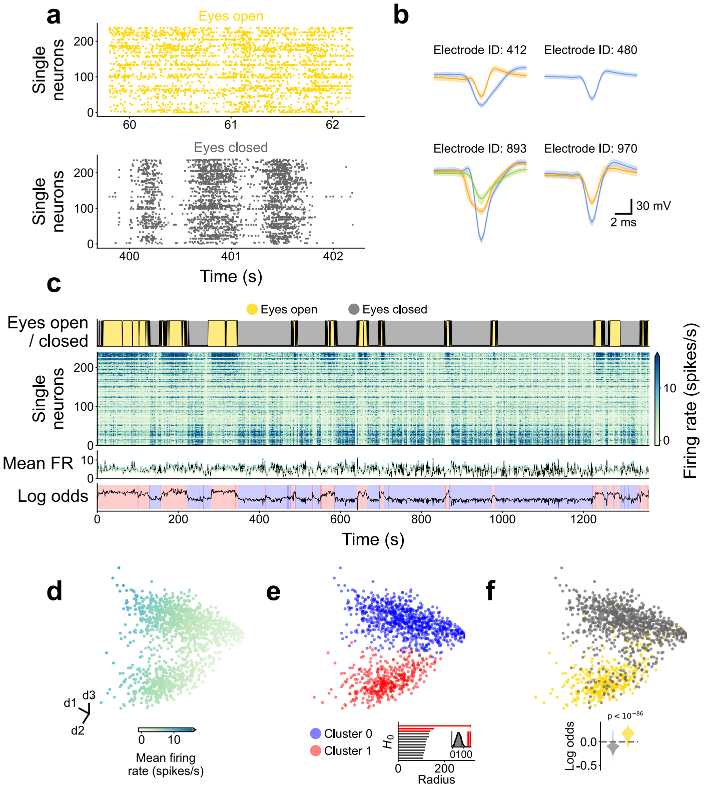


Figure 4.4: Overview of the spiking data from session L_RS_250717. Single neurons were isolated using a semi-automatic spike sorting method. The firing rate was calculated counting the number of spikes in 1-second bins. **a** Sample spike raster plots for eyes-open and eyes-closed periods. **b** Sample waveforms from four electrodes, multiple single units isolated in some electrodes (colour-coded). Median (solid line) and 20-80 percentiles (shading) shown per unit. **c** Time course of multi-unit activity. **d, e, f** First three principal components of the firing rate. Insets show the persistent homology for the H_0 homology group (e) and the violin plots of the eye closure against the clustering log odds (f).

4.3.2 *Dependence of manifold dimensionality on eye closure*

Next, we further explored the properties of the neural manifolds in V1 by studying their dimensionality.
For this, the time-varying participation ration (PR) (see Section 4.2.3) was computed from the z-scored MUAe signals for sliding windows of 30 s width. Higher MUAe activity is typically associated with higher variance, potentially leading to higher dimensionalities. To avoid this bias, we normalized the varying activity levels (via z-scoring) within each window. We find a strong correlation between the log odds and the time-varying PR (Figure 4.5b). Comparing the PR values between the two manifolds using a Mann-Whitney U test reveals a consistent and significant increase of dimensionality during the eyes-open periods for all data sets (Figure 4.5c).
To further support this finding, we also show the distribution of the variance explained by each of the principal components (PC) of the MUAe data, depicted on a log-log scale in Figure 4.5d. We fitted a power law to the PC variances and report the exponent α (Figure 4.5e). A higher α indicates faster decay of the curve, i.e., lower dimensionality. The presence of power laws is consistent with Stringer et al. (2019a), its exponents are in agreement with our sliding window approach: We observe higher dimensionality during eyes-open than during eyes-closed periods (Figure 4.5e).
In order to narrow down the cause of the dimensionality changes, the various quantities the PR is a function of (v^2, $(N-1)m^2$, and $(N-1)s^2$) were computed (see Equation 4.2). We observe that the changes in $(N-1)m^2$—the scaled average cross-covariances divided by the average auto-covariance—dominate the PR differences between the eyes-open and eyes-closed periods (Figure 4.5f). The main reason for the observed dimensionality changes therefore stems from decorrelation of the activity during eyes-open periods.

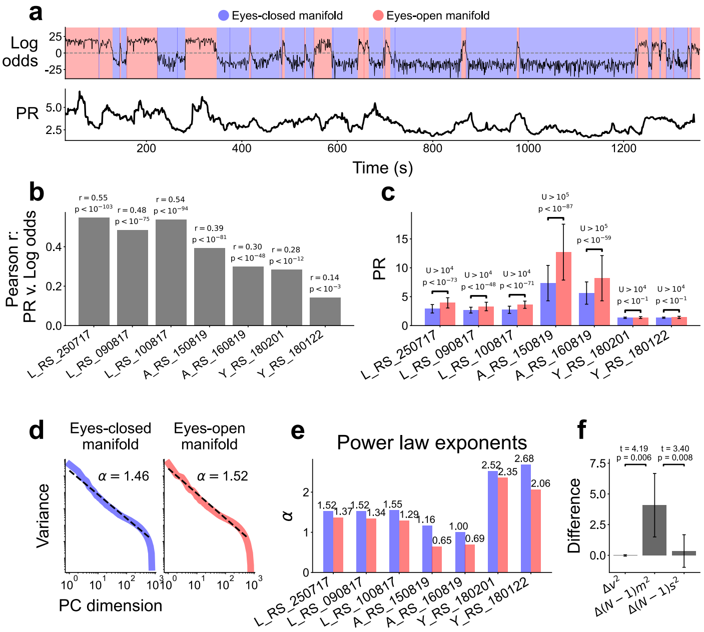


Figure 4.5: Higher dimensionality during eyes-open periods. **a** Log odds and participation ratio (PR) for session L_RS_250717. The PR was calculated on a sliding window of 30 s width. **b** Pearson correlation between log odds and PR. **c** Comparison of PR between neural manifolds (Mann-Whitney U test). **d** Distribution of principal components and their explained variance on a log-log scale for each manifold. Power law exponent α estimated over the ranges where the curves approximate a power law. **e** Comparison of power law exponents for the two neural manifolds in all sessions. The eyes-open manifold always had a smaller exponent, indicating a higher dimensionality. **f** Differences of quantities decomposing the PR (standard deviation of auto-covariances, v^2, average cross-covariances, $(N-1)m^2$, and standard deviation of cross-covariances, $(N-1)s^2$). Results of Welch's t-test across sessions shown.

4.3.3 *Modulation from V4 to V1 in the β-band*

In search of an internal, stimulus-independent mechanism that may modulate the neural manifolds and their dimensionality, we turned our attention to cortico-cortical interactions. Since signatures of top-down activity have previously been reported in the β frequency band (roughly 12 Hz to 30 Hz) (Bastos et al., 2015; Vezoli et al., 2021), we perform a spectral Granger causality analysis to assess top-down modulations.

To determine whether top-down signals are present in our data, we calculated the coherence and Granger causality between every pair of V1-V4 and V1-DP electrodes (see Section 4.2.3) in the local field potential (LFP). Figure 4.6a,b show the coherence and Granger causalities for a sample pair of electrodes. To quantify the cortico-cortical signals, we searched for peaks in the coherence and Granger causality, using an automatic method (see Section 4.2.3). We detected β frequency Granger causality peaks in around 0.5% of all V1-V4 electrode pairs, predominantly in the top-down direction (Figure 4.7). We only found β-band bottom-up interactions in V1-DP electrode pairs.

For the electrodes with a β causality peak, we estimated the causality strength $\mathcal{B}$ (Equation 4.8). The electrodes with their receptive field (RF) closer to the fovea show substantially higher $\mathcal{B}$ (Figure 4.6c–e, Figure S11), in agreement with a previous structural connectivity report (Wang et al., 2022) (Figure 4.6f). To disregard potential spurious Granger causality peaks, we restrict all further analysis of the top-down signals to the strongest interactions, by setting a threshold of $\mathcal{B} > 10$ (Figure 4.6g). We observed no bottom-up V1-to-V4 signals with high strength in the β frequency band.

Thus, we find top-down signals from V4 to V1, in agreement with previous studies (Bastos et al., 2015; Vezoli et al., 2021); but we do not observe strong signals from DP to V1 in our data. V4-to-V1 signals are therefore strong candidates for the modulation of the neural manifolds and their dimensionality.

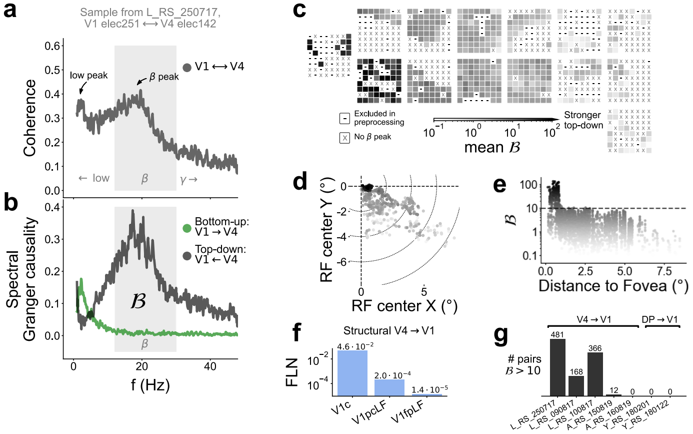


Figure 4.6: Inter-area coherence and spectral Granger causality. **a** Representative sample of coherence between V1 and V4 (electrodes 242 and 142, respectively). Low-frequency and β-band peaks indicated. **b** Representative sample of spectral Granger causality. **c** Schematic representation of the electrode locations overlaid with the mean top-down signal strength $\mathcal{B}$ per electrode (see Section 4.2.3 for a description of $\mathcal{B}$). **d** Receptive field (RF) map overlaid with the mean $\mathcal{B}$ per electrode. Stronger $\mathcal{B}$ is found around the foveal region of V1. **e** Mean $\mathcal{B}$ displayed against the distance from the fovea. **f** Log of fraction of labelled neurons (FLN) from V4 to V1 (data from tract-tracing experiments (Wang et al., 2022)). V1 subdivisions represent c: central (foveal region), LF: lower visual field, pc: peri-central, and fp: far periphery. The strongest connectivity exists from V4 to V1c, in agreement with our measurements. **g** Number of electrode pairs with strong ($\mathcal{B} > 10$) top-down signals detected in each session.

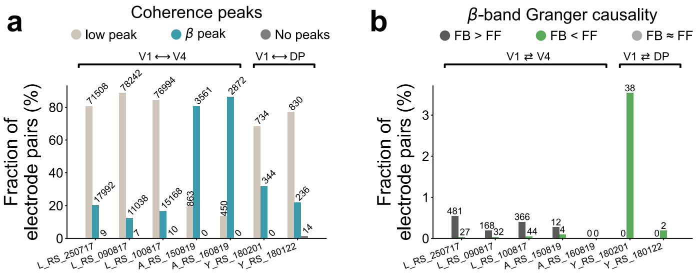


Figure 4.7: Quantification of coherence peaks and β-band spectral Granger causality. **a** Quantification of coherence peaks across all sessions. A substantial portion of all electrode pairs displayed a β peak. Note that the percentages for a session can add up to more than 100% since the same electrode pair can have both a low-frequency and a β peak. **b** Quantification of β-band spectral Granger causality for all sessions. Welch's t-test was used to determine whether top-down Granger causality was greater than, less than, or roughly equal to bottom-up Granger causality, within the β frequency band. The test was only applied to those electrode pairs that showed a β coherence peak. A large portion of V1 $\leftrightarrow$ V4 pairs show stronger causality in the top-down direction, while V1 $\leftrightarrow$ DP did not appear to have prominent top-down causality compared to bottom-up causality.

4.3.4 *Stronger top-down signals from V4 to V1 during eyes-open periods*

To investigate links between behavior and the V4-to-V1 top-down modulation, we examined how the spectral power, coherence, and Granger causality of the LFP change in relation to eye closure.

We extracted the LFP data for each behavioral condition and concatenated the data within the same condition. This approach potentially introduces some artifacts, which we expect to be minor in view of the very small number of transitions in comparison with the number of data samples (500 Hz resolution). Both in V4 and V1, we find that the spectral power at low frequencies (< 12 Hz) is higher during the eyes-closed periods, whereas the power in the β-band (12 Hz to 30 Hz) is slightly higher during eyes-open periods (Figure 4.8a, see Figure S12 for other sessions). Spectrograms of the V1 LFP power confirm the reduction in low-frequency power during eyes-open periods (Figure 4.9). The coherence in the β-band is higher during the eyes-open periods, with the peak shifted to higher frequencies compared to the eyes-closed condition. Notably, the top-down Granger causality is substantially higher in the β-band during the eyes-open periods.

In order to confirm our observations, we also computed spectrograms of the Granger causality using a 10-second sliding window (Figure 4.10a). Performing a Welch's t-test of the difference between bottom-up and top-down Granger causality, ΔGC, confirmed a shift toward top-down interactions during the eyes-open periods compared to the eyes-closed periods, for a vast majority of all electrode pairs (Figure 4.10b,c). Thus, we found higher β-band Granger causality during eyes-open periods using two different approaches.

Additionally, to confirm the interdependence of top-down signals and the neural manifolds, we computed the correlation between the time-varying β-band Granger causality $\mathcal{B}(t)$ (Equation 4.10) and the log odds (Figure 4.8b,c). An overwhelming majority of V1-V4 electrode pairs showed a highly significant correlation ($p < 10^{-6}$, two-sided t-test). Thus, the top-down signals and neural manifolds are co-dependent at a fine temporal scale, as well as within eyes-closed and eyes-open periods.

We further tested whether the top-down signals were correlated with gaze direction and eye movements (Figure S13), to rule out the presence of any visual stimuli—despite the experiments being performed in a dark room. No clear trend could be observed, thus indicating no relation between gaze direction and top-down signals. This finding suggests that the visual scene is not the source of the observed top-down signals.

Summing up, the time-dependent spectral analysis reveals large variations of power and Granger causality. On the one hand, the spectral power at low frequencies decreases during eyes-open periods, consistent with the well-known α blocking phenomenon (Berger, 1929;

Klimesch et al., 2007; Liley and Muthukumaraswamy, 2020). On the other hand, the V4-to-V1 top-down signals are strongest during the eyes-open periods. The time-varying top-down β causality strength did not substantially correlate with gaze direction or eye movements, suggesting no relation between the top-down signals and the visual scene; as expected in a dark room. Taken together, these results suggest that V4-to-V1 signals modulate V1 activity, contributing to a different state-space manifold with increased dimensionality.

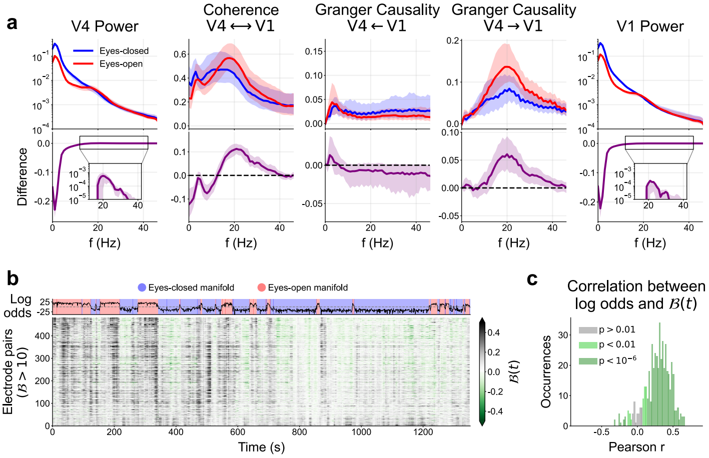


Figure 4.8: Stronger top-down signals from V4 to V1 during eyes-open periods. **a** Spectral power, coherence, and Granger causality of the LFP for the electrodes with high causality strength ($\mathcal{B} > 10$) in session L_RS_250717, see Figure S12 for all other sessions. The data for each behavioural condition (eyes-open/closed) were concatenated and their metrics reported separately (top row). The difference between eyes-open and eyes-closed periods was calculated for each electrode or pair of electrodes (bottom row). In all panels the thick line shows the median across electrodes (or pairs of electrodes) and shading indicates the 25th to 75th percentile. **b** Time evolution of log odds (top) and time-dependent β-band Granger causality difference $\mathcal{B}(t)$ (bottom), for the electrode pairs with top-down signals. **c** Histogram of the Pearson correlation between the log odds and $\mathcal{B}(t)$. Colour indicates the significance levels of the associated two-sided t-test.

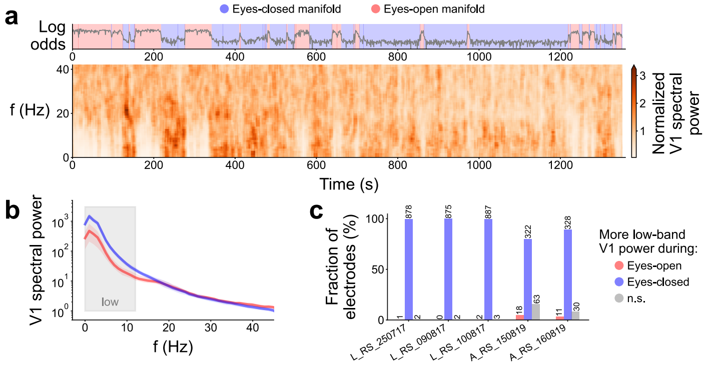


Figure 4.9: Analysis of the V1 LFP spectrogram. **a** Log odds identifying the neural manifolds (as in Figure 4.2a), and time-varying spectrum of a sample V1 electrode (session L_RS_250717, power normalised for each frequency). **b** Spectrum of a sample V1 electrode (session L_RS_250717). Colours indicate the different manifolds. **c** Result of t-test in the low frequency band (less than 12 Hz) for all V1 electrodes. As expected, the overwhelming majority of electrodes displays higher low frequency power when the eyes are closed.

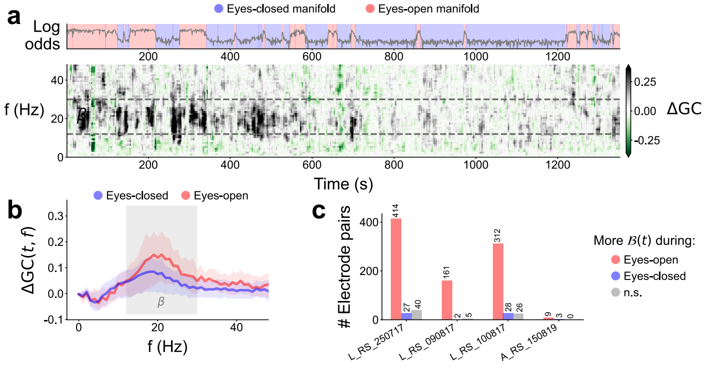


Figure 4.10: Time-dependent spectral Granger causality reveals higher top-down signals in the eyes-open periods. **a** Time evolution of the log odds (top), and the spectral Granger causality difference for a representative sample of V1-V4 electrodes (bottom). The sample electrodes were the same as in Figure 4.8a. **b** Causality difference median (line) and 25th to 75th percentiles (shade) in each manifold for one sample V1-V4 electrode pair. β frequency range highlighted. **c** Quantification of β-band causality difference $\mathcal{B}(t)$ over time (in each manifold) for all V1-V4 and V1-DP electrodes in all sessions—using Welch's t-test.

4.4 DISCUSSION

In this chapter, three findings in the resting-state activity in the primary visual cortex of macaque are presented: two separate neural manifolds in the state space linked to eye closure (Figure 4.2), higher dimensionality due to lower mean cross-correlations in eyes-open periods (Figure 4.5), and the presence of stronger top-down signals from V4 to V1 during the eyes-open periods, primarily targeting the foveal region of V1 (Figure 4.6, Figure 4.7, Figure 4.8). Beyond that we observed lower power at frequencies below 12 Hz during the eyes-open periods (Figure 4.8, Figure 4.9), consistent with the well-known α blocking effect (Berger, 1929).

A separation of neural activity of macaque V1 during resting-state into two neural manifolds in both the MUAe and spiking activity was found (Figure 4.2, S1-S6, Figure 4.4). The manifolds in V1 are correlated with eye closure and could not be found in the activity of V4 or DP. A simple explanation for the separation could be movements of the macaques: indeed, Stringer et al. (2019b) showed that visual cortex in mice in the resting-state represents behaviors as facial movements or running. Talluri et al. (2022), however, showed that this finding does not generalize to the visual cortical activity in macaque. Hence it is plausible to conclude that the modulation of the neural manifolds are mainly driven by visual behavior.

Alternatively, the results presented here could be a simple consequence of the visual activity being driven by strong visual stimuli. However, the recording room was dark, and we furthermore conducted analyses to control for activity levels (Figure 4.2 and Figure S7) and gaze direction (Figure S13). Moreover for macaques L and A extensive evaluations of data quality were performed, which excluded all electrodes that did not strongly respond to visual stimuli (Chen et al., 2022). This implies that in the analysis of macaques L and A all remaining electrodes would strongly respond if a strong visual stimulus was present. Such responses were, however, not observed in the MUAe activity analyzed here (see Figure 4.2).

Additionally, we explored the dimensionality of the neural activity (Figure 4.5). We found that the dimensionality is manifold-dependent and higher in the eyes-open periods across all macaques and all sessions. Our findings are in agreement with previous studies on the visual cortex (Gao and Ganguli, 2015; Stringer et al., 2019a). Note that analogous to the results presented here, Dąbrowska et al. (2021) also showed higher dimensionality in the primary motor cortex during eyes-open periods in comparison with eyes-closed periods.

We hypothesized that top-down signals from higher cortical areas could modulate the neural manifolds and dimensionality in the V1 activity, and thus be responsible for the observed changes. Strong top-down signals from V4 to V1 (Figure 4.6) were indeed found, targeting particularly the foveal region of V1, in agreement with structural connectivity (Wang et al., 2022). This top-down signals vary over time, with increased presence during the eyes-open periods (Figure 4.8). In agreement with our results, previous studies found that cortico-cortical top-down signals between in visual areas are primarily located in the β (12 Hz to 30 Hz) frequency band. Bottom-up signals are to be found in the delta/theta ($<$ 8 Hz) and γ (>30 Hz) bands (Bastos et al., 2015; Vezoli et al., 2021).

We did not observe any γ-band causality (Figure 4.6). This is likely due to the recordings being from the deep cortical layers (in macaque L and A the electrodes were 1.5 mm long, putatively recording mostly from layer 5) since γ oscillations are known to be weak in layer 5 of the visual cortex (Mendoza-Halliday et al., 2022; van Kerkoerle et al., 2014). Contrary to the results presented in this chapter, van Kerkoerle et al. (2014) reported that top-down signals appear in the α (8 Hz to 12 Hz) frequency range. Whether the specific top-down and bottom-up frequencies generalize to the whole neocortex remains elusive. Instead, Vezoli et al. (2021) suggest overlapping modules associated with frequency bands (α, low-β, high-β, and γ) that differ across cortex. Our findings are additionally consistent with the work by Semedo et al. (2022): they suggested that bottom-up signals dominate in a stimulus driven regime while top-down signals dominate in the absence of visual stimuli. It has to be noted, however, that in their work the eyes were always open. We did not find top-down signals from DP to V1. Potential reasons for this include the electrodes used in macaque Y being 1 mm long, thus likely recording from granular layer (layer 4). As a matter of fact, top-down connections do neither stem from nor target this layer of cortex (Markov et al., 2014b).

The change in V1 population activity with eye closure insinuates a functional benefit. We here propose the following line of reasoning: if the eyes are closed, no visual stimuli can be processed and thus the V1 activity can be reduced, potentially to save energy. If the eyes are open, however, higher-dimensional activity might improve the encoding of incoming visual stimuli. Thus, the separation of neural manifolds accompanied by the observed change in dimensionality could facilitate visual processing. Previous work showed that spectral power in the α-band (8 Hz to 12 Hz) is inversely correlated with visual recognition performance in human subjects (Hanslmayr et al., 2005; Klimesch et al., 2007): lower α power was associated with better performance in a visual discrimination task. Our findings suggest that a direct correlation between separation of neural manifolds and dimensionality with the decrease in α power (Figure 4.8, Figure 4.9). Future work could investigate the functional implications of our findings in light of the relation between the dimensionality, α power, and visual performance.

Given the analyzed experimental data, it was not possible to test directly whether the V4-to-V1 signals are in fact responsible for the modulation of V1 dynamics. In view of the research program suggested in Chapter 2, simulations of biologically plausible spiking neural networks offer a means to validate the hypothesis developed above. *In silico* experiments following this line of thought are described in [add autref for chaper here].

In conclusion, we provide *in vivo* evidence for the modulation of neural manifolds by cortico-cortical communication, which we hypothesise could enable more efficient responses to visual stimuli. Our analysis and previous results suggest that the eyes-open manifold—together with the corresponding dimensionality and spectral power changes—constitutes a visual stand-by mode, which is modulated by top-down input from V4 and potentially other internal mechanisms.

Part III

CONSTRUCTION AND SIMULATION OF BIOLOGICALLY PLAUSIBLE SPIKING NEURAL NETWORK MODELS

5 PREFACE

In this part, we turn to the central topic of the neuroscientific research program sketched in Chapter 2: construction of digital twins of cortical tissue. We here focus on large-scale, biologically plausible spiking neural network models.

Here, the author of this thesis wants to highlight the need for simulation studies in general, and the importance of taking biological constraints seriously in particular.

The dynamical properties of networks consisting of spiking model neurons have been analysed mathematically (see e.g. Brunel (2000) and El Boustani and Destexhe (2009)). Research along these lines actively continues to this day (e.g Ostojic (2014), Tartaglia and Brunel (2017), and van Meegen and van Albada (2021)). These approaches rely on approximations, most often based on mean-field theory. This implies that their domain of applicability is limited to a regime where the assumptions of the theory are satisfied. Most importantly, correlations between the constituting neurons must be weak. While for the resting-state spiking activity of cortical neurons it seems to be reasonable to assume that correlations are small, this assumption is nonetheless frequently violated, for example by an increase in firing rate due to an incoming sensory input or an external modulation (Schulz et al., 2015). Thus, simulations of spiking network are necessary to fully investigate the dynamical behavior of spiking neural network models of cortical tissue —beyond what is possible analytically —, for example when driven by an external stimulus.

Moreover, van Albada et al. (2015a) showed that to faithfully represent cortical dynamics, realistic scales of the models matter: the downscaling of neuronal network models is fundamentally limited if first- and second-order statistics of the dynamics are to be preserved. Both quantities are relevant for the activity of cortical tissue ultimately related to function (see e.g. Ito et al. (2011) and Romo et al. (1999)). Since cortical circuits need to have a size of at least on the order of ten thousand neurons to reconcile at the same time a local connection probability of about 0.1, and a local indegree on the order of a thousand (Maksimov et al., 2018), this constrains the smallest possible scale of realistic models.

Taken together, these points illustrate the necessity of simulations of biologically plausible spiking neural network models.

Using such digital twins of cortical tissue as a means of neuroscientific research has, however, not been hailed unanimously. Indeed, they often are regarded somewhat skeptically, as summarized by Traub et al. (2005):

> "Any model, even of a small bit of cortex, is subject to difficulties and hazards: limited data, large number of parameters, criticism that models with complexity comparable to the modeled system cannot be scientifically useful, the expense and slowness of the necessary computations, serious uncertainties as to how a complex model can be compared with experiment and shown to be predictive. The above hazards are too real to be dismissed readily. In our opinion, the only way to proceed is through a state of denial that any of the difficulties need be fatal. The reader must then judge whether the results, preliminary as they must be, help our understanding."

The author of this thesis shares the concerns —as well as the hope —raised by Traub et al. Indeed, in light of the above considerations, and taking the integrative power of biologically plausible neuroscientific modeling into account (cf. Chapter 2), we here bring forward the argument that this approach is a necessary step in bridging the gap between experiment and theory in Neuroscience.

In the remainder of this part, we employ biologically plausible spiking neural network models as a means for neuroscientific research.

In Chapter 6, simulations of a generic spiking network model of cortical tissue are performed to explore to what extent the hypothesis established in Chapter 4 (change of neural manifold induced by top-down modulation) is consistent with numerical simulations. We find that indeed a modulation provided as a weak, additional input targeting a random subset of the neurons causes such a change. Thereby, we give credibility to the mechanism proposed in Chapter 4.

The model used in Chapter 6 is minimal in the sense that it consisted of just enough neurons and synapses to satisfy the criteria for biologically plausible networks stated above. This motivates the construction of a more detailed spiking neural network model of macaque V1 in Chapter 7. In this chapter, we aim at taking the laminar as well as lateral structure of cortex into account. The model includes biologically constrained distance-dependent connection probabilities. Additionally, higher-order connectivity structures such as patchy connectivity mediated by orientation preference are embedded in the network.

Simulations of the model developed in Chapter 7 fail to exhibit reasonable activity given biologically plausible parameters for single cell dynamics. Indeed, an additional analysis reveals that the local structure of cortical connectivity cannot be reconciled with realistic activity without ad-hoc changes of the structure of the network. This necessitates a more detailed investigation of the architecture of local cortical circuits. Using recent reconstructions of the cortical tissue in mouse visual and human temporal cortex, we construct connectivity maps in Chapter 8 to facilitate this investigation. We find stark differences between the cortical connectivity derived in Chapter 7 and in Chapter 8: the connectivity maps derived in the latter chapter exhibit architectural features that can help mitigate the problems in dynamical simulations encountered when using the former one.

6 TOP-DOWN MODULATIONS IN SPIKING NEURAL NETWORK MODELS

The chapter is in part based on the following publication:

Morales-Gregorio, A., Kurth, A. C., Ito, J., Kleinjohann, A., Barthélemy , F. V., Brochier, T., Grün, S., & van Albada, S. J. (2023). Neural manifolds in V1 change with top-down signals from V4 targeting the foveal region. bioRxiv, doi: 10.1101/2023.06.14.544966

Author contributions:
Under the supervision of SvA, SG and JI, the author and AMG conceptualized and designed the *in silico* experiments. The author wrote the simulation code. The author and AMG performed the analysis of the simulated data which was refined through discussions between the author, AMG, JI, SG, and SvA.

6.1 INTRODUCTION

Sparsely connected random networks of excitatory and inhibitory neurons are simple models of neural circuits. If the recurrent excitatory and inhibitory inputs cancel each other on average (leading to a mean zero input current with finite fluctuations), these networks are called *balanced*. Provided that certain conditions regarding the size of the network, the connection probability between the neurons and the average indegree of as well as the evoked post-synaptic potentials in single-cells are satisfied (cf. Chapter 5), these networks can be considered as one of the simplest instances of biologically plausible spiking neural network models of local cortical circuits. As such, they are able to capture some fundamental properties of neuronal activity *in vivo* and *in vitro* (Maksimov et al., 2018).

Balanced networks and their variations have been occupying prominent places in Computational Neuroscience (e.g. Brunel (2000), Rosenbaum and Doiron (2014), Schaub et al. (2015), and van Vreeswijk and Sompolinsky (1996) to list some). Among others things, they have been studied analytically with binary (van Vreeswijk and Sompolinsky, 1996) and spiking (Brunel, 2000) model neurons. Van Vreeswijk and Sompolinsky (1996) showed the chaotic nature of neuronal dynamics in such networks of binary neurons. Moreover, they highlighted that the response time to an external perturbation of a network in the balanced state is shorter than the response time of single units. Brunel (2000) explored the repertoire of neuronal activity. The author described a stationary, asynchronous state with irregular single-cell firing that resembles cortical activity, alongside asynchronous and synchronous regular, as well as synchronous irregular states. Monteforte and Wolf (2012) investigated the sensitivity of the activity in balanced spiking networks to perturbations on the single spike level.

In this chapter, we set out to use simulations of a balanced spiking neural network model to reproduce the experimentally observed change in dynamical state shown in Chapter 4. There, we related the states to the eye closure of the monkeys. Moreover, top-down modulation from V4 to the primary visual cortex was identified as a potential trigger for the observed change in state. However, due to the experimental setup and limitations in the available recordings, this hypothesis could not be tested directly.

We here simulate a spiking network model of the local cortical circuit in macaque primary visual cortex to investigate the effects of a simple, external modulations on its dynamical state. The approach taken in this chapter is minimal in the sense that it is neither the goal to exhaustively investigate plausible changes in model architecture, nor to explore a wide range of potential effects on single neurons induced by top-down modulation influencing the network dynamics. Rather, we focus on a variation of an established model in a simple setup to ascertain the consistency of our hypothesis with *in silico* experiments.

6.2 METHODS

To investigate whether top-down signals in the β-band can induce a change in the population dynamics, we conduct simulations of spiking neural network models inspired by (Brunel, 2000). The network consists of an excitatory population of 10.000 and an inhibitory population of 2.000 leaky integrate-and-fire (LIF) model neurons with alpha-function shaped post-synaptic currents. Pairs of neurons are randomly connected with a connection probability of $p = 0.1$, leading to an average indegree of about 1000 excitatory and 200 inhibitory connections. The spike transmission delay is randomly sampled for each connection following a log-normal distribution.

The simulation experiments consist of two parts corresponding to the eyes-open and eyes-closed states, modeled via different input regimes. In the first input regime (background input), the input consists of spike trains sampled from inhomogeneous Poisson processes with a baseline rate of ν_{bg} Hz that is modulated with a 10 Hz sinusoidal oscillation. In the second input regime, the network additionally receives input consisting of spike trains from inhomogeneous Poisson processes with rates oscillating at 20 Hz representing the top-down innervation. The first state is meant to represent the eyes-closed, the second the eyes-open condition. Both input regimes provide independent, identically distributed input to each neuron, based on the corresponding rate profiles. During the simulation, the spiking activity of $1,000$ excitatory and 200 inhibitory neurons is recorded.

We explore three different top-down innervation modes by providing input to different subsets of neurons in the network. In the first innervation mode, projections target a subset consisting of 50% of neurons in the network in both the excitatory and inhibitory population. Secondly, we provide top-down modulations targeting the subset described above as well as all neurons in the network in an alternating way. Finally, neurons are assigned to five overlapping subsets, each consisting of 50% of the neurons in the populations. In this mode, each subset is innervated independently, leading to five different pathways. The modulations are only applied along one pathway at a time.

See Table 6.1, Table 6.2, and Table 6.3 for a full description of the network and the experiments in the style of Nordlie et al. (2009a).

To compare the simulated activity with the neural activity analyzed in Chapter 4, we calculate the time resolved firing rate of the single-neuron activities with a bin-size of 1 s (cf. Figure 4.2). Additionally, we computed a detrended version of this firing rate, subtracting the mean firing rate of that neuron during the different input regimes. Again, akin to Chapter 4 we employed a dimensionality reduction

based on principal component analysis (PCA) to visualize the activity in the different states (see Section 4.2.2). When multiple modulations are provided, we compute the PCA in a innervation mode specific manner. Secondly, when the entire activity was used for the principal component analysis, we call this full PCA.
For the simulations we used NEST (version 3.3, Spreizer et al. (2022)).

Table 6.1: General model description of balanced spiking neural network model

Model Summary	
Populations	two populations, one excitatory, one inhibitory
Connectivity	random connectivity
Neuron model	leaky integrate-and-fire model
Synapse model	alpha-function shaped postsynaptic current
Input	independent spike trains from inhomogeneous Poisson processes with given rate $r(t)$
Neuron and synapse model	
Subthreshold dynamics	$\frac{dV}{dt} = -\frac{V}{\tau_m} + \frac{I_{\text{syn}}(t)}{C_m}$, $I_{\text{syn}}(t) = J \frac{e}{\tau_{\text{syn}}}(t - t^* - d)e^{-(t-t^*-d)/\tau_{\text{syn}}} \times H(t - t^* - d)$
Spiking	If $V(t-) < \theta$ and $V(t+) \geq \theta$, 1. Set $t^* = t$ and $V(t) = V_0$, and 2. Emit spike with time stamp t^*.
Connectivity	
Type	pairwise Bernoulli, i.e., for each pair of neurons generate a synapse with probability p
Weights	fixed source- and target-population specific weights
Delays	log-normally distributed delays for excitatory and inhibitory neurons
Input	
Background	$r(t) = \max(0, \nu^{\text{bg}}_{\text{base}} + \nu^{\text{bg}}_{\text{amp}} \cdot \sin(2\pi f^{\text{bg}} \cdot t))$
Top-down modulation	$r(t) = \max(0, \nu^{\text{td}}_{\text{base}} + \nu^{\text{td}}_{\text{amp}} \cdot \sin(2\pi f^{\text{td}} \cdot t))$

Table 6.2: Simulation parameters of balanced spiking neural network

Population Parameters		
N_{ex}	10,000	number of excitatory neurons
N_{in}	2,000	number of inhibitory neurons
Connectivity Parameters		
p	0.1	connection probability
Neuron parameters		
τ_{m}	20 ms	membrane time constant
τ_{r}	2 ms	absolute refractory period
τ_{syn}	2 ms	postsynaptic current time constant
C_{m}	250 pF	membrane capacity
V_{m}	0 mV	resting potential
E_{L}	0 mV	membrane capacity
V_{reset}	0 mV	reset membrane potential
V_{th}	20 mV	threshold
Synapse parameters		
J_{EE}	6.4 pA	synaptic efficacy excitatory to excitatory
J_{IE}	9.5 pA	synaptic efficacy excitatory to inhibitory
g	4	relative inhibitory synaptic efficacy
J_{EI}	$-g \cdot J_{EE}$	synaptic efficacy inhibitory to excitatory
J_{II}	$-g \cdot J_{EE}$	synaptic efficacy inhibitory to inhibitory
Delay parameters		
μ_{ex}	1.5 ms	mean of lognormal distribution for excitatory connections
μ_{in}	0.75 ms	mean of lognormal distribution for inhibitory connections
$\sigma_{\mathrm{ex,in}}$	0.5 ms	standard deviation of lognormal distribution for all connections

Table 6.3: Stimulation parameters of balanced spiking neural network

Stimulus parameters: Background		
$\nu^{\mathrm{bg}}_{\mathrm{base}}$	8682 spikes/s	baseline rate
$\nu^{\mathrm{bg}}_{\mathrm{amp}}$	2170 spikes/s	amplitude
f	10 Hz	sinusoidal oscillation frequency
Stimulus parameters: Top-down signal		
$\nu^{\mathrm{td}}_{\mathrm{base}}$	0 spikes/s	base line rate
$\nu^{\mathrm{td}}_{\mathrm{amp}}$	723 spikes/s	amplitude
f	20 Hz	sinusoidal oscillation frequency
p^{td}	0.5	fraction of neurons targeted by top-down modulation in setup 1 and 3
p^{td}	1	fraction of neurons targeted by top-down modulation in setup 2

6.3 RESULTS

In the first innervation mode, where the additional external drive targets a subset consisting of 50% of all neurons (Figure 6.1a), the modulation leads to two distinct cluster in the PCA of the network activity (Figure 6.1d). Note that over time, the population-averaged firing rate varies only slightly with the modulation (Figure 6.1b). The observed shift in the neural manifold is not due to an increase in firing rate caused by additional external input (Figure 6.1e). Rather, it appears to be caused by a change in firing rate in some neurons, changing neurons with the highest activity (Figure 6.1f).

Simulation with the second innervation mode (two alternating inputs) support the notion that the change in state is not just due to an increase in the average firing rate of the network. The first input is the same as in innervation mode one (targeted top-down modulation); the other targets all neurons (untargeted top-down modulation, Figure 6.2a). We first note that the population-averaged firing rate varies with both additional input modes. In the untargeted case, the increase is smaller than when the targeted modulation is applied (Figure 6.2b). Separating both top-down modulation inputs, we observe that in the targeted case again two clusters appear in the modulation-specific PCA projection (Figure 6.2c). However, the untargeted modulation does not lead to a separation of manifolds in the PCA projection (Figure 6.2d). When applying the PCA to the entire simulation, the observations remain identical (Figure 6.2e). Since in the untargeted case the mean firing rate is not expected to change with respect to the background state, this provides evidence that indeed the change in the dominant groups of neurons causes the change of the neural manifold.

To further investigate the relationship between the highly active neurons and the neural manifolds, we analyze the activity of a simulation applying the third innervation mode. Here, five subsets each consisting of 50% of the neurons of both population are targeted alternately. In the full PCA projection, we observe six clusters, one belonging to each targeted subset, and one for the background state (Figure 6.3c). Again, the mean firing rate distribution remains similar across all modulations and the background input regime (Figure 6.3d). The change of the mean activity of single-neurons is, however, distinct for each of the five pathways (Figure 6.3e). From this, we conclude that different neural manifolds can be induced by an alteration of the most active neurons in a network.

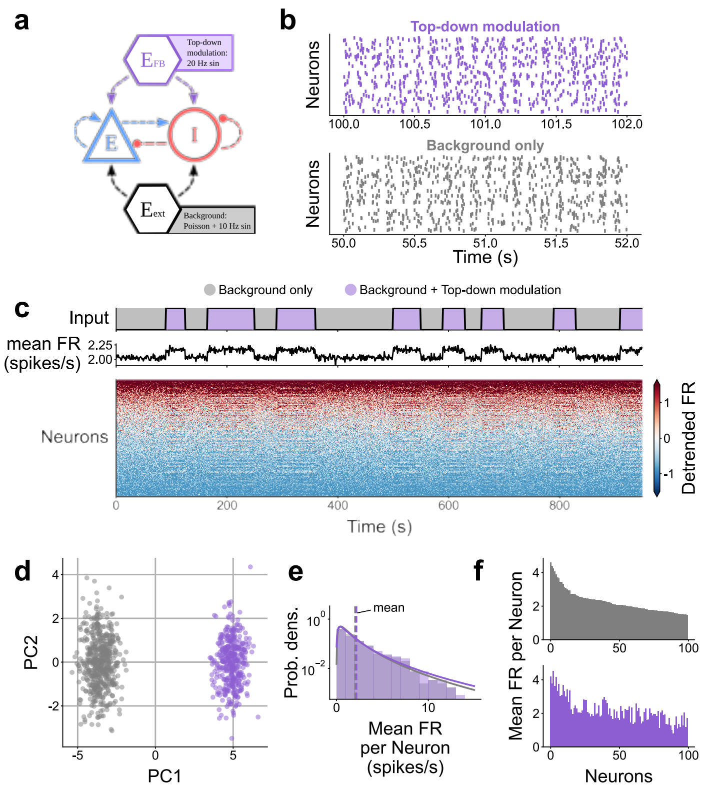


Figure 6.1: Simulation of a balanced spiking neural network with top-down modulation in the first innervation mode, in which the external drive targets a subset consisting of 50% of neuron in both populations. Different colors indicate the different input regimes. **a** Diagram of balanced random spiking neural network. Background input is provided constantly, top-down signals are provided intermittently. **b** Sample raster plots show spiking activity in the different input regimes when only the background input is provided. **c** Time evolution of input regimes, mean firing rate (FR) and detrended firing rates ordered by magnitude of the rate. **d** First two principal components of the firing rate (binsize = 1 s). Colours indicate the different input regimes. **e** Distribution of mean firing rate per neuron is almost identical between the two regimes. **f** Mean firing rate of the 100 most active neurons. The top-down modulation changes the mean firing rates of certain neurons, in both the positive and negative directions.

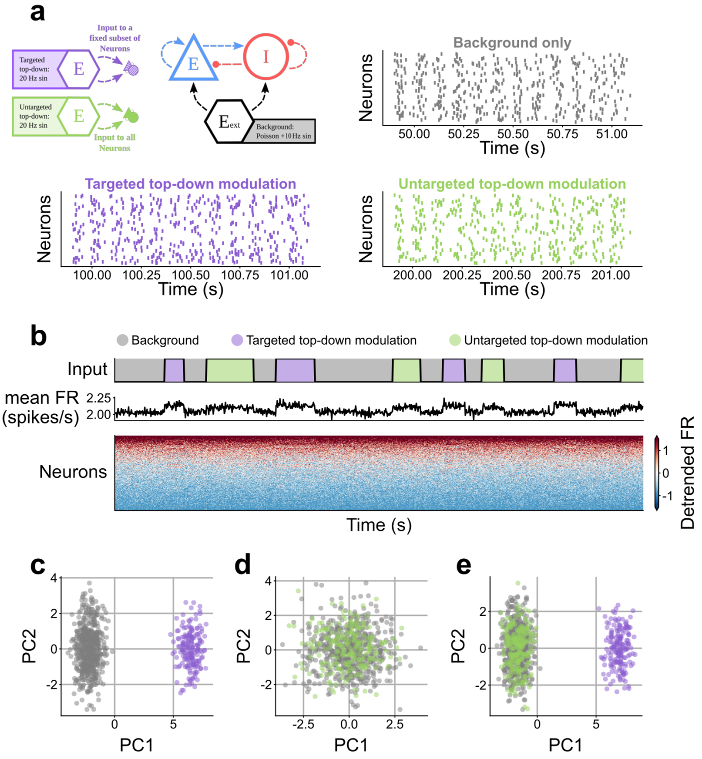


Figure 6.2: Simulation of a balanced spiking neural network with top-down modulations in the second innervation mode, in which an alternating external input to a subset as in the first innervation mode and to all neurons in the network is provided. Different colors indicate the different input regimes. **a** Diagram of balanced random spiking neural network with background input and both modulations. Sample raster plots show spiking activity in different input regimes. **b** Time evolution of input regimes, mean firing rate, and detrended firing rates ordered by magnitude of the rate in the when only the background input is provided. **c** First two principal components of the modulation-specific PCA of the firing rate of background state and the targeted modulation. **d** First two principal components of the modulation-specific PCA of the firing rate of background state and the untargeted modulation. **e** First two principal components of the full PCA of the firing rate of background state and both modulations states.

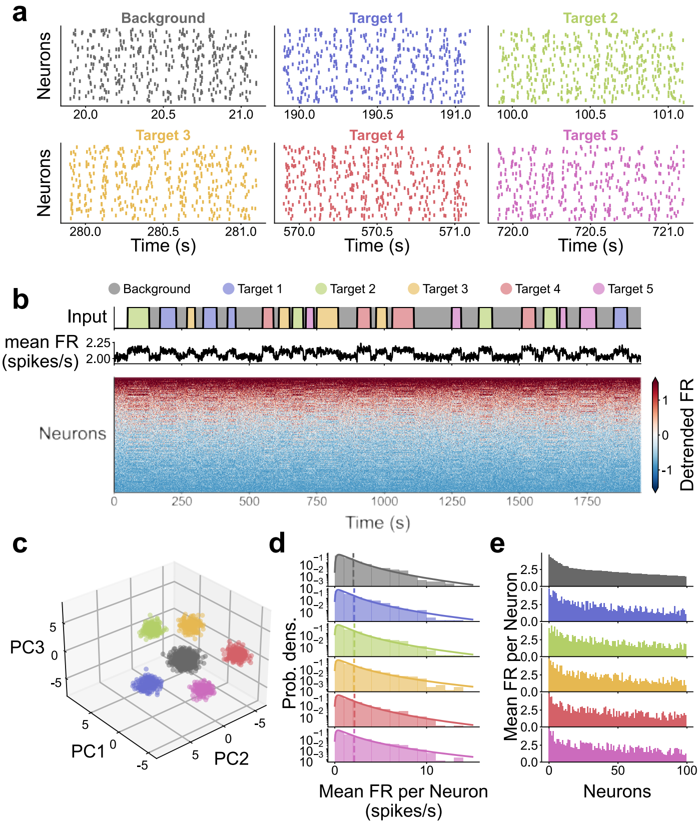


Figure 6.3: Simulation of a balanced spiking neural network with top-down modulations in the third innervation mode, in which alternating external input target five different, overlapping subsets of the network. Each subset consists of 50% randomly sampled neurons in both populations. **a** Sample raster plots show spiking activity in different input regimes. **b** Time evolution of input regimes, mean firing rate (FR), and detrended firing rates ordered by magnitude of the rate when only the background input is provided. **c** First three principal components of the full PCA of the firing rate of background state and the five targeted modulations. **e** Distribution of mean firing rate per neuron is almost identical between different input regimes and projection patterns. **f** Mean firing rate of the 100 most active neurons in different input regimes and projection patterns. The top-down modulation changes the mean firing rates of certain neurons, giving rise to a distinct activity profile.

6.4 DISCUSSION

We here show that a change in neural manifold can be induced by a targeted top-down modulation in a spiking neural network model. This finding is consistent with the hypothesis developed in Chapter 4. Thereby, we give credibility the mechanism suggested there of top-down modulations from V4 causing the observed shift in the neural manifold of resting-state activity in V1. Furthermore, the different manifolds in the *in silico* experiments are not due to an increase in firing rate. Our results show that they are caused by different groups of most active neurons in the network. This can be understood via the detrended single-neuron firing rates (Figure 6.1c, Figure 6.2b, Figure 6.3c): we observe that the targeted top-down modulation alters the activity of the neurons. Targeting different subsets leads to different alterations in the activity pattern. The untargeted input additional to the background stimulation leads to no discernible change. These alterations cause a shift in the high-dimensional neural activity, leading to separate neural manifolds. Additionally, innervating the network along the same pathway leads to the same high-dimensional activity states. This finding in the *in silico* experiments supports the *in vivo* observation that the V4-to-V1 signals induce the neural manifolds, and suggest that the top-down signals from V4 target always a similar subset of V1 neurons in the macaque brain.

The spiking neural network used here is inspired by the well-known model studied by Brunel (2000). We limited the analysis of the spiking activity to the presence of neural manifolds, because our model was ill-suited to study the dimensionality, given that average cross-correlation is known to cancel out in balanced spiking networks (Doiron et al., 2016; Tetzlaff et al., 2012). Future work could use more complex models —such as clustered networks, see e.g. Doiron et al. (2016), Rostami et al. (2022), and Schaub et al. (2015) —to study the effects of correlated inputs with realistic power spectra on the dimensionality and elucidate whether the top-down signals can directly induce the observed increase in the dimensionality during the eyes-open periods.

Manifolds in the primary visual cortex of macaque *in vivo* might be additionally sustained by other means. Potential mechanisms include N-methyl-D-aspartate (NMDA) receptors (central to the top-down communication from V4 to V1, see e.g. Self et al. (2013) and Van Loon et al. (2016)) or cell-type specific recurrent connectivity (Dahmen et al., 2020). The former is believed to contribute to decorrelation during eyes-open states (Herrero et al., 2013). Additionally, the slow timescales of NMDA receptors could support the network dynamics in remaining in the corresponding manifold, especially in view of non-constant top-down signals during eyes-open periods (cf. Figure 4.10). The latter has been shown to control the dimensionality of neural networks, and could play a role in controlling distinct effective connec-

tivities in a rigid (at least at the time-scales relevant for the analysis conducted in Chapter 4) anatomical network.
The exploration of these potential mechanism, their viability for supporting neural manifolds in the V1 eyes-open state, are possible future extensions of the work presented here.

7

A SPATIAL MODEL OF THE MACAQUE PRIMARY VISUAL CORTEX

Author contributions:

Under the supervision of Prof. Dr. Sacha van Albada (SvA) and Prof. Dr. Markus Diesmann (MD), the author and Jasper Albers (JA) conceptualized the work presented in this chapter. The author and JA jointly developed a software pipeline to derive the network connectivity, estimated characteristic length scales of the medium-range connectivity, obtained constraints for the orientation tuning map as well as patchy connectivity and derived a push-pull connectivity scheme. The author and JA jointly wrote simulation code based on the Mesocircuit model code[1]for a spiking neural network in NEST implementing the derived connectivity. The author conducted the analysis of the obtained connectivity as well as the numerical simulation, and the analyses were refined in ongoing discussions with JA, SvA, and MD.

7.1 INTRODUCTION

In the last chapter, we employed a spiking neural network model with a realistic connection probability, plausible single-neuron indegrees and biologically constrained evoked post-synaptic potentials to investigate the hypothesis whether a top-down modulation from V4 can induce a change in the neural manifold in V1. The simple model used there could indeed confirm this mechanism as a potential cause for the separation of neural manifolds observed in Chapter 4 with *in silico* experiments. While suitable for the addressed research question, the model ignores features that are fundamental to the architecture of cortical circuits. The most apparent neglected organizational principles are the *laminar* and *lateral* (i.e. vertical and horizontal with respect to the cortical surface) structure of connectivity within a cortical area. The simple model thus has limited validity when investigating the relationship between the structure of the cortex and its dynamics. In the following, we explore the aforementioned organizational principles in more detail.

The neocortex is classically subdivided into six *layers* or *laminae* based on occurring cell-types and their densities (Abeles, 1991). The size of these layers, the number of neurons and synapses they contain, as well as the number of sub-layers vary with cortical area. The primary visual cortex (V1) of macaque, for example, has a dominant layer 4 (also referred to as *granular layer*) which is strongly innervated by projections from the thalamic lateral geniculate nucleus (LGN) (Kandel et al., 2013). Via this pathway, visual input is provided to V1. The laminar organization is believed to underlie the cortical computation. *Cortical columns*, comprising the neurons in all layers under a small patch of brain surface, have been hypothesized to be the functional building block of the cortex (Mountcastle, 1957).

Additionally to the laminar structure, cortex also exhibits a prominent lateral organization. We here distinguish three types of horizontal connections: short- (approximately < 200 mm), medium- (dominating between 200 mm to 500 mm), and long-range (appearing at distances >500 mm) connectivity. The former two comprise the connections within the local cortical network that are established by the axon collaterals close to the cell body (Voges et al., 2010). Short-range connectivity often is cell-type specific and is organized via motifs in the network topology, see e.g. Peng et al. (2021) and Perin et al. (2011). On a medium-range, less is known about the specificity of neuronal connectivity. The probability of establishing connections on this scale can be assumed to decay with an exponential profile (Markov et al., 2011). Long-range connections often terminate in quasi-periodic patches separated by a positive distance, henceforth named patchy connectivity (Angelucci et al., 2002; Gilbert and Wiesel, 1989; Rockland and Lund,

[1] https://github.com/INM-6/mesocircuit-model

1982; Tanigawa et al., 2005). Patchy connections are made by axons that travel up to several millimeters parallel to the cortical surface and branch out at certain sites (Voges et al., 2010). They are best investigated in visual areas but occur also in other sensory cortices as well as the motor cortex (Lund et al., 1993). It is noteworthy that they are found in the primary visual cortex only in some species, including all primates and carnivores but not in for example gray squirrels (Van Hooser et al., 2006).

In this chapter, we set out to construct a biologically constrained spiking neural network model of macaque primary visual cortex integrating both its layered and its spatial structure. V1 is the largest cortical area in the macaque (Sincich and Horton, 2005). The primary visual cortex not only receives strong visual input from the LGN. It also sends feed-forward projections to and receives feedback projections from other visual areas, see e.g. Maunsell and Newsome (1987). It is considered to be the first stage of cortical visual processing and the lowest area in the visual hierarchy (Felleman and Van Essen, 1991). In V1 (but also in e.g. V2, V4) the visual scene is mapped in a continuous but non-linear way onto the cortex, leading to nearby neurons having overlapping receptive fields (Tootell et al. (1982) and Vanni et al. (2020), known as *retionotopy*).

Beyond this direct spatial representation, Hubel and Wiesel (1959) showed (initially in cat) that neurons in V1 have more intricate response properties. Those include stimulus selectivity, e.g. to orientations, spatial/temporal frequency and direction. Here, we focus on the orientation selectivity, i.e. the dependence of the firing rate of single-neurons on the angle of a grating moving across the visual scene. Cells of given orientation preference —determined by their strongest response in such experiments —cluster quasi-periodically in so-called iso-orientation domains. This organization gives rise to the orientation map, which also present in some higher visual areas such as V2 or V4 (Blasdel and Salama, 1986). Experimental work suggests that in V1 no orientation is overrepresented (Ju et al., 2021). In the primary visual cortex of macaque, the orientation map mediates the cortical organization: it has been shown that patchy connections preferentially link neurons with similar orientation preference over long distances (Stettler et al., 2002). Iso-orientation domains thus are of great importance for long-range patchy connections.

Taken together, macaque V1 is a good candidate for building a digital twin in the form of a biologically plausible, large-scale spiking neural network: its anatomy and physiology have been investigated in a long line of research that continues to this day, see e.g. Trepka et al. (2022). Additionally, there is a close link between the structure of the primary visual cortex (e.g. retionotopy, patchy connectivity, etc.) and its (putative) function.

Multiple spiking neural network models of cortical sensory areas taking into account the layered and/or spatial structure of cortex have been developed in the past years. Potjans and Diesmann (2014) constructed a spiking neural network model of the cortical microcircuit below 1 mm^2 patch of brain surface. To obtain the underlying connectivity map of the local cortical network, they integrated anatomical data across multiple species. Simulations of the network exhibit asynchronous and irregular spiking activity and realistic, cell-type specific firing rates in the resting-state. Building on this, Senk et al. (2023) derived an upscaled version of the cortical microcircuit model using horizontally distance-dependent connection profiles specific to excitatory and inhibitory connections. As a scaffold for the local connectivity, they used a version of the model by Potjans and Diesmann (2014) adapted to neuron and synapse numbers in macaque V1 (Schmidt et al., 2018b). The model by Senk et al. (2023) represents the neural tissue below a 4 mm × 4 mm patch of cortex. The authors computed LFP predictions of the cortical activity using the simulated spiking activity with a hybrid scheme (Hagen et al., 2016). There, LFP predictions are determined using multicompartmental neuron models with a realistic distribution of synapses to faithfully model the synaptic and ionic currents, while the network activity is based on point neuron model simulations. The spiking activity in Senk et al. (2023) exhibits heterogeneous firing rates as well as spiking variability in agreement with experimentally recorded activity, and is excitable in the sense that upon a focused stimulation a spatial spread of activity with realistic propagation speed can be observed. Additionally, in a state of spontaneous network activity, the computed LFP are spatially correlated and coherent, their amplitudes agree with experimentally observed ones.

A different approach was undertaken by Antolík et al. (2018). There, the authors built a spatially organized spiking neural network model of layer 2/3 and 4 of cat visual cortex, sparing the infragranular layers. The network is recurrently connected in a layer- and population-specific way with distance dependent connection probabilities. Among other things, endowing the circuit with a pre-computed orientation map, they additionally include long-range patchy connectivity. Antolík et al. (2018) suggest a list of validation experiments to be performed with the model. The *in silico* spiking activity is used to check for consistency with data from neural recordings. Their network satisfies these tests. In contrast to the work of Potjans and Diesmann (2014) and Senk et al. (2023), however, the model is not full-scale. The density of both neurons and synapses are reduced in comparison with the tissue in cat V1.

Billeh et al. (2020), on the other hand, constructed a full-scale model of the mouse primary visual cortex, including both the layered and spatial structure. The network model combines a large body of anatomical and electrophysiological data, primarily obtained in the Allen Institute for Brain Science[2]. They simulate the network in two ways: using single-neuron models taking into account the cells' morphologies as well as point neuron models. The simulated activity using either approach is consistent with experimental data recorded in mouse visual cortex and lead to qualitatively similar conclusions.

The efforts presented here aim at incorporating the laminar structure of cortex, the spatial organization of intra-areal connectivity on both local and long-range scales, and realistic numbers of neurons and synapses into a comprehensive model macaque primary visual cortex. We thus directly extend the work by Senk et al. (2023). In comparison with Antolík et al. (2018), our model represents the full neural tissue below the cortical surface with biologically plausible numbers of neurons and synapses. In contrast to Billeh et al. (2020), our network incorporates the orientation preference map as well as long-range patchy connections mediated by the iso-orientation domains. The latter are not present in mouse visual cortex which exhibits a "salt-and-pepper" organization (Ohki and Reid, 2007).

To constrain our model of macaque V1, we preferentially use experimental results from that species. However, this is not always possible due to lacking data. When necessary, insights gained from studies conducted on other species are employed.

7.2 METHODS

We integrate a large body of anatomical and physiological data to construct a biologically plausible model of macaque primary visual cortex.

In the model, the cortical layers 2/3A, 3B, 4AB, 4C, 5, and 6 are distinguished. We thus use a finer laminar resolution than in the works by Potjans and Diesmann (2014), Schmidt et al. (2018b), and Senk et al. (2023). The separation into layers 2/3A and 3B is motivated by the cortical counterstream hypothesis, suggesting that layers play different roles in receiving and sending feed-forward and feed-back projections (Markov et al., 2013). We moreover split layer 4 into 4AB and 4C: layer 4C is the primary target of thalamic projections in the granular layer (Garcia-Marin et al., 2017) and has different lateral connectivity in comparison with layer 4AB.

In each layer there is an excitatory and an inhibitory population, denoted by e.g. L2/3AE or L2/3AI for the respective populations. We generally aim at representing the tissue below an at least 4 mm × 4 mm patch of cortical surface, although the side length can be regarded as a free parameter in the subsequent derivations. However, a sufficiently large side length is necessary for incorporating spatial connections in a reasonable way in the model. Hence, only some results regarding the network architecture remain valid in smaller models.

[2] https://alleninstitute.org/division/brain-science/

Synapses are classified as intra-areal, inter-areal (also referred to as cortico-cortical) or thalamo-cortical.
Regarding the lateral structure of cortical connectivity, we differentiate between long-range patchy, medium-range isotropic and short-range push-pull connectivity (see Section 7.1).
In the following, neural populations in our model are denoted by $A, B, ... \in \{\text{L2/3AE}, \text{L2/3AI}, ..., \text{L6I}\}$, layers by $u, v, ... \in \{\text{L2/3A}, ..., \text{L6}\}$ and the excitatory or inhibitory nature of synapses by $X, Y \in \{\text{E}, \text{I}\}$. We write "synapse of type X" to indicate that a synapse stems from an excitatory or inhibitory neuron (the appropriate term has to be chosen depending on X).

7.2.1 *Number of neurons and synapses*

Given a cortical layer u in V1, the data provided in Vanni et al. (2020) (cf. Table 2, combining multiple sources, Beaulieu et al. (1992), Fitzpatrick et al. (1987), Giannaris and Rosene (2012), Hendry et al. (1987), and O'Kusky and Colonnier (1982)) yields the number of neurons per layer, N^u (as well as the number of excitatory and inhibitory neurons N^u_E, N^u_I), and the total number of synapses per layer S^u. Each synapse can be assigned to one of the aforementioned classes, leading to the decomposition:

$$S^u = S^{\text{ia},u} + S^{\text{cc},u} + S^{\text{tha},u} \tag{7.1}$$

Here, $S^{\text{ia},u}$ denotes the intra-areal, $S^{\text{cc},u}$ the cortico-cortical, and $S^{\text{tha},u}$ the thalamo-cortical number of synapses in layer u.
In Vanni et al. (2020) layer 3A and 3B are not distinguished, only information for layer 3 is provided. Assuming a uniform distribution of cells and synapses in layer 3, we subdivide the number of neurons and synapses according to the mean thicknesses of layers 3A and 3B, combining data collected by Schmidt et al. (2018b).

7.2.2 *Laminar structure of cortical connectivity*

To derive a layer-resolved connectivity map for the intra-areal synapses, we combine connection probabilities (data from Binzegger et al. (2004) of cat V1, there is no equally complete list of connection probabilities for macaque to be found in the literature) with the "anatomical connection strength" (here referred to as projection strength) in macaque primary visual cortex compiled by Vanni et al. (2020) (cf. their Figure 2).

7.2.2.1 *Connection probabilities*

Binzegger et al. (2004) reconstructed 39 neurons in cat primary visual cortex using light microscopy. Each neuron was assigned to a neuron type determined by its layer of origin, the morphology of its axonal and dendritic trees, and whether it was excitatory or inhibitory. In this way, for example, pyramidal and spiny stellate cells in layer 4 are distinguished in their data. Thus, the neuron types used by Binzegger et al. (2004) are more fine grained than the distinction in neural populations used here: post-synaptic types can be assigned to populations in cortical layers, pre-synaptic types additionally include cortico-cortical and thalamic connections. The laminar resolution of the data from Binzegger et al. (2004) comprises layers 1, 2/3, 4, 5, 6. Employing a modified version of Peters' rule —synapses between neurons are established by chance depending only on the vicinity of a presynaptic axon and postsynaptic dendrite (Braitenberg and Schüz, 1991) —Binzegger et al. (2004) derived the average number of synapses between pre- and postsynaptic neurons of given types i and j. This approach rests on the assumption that synapses are evenly distributed on the target processes. Taking into account numbers of neurons and synapses in cat V1 (Beaulieu and Colonnier, 1985; Beaulieu and Colonnier, 1983), they computed the average number of synapses between neurons of type i and j in layer ν, denoted by $s_{ij}^{|\nu}$. From this, the following quantities can be obtained (cf. Izhikevich and Edelman (2008)):

- the probability for an existing connection onto a neuron of type i with synapse in layer u to stem from a neuron of type j, $p_{|ij}^{|u}$
- the total number of synapses onto neurons of type i that reside in layer u, s_i^u
- relative occurrence of neurons of type i, c_i

In the following, we employ these quantities in our derivations. Regarding notation: Here and in the following the vertical bar | indicates that a quantity under consideration is conditional on the variable after the bar. In case that the quantity is a probability, e.g. $p_{|ij}^{|v}$, this implies that $\sum_j p_{|ij}^{|v} = 1$.

INTRA-AREAL CONNECTIONS We ultimately aim at calculating the probability that an intra-areal synapse in layer u targets population A and is excitatory or inhibitory, X, denoted by $\bar{p}_A^{|u,X}$. The dependence on u is important here: while the cell body of a neuron from population A resides in a certain layer, say v, it does not necessarily follow that $v = u$. Indeed, intra-areal connections onto a, e.g., pyramidal neuron are in part established on its apical dendrite outside its layer of origin. To calculate $\bar{p}_A^{|u,X}$, we first consider probability for an intra-areal synapse of type X in layer u to establish a connection that targets population A, which can be written as

$$p_A^{|u,|X} = \frac{\#\text{ intra-areal } X \text{ synapses in layer } u \text{ targeting population } A}{\#\text{ intra-areal } X \text{ synapses in layer } u} . \tag{7.2}$$

Here and in the following "#" is shorthand notation for "number of". Given the total number of neurons N and the above quantities we have

$$\begin{aligned} N \cdot c_i &= \text{total number of neurons of type } i \\ N \cdot c_i \cdot s_i^u &= \text{total number of synapses in layer } u \text{ with} \\ &\quad\ \text{a postsynaptic neuron of type } i \\ N \cdot c_i \cdot s_i^u \cdot p_{|ij}^{|u} &= \text{total number of synapses in layer } u \text{ with} \\ &\quad\ \text{a postsynaptic neuron of type } i \text{ that have} \\ &\quad\ \text{a presynaptic neuron of type } j \end{aligned}$$

Thus

$$p_A^{|u,|X} = \frac{\sum\limits_{i\in A} \sum\limits_{j\in X\cap\text{ia}} N\, c_i\, s_i^u\, p_{|ij}^{|u}}{\sum\limits_{i} \sum\limits_{j\in X\cap\text{ia}} N\, c_i\, s_i^u\, p_{|ij}^{|u}} = \frac{\sum\limits_{i\in A} \sum\limits_{j\in X\cap\text{ia}} c_i\, s_i^u\, p_{|ij}^{|u}}{\sum\limits_{i} \sum\limits_{j\in X\cap\text{ia}} c_i\, s_i^u\, p_{|ij}^{|u}} , \tag{7.3}$$

where $j \in X \cap \text{ia}$ is a symbolic notation for all cortical neuron types j (and thus establishing intra-areal connections) that have X synapses. Using the expression for the total number of intra-areal X synapses in layer u (the denominator of Equation 7.3)

$$S_X^{\text{ia},u} = \sum_i \sum_{j\in X\cap\text{ia}} N\, c_i\, s_i^u\, p_{|ij}^{|u} \tag{7.4}$$

we determine the fraction of intra-areal synapses in that layer:

$$f^{\text{ia},u} = \frac{\sum\limits_{i} \sum\limits_{j\in\text{ia}} N\, c_i\, s_i^u\, p_{|ij}^{|u}}{\sum\limits_{i} \sum\limits_{j} N\, c_i\, s_i^u p_{|ij}^{|u}} = \frac{\sum\limits_{i} \sum\limits_{j\in\text{ia}} N\, c_i\, s_i^u\, p_{|ij}^{|u}}{\sum\limits_{i} N\, c_i\, s_i^u} . \tag{7.5}$$

Here we used that $\sum_j p_{|ij}^{|u} = 1$. Similarly, the fraction of cortico-cortical synapses in layer u, $f^{\text{cc},u}$, as well as the fraction of thalamo-cortical synapses in layer u, $f^{\text{tha},u}$ are calculated. We additionally calculate the fraction of intra-areal resp. cortico-cortical synapses in layer u relative to the sum of intra-areal and cortico-cortical synapses in that layer as

$$\bar{f}^{\text{ia},u} = \frac{f^{\text{ia},u}}{f^{\text{ia},u} + f^{\text{cc},u}} , \qquad \bar{f}^{\text{cc},u} = \frac{f^{\text{cc},u}}{f^{\text{ia},u} + f^{\text{cc},u}} \tag{7.6}$$

With the absolute number of synapses the fraction of intra-areal excitatory/inhibitory synapses in a given layer u can be determined by

$$\$_E^{|u} = \frac{S_E^{\mathrm{ia},u}}{S_E^{\mathrm{ia},u} + S_I^{\mathrm{ia},u}} \qquad \text{and} \qquad \$_I^{|u} = 1 - \$_E^{|u} \, . \tag{7.7}$$

Noting that

$$S_E^{\mathrm{ia},u} + S_I^{\mathrm{ia},u} = S^{\mathrm{ia},u} = \sum_i \sum_{j \in \mathrm{ia}} N \, c_i \, s_i^u \, p_{|ij}^{|u} \, , \tag{7.8}$$

we finally calculate

$$\begin{aligned} & \$_X^{|u} \cdot p_A^{|u,|X} \\ = & \frac{\sum\limits_i \sum\limits_{j \in X \cap \mathrm{ia}} N \, c_i \, s_i^u \, p_{|ij}^{|u}}{\sum\limits_i \sum\limits_{j \in \mathrm{ia}} N \, c_i \, s_i^u \, p_{|ij}^{|u}} \cdot \frac{\sum\limits_{i \in A} \sum\limits_{j \in X \cap \mathrm{ia}} c_i \, s_i^u \, p_{|ij}^{|u}}{\sum\limits_i \sum\limits_{j \in X \cap \mathrm{ia}} c_i \, s_i^u \, p_{|ij}^{|u}} \\ = & \frac{\sum\limits_{i \in A} \sum\limits_{j \in X \cap \mathrm{ia}} c_i \, s_i^u \, p_{|ij}^{|u}}{\sum\limits_i \sum\limits_{j \in \mathrm{ia}} c_i \, s_i^u \, p_{|ij}^{|u}} = \bar{p}_A^{|u,X} \, , \end{aligned} \tag{7.9}$$

which we identify with the desired quantity, i.e. the probability that an intra-areal synapse in layer u targets population A and is of type X.

Since the laminar granularity of the used data is coarser than the one of the model, we need to adapt $\bar{p}_A^{|u,X}$ to account for the case of sublayers and sub-populations of layers 2/3 and 4. Denoting by $a \uplus b$ the union of disjoint objects (here layers or populations) a and b, we decompose the respective populations and layers as

$$A = A_1 \uplus A_2 \tag{7.10}$$

$$u = u_1 \uplus u_2 , \tag{7.11}$$

where A is a population in layer u, A_1 and A_2 are the sub-populations of A in the respective sublayers u_1 and u_2. Consider a synapse that resides in a different layer v. We note that

$$\bar{p}_A^{|v,X} = \bar{p}_{A_1}^{|v,X} + \bar{p}_{A_2}^{|v,X} \tag{7.12}$$

and define

$$\bar{p}_{A_1}^{|v,X} = \frac{N_1}{N_1 + N_2} \, \bar{p}_A^{|v,X} , \qquad \bar{p}_{A_2}^{|v,X} = \frac{N_2}{N_1 + N_2} \, \bar{p}_A^{|v,X} \tag{7.13}$$

with N_1 and N_2 being the number of neurons of the respective sub-populations. This is based on the assumption that synapse of type X that reside in layer v uniformly target neuron in the subpopulation of

A. In the case that v itself consists of two sublayers, $v = v_1 \uplus v_2$, we assume no dependence of the connection probability on the sub-layer that the synapse lies in, i.e.

$$\bar{p}_A^{|v_1,X} = \bar{p}_A^{|v_2,X} = \bar{p}_A^{|v,X} \tag{7.14}$$

Thus, Equation 7.12 becomes

$$\bar{p}_A^{|v,X} = \underbrace{\bar{p}_{A_1}^{|v_1,X}}_{\frac{N_1}{N_1+N_2}\bar{p}_A^{|v,X}} + \underbrace{\bar{p}_{A_2}^{|v_1,X}}_{\frac{N_2}{N_1+N_2}\bar{p}_A^{|v,X}} = \bar{p}_{A_1}^{|v_2,X} + \bar{p}_{A_2}^{|v_2,X} \tag{7.15}$$

Finally, we consider the case where the synapse resides in the same (sub-)layer as the postsynaptic neuron, denoted by u. While the above described approach is sensible for connections targeting a population not located in the same sublayer, it leads to unexpected results if the target population is in that sublayer.

Indeed, if decomposed according to the number of neurons in u_i relative to the number of neurons in u, we have, denoting by A_i a subpopulation in u_i, either

$$\bar{p}_{A_1}^{|u_1,X} > \bar{p}_{A_1}^{|u_2,X} \text{ and } \bar{p}_{A_2}^{|u_1,X} > \bar{p}_{A_2}^{|u_2,X} \tag{7.16}$$

or

$$\bar{p}_{A_1}^{|u_1,X} < \bar{p}_{A_1}^{|u_2,X} \text{ and } \bar{p}_{A_2}^{|u_1,X} < \bar{p}_{A_2}^{|u_2,X} \tag{7.17}$$

since

$$\bar{p}_{A_j}^{|u_i,X} = \frac{N_j}{N_1+N_2}\bar{p}_A^{|u_i,X} = \frac{N_j}{N_1+N_2}\bar{p}_A^{|u,X} \tag{7.18}$$

is independent of i and thus

$$\bar{p}_{A_1}^{|u_1,X} = \bar{p}_{A_1}^{|u_2,X}, \quad \bar{p}_{A_2}^{|u_1,X} = \bar{p}_{A_2}^{|u_2,X}. \tag{7.19}$$

In Equation 7.16, for example, the second part is problematic since one expects the connection probability to be largest to the population within the layer, and vice versa for Equation 7.17.

To circumvent this problem, we introduce an ad-hoc splitting factor $\alpha = 0.8$ replacing the relative number of neurons in the sublayers. We hence set

$$\bar{p}_{A_1}^{|u_1,X} := \alpha \cdot \bar{p}_A^{|u,X}, \quad \bar{p}_{A_2}^{|u_1,X} := (1-\alpha) \cdot \bar{p}_A^{|u,X} \tag{7.20}$$

and

$$\bar{p}_{A_1}^{|u_2,X} := (1-\alpha) \cdot \bar{p}_A^{|u,X}, \quad \bar{p}_{A_2}^{|u_2,X} := \alpha \cdot \bar{p}_A^{|u,X}. \tag{7.21}$$

This concludes the derivation of intra-areal connection probabilities.

CORTICO-CORTICAL CONNECTIONS Similarly, we calculate the respective probability for cortico-cortical connections. We assume all cortico-cortical connections to be excitatory. The probability for a cortico-cortical synapse in layer u to establish a connection with population A is

$$p_A^{|u,\mathrm{cc}} = \frac{\#\text{ cortico-cortical synapses in layer } u \text{ targeting population } A}{\#\text{ cortico-cortical synapses in layer } u} \tag{7.22}$$

and thus

$$p_A^{|u,\mathrm{cc}} = \frac{\sum_{i\in A} N\, c_i\, s_i^u\, p_{|i,\mathrm{cc}}^{|u}}{\sum_i N\, c_i\, s_i^u\, p_{|i,\mathrm{cc}}^{|u}} = \frac{\sum_{i\in A} c_i\, s_i^u\, p_{|i,\mathrm{cc}}^{|u}}{\sum_i c_i\, s_i^u\, p_{|i,\mathrm{cc}}^{|u}}\,, \tag{7.23}$$

where $p_{|i,\mathrm{cc}}^{|u}$ is the probability for a synapse in layer u that targets population i has a presynaptic neuron of cortico-cortical origin. Again, the probabilities $p_A^{|u,\mathrm{cc}}$ have to be split due to the different layer resolution used Binzegger et al. (2004). We proceed as in Equation 7.12 and Equation 7.14 for the relevant layers.

THALAMO-CORTICAL CONNECTIONS Finally, the probability for a thalamo-cortical synapse (i.e. a synapse terminating in the cortex with presynaptic neuron residing in the thalamus) in layer u to establish a connection with population A is determined by

$$p_A^{|u,\mathrm{tha}} = \frac{\#\text{ thalamo-cortical synapses in layer } u \text{ targeting population } A}{\#\text{ thalamo-cortical synapses in layer } u}. \tag{7.24}$$

As for cortico-cortical connections, we assume all thalamo-cortical connections to be excitatory. We again modify the laminar resolution of the connection probabilities $p_A^{|u,\mathrm{tha}}$ as above (Equation 7.12 and Equation 7.14), except for layer 4. Assuming that synapses terminating in one of the sub-layers of layer 4 establish only connections within this sub-layer, we set

$$p_{\mathrm{L4A/BE}}^{|\mathrm{L4A/B,th}} = p_{\mathrm{L4E}}^{|\mathrm{L4,th}}, \quad p_{\mathrm{L4CE}}^{|\mathrm{L4A/B,th}} = 0 \tag{7.25}$$

as well as

$$p_{\mathrm{L4CX}}^{|\mathrm{L4C,th}} = p_{\mathrm{L4X}}^{|\mathrm{L4,th}}, \quad p_{\mathrm{L4A/BX}}^{|\mathrm{L4C,th}} = 0. \tag{7.26}$$

7.2.2.2 *Anatomical projection strengths*

Vanni et al. (2020) reviewed the anatomy and physiology of (among other areas) macaque V1 and compiled a “connection strength” matrix of the intra-area and thalamo-cortical connectivity. In the following we refer to this as *projection strength*. For each of the investigated connections, they estimated the strength as either absent, sparse, medium, or dominant by evaluating the number of stained presynaptic somata or axon terminals in a large body of retro- and anterograde tracing studies.

To get a quantitative estimate of this strength, we map these qualitative categories to the numeric values 0, 1, 2, and 4. Thus, in the following calculations absent connections are accounted for with a value of 0, sparse with a value of 1, medium with value of 2 and dominant with a value of 4. The specific values are to be interpreted as order of magnitudes in the base 2. After this preprocessing we obtain:

- $\eta_{u'v'}^{\text{fine},X}$ = the anatomical projection strength for connections of X synapses from $v' \to u'$, where u' denotes the target and v' in the resolution used by Vanni et al. (2020)

INTRA-AREAL CONNECTIONS The layer resolution for the reported data is finer than the one of our model. Thus, the anatomical projection strengths are aggregated to match the layers used here:

$$\eta_{uv}^{X} = \frac{\sum_{u'\in u, v'\in v} \eta_{u'v'}^{\text{fine},X}}{\#u\#v}, \tag{7.27}$$

i.e. summing the finer relative connections strengths over the sublayers of layers u and v respectively, and normalising by the product of the numbers of sublayers of u and v.

From η_{uv}^{X} we derive the probability for an X synapse terminating in u to originate in v:

$$q_{|uv}^{|X} = \frac{\eta_{uv}^{X}}{\sum\limits_{v} \eta_{uv}^{X}} \tag{7.28}$$

Due to the relative anatomical projection strength being obtained from tracing studies, pre-synaptic neurons within the uptake zone of the stain are systematically underestimated. This mainly leads to a reduced projection strength in intra-layer connections, which we amend in the following. Writing $l_{uv}^{X}(r)$ for the density of labeled neurons in layer u with X synapses after injection of a hypothetical tracer into layer v, we assume an exponential decrease in density (Markov et al., 2011) of labeled cells:

$$l_{uv}^{X}(r) \sim e^{-\frac{r}{\lambda_{uv}^{X}}} . \tag{7.29}$$

Here r is the distance between neurons in the three-dimensional cortical space. (Markov et al., 2011) show that the density of labeled neurons in a two-dimensional projection parallel to the cortical surface (i.e. "looking from above" onto the cortex) decays according to Equation 7.29, where the distance here is determined as the distance on the projection plane. Since we in the following will only focus on

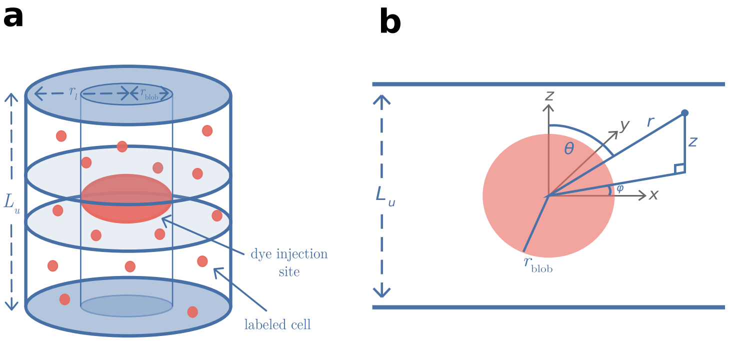


Figure 7.1: Schematics for amendment of projection strength **a** Model assumption and meaning of r_l and r_{blob}. **b** Coordinate system used for integration, shows the relation between r, θ and z.

the case where $u = v$, we argue that Equation 7.29 is a reasonable approximation, even in the three-dimensional case.
Amending η^X_{uu} rests on the following assumptions (Figure 7.1a):

- the uptake zone of the dye can be approximated by a sphere with radius r_{blob}
- the layer u can be regarded as a cylinder with radius r_l and height L_u
- the uptake sphere is centered at the middle of layer u

To enhance readability, we write λ instead of λ^X_{uv}. Setting formally $r_l = \infty$, the number of X synapses in layer u that is measured by tracing studies is given by

$$S^X_u \sim \underbrace{\lim_{r_l \to \infty} \int_{Z(r_l)} e^{-\frac{r}{\lambda}} \mathrm{d}V - \int_{S(r_{\text{blob}})} e^{-\frac{r}{\lambda}} \mathrm{d}V}_{=:\, I(r_{\text{blob}})} \,. \tag{7.30}$$

Here, $Z(r_l)$ denotes the cylinder with radius r_l approximating layer u and $S(r_{\text{blob}})$ denotes the sphere with radius r_{blob} approximating the uptake zone of the dye in the center of the cylinder. Accounting for the missing connections inside of the blob of dye, the total number of connections with X synapses within layer u is calculated by

$$S^{X,\text{corr}}_u = S^X_u \cdot \frac{I(0)}{I(r_{\text{blob}})} \,. \tag{7.31}$$

We thus replace the intra-layer relative anatomical projection strength by a quantity scaled with the same factor to correct for the connections not accounted for, and obtain

$$\eta^{X,\text{corr}}_{uu} = \eta^X_{uu} \cdot \frac{I(0)}{I(r_{\text{blob}})} \,. \tag{7.32}$$

To evaluate $I(r_{\text{blob}})$ we use spherical coordinates. Since the domain of integration is contained in a cylinder, the respective constraints for the integration variables needs to be derived first (Figure 7.1b). The exclusion of the dye sphere at the center of our coordinate system simply imposes the constraint $r > r_{\text{blob}}$. In the case that $r < L_u/2$ the layer boundaries do not restrict the integration. If $r > L_u/2$ the height needs to be restricted: $|z| \overset{!}{\leq} \frac{L_u}{2}$. In the chosen coordinates, z is expressed as $z = r\cos\theta$. Using this formulation we derive a constraint on the azimuth angle θ:

$$r\cos\theta \leq \frac{L_u}{2} \quad \Rightarrow \quad \arccos\left(\frac{L_u}{2r}\right) \leq \theta \leq \pi/2\,, \tag{7.33}$$

where we used that $0 \leq \theta \leq \pi/2$ holds per definition of the azimuth angle. For the integration we distinguish two cases: If the blob radius is smaller than or equal to half the laminar thickness, $r_{\text{blob}} \leq \frac{L_u}{2}$, the entire blob is contained inside of the layer (*spherical case*). If the blob radius exceeds half the laminar thickness, $r_{\text{blob}} > \frac{L_u}{2}$, only parts of the spherical blob will be inside the layer. As the extent of the layer defines our integral boundaries, we need to consider a spherical blob where the top and bottom are cut off (*frustum case*).

First, consider the solution to two generic integrals, $I_1(r_1)$ and $I_2(r_1, r_2)$:

$$\begin{aligned} I_1(r_1) &:= \int_{r_1}^{\infty} re^{-r/\lambda}\,\mathrm{d}r = -\lambda re^{-r/\lambda}\Big|_{r_1}^{\infty} + \lambda\int_{r_1}^{\infty} e^{-r/\lambda}\,\mathrm{d}r \\ &= \left[\lambda r_1 + \lambda^2\right] e^{-r_1/\lambda} \end{aligned} \tag{7.34}$$

$$\begin{aligned} I_2(r_1, r_2) &:= \int_{r_1}^{r_2} r^2 e^{-r/\lambda}\,\mathrm{d}r = -\lambda r^2 e^{-r/\lambda}\Big|_{r_1}^{r_2} + 2\lambda\int_{r_1}^{r_2} re^{-r/\lambda}\,\mathrm{d}r \\ &= \left[\lambda r^2 + 2\lambda^2 r\right] e^{-r/\lambda}\Big|_{r_2}^{r_1} + 2\lambda^2\int_{r_1}^{r_2} e^{-r/\lambda}\,\mathrm{d}r \\ &= \left[\lambda r^2 + 2\lambda^2 r + 2\lambda^3\right] e^{-r/\lambda}\Big|_{r_2}^{r_1} \end{aligned} \tag{7.35}$$

Spherical case The integral can be split into two regions.

For $r_{\text{blob}} \leq r \leq \frac{L_u}{2}$, we integrate over the whole sphere shell and θ is unconstrained. For $r > \frac{L_u}{2}$ we need to enforce Equation 7.33. Adding the two parts, exploiting the symmetry of the system, and performing some simplifications we obtain

$$\begin{aligned}
I_{\text{sphere}}(r_{\text{blob}}) = &\int_{r_{\text{blob}}}^{\frac{L_u}{2}} \int_0^{2\pi} \int_0^{\pi} e^{-\frac{r}{\lambda}} r^2 \sin(\theta)\, \mathrm{d}\theta \mathrm{d}\varphi \mathrm{d}r \\
&+ 2 \int_{\frac{L_u}{2}}^{\infty} \int_0^{2\pi} \int_{\arccos\left(\frac{L_u}{2r}\right)}^{\pi/2} e^{-\frac{r}{\lambda}} r^2 \sin(\theta)\, \mathrm{d}\theta \mathrm{d}\varphi \mathrm{d}r \\
=&4\pi \int_{r_{\text{blob}}}^{\frac{L_u}{2}} e^{-\frac{r}{\lambda}} r^2 \mathrm{d}r \\
&+ 2 \cdot 2\pi \int_{\frac{L_u}{2}}^{\infty} e^{-\frac{r}{\lambda}} r^2 \left[-\cos(\pi/2) + \cos\left(\arccos\left(\frac{L_u}{2r} \right) \right) \right] \mathrm{d}r \\
=&4\pi \left\{ \underbrace{\int_{r_{\text{blob}}}^{\frac{L_u}{2}} e^{-\frac{r}{\lambda}} r^2 \mathrm{d}r}_{I_2(r_{\text{blob}}, \frac{L_u}{2})} + \frac{L_u}{2} \underbrace{\int_{\frac{L_u}{2}}^{\infty} e^{-r/\lambda} r \, \mathrm{d}r}_{I_1(\frac{L_u}{2})} \right\} \\
=&4\pi \left\{ \left[\lambda r^2 + 2\lambda^2 r + 2\lambda^3 \right] e^{-r/\lambda} \Big|_{r_{\text{blob}}}^{\frac{L_u}{2}} + \frac{L_u}{2} \left[\lambda \frac{L_u}{2} + \lambda^2 \right] e^{\frac{L_u}{2\lambda}} \right\} .
\end{aligned} \tag{7.36}$$

Frustum case Since $r \geq r_{\text{blob}} > \frac{L_u}{2}$ holds in this case, the integrals need not to be split here. Thus,

$$\begin{aligned}
I_{\text{frustum}}(r_{\text{blob}}) &= 2 \int_{r_{\text{blob}}}^{\infty} \int_0^{2\pi} \int_{\arccos\left(\frac{L_u}{2r}\right)}^{\pi/2} e^{-\frac{r}{\lambda}} r^2 \sin(\theta)\, \mathrm{d}\theta \mathrm{d}\varphi \mathrm{d}r \\
&= 4\pi \frac{L_u}{2} \underbrace{\int_{r_{\text{blob}}}^{\infty} r e^{-r/\lambda} \, \mathrm{d}r}_{I_1(r_{\text{blob}})} \\
&= 4\pi \frac{L_u}{2} \left[\lambda r_{\text{blob}} + \lambda^2 \right] e^{r_{\text{blob}}/\lambda} .
\end{aligned} \tag{7.37}$$

$I(0)$ can be trivially obtained for both cases via

$$I(0) = \lim_{r_{\text{blob}} \to 0} I(r_{\text{blob}}) \tag{7.38}$$

When calculating the correction for the number of synapses we choose $r_{\text{blob}} = 0.16\,\text{mm}$. To complete the calculations and thus the amendment of the anatomical projection strengths, the decay constants λ need to be known. They are derived in Section 7.2.3.

thalamo-cortical connections Denoting by η_{ub}^{tha} the projection strength of the thalamic projection from the thalamic population b (multiple reported by Vanni et al. (2020), corresponding to the magno- and parvocellular layers) to the cortical layer u, we continue similarly as above. Since the data from Vanni et al. (2020) (these are layers 2/3, 4 and layer 6) are coarser than what we use here, we define the fraction of thalmo-cortical projection strength targeting the layers L2/3A and L3B as

$$g^{\mathrm{tha,L2/3A}} = \frac{\sum_b \sum_{u \in \mathrm{L2/3A}} \eta_{ub}^{\mathrm{tha}}}{\sum_b \sum_{u \in \mathrm{L2/3}} \eta_{ub}^{\mathrm{tha}}}, \quad g^{\mathrm{tha,L3B}} = \frac{\sum_b \sum_{u \in \mathrm{L3B}} \eta_{ub}^{\mathrm{tha}}}{\sum_b \sum_{u \in \mathrm{L2/3}} \eta_{ub}^{\mathrm{tha}}} \tag{7.39}$$

and similarly for $g^{\mathrm{tha,L4A/B}}, g^{\mathrm{tha,L4C}}$. Garcia-Marin et al. (2017) additionally provide the fraction of thalamo-cortical synapses in layer 4C (relative to all synapses), β_{th}. Setting

$$S^{\mathrm{tha,L4C}} = S^{\mathrm{L4}} \beta_{\mathrm{th}}, \quad S^{\mathrm{tha,L4A/B}} = S^{\mathrm{tha,L4C}} \frac{g^{\mathrm{tha,L4A/B}}}{g^{\mathrm{tha,L4C}}} \,. \tag{7.40}$$

We further calculate

$$S^{\mathrm{tha,L2/3A}} = g^{\mathrm{tha,L2/3A}} \, S^{\mathrm{tha,L2/3}} \tag{7.41}$$

(analogously for layer 3B), and set $g^{\mathrm{th,L6}} = 1$. This allows us to write the number of thalamo-cortical synapses terminating in all layers u except 4A/B and 4C as

$$S^{\mathrm{tha},u} = g^{\mathrm{tha},u} \, f^{\mathrm{tha},u} S^u \,. \tag{7.42}$$

7.2.2.3 *Combing connection probabilities and strengths*

Using the quantities obtained above, we derive the connectivity map for the for the laminar cortical network of macaque V1.

INTRA-AREAL CONNECTIONS We calculate the number of synapses between to populations, also referred to as connections, as

$$\begin{aligned} S_{AB}^{\mathrm{ia}} &= \sum_u \bar{f}^{\mathrm{ia},u} \left(S^u - S^{\mathrm{th},u} \right) \not{s}_X^{|u} \, q_{|uv}^{|X} \, p_A^{|u,|X} \\ &= \sum_u \bar{f}^{\mathrm{ia},u} \left(S^u - S^{\mathrm{th},u} \right) \, q_{|u,v}^{|X} \, \bar{p}_A^{|u,X} \end{aligned} \tag{7.43}$$

where v is the layer that the source population B resides in, and B has synapses of type X.

CORTICO-CORTICAL CONNECTIONS Similarly, the number of cortico-cortical connections is determined as

$$S_A^{\mathrm{cc}} = \sum_u \bar{f}^{\mathrm{cc},u} \left(S^u - S^{\mathrm{th},u} \right) p_A^{|u,\mathrm{cc}} \,. \tag{7.44}$$

THALAMO-CORTICAL CONNECTIONS Finally, the number of thalamo-cortical connections can be obtained by

$$S_A^{\mathrm{th}} = \sum_u S_{\mathrm{th}}^u \, p_A^{|u,\mathrm{th}} \,. \tag{7.45}$$

7.2.2.4 *Rescaling the number of synpases*

With the quantities derived thus far, we also can calculate the total numbers of intra-areal, cortico-cortical and thalamo-cortical synapses:

$$S^{\text{ia}} = \sum_u S^{\text{ia},u},\ S^{\text{cc}} = \sum_u S^{\text{cc},u},\ S^{\text{th}} = \sum_u S^{\text{th},u} \tag{7.46}$$

The relative value of these quantities with respect to the total number of synapses were determined experimentally by Markov et al. (2011):

$$p^{\text{ia}}_{\text{Markov}} = 85\%,\ p^{\text{cc}}_{\text{Markov}} = 13\%,\ p^{\text{th}}_{\text{Markov}} = 2\% \tag{7.47}$$

To harmonize the respective the relative frequencies of the model with the experimentally constrained ones, we scale the number of synapses in those layers in which the percentage of intra-areal synapses is below 90%. Scaling those layers as well would result in greater than 100% relative frequencies of intra-areal synapses in some layers. The described scaling procedure leads to the desired consistency and concludes the derivations of the laminar organization of macaque V1. Note that Markov et al. (2011) only studied one hemisphere of the macaque cortex. Potentially, there could be interhemispheric connections between the two parts of V1, altering the values in Equation 7.47. The results of Dehay et al. (1988), however, indicate that such connections do not exist in macaque.

7.2.3 *Spatial structure of intra-areal connectivity*

Up to this point, the main quantity with which we described the network connectivity of our model was the number of synapses between populations. In the following sections, we use the obtained quantity to derive a probabilistic description based on *connection probabilities*. Here, the term connection probability refers to *pairwise-Bernoulli* connectivity (Senk et al., 2022). This means that a connection from a neuron of a source populations is established with a neuron of a target population with a given probability. Especially, multiple synapses between the pre- and post-synaptic neurons are excluded. This description of the connectivity is important for instantiating the derived network as a spiking neuron network model.

7.2.3.1 *Medium range isotropic connectivity - Preliminary remarks*

As an underlying model for the medium-range connectivity, we assume an exponential decay of connection probability with distance in a plane on which the neurons are located (Markov et al., 2011):

$$P_{AB}(r) = P^0_{AB}\, e^{-\frac{r}{\lambda_{AB}}} \tag{7.48}$$

Here, A, B are the post- and presynaptic populations, P^0_{AB} is the peak probability at zero distance, and λ_{AB} is the characteristic length of the

spatial decay.
Note that the quantities derived in the above section and *given* the characteristic length enables the calculation of P_{AB}^{0} with the following ansatz:

$$\begin{aligned} S_{AB}^{\mathrm{ia}} &= \rho_A N_B \int_0^{2\pi} \int_0^{\infty} P_{AB}^{0}\, r e^{-\frac{r}{\lambda_{AB}}} \mathrm{d}r\mathrm{d}\varphi \\ &= 2\pi \rho_A N_B {\lambda_{AB}}^2 P_{AB}^{0} \end{aligned} \tag{7.49}$$

Solving for P_{AB}^{0} yields the peak connection probability.
Here, ρ_A is the density of postsynaptic neurons and N_B is the number of presynaptic neurons. This leaves us with the task to determine the numeric values for λ_{AB}.
Due to

$$\rho_A N_B = \rho_A \rho_B \cdot \mathrm{A} = N_A \rho_B, \tag{7.50}$$

where ρ_B is the density of the post-synaptic neurons and A is the area under consideration, the interpretation of the connection probability is independent from whether the pre- or postsynaptic neurons are taken as reference points.
The characteristic lengths are, in principle, specific to all pairs of populations A, B. Since, however, data fully determining the matrix $(\lambda_{AB})_{A,B}$ is lacking, we introduce some simplifying assumptions.
We estimate the characteristic length primarily from retro- and anterograde tracing studies. With retrograde tracing, the pre-, but not the post-synaptic population can be resolved. The target layer of a connection can nevertheless be obtained. For anterograde tracing, we take the density of stained axonal processes as a proxy for connection between neurons —assuming here a uniform distribution of the dendrites of potential target neurons as well as Peters' rule (Braitenberg and Schüz, 1991). Thus, we base our derivations on the potential connectivity between neurons (cf. Stepanyants et al. (2007)). Again, only the target layer can be resolved by this approach. Thus, with our approach we can only derive characteristic lengths λ_{uB} where u denotes the target layer and B the presynaptic population.

7.2.3.2 *Medium-range isotropic connectivity - Estimating the characteristic lengths*

INHIBITORY PRESYNAPTIC POPULATIONS Precise data about the extent of lateral connections of inhibitory neurons is sparse. Packer and Yuste (2011) measured the dependence of the connection probability between paravalbumin-positive neurons (largest class of GABAergic inhibitory interneurons in the cortex) and pyramidal cells in mouse cortex (layer 2/3) on the distance of the cells. Fitting Equation 7.48 to this data reveales a good agreement of connection probabilities with our modeling assumption as well the numeric value for the characteristic length (Figure S1a).

Data of this quality are lacking for macaque V1. Kritzer et al. (1992) studied the termination patterns of inter- and intra-laminar connections established by GABAergic neurons in macaque V1 by retrogradely tracing neurons of this class relative to injections into supragranular, granular and infragranular layers. The authors observed a circular spread of labeled neurons relative to the injection site and irrespective of layer where most heavily labeled neurons cluster in a small disc centered at the injection site. For injection sites in supragranular layers the spatial spread is largest in the supragranular layers and decreases with laminar distance, injections in infragranular layers mirror this pattern, and for injection sites in the granular layers the spatial spread is equal across all layers. Moreover, they found far reaching lateral projection in the supragranular layers. We discard the latter in the subsequent considerations.

We estimate the extent of the lateral spread of connectivity for inhibitory neurons from Kritzer et al. (1992), summarized in Table S1. Denoting the number of intra-areal synapses within a radius R by $S_{AB}^{\text{ia},R}$ we calculated the fraction of connections within this radius, p_{AB}^{R} as

$$p_{AB}^{R} = \frac{S_{AB}^{\text{ia},R}}{S_{AB}^{\text{ia}}} = 1 - (\frac{R}{\lambda_{uB}} + 1).\exp(-\frac{R}{\lambda_{uB}}) \tag{7.51}$$

where u is the layer in which A resides. In the derivation we used

$$S_{AB}^{\text{ia},R} = \rho_A N_B \int_0^{2\pi} \int_0^R P_{AB}^0\, r e^{-\frac{r}{\lambda_{vB}}} \mathrm{d}r \mathrm{d}\varphi \tag{7.52}$$

$$= 2\pi \rho_A N_B \left({\lambda_{vB}}^2 - \lambda_{vB}(\lambda_{vB} + R) e^{-\frac{R}{\lambda_{vB}}} \right) \tag{7.53}$$

combined with Equation 7.49. Taking the characteristic length determined estimated from Packer and Yuste (2011) as the value for inhibitory connections originating in L2/3 and terminating in the same layer, $\lambda_{\text{L2/3,L2/3I}} = 123.9\mu\text{m}$, and plugging in the corresponding spatial spread from Table S1, $R = 600\mu\text{m}$, we arrive at:

$$p_{\text{L2/3I,L2/3}}^{600\mu\text{m}} \approx 0.95 \tag{7.54}$$

This means that 95% of labeled neurons reside in the cortical tissue below the estimated radius. Assuming now that the same relation also holds true for the other measurements, the lacking characteristic lengths can be determined by solving

$$1 - (\frac{R_{vB}}{\lambda_{vB}} + 1) \exp(-\frac{R_{vB}}{\lambda_{vB}}) = 0.95 \tag{7.55}$$

for λ_{uB} where R_{vB} is obtained from Table S1.

EXCITATORY PRE-SYNAPTIC POPULATIONS For quantitative estimates of the characteristic lengths of excitatory projections, we primarily rely on the anterograde tracing study by Sincich and Blasdel (2001)

(Figure 7.3a). Assuming the density of stained axons as proportional to the connection probability, we fit

$$f(c, x_0, y_0, \lambda, d) = c \cdot e^{-\frac{\sqrt{(x-x_0)^2+(y-y_0)^2}}{\lambda}} + d \tag{7.56}$$

to $1 -$ relative luminance of the stains. Parameters optimized alongside λ are the pre-factor c and the offset d. The latter is needed since in the numeric representation, the luminance value far away from the injection site does not decay to zero. The estimated values, gathered in Table S2, are only valid for excitatory projections originating in layer 3. To extend the derivation, we analyse additional retro- and anterograde tracing studies investigating projections originating outside (Blasdel et al., 1985) and inside (Fitzpatrick et al., 1985) layer 4C. Since the data is not presented in a form that is immediately amenable to a quantitative analysis as conducted above, we need to assess the range of intra-areal connectivity qualitatively. Each projection from a pre-synaptic population to a target layer is assigned to be short, medium or long, depending on whether the stained axonal processes were confined within a radius of 300 µm, or 600 µm from the injection site, or extended beyond 600 µm (Table S3.)
To compile a table of characteristic lengths for excitatory connections - combining the quantitative estimates based on anterograde tracing (Sincich and Blasdel, 2001) with the qualitative insights from (Blasdel et al., 1985; Fitzpatrick et al., 1985) - we define the spatial decay constant of long horizontal connections as 0.261 mm, medium horizontal connections as 0.215 mm, and for short horizontal connections as 0.141 mm (numeric values derived from Table S2).

7.2.3.3 *Long-range patchy connectivity*

FUNDAMENTAL CONSTRAINTS ON PATCHY CONNECTIONS In contrast to the medium-range connectivity (see Section 7.2.3.1), long-range patchy connectivity in macaque V1 is anisotropic in space and preferentially links neurons with similar orientation preference organized in orientation maps (Stettler et al., 2002). The appearances of patches is confined to an ellipse that is aligned to the preferred orientation of the neuron that establishes these long-range connections (Bosking et al., 1997). The length of the semi-major and semi-minor axes of the ellipse vary across layers (Angelucci et al., 2002). In this study, the authors obtained the length of these axes resolved to infragranular, granular, and supragranular after injecting a bi-directional tracer in macaque primary visual cortex. We use the determined values as numeric constraints in our model (see Table 7.10). Angelucci et al. (2002) additionally show that patchy connectivity between neurons exhibits a reciprocal structure.
Patches, determined by dense tufts of axonal collaterals far away from

the soma, appear to preserve their location across their termination layers (Gilbert and Wiesel (1983), also see Figure 7.3a).

generating an orientation map To endow our model with long-range patchy connections, we generate an orientation map for mediating patchy connections. We follow the approach suggested by Sadeh and Rotter (2013). The authors showed that plausible orientation maps can originate from randomly jittered thalamic projections to the neocortex if both the thalamus and V1 are modeled as superimposed sheets.

We first place N cortical dummy neurons on a regular grid in an square with side length L. The threshold of these spiking neurons is set to infinity. Thus, these neurons do not emit spikes; their membrane potential reflects the input each neuron receives in the cortical sheet. Next, we create a layer of N_{Th} regularly spaced effective thalamic neurons on a square with side length L'. We set $L' > L$ to avoid boundary effects in the generation of the orientation map. The cortical and the thalamic sheet share the central point. Jittering both, horizontal and vertical components of the positions of the thalamic neurons with displacements uniformly drawn from $[-a, a]$ yields the *base target location* of these neurons in the cortical layer.

The preferred orientation of a cortical dummy neuron is determined from its strongest response to stimulations from the effective thalamic neurons. The latter provide an input to the former with temporal and spatial characteristics of moving gratings for orientations $\theta \in \{0°, 30°, 60°, 90°, 120°, 150°\}$.

The effective thalamic neurons are connected to the dummy cortical neurons with distance-dependent weights:

$$w \sim w_0 \cdot e^{-\frac{d^2}{2\sigma^2}} \tag{7.57}$$

Here w_0 denotes the peak weight, d denotes the horizontal distance of cortical neurons to the *base target location* of the thalamic neuron, and σ the spatial width of connections. Only those neurons with a distance smaller than 2σ from the base target location are connected (Figure S2a). The effective thalamic neurons inject a sinusoidal current into the cortical dummy neurons,

$$I(t) = w \cdot \hat{I} \sin(2\pi \cdot f_t + \phi_o) \ . \tag{7.58}$$

Here, w is the weight as determined by Equation 7.57, $\hat{I}$ scales the peak amplitude, f_t the temporal frequency and ϕ_o the phase. Changes in the latter for all effective thalamic neurons result in activity emulating a moving bar experiment of varying orientations. It is calculated depending on the location of the thalamic neuron in the thalamic layer (not the base target location), (x, y) and the orientation θ:

$$\phi(\theta) = \frac{360}{2\pi} \cdot f_s \cdot \langle \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix}, \begin{pmatrix} x \\ y \end{pmatrix} \rangle, \tag{7.59}$$

Here f_s denotes the spatial frequency. Simulating for all θ and recording the response in the cortical dummy neurons yields the *raw orientation map* (Figure S2b). To generate the orientation map for the model of macaque V1, we first randomly place neurons belonging to each population in the modeled cortical sheet. Then the closest cortical dummy neuron for each neuron in the modeled populations is determined. The preferred orientation is then identical with the one of the closest cortical dummy neuron. See Table 7.1 and Table 7.2 for the parameters used in the orientation map generation.

Simulation parameters

L	4 mm	Side length of cortical sheet
L'	4.8 mm	Side length of thalamic sheet
N	3600	Number of cortical dummy neurons
N_{Th}	196	Number of effective thalamic neurons
a	$[-0.1\,\text{mm}, 0.1\,\text{mm}]$	Interval from which random jitter is drawn
w_0	10	Peak weight of thalamic stimulation
σ	0.25 mm	Standard deviation of Gaussian kernel describing weight decay

Table 7.1: Summary of network parameters for generation of orientation map.

Simulation parameters

$\hat{I}$	30 pA	Scaling factor for amplitude of injected current
f_t	1 Hz	Temporal frequency of injected current
f_s	$4\,\text{mm}^{-1}$	Spatial frequency of injected current

Table 7.2: Summary of stimulation parameters for generation of orientation map.

PATCHY CONNECTIONS Patchy connections are established between neurons of the same preferred orientation. Since all layers share one orientation map, the organization of patchy connectivity is preserved across layers. This is in agreement with the findings of e.g. Gilbert and Wiesel (1983). Additionally, each pre-synaptic neurons establishing a patchy connections is the center of an ellipse aligned along its preferred orientation (Stettler et al., 2002). The ellipse covers multiple iso-orientation domains and constrains the area in which potential post-synaptic neurons reside. In our model neurons from pre-synaptic population B to neurons of a post-synaptic population A satisfying the above constraints are connected randomly with probability P_{AB}^{patchy}.

The *patchiness parameter* $\psi \in [0,1]$ controls the relative number of synapses stemming from a patchy connection between B and A. Denoting the synapses established with this connectivity scheme by $S_{AB}^{\text{ia,patchy}}$, we have

$$S_{AB}^{\text{ia}} = \underbrace{\psi S_{AB}^{\text{ia}}}_{=S_{AB}^{\text{ia,patchy}}\text{, synapses of patchy conn.}} + \underbrace{(1-\psi) S_{AB}^{\text{ia}}}_{=S_{AB}^{\text{ia,iso}}\text{, synapses of isotropic conn.}} . \tag{7.60}$$

Here, S_{AB}^{ia} denotes the total number of intra-areal synapses from B to A. With this, the respective connection probabilities $P_{AB}^{\text{iso,0}}$ and P_{AB}^{patchy} can be calculated:

$$\begin{aligned}
& S_{AB}^{\text{ia,patchy}} = \frac{1}{N_O}\int_{\text{ellipse}} \rho_A N_B P_{AB}^{\text{patchy}} \mathrm{d}x \overset{!}{=} \psi S_{AB}^{\text{ia}} \\
\Rightarrow\quad & \frac{ab\pi}{N_O}\rho_A N_B P_{AB}^{\text{patchy}} = \psi S_{AB}^{\text{ia}} \\
\Rightarrow\quad & P_{AB}^{\text{patchy}} = \frac{\psi S_{AB}^{\text{ia}} N_O}{ab\pi\rho_A N_B}
\end{aligned} \tag{7.61}$$

and

$$\begin{aligned}
& S_{AB}^{\text{ia,iso}} = \int_0^{2\pi}\int_0^{\infty} \rho_A N_B P_{AB}^{0,\text{iso}} r e^{-r/\lambda_{vX}} \mathrm{d}r \mathrm{d}\varphi \overset{!}{=} (1-\psi) S_{AB}^{\text{ia}} \\
\Rightarrow\quad & \frac{P_{AB}^{0,\text{iso}}}{P_{AB}^{0}} \underbrace{\int_0^{2\pi}\int_0^{\infty} \rho_A N_B P_{AB}^{0} r e^{-r/\lambda_{vX}} \mathrm{d}r \mathrm{d}\varphi}_{=S_{AB}^{\text{ia}}} = (1-\psi) S_{AB}^{\text{ia}} \\
\Rightarrow\quad & P_{AB}^{0,\text{iso}} = (1-\psi) P_{AB}^{0}
\end{aligned} \tag{7.62}$$

Here ρ_A denotes the neuron density of population A, N_B is the number of neurons in population B, a and b are the semi-major and semi-minor axes of the confining ellipse, respectively, and N_O is the number of orientations used in the model. In our model $N_O = 6$.

Patchy connections are established only when the pre-synaptic population is excitatory. In our model, they connect the supragranular layers 2/3A, 3B, 4AB as well as layer 5 with itself and layer 5 to layer 6, there are no patchy connection from layer 6 to layer 5.

7.2.3.4 *Short-range push-pull connectivity*

The model of macaque V1 presented here is additionally endowed with *push-pull* connections. Push-pull connections are a suggested mechanism underlying contrast invariant responses of neurons in the primary visual cortex. This means that the shape of orientation tuning curves, i.e. the response of a neuron in a moving grating experiment as a function of the angle of the grating, is invariant under the contrast of the presented grating —in contrast to the magnitude of the response (Sclar and Freeman, 1982; Skottun et al., 1987). Here

we follow the approached by Troyer et al. (1998). There, the authors suggested a generative model for push-pull connections based on thalamic projections from the LGN to the primary visual cortex.
In adding these connections to our model, we not only a include local, functional connectivity scheme in layer 4C, but also build a simple model for the thalamic projections. This in principle allows us to provide our network with plausible visual input activity.

LGN PROJECTIONS The LGN is represented by two independent sheets of neurons being randomly distributed on an $L' \times L'$ square (cf. Table 7.1). One sheet contains thalamic neurons with ON, the other sheet with OFF receptive subfields. ON-cells respond to a bright spot surrounded by dark sports, OFF-cells to a dark spot surrounded by bright spots. Each thalamic neuron has the same circular center-surround receptive field size. Jointly, they cover the visual scene completely. Connections from the thalamic sheets to the cortical neurons are established in a *target-specific* way: Given a cortical neuron in layer 4C with preferred orientation $\theta \in \{0°, 30°, 60°, 90°, 120°, 150°\}$ we randomly draw a phase $\psi \in \{0°, 180°\}$ for that neuron and assign it to have an either ON or OFF receptive subfield with 50% probability. We then connect ON/OFF thalamic neurons to the cortical neurons in L4C with a probability following a Gabor filter

$$p(x, y; \lambda, \psi, \gamma, \sigma) = \left[\cos\left(2\pi \cdot \frac{y'}{\lambda} + \psi\right)\right]^{+/-} \exp\left(-\frac{\gamma^2 x'^2 + y'^2}{2\sigma^2}\right) \quad (7.63)$$

where $\left[\,\cdot\,\right]^{+/-}$ denotes the positive/negative part of a function satisfying for a function f with real values

$$f(x) = \left[f\right]^{+}(x) - \left[f\right]^{-}(x) \quad (7.64)$$

(especially implying that $\left[f\right]^{-} \geq 0$) and

$$x' = x\cos(\theta) + y\sin(\theta), \quad y' = -x\sin(\theta) + y\cos(\theta)\,. \quad (7.65)$$

The vector (x, y) is the difference between the location of the source and target neuron. Here, we chose $\gamma = 0.60$ and for the wavelength $\lambda = 0.66\,\text{mm}$ (derived from Jones and Palmer (1987)).
For cortical ON-cells, ON thalamic neurons are connected with the positive, OFF cells with the negative part of the cos function. The roles are reverse for layer 4C neurons with OFF receptive subfields.

PUSH-PULL CONNECTIONS Using a similar scheme as for the connections from LGN to the neocortex, Troyer et al. (1998) derived a push-pull connectivity rule for cortical connections based on a correlation measure derived from thalamic projections targeting cortical neurons: The correlation is positive if single cells' ON and ON or OFF and OFF thalamic projections patterns overlap. It is negative when single cells' ON and OFF or OFF and ON thalamic projections overlap, and zero otherwise. Moreover, the correlation is multiplied by -1 if the pre-synaptic neuron is inhibitory. After rectifying, it was used as a proxy for connection probability between the neurons. Similarly, Antolík et al. (2018) proposed a correlation-based connectivity scheme where the connection probability is given by

$$\frac{1}{\sqrt{2\sigma^2}} e^{-\frac{(c-\mu)^2}{2\sigma^2}}. \tag{7.66}$$

Here, c is the correlation (which is determined as above) and $\mu = 1$ if the pre-synaptic neuron is excitatory, and -1 if inhibitory.
Due to the high number of neurons in our model, calculating the correlation coefficient between all neurons in layer 4C is computationally not feasible. We thus make the following approximations based on the projection from the LGN to the cortex described above:
push-pull connections are only established between neurons with the same preferred orientation. Indeed, assuming that neurons have different preferred orientations, the correlation coefficient is expected to be small even if the overlap of thalamic projections is large —the projections of ON and OFF cells are rotated, thus overlap in a checkerboard pattern and hence cancel each other. For the case of two neurons with the same preferred orientation, we note that the convolution of two Gabor filters with the same standard deviation σ and phase can be approximated by a Gabor filter (up to a normalization) with a standard deviation of $\sqrt{2}\sigma$ and twice the phase. With this, we can derive the dependence of the connection probabilities for all combinations of push-pull connections (see Equation 14.1). The spatially dependent connection probabilities are thus determined up to a pre-factor. Note that the number of established synapses scales linearly with this pre-factor. In contrast to patchy connections, there is no direct way to approximate the number of synapses arising from the push-pull scheme. We define the fraction of push-pull connections relative to all synapses between the relevant populations as ξ. Instantiating a network of layer 4C with connection probabilities as described in Equation 14.1 allows us to obtain the number of isotropic and push-pull connections. With this, given ξ the pre-factor is calculated.

7.2.4 *Target specificity and EI-Ratio*

Additionally to the connection probabilities $P_{AB}^{0,\text{iso}}$ and P_{AB}^{patchy} used in defining the network (as well as instantiating and simulating it in the subsequent sections), we calculate differently defined connection probabilities between the modeled populations in the tissue below a 1 mm × 1 mm patch of cortex, denoted by $P_{AB}^{1\times1}$. At this scale, an approximation neglecting the spatial decay of connection probabilities is reasonable (Potjans and Diesmann, 2014). These are employed to define the *target specificity* and the *EI-fraction*. The latter quantities are exclusively used for a comparison of the layer resolved connectivity derived here with another model of macaque V1 established by different means (Schmidt et al., 2018b) at the 1 mm × 1 mm scale.

The population specific connection probabilities are given by:

$$P_{AB}^{1\times1} = 1 - \left(1 - \frac{1}{N_A^{1\times1} N_B^{1\times1}}\right)^{S_{AB}^{\text{ia},1\times1}} \tag{7.67}$$

Here, $S_{AB}^{\text{ia},1\times1}$ denotes the number of synapses from population B to A under the surface considered here, and mutatis mutandis for $N_A^{1\times1}, N_B^{1\times1}$. Thus, $P_{AB}^{1\times1}$ is the probability of at least one connection from a source neuron in population B to a target neuron in population A. With this, we define the target specificity (Potjans and Diesmann, 2014) as

$$\text{TS}_{vB} = \frac{P_{vEB}^{1\times1} - P_{vIB}^{1\times1}}{P_{vEB}^{1\times1} + P_{vIB}^{1\times1}} \tag{7.68}$$

where v is a target layer and vE denotes the excitatory population in that layer (similarly for vI). The target specificity thus is the normalized difference in connection probabilities and reveals which populations is preferentially targeted in a given layer. Note that this measure depends on the source population but only on the target layer.

The EI-fraction of a population A consists of two numbers

$$\frac{\sum_v S_{AvE}^{\text{ia}}}{\sum_B S_{AB}^{\text{ia}}}, \qquad \frac{\sum_v S_{AvI}^{\text{ia}}}{\sum_B S_{AB}^{\text{ia}}}, \tag{7.69}$$

i.e. the fraction of excitatory and inhibitory intra-areal synapses targeting a population A relative to all intra-areal synapses that establish a connection with A. By definition, adding the two numbers yields 1.

7.2.5 *Implementational considerations*

For instantiating and simulating the spiking neural network specified by our biologically plausible model of macaque V1 we use NEST (Gewaltig and Diesmann, 2007). Since the Gabor-filter determining the connections probabilities for the push-pull connectivity is not included in the standard NEST package, we add this function to the simulation engine[3].

When instantiating the model at a scale of 4 mm × 4 mm, it becomes impractical to use a strict model interpretation for intra-area connections. Given a presynaptic population B as well as a postsynaptic

[3] See https://github.com/ackurth/nest-simulator/tree/V1V2V4_nest for the corresponding code

population A, the connection routine connects pairs of neurons in the pre- and post-synaptic populations with probability $P_{AB}(r)$, where r is the distance between the source and the target neurons. Thus, the number of neuron pairs for which $P_{AB}(r)$ has to be evaluated scales quadratically with the number of neurons in the simulation, leading to long network construction phases. Since by assumption the connection probability decays exponentially, we can restrict potential target neurons given a source neuron to a disc with radius R. This clearly leads to a reduced number of synapses between populations A and B. To correct for this, the peak connection probability needs to be increased. The number of intral-areal synapses between two populations stemming from medium-range isotropic connections within a radius R obeys

$$S_{AB}^{\text{ia,iso},R} = \rho_A N_B \int_0^{2\pi} \int_0^R P_{AB}^0 (1-\psi)\, r e^{-\frac{r}{\lambda_{vX}}} \mathrm{d}r \mathrm{d}\varphi \tag{7.70}$$

$$= 2\pi \rho_A N_B P_{AB}^0 (1-\psi) \left({\lambda_{vX}}^2 - \lambda_{vX}(\lambda_{vX} + R) e^{-\frac{R}{\lambda_{vX}}} \right) . \tag{7.71}$$

The total number of synapses between A and B, S_{AB}^{ia}, can be obtained by letting $R \to \infty$. The required scaling factor $\alpha(R)$ is now given by

$$\alpha(R) = \frac{S_{AB}^{\text{ia}}}{S_{AB}^{\text{ia},R}} = \frac{{\lambda_{vX}}^2}{{\lambda_{vX}}^2 - \lambda_{vX}(\lambda_{vX} + R) e^{-\frac{R}{\lambda_{vX}}}} \tag{7.72}$$

$$= \frac{1}{1 - \left(1 + \frac{R}{\lambda_{vX}}\right) e^{-\frac{R}{\lambda_{vX}}}} . \tag{7.73}$$

If R is chosen proportional to the λ_{vX}s, α is independent of the populations. See Table S4 for different values of α under this assumption. We choose $R = 4\lambda_{vX}$ balancing small adaptations of the peak zero-distance probability and computational efficiency.

7.2.6 *Simulation parameters*

Instantiating the network derived here requires the specification of parameters regarding the network itself, but also of the constituting neurons and their interactions. We gather the parameters used here in tables Table 7.3, Table 7.4, Table 7.5, and Table 7.6 in the style of Nordlie et al. (2009b).

Each neuron is represented by a leaky integrate-and-fire model neuron with exponential postsynaptic currents. The numeric values chosen for the parameters of the single-neurons and their interactions are taken from Potjans and Diesmann (2014), for the delays from Senk et al. (2023).

Table 7.3: General description of model of macaque primary visual cortex

Neuron and synapse model	
Subthreshold dynamics	$\frac{dV}{dt} = -\frac{V-E_L}{\tau_m} + \frac{I_{\text{syn}}(t)}{C_m}$, $I_{\text{syn}}(t) = Je^{-(t-t^*-d)/\tau_{syn}} H(t-t^*-d)$, here H denotes the Heaviside function.
Spiking	If $V(t-) < V_{\text{th}}$ and $V(t+) \geq V_{\text{th}}$, 1. Set $t^* = t$ and $V(t) = V_{\text{reset}}$ in $(t^*, t^* + \tau_r]$ 2. Emit spike with time stamp t^*.
Delays	
Delay	Synaptic delay is $d_0 + r \cdot v_c + \mathcal{N}(0, \sigma_d)$ where r is the Euclidean distance between pre- and post-synaptic neurons.
Stimulation	
External input	Independent Poisson spike trains with fixed rate $\nu = \nu_0 + \delta\nu$ and population-specific indegrees.
Connectivity	
Isotropic connections	Establish connection with probability $P_{AB}^{0,\text{iso}} e^{-\frac{r}{\lambda_{vX}}}$ where B is the pre-, A the post-synaptic population, v of A resides, B has X synapses, and r the Euclidean distance between neurons that are to be connected.
Patchy connections	Connect neurons with identical orientation preference with pair-wise Bernoulli connectivity with connection probability P_{AB}^{patchy} where B denotes the and A the post-synaptic population.
Push-pull connections	Connect neurons with identical orientation preference with spatially dependent connection probability according to Equation 14.1.

Table 7.4: Network size under a 4 mm × 4 mm patch of cortex and external indegrees

Network size		
Layer	Number of excitatory neurons	Number of inhibitory neurons
2/3A	323 466	80 866
3B	394 614	98 654
4A/B	546 816	141 184
4C	635 712	121 088
5	317 440	79 360
6	409 360	72 240
Inter-areal single-neuron indegrees		
Layer	Indegree of excitatory neurons	Indegree of inhibitory neurons
2/3A	169	117
3B	169	108
4A/B	253	136
4C	197	139
5	184	44
6	699	577
Thalamic single-neuron indegrees		
Layer	Indegree of excitatory neurons	Indegree of inhibitory neurons
2/3A	0	0
3B	0	0
4A/B	51	30
4C	238	187
5	11	0
6	139	54

Table 7.5: Parameters for the network connectivity

Connectivity		
P^0_{AB}	Table 7.9	Peak connection probabilities
ψ	0.2	Patchiness parameter
ξ	0.5	Relative frequency of push-pull connections
λ_{vX}	Table 7.7, Table 7.8	Characteristic lengths of isotropic connectivity
a, b	Table 7.10	Size of ellipses parametrized by lenght of semi-major and semi-minor axis

Table 7.6: Neuron, synapse and delay parameters for network simulation

Neuron parameters		
τ_{m}	10 ms	Membrane time constant
τ_{r}	2 ms	Absolute refractory period
τ_{syn}	0.5 ms	Postsynaptic current time constant
C_{m}	250 pF	Membrane capacity
E_{L}	-65 mV	Reversal potential
V_{reset}	-65 mV	Reset membrane potential
V_{th}	-50 mV	Threshold
Synapse parameters		
J_{XE}	87.8 pA	Synaptic weight excitatory neurons
g	4	Relative inhibitory synaptic efficacy
J_{XI}	$-g \cdot J_{XE}$	Synaptic weight inhibitory neurons
Delay parameters		
d_0	0.5 ms	Delay offset
v_c	0.3 mm/ms	Conduction speed
σ_d	0.05 ms	Temporal jitter of delay distribution
Stimulation		
ν_0	$9\,\frac{\mathrm{spikes}}{\mathrm{s}}$	Baseline rate

7.3 RESULTS

Under a 1 mm × 1 mm patch of cortical surface, our model of the primary visual cortex of macaque comprises 164 thousand excitatory and 37 thousand inhibitory neurons. They are recurrently connected by in total 393 million intra-cortical synapses and externally innervated by 52 million cortico-cortical and 17 million thalamo-cortical connections.

7.3.1 *Laminar connectivity*

To analyse the laminar connectivity, we investigate the number of connections between different populations (Figure 7.2). We compare our results with the connectivity data from the adaptation of the model of Potjans and Diesmann (2014) to macaque V1 by Schmidt et al. (2018b), in the following named MAM V1. This model represents the cortical tissue below 1 mm × 1 mm of cortical surface as well. It consists of 158 thousand excitatory and 39 thousand inhibitory neurons, connected by 381 million synapses. Thus, representing the same cortical tissue, the respective quantities in our model of the primary visual cortex and MAM V1 are consistent (deviations are due to different data used in the derivations). Note, however, that the ratio between excitatory and inhibitory neurons in the model derived here is 4.43, while it is 4.05 for MAM V1.

The model of macaque V1 presented here exhibits population-specific connectivity (Figure 7.2a). In direct comparison with the MAM V1 model (Figure 7.2b), the number of connections between two populations is reduced. This is generally to be expected, since we here display the total number of synapses between the different populations. As our model has a finer laminar granularity, there are fewer neurons in certain populations and hence a smaller number of synapses between them.

The connectivity between excitatory populations suggests a flow of information after a stimulation of layer 4C (e.g. by visual input transmitted to V1 via LGN projections): activity entering there is transferred to upper layer 4 and layer 3AB, as well as layer 6. Subsequently, also layer 2/3A and 5 are innervated. This putative flow of activity derived from the excitatory projections is consistent with the respective anatomical data in MAM V1 (Figure 7.2b), simulations (see Potjans and Diesmann (2014), Figure 11), and indeed in experimentally recorded neural activity, e.g. Wang et al. (2020).

The average excitatory and inhibitory indegrees to neurons in the populations are similar in magnitude between the model presented here and MAM V1 (Figure 7.2c,d). In our model, excitatory populations have an higher excitatory indegree as compared with inhibitory populations. For populations in MAM V1, this is reversed, except for layer 5. Note also the relatively low excitatory indegree for subpopulations of

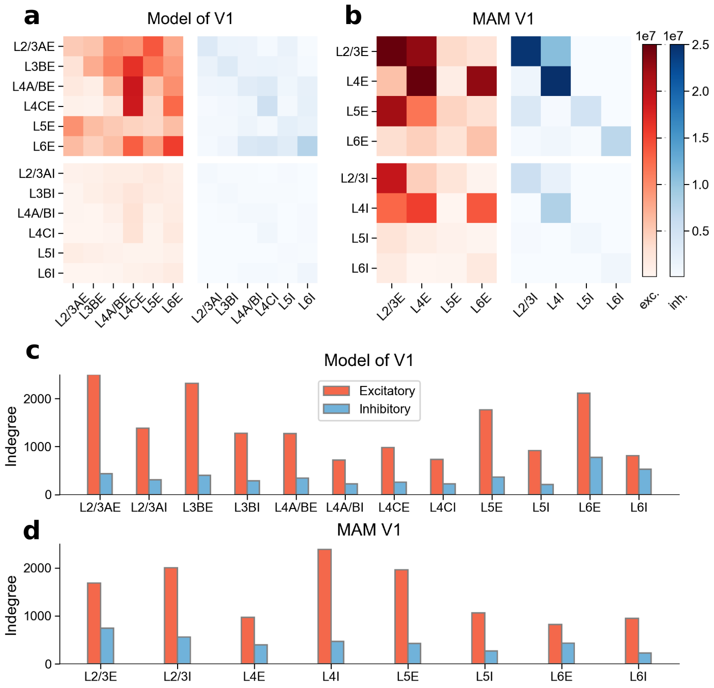


Figure 7.2: Connectivity of our model of macaque V1 and the adaptation of the cortical microcircuit model by Potjans and Diesmann (2014) to macaque primary visual cortex by Schmidt et al. (2018b). **a, b** Number of connections between the populations in our model and between the populations in MAM V1 under a 1 mm × 1 mm patch of cortical surface. Excitatory connections are represented in red, inhibitory connections in blue. **c, d** Average excitatory and inhibitory indegrees to neurons in the different population in our model and MAM V1.

layer 4, both in comparison with other populations of the same model and MAM V1. The inhibitory indegree for especially the inhibitory population in layer 6 is high for the model derived here.

7.3.2 *Spatial connectivity*

Spatial connections are classified as either long-range patchy, medium-range isotropic and short-range push-pull connections. The three classes are introduced in detail in Section 7.2.3.

7.3.2.1 *Medium-range isotropic connectivity*

The central quantity for assessing the medium-range isotropic connectivity is the characteristic length λ, describing the decay of connection probability with distance d:

$$P(r) \sim e^{-\frac{d}{\lambda}}$$

The estimation of λ is exemplified in Figure 7.3b. Combining these quantitative estimates with qualitative data (as described in Section 7.2.3.2) we obtain characteristic lengths specific to the pre-synaptic populations and the target layer. Numeric values are gathered in Table 7.7 for excitatory and Table 7.8 for inhibitory projections.

We find that excitatory neurons generally project farther than inhibitory ones. This is in agreement with experimental findings showing that excitatory neurons (especially pyramidal cells) develop axon collaterals that reach over longer distances than inhibitory neurons enabling connections between neurons separated by greater distances, see e.g. Binzegger et al. (2004) and Stepanyants et al. (2007). Numeric values of the characteristic lengths for both excitatory and inhibitory source neurons are moreover consistent with independent measurements for macaque V1 (Budd and Kisvárday, 2001; Markov et al., 2011). Generally, the characteristic lengths is maximal for connections within a layer. The spatial spread of connections from and to the layer 4C excitatory population is small compared with other populations, resulting in localized projections. This is consistent with the layer's role as primary input layer of thalamo-cortical projections, putatively performing few computations on its own and rather spreading the activity to supra- and infra-granular layers.

Using the estimated characteristic lengths, we are able to derive the peak connection probabilities P^0_{AB} according to Equation 7.49 where we use the correction as described by equation Equation 7.73. See Table 7.9 for the numeric values.

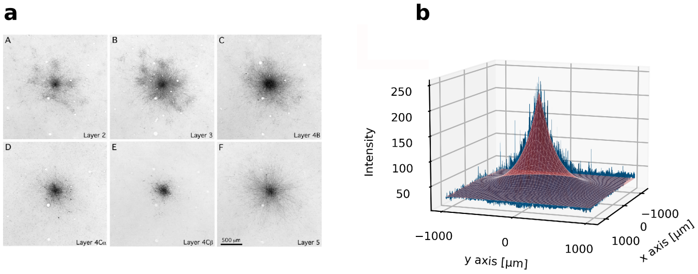


Figure 7.3: **a** Axonal projection from Sincich and Blasdel (2001), Figure 3, upon injection of a bi-directional tracer in layer 3. Copyright 2001 Society of Neuroscience, **b** Exponential fit for layer 5, as described in Equation 7.56, see also Korcsak-Gorzo et al. (2022).

λ_{vB} **[mm]**

Target Layer	**Source Layer**					
	L2/3A	**L3B**	**L4AB**	**L4C**	**L5**	**L6**
L2/3A	0.261	0.215	0.178	0.141	0.212	0.141
L3B	0.215	0.261	0.249	0.141	0.215	0.215
L4AB	0.141	0.261	0.261	0.215	0.171	0.261
L4C	0.121	0.121	0.121	0.141	0.171	0.215
L5	0.261	0.215	0.249	0.141	0.261	0.121
L6	0.215	0.141	0.212	0.141	0.249	0.261

Table 7.7: Estimated characteristic lengths for projections from excitatory neurons to the different cortical layers. Numeric values are derived from experiments by Sincich and Blasdel (2001) and - for lack of data - refined by qualitative inspections of the tracing studies by Blasdel et al. (1985) and Fitzpatrick et al. (1985).

λ_{vB} **[mm]**

Target Layer	**Source Layer**					
	L2/3A	**L3B**	**L4AB**	**L4C**	**L5**	**L6**
L2/3A	0.123	0.123	0.103	0.051	0.062	0.062
L3B	0.123	0.123	0.103	0.051	0.062	0.062
L4AB	0.062	0.062	0.062	0.062	0.062	0.062
L4C	0.062	0.062	0.062	0.062	0.062	0.062
L5	0.092	0.092	0.103	0.123	0.144	0.144
L6	0.092	0.092	0.103	0.123	0.144	0.144

Table 7.8: Estimated characteristic lengths for projections from inhibitory neurons to the different cortical layers. Numeric values are derived from estimates based on Kritzer et al. (1992) combined with data from Packer and Yuste (2011).

	Peak connection probability P_{AB}^{0}											
Population	**2/3AE**	**2/3AI**	**3BE**	**3BI**	**4A/BE**	**4A/BI**	**4CE**	**4CI**	**5E**	**5I**	**6E**	**6I**
2/3AE	0.031	0.386	0.042	0.147	0.077	0.139	0.098	0.173	0.148	0.781	0.130	0.189
2/3AI	0.015	0.272	0.024	0.105	0.043	0.099	0.055	0.124	0.080	0.548	0.073	0.136
3BE	0.020	0.154	0.031	0.256	0.039	0.115	0.146	0.567	0.088	0.512	0.052	0.290
3BI	0.007	0.105	0.018	0.184	0.022	0.082	0.083	0.407	0.047	0.355	0.029	0.208
4A/BE	0.015	0.272	0.024	0.105	0.043	0.099	0.055	0.124	0.081	0.548	0.073	0.136
4A/BI	0.012	0.189	0.001	0.170	0.008	0.293	0.031	0.395	0.024	0.176	0.016	0.408
4CE	0.013	0.087	0.008	0.143	0.027	0.139	0.103	0.819	0.027	0.151	0.047	0.566
4CI	0.006	0.055	0.002	0.113	0.017	0.119	0.081	0.738	0.018	0.122	0.037	0.510
5E	0.061	0.108	0.048	0.097	0.024	0.153	0.052	0.076	0.027	0.226	0.187	0.252
5I	0.035	0.051	0.028	0.051	0.013	0.095	0.023	0.043	0.012	0.141	0.102	0.165
6E	0.046	0.076	0.061	0.135	0.029	0.251	0.110	0.226	0.048	0.186	0.057	0.512
6I	0.004	0.000	0.008	0.066	0.007	0.167	0.033	0.135	0.028	0.091	0.030	0.417

Table 7.9: Peak connection probabilities for our model of macaque primary visual cortex. Target populations are given in rows, source populations in columns.

7.3.2.2 *Long-range patchy connectivity*

Long range patchy connectivity is confined to an ellipse centered at the source neuron with an orientation aligned to the preferred orientation of that neuron (see Section 7.2.3.3). The semi-major and minor axes of these ellipses can be constrained from experimental data (Table 7.10). This connectivity scheme exclusively connects neurons with the same orientation preference, clustered in iso-orientation domains. The fraction of synapses that belong to a patchy connection is unknown. We quantify this in the patchiness parameter ψ that can attain values in the unit interval $[0, 1]$. Figure 7.4 shows intra-areal connections for a high patchiness parameter where the confining ellipse can be identified visually.

Semi-minor and major axes [mm]

Source Layer	Semi-minor axis	Semi-major axis
L2/3A or L3B	1.9	3.0
L4AB	2.2	3.3
L5 or L6	2.3	4

Table 7.10: Semi-minor and major axes describing an ellipse centered at a source neuron confining the region in which patchy connections to neurons with the same preferred orientation can be established. Data from Angelucci et al. (2002).

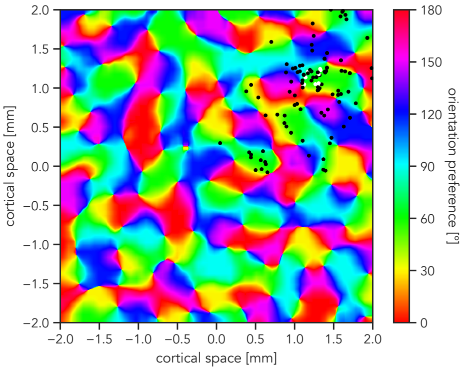


Figure 7.4: Processed orientation map for the orientation preference of neurons under a 2 mm × 2 mm patch of cortical tissue. White dots are selected pre-synaptic neurons with an orientation preference of 60°. Black dots are post-synaptic neurons highlighting long-range patchy connections confined to an ellipse.

7.3.2.3 *Short-range push-pull connectivity*

In our model of macaque V1, neurons in layer 4C are moreover endowed with push-pull connectivity (Section 7.2.3.4). In this scheme, neurons are subdivided into cells with center ON- and OFF receptive subfields, additionally to their orientation preference. This distinction is related the specific thalamic input pattern (Martinez et al., 2005; Troyer et al., 1998), see Figure 7.5a. Based on this, push-pull connections between neurons of the same orientation preference are established (Figure 7.5b).

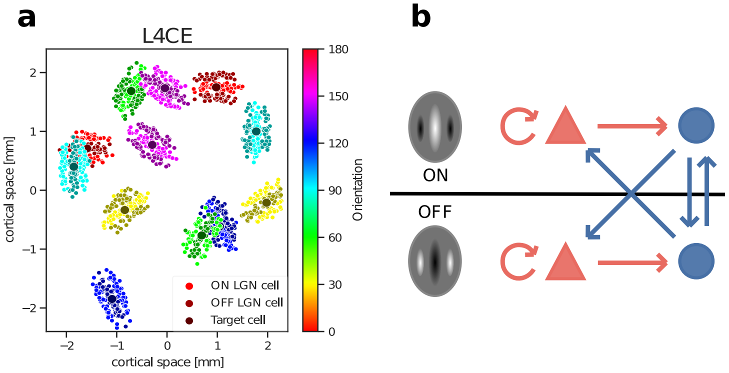


Figure 7.5: **a** Thalamic projections to neurons in the excitatory population in layer 4C having ON and OFF receptive subfields of various preferred orientations. **b** Sketch of push-pull connectivity. Left: ON and OFF receptive subfields. Right: Excitatory neurons with either ON or OFF receptive subfield innervate corresponding inhibitory neurons that suppress firing in the neurons of the opposite receptive subfield.

7.3.3 *Simulations of the network*

We instantiate and simulate our spiking neural network model of the macaque primary visual cortex on the JURECA-DC super computer (Thörnig and von St. Vieth, 2021). The spatial extent of the network used for the simulations is 4 mm × 4 mm. Choosing standard parameters for the single-neuron dynamics and their interactions (cf. Potjans and Diesmann (2014) and Senk et al. (2023), Section 7.2.6), we perturb the baseline rate (leading to an external input firing rate of $\nu = \nu_0 + \delta\nu$) and vary the excitation-inhibition factor g. First, the external cortico-cortical indegrees from our derivations are used to stimulate the network (cf. Table 7.3). Simulations reveal that the network does not exhibit biologically plausible firing rates on the order of $\sim 10\,\frac{\text{spikes}}{\text{s}}$ (Figure 7.6a). Rather, the population averaged firing rates transitions rapidly from a quiescent state to a regime with implausibly high spiking activity. Such behavior is typical for non-balanced networks where a net-positive feedback loop causes a divergence in the activity. In order to control the network, we increase the excitation-inhibition factor g. Doing so strengthens the inhibition and should —thinking naively —counteract the positive feedback of excitatory neurons. For a reasonable numeric values of g no balanced state can be achieved, the rapid transition still occurs.

The non-balanced state of the network can be also induced by an asymmetry in external input (van Vreeswijk and Sompolinsky, 1998): if, for example, the excitatory populations of an otherwise balanced (meaning in the presence of identical input to excitatory and inhibitory

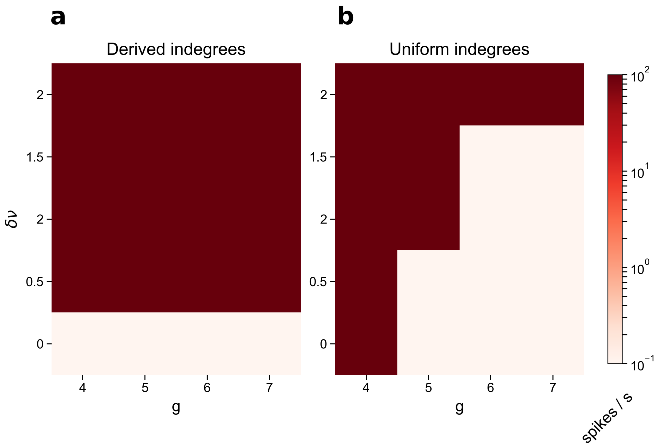


Figure 7.6: Population averaged firing rates of the model of the primary visual cortex of macaque monkey. Two parameters are varied between simulations: $\delta\nu$, the absolute increase in the firing rate of the Poisson generators externally driving, and the excitation-inhibition balance factor g **a** External indegrees derived for our model of macaque V1 as described in Table 7.4. **b** Uniform external indegrees of 600 incoming synapses. Each neuron in each population receives the same external indegree.

populations) is driven more strongly by external input, diverging firing rates may occur. To test for this in our case, we simulate the network with uniform external indegrees to both excitatory and inhibitory populations (Figure 7.6b). While in this case the onset of the transition shifts with increasing excitation-inhibition factor, the picture remains qualitatively similar. The derived network does not allows for bio-physiologically reasonable firing rates (for moderate g), the activity rapidly changes from a quiescent to a diverging state.
This motivates a closer examination of the recurrent connectivity of our model of macaque V1.

7.3.3.1 *Connectivity analysis revisited*

To understand why our model of macaque V1 does not admit physiologically plausible activity —in contrast to the network of Potjans and Diesmann (2014), Schmidt et al. (2018b), and Senk et al. (2023) —we consider the target specificity (Section 7.2.4) of the intra-areal connectivity. The target specificity attains values from -1 to 1 and assesses whether neurons of a given pre-synaptic population preferentially connected to the excitatory or inhibitory population in a target layer. -1 means that only inhibitory neurons, 1 that only excitatory neurons are targeted. A target specificity of 0 implies equal probabilities. Our model of the macaque primary visual cortex exhibits exclusively positive target specificities (Figure 7.7a). If the pre-synaptic population is inhibitory, the values are generally smaller. In contrast, the connectivity of MAM V1 also shows negative target specificities, especially for excitatory source populations (Figure 7.7b). Additionally, inhibitory pre-synaptic populations innervate excitatory target populations with high probability for a larger number of projections.

Studying the EI-fractions of our model of V1 and MAM V1 reveals that —consistent with the observations reported in Section 7.3.1 —in our model the excitatory populations generally have a higher relative frequency of incoming excitatory synapses than their inhibitory counterparts (in the same layer). The situation is reversed for MAM V1.

This suggests an explanation for why our model (in contrast to MAM V1, choosing the same single-neuron parameters) does not show physiologically realistic firing rates across layers and rather transitions from an almost quiescent to a diverging network state: excitatory neurons are preferentially targeted by excitatory populations (and strongly so), leading to a higher level of excitation than in their inhibitory counterparts. While also inhibitory neurons preferentially connect to excitatory populations, the preference is less strong. This might induce a positive feedback loop that cannot be caught by the inhibitory activity, leading to divergence in the network activity —the attained network state exhibits implausibly high firing rates.

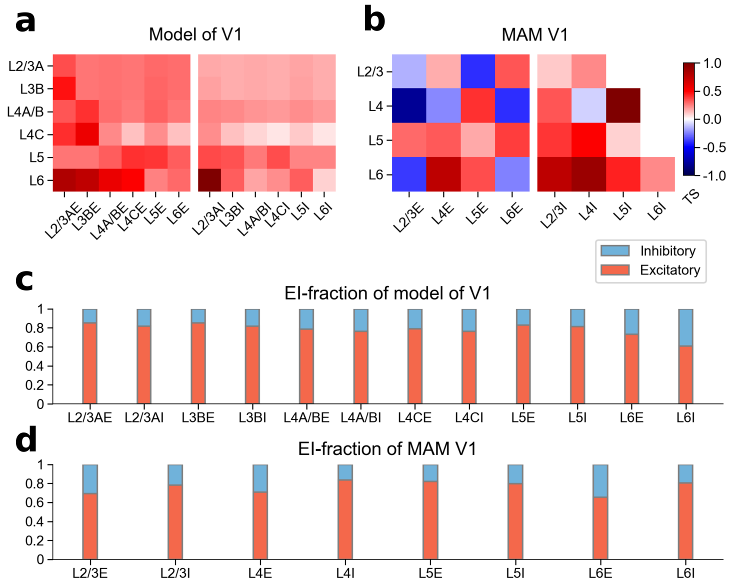


Figure 7.7: Target specificity and EI-fractions for models of primary visual cortex of macaque **a, b** Target specificity of our model of macaque V1 and MAM V1. **c, d** EI-fractions of our model of macaque V1 and MAM V1.

7.4 DISCUSSION

In this chapter, we constructed a biologically plausible spiking neural network model of the primary visual cortex in macaque. To this end, a large number of anatomical studies were analysed and brought together in a comprehensive description of laminar and lateral cortical connectivity.

In particular, we developed a modular workflow in which we combined connection probabilities (Section 7.2.2.1) from cat V1 with anatomical projection strength from macaque primary visual cortex (Section 7.2.2.2). The former is used due to a lack of similar data for macaque V1. The latter is intended to introduce connection structures specific for macaque. The obtained connectivity exhibits population specific projection patterns (suggesting a certain flow of activity after a suitable stimulation) and realistic neuronal indegrees (Section 7.3.1). Moreover, we estimate the characteristic lengths for intra-areal connections and endow the model with functional long-range patchy and short-range push-pull connections. Thereby, we extend the work by Senk et al. (2023) in multiple ways: first, by providing more detailed estimates for the characteristic lengths. While Senk et al. (2023) employ characteristic lengths that depend only on whether the pre- and postsynaptic neurons are excitatory or inhibitory (and thus are not layer-specific), we derive values that are specific to the source population and the target layer. The values reported here are consistent with independently derived estimates for macaque (Markov et al., 2011) and qualitatively agree with respective measurements for rat cortex (Schnepel et al., 2015). Additionally, the connectivity is extended by functional long-range patchy (Section 7.3.2.2) and local push-pull connectivity (Section 7.3.2.3). For this, we endow neurons of our model with preferred orientations and preferentially link neurons with the same orientation preference in certain conditions. Furthermore, we build an input model that takes into account ON and OFF receptive subfields of thalamic neurons that is the basis for push-pull connectivity.

Our model thus takes into account certain features of the organization of macaque V1 that are believed to underlie cortical information processing in this area. Thus, the work presented here paves the way for *in silico* experiments exploring the dynamical implications of the embedded organizational principles.

Simulating our model of macaque primary visual cortex, however, shows that the network in its current form does not allow for population specific, physiologically realistic firing rates using standard parameters for the single-neuron dynamics and their interactions. Such firing rates are attained in the comparable model of Potjans and Diesmann (2014), its adaptation to macaque primary visual cortex by Schmidt et al. (2018b) and its spatial extension by Senk et al. (2023). Studying the target specificity of our model reveals a lack of excitation on inhibitory and of inhibition on excitatory neurons, putatively leading to the diverging network activity.

This property of the network is inherited from the connection probabilities (Binzegger et al., 2004) and the anatomical projection strengths (Vanni et al., 2020) used in the derivations. Especially the connection probabilities potentially exhibit certain inaccuracies due to the small number of reconstructed neurons on which the estimates of Binzegger et al. (2004) are based. Large-scale network reconstructions using electron microscopy promise to mitigate this shortcoming and promise a more detailed view on the cortical connectivity (MICrONS Consortium et al., 2021; Shapson-Coe et al., 2021). The modular workflow we developed here allows for constructing new networks once new estimates for the local connection probabilities are provided. These new estimates might remove the imbalance in excitatory connectivity and thus the divergence in network activity observed here.

However, it has to be noted that potentially the network connectivity is not terminally flawed and that the problem lies in neglecting balance on the single-neuron level, homeostatic or plastic mechanisms (see e.g. Vogels et al. (2011), Xue et al. (2014), and Znamenskiy et al. (2018)). Given that a reasonable resting-state activity is achieved, future work will investigate the activity upon stimulation with visual inputs and architectural extensions. First, properties relying on architectural features built into the network need to be retrieved from *in silico* experiments. This includes the correct (i.e. embedded) orientation preference and the contrast invariance of the orientation tuning. Since such experiments drive the network beyond the resting-state, conductance- instead of current-based neuron models might be needed for more realistic simulated activity. After validating the built-in features, the influence of patchy connectivity on network dynamics can be investigated. The patchiness parameter ψ is experimentally not well constrained. It controls to what extent sub-networks are clustered (see Rostami et al. (2020) for a different approach to clustered networks). This potentially has a strong influence on the network dynamics which underlie cortical computations.

Future work also might include extending the developed workflow to other areas of macaque visual cortex. V2 and V4 in the macaque visual system are reasonable choices for this endeavor. Not only do they belong to cortical areas for which anatomical and physiological data is readily available; they also can be embedded into a larger model comprising V1, V2 and V4. These areas form the first three stages of visual processing along the ventral stream, which is associated with the recognition of objects and their representation in an abstracted form in the neural activity (Kandel et al., 2013). The cortico-cortical connections between V1, V2, and V4 respect retionotopy and are convergent for feedforward and divergent for feedback connections (Zeki and Shipp, 1988). Adding this requires the correct relative sizes for the modeled areas. Moreover, the employed split of layer 2/3 into 2/3A and 3B introduced in this model might play an important role (Markov and Kennedy, 2013). Such a network would in a natural way extend the model of macaque primary visual cortex presented here and would fit right between biologically plausible large-scale spiking neural networks of Senk et al. (2023) (one area, taking cortex' spatial structure into account) and Schmidt et al. (2018b) (modeling all vision related areas in the macaque while ignoring the spatial structure of single areas).

8

ARCHITECTURAL PRINCIPLES OF LOCAL CORTICAL CIRCUITS

Author contributions:
Under the supervision of Prof. Dr. Sacha van Albada (SvA) and Prof. Dr. Markus Diesmann (MD), the author conceived the project, and the author and Jasper Albers (JA) conceptualized the work presented in this chapter. The author analyzed the H01 data set, JA analyzed the MICrONS data set, both jointly analyzed the Binzegger data set. Both jointly analyzed and compared the resulting connectivity maps. The analyses were refined in ongoing discussions of the author with JA, SvA, and MD.

8.1 INTRODUCTION

The previous chapter was concerned with the construction of a biologically plausible model of macaque V1. The intra and inter-laminar connectivity derived there (cf. Section 7.3.1) was based on two data sets: estimates of the layer-resolved connection probabilities in cat (Binzegger et al., 2004) and layer-specific projection strengths of various neural populations in macaque (Vanni et al., 2020) primary visual cortex. Instantiating and simulating the network with the derived connectivity did not allow for a stable ground state activity with realistic firing rates (Figure 7.6). A subsequent analysis of the derived circuit (Figure 7.7) revealed that connectivity appears to be prone to introducing positive feedback loops leading to diverging activity. Thus, under the given choice for establishing connections, of parameters for single neurons and their interactions, the derived structure of the network cannot be reconciled with biologically plausible dynamics. This hints at a fundamental bias in the employed anatomical data (or its interpretation) that cannot be easily mitigated without introducing additional anatomical assumptions.

Changes along these lines were introduced by Potjans and Diesmann (2014), which is the basis of the MAM V1 discussed in Chapter 7 alongside our model of macaque primary visual cortex. The authors first derived a raw connectivity map solely based on the data by Binzegger et al. (2004). Then, they reviewed a large body of electrophysiological connectivity studies across multiple species, derived a physiological connectivity map and unified this map with the raw connectivity map under a Gaussian assumption for the distance dependence of connection probabilities. This led to what they refer to as integrated connectivity map. Finally, specific connections of this map were modified to meet a target specificity obtained again from electrophysiological connectivity studies.

Such modifications—while in the case of Potjans and Diesmann (2014) leading to a model of the cortical microcircuit exhibiting plausible, cell type specific activity—are potentially problematic: assumptions regarding the unification of multiple studies with differences in methodology have to be made, and architectural features foreign to the cortical circuit under investigation can be easily introduced. This bears the risk of rendering the connectivity of the local network inconsistent. Thus, new insights are needed in order to consistently construct biologically plausible network models of such circuits.

The aim of this chapter is to take a closer look at the most consistent estimates of microcircuit connectivity we are able to obtain. To this end, we analyze three data sets providing descriptions of local cortical networks. We employ the data provided by Binzegger et al. (2004) (already used in Chapter 7) as well as the new reconstructions of cortical tissue using electron microscopy (EM) by Shapson-Coe et al.

(2021) and MICrONS Consortium et al. (2021).
In recent years, EM has become a powerful tool for the large-scale, three-dimensional reconstruction of cortical circuits (Kubota et al., 2018). The data used here offer unprecedented levels of detail regarding the reconstructions of the local circuitry of the mammalian brain. This was enabled by recent advances in the analysis of large-scale EM data of cortical tissue based on methods from machine learning (MICrONS Consortium et al., 2021; Shapson-Coe et al., 2021). Even though the new data allow for the most accurate reconstructions to this day, the complete network graph still cannot be recovered. Thus, probabilistic descriptions in the form of connectivity maps as in Chapter 7 are still needed. Here, we derive such connectivity maps of the local circuitry based on the three data sets. We compare the maps based on the newer data with the connectivity map derived from Binzegger et al. (2004). The goal is to elucidate potential shortcomings in the latter and obtain biologically plausible models of the local cortical circuitry. This eventually might pave the way towards a reconciliation between consistent models of cortical structure and realistic neuronal dynamics.

8.2 METHODS

We here construct and compare connectivity maps consisting of the connection probability from the different anatomical data sources. In contrast to the previous chapter, the laminar resolution used here distinguishes between layers 2/3, 4, 5, 6. The neurons in each layer are assigned to either an excitatory or inhibitory population. This resolution used in the data sets provided by Binzegger et al. (2004) (referred to as `Binzegger`) and MICrONS Consortium et al., 2021 (in the following referred to as `MICrONS`) is identical with the one employed here. Shapson-Coe et al. (2021) treat layers 2 and 3 separately. This data set will be referred to as `H01`.
The choice of the laminar resolution is motivated by minimizing the additional assumptions when estimating the connectivity maps. For the `H01` data, layer 2 and 3 are combined in the following derivations. First, we again determine the layer-resolved number of connections S^u_{AB} between the pre- and post-synaptic cortical populations B and A under a patch of cortical tissue below 1 mm × 1 mm.

BINZEGGER A brief description of the data is given in Section 7.2.2.1. Using the notation introduced there, for `Binzegger` we calculate the desired quantity as

$$S^{u,\mathrm{Binzegger}}_{AB} = \sum_{i\in A}\sum_{j\in B} N c_i s^u_i p^{|u}_{|ij} \tag{8.1}$$

Here, Nc_i is the total number of neurons of type i, s_i^u the number of synapses targeting a neuron of type i in layer u, and $p\big|_{ij}^{u}$ the probability for a connection to exist from a neuron of type j onto a neuron of type i with synapse in layer u.

EM RECONSTRUCTIONS In the reconstructions based on EM, the brain tissue is cut into thin (on the order of 30 nm) sections that are subsequently analyzed with a resolution of 8 nm × 8 nm. The obtained cuboids are referred to as segments. The main technical difficulty in reconstructing cells is their agglomeration (that is association of segments) across sections as well as the identification of synapses. This is a challenging task, especially in light of the size of the obtained data (on the order of petabytes). Hence, advanced computational methods have to be employed. Still, merge errors across sections cannot be excluded. For dendritic processes, however, automatic merging leads to satisfying results with low error rates (Shapson-Coe et al., 2021). More problematic is the agglomeration of axons. This is due to their small diameter in comparison with dendrites. To overcome this, certain neurons are manually proofread, i.e. their axons and dendrites are traced manually across sections, leading to a significantly better reconstruction.

We employ only proofread cells as pre-synaptic neurons in our derivation in order to avoid systematic errors induced axon break overestimating very local connections. On the target side, all identified neurons are potential candidates for post-synaptic cells. This is justified by the well agglomerated dendritic processes. Additionally, we exclude self-connections in our analysis.

In the `H01` dataset, the authors reconstructed an approximately 1 mm^3 volume of human temporal lobe from a resection of an epileptic patient. They identified 16 087 neurons as well as 134 synapses in the tissue. The resected volume covers a depth of up to 3 mm (from pia to white matter), a breadth of up to 2 mm and a width of 0.175 mm. The authors distinguished the cortical layers 1, 2, 3, 4, 5, as well as 6 and classified neural cells into three excitatory neuron types as well as inhibitory neuron types. In total, around 27% of the synapses were judged to be inhibitory, the rest excitatory. 104 randomly selected neurons close to one boundary of the long vertical side of the resection were proofread by the authors. Neurons of all layers and all populations except layer 6 inhibitory neurons were proofread. For more detailed information, we refer the reader to the original study (Shapson-Coe et al., 2021).

The `MICrONS` data provide a reconstruction of around 1 mm^3 volume of mouse visual cortex (VISp, VISrl, VISal and VISlm). The researchers identified around 17 700 neurons and 500 million synapses. The reconstructed volume consists of two subvolumes that were processed independently and aligned later on. Multiple different inhibitory neuron types were distinguished. In our analysis, we only differentiate between excitatory and inhibitory neurons. In this dataset, 349 individual neuronal cells from the larger subvolume were proofread. See MICrONS Consortium et al. (2021) for a more detailed description.

To determine $S_{AB}^{u,\{\text{MICrONS,H01}\}}$, we count the number of connections established by proofread cells of a source population B onto reconstructed neurons of a target population A. Since the cell types of the proofread neurons are not sampled uniformly, we correct for this by first calculating the number of connections established by one cell of the respective type. This number is then scaled up by the total number of cells of this type in the reconstructed volume. Finally, we scale up the counted number of connections so that $\sum_{A,B} S_{AB}^{u,\{\text{MICrONS,H01}\}}$ equals the total number of synapses located in layer u in the `MICrONS` and `H01` data set respectively. This ignores that a certain fraction of synapses in each layer may stem from thalamo-cortical or cortico-cortical projections. Since the total number of these synapses is small in comparison with all synapses in the reconstructed volume, and since precise figures estimating the respective fractions are not available, we choose the described strategy. Additionally, for the subsequent analyses the thus introduced error seems to be of little importance since the fractions of synapses where the pre-synaptic neuron resides in the same layer can be assumed to roughly range from 75% to 80% (Markov et al., 2011). In the case of the `H01` data, we sum over layers 2 and 3 and combine the values to obtain the respective numbers for layer 2/3.

Using the number of neurons from Beaulieu and Colonnier (1985) and Beaulieu and Colonnier (1983) when further processing $S_{AB}^{u,\text{Binzegger}}$, and the respective reported values from the reconstructions of `MICrONS` and `H01`, we determine the population-resolved connection probability as in Equation 7.67. This leads to three different models of the local cortical circuit denoted by $\mathcal{M}[\text{Binzegger}]$, $\mathcal{M}[\text{MICrONS}]$ and $\mathcal{M}[\text{H01}]$. Additionally, the target specificity Equation 7.68 and the EI-fractions Equation 7.69 of the three different models are calculated.

8.3 RESULTS

We first turn to an analysis of the connectivity (Figure 8.1a). All three models exhibit source- and target-specific connection probabilities. Generally, in most populations within-layer connection probabilities are comparatively strong. $\mathcal{M}[\text{MICrONS}]$ has the smallest pairwise connection probabilities, while $\mathcal{M}[\text{H01}]$ shows the largest ones. In $\mathcal{M}[\text{Binzegger}]$ and $\mathcal{M}[\text{MICrONS}]$, we observe a similar motive in the connectivity from excitatory to excitatory populations: There appears to be a strongly connected subnetwork linking L2/3E and L5E with themselves and among each other. While also present in $\mathcal{M}[\text{Binzegger}]$ between excitatory and inhibitory populations, this is not the case in $\mathcal{M}$[MICrONS]. In all models, inhibitory neurons show a slight preference towards targeting excitatory neurons within or above the layer of the pre-synaptic neurons. This is consistent with the greater abundance of specific inhibitory cell-types projecting from deeper to upper layers (e.g. Martinotti cells) compared to neurons projecting from upper to deeper layers (as e.g. bipolar cells) (Jiang et al., 2015). In case of inhibitory post-synaptic populations, this structure of the connectivity is less salient (for $\mathcal{M}$[Binzegger], $\mathcal{M}$[MICrONS]), or vanishes completely (for $\mathcal{M}$[H01]).

The layer-resolved number of neurons as well as the ratio between excitatory and inhibitory neurons vary across the different models (Figure 8.1b). We calculate the average indegree of the neurons in a given population (Figure 8.1c). In $\mathcal{M}$[MICrONS] and $\mathcal{M}$[H01], neurons belonging to the inhibitory population in a given layer receive more incoming connections than their excitatory counterparts. This is consistent with the significantly higher number of synapses per soma and per µm dendrite on inhibitory neurons as compared with excitatory neurons in mouse V1 (Wildenberg et al., 2021). Note, however, that this relation is reversed in $\mathcal{M}$[Binzegger].

Additionally, we assess connectivity patterns between excitatory and inhibitory neurons with the target specificity (Figure 8.2a). As already indicated in Chapter 7, in $\mathcal{M}$[Binzegger] only excitatory populations are preferentially targeted, irrespective whether the pre-synaptic population is itself excitatory or inhibitory. This is in stark contrast with the target specificity of $\mathcal{M}$[MICrONS] and $\mathcal{M}$[H01]: in the former, all excitatory populations preferentially target inhibitory neurons. In the latter, there is a diverse specificity to be observed for excitatory pre-synaptic populations where some establish connections onto inhibitory, some onto excitatory neurons with a higher probability. Similarly, inhibitory populations exhibit both preferential targeting of excitatory an inhibitory populations. Within one layer, the target specificity is always positive in $\mathcal{M}$[MICrONS] and $\mathcal{M}$[H01]. Comparing the target specificity of the models based on the EM reconstruction with the connectivity derived by Potjans and Diesmann (2014), we note a similar diversification of target specificity that cannot be found in $\mathcal{M}$[Binzegger].

Examining the EI-fractions (Figure 8.2b) reveals that in all models except $\mathcal{M}$[Binzegger] inhibitory neurons have a higher relative frequency of excitatory pre-synaptic partners than excitatory neurons. Thus, by preferentially targeting inhibitory neurons, in $\mathcal{M}$[MICrONS] and $\mathcal{M}$[H01] and the microcircuit model of Potjans and Diesmann (2014), increasing activity of excitatory neurons should increase the firing rate of some of the inhibitory populations, potentially leading to dynamical balance of excitation and inhibition. This is further supported by the high relative excitatory indegrees on inhibitory neurons. In contrast, in $\mathcal{M}$[Binzegger] it appears unlikely that a similar

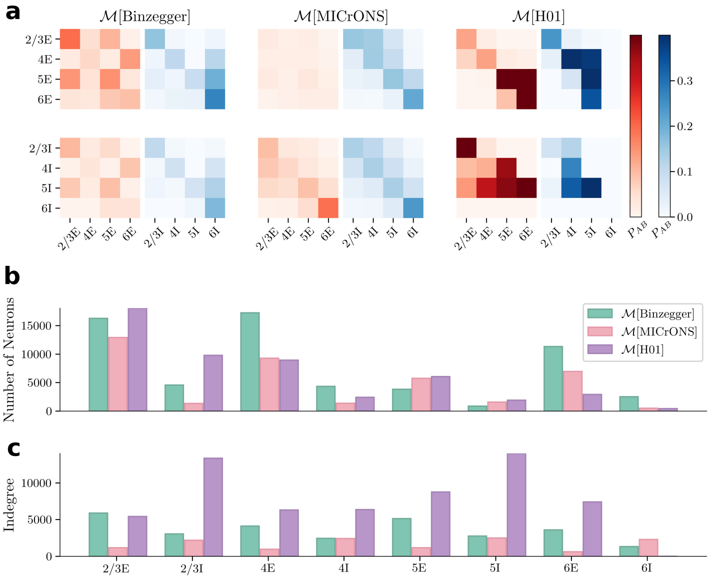


Figure 8.1: Connectivity of reconstructions based on three anatomical data sets. **a** Connection probabilities based on number of synapses between neurons of the respective populations. For better visibility, connection probabilities are capped at 0.4. **b** Number of neurons in different populations obtained from different datasets. **c** Indegree of neurons in different populations based on connectivity maps.

effect could catch run-away excitatory activity in models similarly constructed to the one in the previous chapter.

It is noteworthy that the EI-fractions reported by Shapson-Coe et al. (2021) differ from the ones we obtain for our model $\mathcal{M}$[H01]. In their work, they count the synapses onto excitatory and inhibitory neurons and do not construct a model integrating the anatomical information of their data. The counting procedure relying on the synapse being identified correctly as excitatory or inhibitory (instead of the neuron, as used in the derivation of our model) or the self-connections neglected in our analysis may explain the difference.

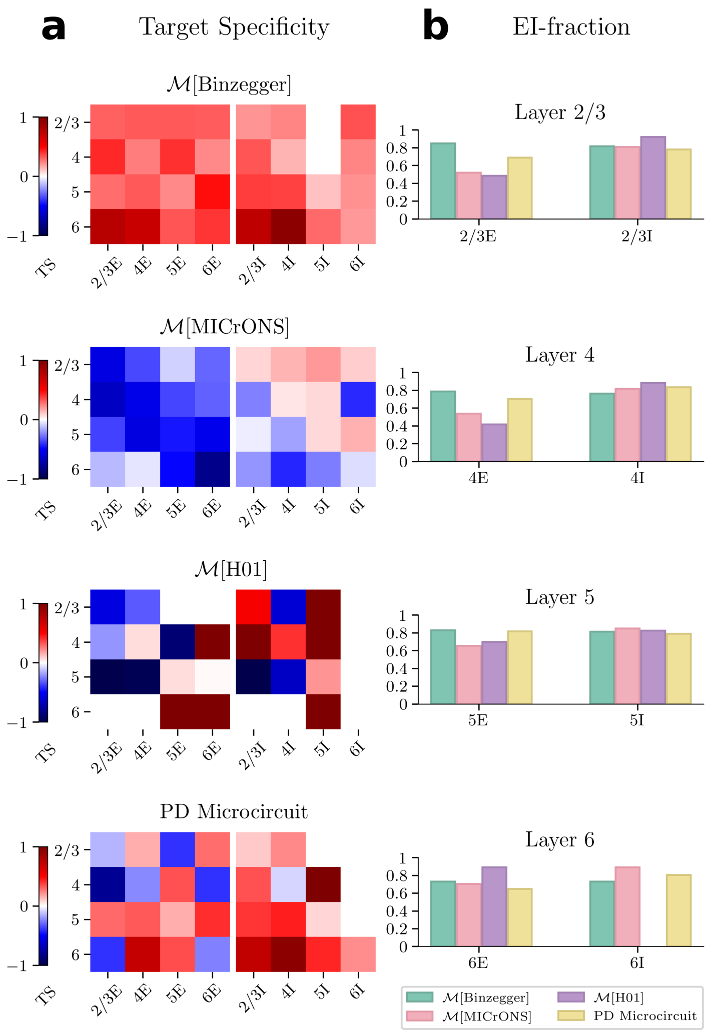


Figure 8.2: Target specificity and EI-fractions of models of local cortical circuits and model of Potjans and Diesmann (2014) (denoted by PD Microcircuit) **a** Target specificity of different models. **b** EI-fractions across layers of different models. Here, only the fraction for excitatory connections are presented. For $\mathcal{M}$[H01] no inhibitory neurons in layer 6 were proofread resulting in no data for this population.

8.4 DISCUSSION

In this chapter, we derived connectivity maps of the local cortical circuits based on three data sets obtained with different methodologies. First, we employed the connection probabilities obtained via the single cell reconstructions in cat V1 of Binzegger et al. (2004) combined with neuron and synapse counts by Beaulieu and Colonnier (1985) and Beaulieu and Colonnier (1983). The data was already used in the previous chapter for our model of the primary visual cortex of macaque. Additionally, two recent, large-scale reconstructions obtained with EM of mouse visual cortex and human temporal lobe were analyzed (MICrONS Consortium et al., 2021; Shapson-Coe et al., 2021). This technology enables a detailed reconstruction of the objects in the cortical tissue. Even though advancements in classifications and agglomeration of these objects across sections of cortical tissue allows for unprecedented insight into local cortical networks, the data is far from being a complete or correct representation. Nonetheless, it is reasonable to assume that these data offer the least biased view on cortical microcircuitry to this day.
The analysis presented here is motivated by the diverging dynamical activity in our model of macaque V1 (Figure 7.6). In the previous chapter, we traced this phenomenon back to excitatory populations preferentially target excitatory neurons in all layers. Here, we could indeed confirm that this property directly stems from the data by Binzegger et al. (2004) (Figure 8.2a). In the models based on the more recent and complete EM data, a more nuanced view on local cortical architectures emerges. While there are consistencies in the connection probabilities of $\mathcal{M}[\text{Binzegger}]$ and $\mathcal{M}[\text{MICrONS}]$, or the target specificity of $\mathcal{M}[\text{Binzegger}]$ and $\mathcal{M}[\text{H01}]$, there are also apparent differences. Especially the target specificity of $\mathcal{M}[\text{H01}]$ exhibits diverse patterns where both excitatory and inhibitory populations preferentially target excitatory or inhibitory neurons, depending on the layer. Additionally, for the connectivity maps based on the EM data, in a given layer the neurons of the inhibitory population have a higher EI-fraction than the excitatory ones. This is in contrast to the respective quantities in $\mathcal{M}[\text{Binzegger}]$.
The structural changes observed in the connectivity maps constructed here are similar to the amendments introduced by Potjans and Diesmann (2014). There, the authors identified candidates for the amendment based on the target specificity and altered an integrated connectivity map derived from anatomical and electrophysiological connectivity data. Since simulations of this model exhibit plausible, cell type specific firing rates (as is the case for the MAM V1, cf. Chapter 7), this suggests that these changes are critical for non-diverging network dynamics. Indeed, the differences in target specificity and EI-fractions for inhibitory neurons —in comparison with $\mathcal{M}[\text{Binzegger}]$ —indicate

that a stronger control of the network activity by inhibitory neurons is needed and can be achieved via a increased innervation of certain inhibitory subpopulations. This increased innervation on the structural level relative to the model based on the `Binzegger` data is suggested by the EM data.

We thus offer a consistent (with respect to the constructed model) view on why models based on the data by Binzegger et al. (2004) might be prone to diverging activity, and what might be different in biological cortical circuits. Additionally, new connectivity maps —especially from $\mathcal{M}$[MICrONS] —that can be used as an alternative basis for the derivation of our model of macaque primary visual cortex are presented. Indeed, $\mathcal{M}$[MICrONS] offers itself as the most promising candidate: the data used in the construction is the most complete and represents the cortical circuitry of V1 (albeit in mice), and the target specificity (Figure 8.2) suggests that a spiking network model based on this connectivity would have the smallest likelihood of exhibiting positive feedback loops and thus diverging activity in dynamical simulations (cf. Chapter 7). This raises the hope that the problems apparent in the spiking activity of our model of macaque V1 can be overcome by incorporating the anatomical insights gained in this chapter.

The obtained maps are, however, interesting unto themselves and can be investigated in their own right: since they provide models for different cortical areas in different species, their structure and thus spiking activity might reflect a functional specialization. In future work, quantities like the excitability (Maksimov et al., 2018) or the intrinsic timescales (Gao et al., 2020; Siegle et al., 2021) could be analyzed to investigate the structure-dynamics relationship in these models.

Additionally, in EM reconstructions the precise locations of pre- and post-synaptic neurons are known. Thus, better estimates for spatial decay constants might be obtained. These can be additionally used to investigate the relationship between the potential connectivity (e.g. (Stepanyants et al., 2008) or Section 7.2.3.2) and the actual connectivity. However, comparing the spatial decay constants directly, for example between the EM reconstructions and Section 7.2.3.2, is not straightforward due to differences in e.g. the size of neurons across species or structural differences in the medium-range connectivity across brain areas (Tanigawa et al., 2005).

Part IV

PERFORMANCE ASSESSMENT OF SIMULATION TECHNOLOGY

9 PREFACE

In order to conduct a research program as the one sketched in Part i where digital twins of cortical circuits are used to explore the relationship between the structure of the brain and its activity as well as function, simulation technologies are of great importance. In the last decades, the Computational Neuroscience community achieved a separation between the underlying mathematical models of a neuronal network and the employed technology for its simulation (Einevoll et al., 2019). With this, it became possible to execute the same neuroscientific model with different simulation technologies, ranging from various classical CPU approaches (Brette et al., 2007) to for example neuromorphic systems (Brüderle et al., 2009; Golosio et al., 2021; Knight et al., 2021; Rhodes et al., 2018).

The aforementioned separation highlighted the role of simulators as research infrastructure —akin to for example hardware in a laboratory —, that deserve attention and tendance in their own right. This view justifies, and indeed suggests giving that attention in a formalized way: simulation technologies need to be *reliable*, *efficient*, and *continuously developed*. In order to understand how this can be achieved, we briefly expand on each of these points:

Reliability ensures the correctness of the results making reproducible spiking neural network simulations possible —an only seemingly simple task with pitfalls (Pauli et al., 2018) and inherent difficulties (Gutzen et al., 2018).

Efficiency highlights the performance of the simulation technology —usually quantified by simulation speed and consumed energy or required memory during the execution of a simulation. To assess these quantities, strong and weak scaling experiments need to be conducted: in the former, the problem size remains fixed while the computational resources are increased allowing for investigating speed-ups gained from parallelization. In the latter, the problem size scales linearly with the resources employed. This allows to assess potential penalties induced just by solving larger problems. For a detailed discussion of weak and strong scaling see e.g. Hager and Wellein (2010).

Finally, **continuous development** focuses on the need to advance the simulation technology by satisfying new, ever changing requirements from the hard- and software side, competing with alternative approaches —in our case neuromorphic computing —and meeting the demands of the research community. This aspect is of paramount importance: Ultimately, the continuous development determines whether or not a simulation technology stands the test of time and can be used long term as a reliable, efficient tool for conducting research.

The need for appropriately addressing these issues is reflected in the emergence and professionalization of the field of *research software engineering* in the last decade or so (Baxter et al., 2012). Nonetheless, challenges remain: Research software engineers on their own often lack domain knowledge and require frequent exchange with domain scientists to guide the assessment and development of simulation technology. On the other hand, also researchers need to be involved more actively and professionally in the development process. Only with joint forces, addressing the aforementioned items enables incremental improvements in software infrastructure, ultimately facilitating neuroscientific research as the one conducted in Part iii.

In this thesis, in the context of High-performance computing (HPC) benchmarking, two aspects of the aforementioned points are addressed: In Chapter 10 we turn to performance benchmarks of a standardized benchmarking model of the general purpose simulation engine for large heterogeneous networks of point neurons NEST (Gewaltig and Diesmann, 2007) simulator on a recent conventional compute node. Here, we focus especially on the time-to-solution as well as the energy-to-solution. Using these metrics, we identify performance bottlenecks of the used version of the NEST simulator and compare its performance with alternative simulation technologies.

Motivated by the difficulties inherent to HPC benchmarking —also experienced by the author of this thesis in the work expounded in Chapter 10 —, in Chapter 11 we analyze the situation and propose a benchmarking workflow mitigating these problems. The theoretical workflow is supplemented with a reference implementation. Finally, we demonstrate how to use the devised workflow (in form of our reference implementation) to aid continuous simulator development: assessing whether certain features to be added to a new version of the NEST simulator indeed improve the chosen metrics, and retain performance and correctness.

10

SUB-REALTIME SIMULATION OF A NEURONAL NETWORK OF NATURAL DENSITY

The chapter is based on the following publication:

Kurth, A. C., Senk, J., Terhorst, D., Finnerty, J., & Diesmann, M. (2022). - Sub-realtime simulation of a neuronal network of natural density. Neuromorphic Computing and Engineering, doi: 10.1088/2634-4386/ac55fc

Author contributions:
Under the supervision of MD, the author conducted the benchmarking experiments and analyzed the data. An initial version of the benchmarking workflow was designed by JS. JF and DT provided technical support. The first version of the manuscript was written by the author and jointly revised by the author, DT, JF, JS, and MD.

10.1 INTRODUCTION

Simulations of spiking neural networks require efficient simulators to meet the demands of the research community.

Indeed, in order to investigate large-scale neuroscientific models (see e.g. van Albada et al. (2021)) or neuronal phenomena like learning, the temporal dynamics of which may unfold over hours, days and even years, simulations need to become faster. Similar, but even more pressing arguments hold true in the more application orientated fields of artificial intelligence and machine learning. Here, reducing the consumed energy —for example during training —is of paramount importance (Strubell et al., 2019).

Both, an increase in simulation speed as well as energy efficiency is a promise of neuromorphic computing (Furber, 2016) for both neuroscientific research and consumer orientated applications.

The progress in this field must, however, be compared with and validated against continuously advancing conventional approaches, which offer higher flexibility at potentially lower costs. Moreover, these conventional approaches are usually easier to use and require less dedicated training for the researchers, implying a broader group of potential users.

Comparisons between conventional and more dedicated approaches require standardized benchmark models to ensure meaningful assessments of the performance of the different technologies (see also Chapter 11 for a more detailed discussion of this topic). For spiking neural network models with biologically plausible neuron- and synapse-densities, the local cortical microcircuit model of Potjans and Diesmann (2014) (see Figure 10.4 a) has become such a standard benchmark model.

The model represents the neural tissue below a 1 mm^2 patch of cortical surface and incorporates four cortical layers. Each layer consists of an excitatory and inhibitory population of integrate-and-fire model neurons with exponential postsynaptic currents. Cell-type specific connection probabilities replace the more involved distance dependent cortical connectivity. The microcircuit model can be routinely simulated using different simulation technologies with only moderate hardware requirements (Knight and Nowotny, 2018; Rhodes et al., 2019; van Albada et al., 2018). The realistic number of about $10,000$ synapses per neuron and, at the same time, a connection probability of about 0.1 renders questions of downscaling irrelevant and maximizes the connection density in biological plausible models of cortical networks (van Albada et al., 2015a). Thus, larger biologically plausible network models are necessarily less densely connected and hence are, relative to the problem size, easier to simulate.

In this chapter, we set out to investigate the performance —assessed with simulation speed and consumed energy —of the general purpose simulation engine NEST (Gewaltig and Diesmann, 2007) on a recent conventional computing system.

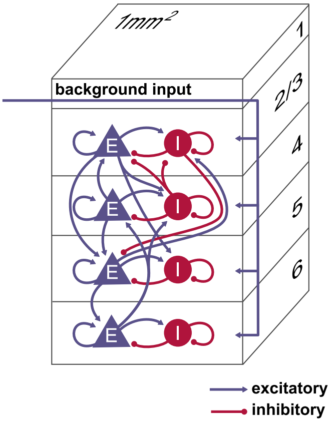


Figure 10.1: Sketch of the microcircuit model with about 80,000 neurons and 300 million synapses organized into four layers of excitatory (blue) and inhibitory (red) populations of neurons.

10.2 METHODS

We simulate the microcircuit model on 128 core dual socket AMD EPYC Rome 7702 compute nodes. Each processor is composed of 8 chiplets, each chiplet holds 8 cores, resulting in 64 cores per socket (see Figure 10.2) Each core has its own L1 and L2 cache, 4 cores share an L3 cache (see Figure 10.3). Two nodes are coupled by a point-to-point Mellanox ConnectX-6 HDR100 interconnect. The software is NEST 2.14.1 (Peyser et al., 2021) (compiled with GCC 6.3.0 and using jemalloc 3.6.0-9.1, Evans (2006)), providing, in contrast to some neuro-

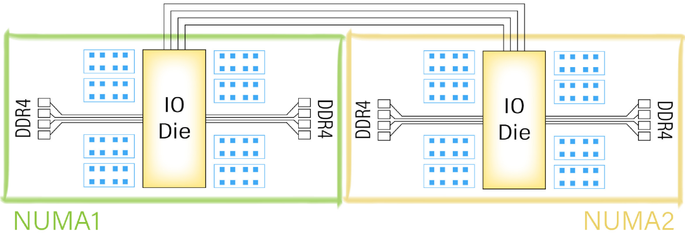


Figure 10.2: Sketch of hardware architecture of the dual socket AMD EPYC Rome 7702 system used in this chapter. Solid blue squares indicate compute cores. 8 compute cores are combined into one chiplet.

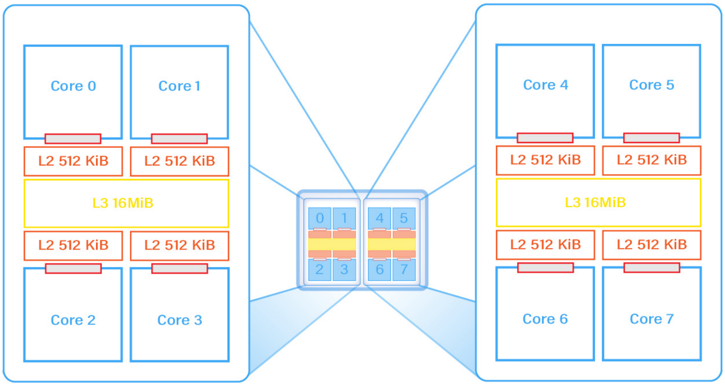


Figure 10.3: Sketch of one chiplet of the AMD EPYC Rome 7702. 4 cores are grouped into one core complex sharing an L3 cache.

morphic systems, double precision numerics and weight resolution. NEST utilizes the Message Passage Interface (MPI, here OpenMPI 4.0.3rc4, Gabriel et al. (2004)) and employs hybrid parallelization with multithreading (OpenMP, OpenMP Architecture Review Board (2008)) for shared memory parallelization where a core never runs more than one thread. Timers monitor the different phases of the simulation: *update* of neuronal states, *communication* of the spikes between MPI processes, and *delivery* of the received spikes to the thread-local targets. By definition, communication is zero if no hybrid parallelization is employed. The employed alternative memory allocator jemalloc has no influence on the time required to propagate the dynamical state of the neuronal network but reduces the time needed to construct the network. This is due to the non-blocking distribution of memory to threads (Ippen et al., 2017).

We perform strong scaling experiments (see Chapter 9). This means, we keep the task size fixed while systematically increasing the computational resources (Figure 10.4b). The task is a simulation of 10 s of model time (T_{Model}), referring to the span of biological time described by the model, if not stated otherwise. Measurements start after model instantiation with optimized initial conditions (Rhodes et al., 2019) and an initial interval of 0.1 s of model time to ensure that potential transients of the network dynamics are discarded. To assess simulation speed we use the realtime factor:

$$\text{RTF} = \frac{T_{\text{Wall}}}{T_{\text{Model}}}$$

Here, T_{Wall} denotes the wall-clock time; the time passed in the machine hall until the simulation completes. A realtime factor smaller than 1 implies sub-realtime performance.

To obtain an optimal performance, the threads need to be bound to the compute cores. This means that each thread is local to one compute core and cannot change the core it runs on. To bind threads to cores on our system we, export OpenMP variables as follows:

```
export OMP_NUM_THREAD = $ CPUSPERTASK
export OMP_PROC_BIN = TRUE
export OMP_PLACES = {0},{8},{15}
```

Here `$CPUSPERTASK` is the number of cores used in a given setup (in this example 3) and `{0},{8},{15}` indicate the first core on the first, second and third chiplet.

We employ two different thread binding (also called placing) schemes on one node to assess the performance given a fixed number of threads (or equivalently, compute cores):

In the *sequential* placing scheme, threads are bound onto physically consecutive cores. Thus, first a chiplet and subsequently a socket is filled when increasing the number of threads used before starting with the next.

In the *distant* placing scheme, threads are bound such that L3 cache sharing given a number of threads is minimized. To be precise, we number the chiplets $0, ..., 15$ where $0, ..., 7$ identify consecutive chiplets on one socket and $8, ..., 15$ on the other (see Figure 10.2). The numbering is induced by the standard output of `lstopo`, which is included in several Linux distributions. The command returns a numbered list of the cores on the compute node hierarchically structured by the NUMA nodes (in our case equivalent to the sockets), the L3 cache and the L1/L2 cache. Cores 0 to 63 and L3 caches 0 to 15 are located on NUMA node 0, cores 64 to 127 and L3 caches 16 to 31 on NUMA node 1. Since on one chiplet two L3 caches are located, one obtains the number of a chiplet by an integer division of the number of the respective L3 cache by 2. We denote the $k-$th core, $k \in \{0, ..., 7\}$ (sketched in Figure 10.3), on the $n-$th chiplet by $n : k$. In the distant placing scheme the filling of a compute node from 1 to 128 threads on one node is split into 8 rounds each addressing a particular core k of the chiplets and successively adding this core of chiplet n (16 in total) to the simulation. This results in $8 \times 16 = 128$ threads being bound to cores. The filling procedure starts with core 0 of each chiplet on the first socket before populating the second socket, i.e. the first 16 simulations use the cores $\{0 : 0\}, \{0 : 0, 1 : 0\}, ..., \{0 : 0, ..., 15 : 0\}$. Here, we employ the $n : k$ notation with the first index (chiplet) changing before the second (core) does, and denote by a set $\{...\}$ of cores their simultaneous use in a benchmarking simulation. The next round uses cores still not sharing an L3 cache with cores already in use. We chose the $4-$th, resulting in consecutively adding the cores $0 : 4, 1 : 4, ..., 15 : 4$ to the set of simultaneously used cores. The following rounds continue with the $2-$nd, $6-$th, $1-$st, $5-$th, $3-$rd and $7-$th core respectively, minimizing shared use of L3 cache.

Simulations on one node are launched by

```
python3 run_microcircuit.py
```

When using two nodes, simulations are launched by

```
mpirun --n 2 --npernode 1 --mca pml ucx -x UCX_NET_DEVICES=mlx
    5_1:1 --bind-to board python3 run_microcircuit.py
```

in this example with 1 MPI process per node.

We moreover determine the number of cache misses in some simulations. For this we employ the perf performance analysis tool of the Linux operating system. We use the command options

```
perf stat -ae task-clock,cycles,instructions,cache-references,
    cache-misses
```

and increase the simulation time to 100 s. Thereby we ensure that approximately 80% of the run time of the program is spent in the simulation phase guaranteeing a reliable assessment of the percentage of cache misses during that phase.

A common measure for comparing the energy consumption of neuromorphic systems is energy per synaptic event defined as total consumed energy divided by the total number of transmitted spikes. Here, power was measured with a Raritan Dominion PX and a Raritan PX3-5190 power distribution unit (PDU). The units have an accuracy of ±5% and data collection frequency of 1 Hz. The power measurement has a delay of 1 s, so that the power readings need to be shifted by 1 s to be aligned to wall-clock time. Since the nodes are connected point-to-point, we do not need to take additional passive energy consumption by an interconnect into account.

For conducting the benchmarks we employ the JUBE (Lührs et al., 2016) benchmarking environment.

10.3 RESULTS

We assess the strong scaling performance of microcircuit model simulations by using 1 MPI process per node and increasing the number of threads on up to two compute nodes with the two different placing schemes introduced above. Simulations on two full nodes employ 256 threads and in total 2 MPI processes.

For sequential placing, we observe linear scaling for a thread count between 1 and 32 as well as super-linear scaling between 32 and 64 (Figure 10.4(a)). A full compute node achieves sub-realtime performance with an RTF of 0.72. Two nodes reduce the realtime factor to 0.63; the simulation runs 1.6 times faster than realtime. The distant placing scheme exhibits super-linear scaling already for a small number of threads. At 33 threads, we note a sudden rise of the realtime factor. At this point, the L3 cache is shared for the first time. Nevertheless, sub-realtime performance is already achieved when using only 64 threads. Comparing the two placings at 128 and 256 threads respectively, we observe identical performance. The reason is that due to the round-robin distribution of neurons across threads (Morrison et al., 2005a), the sequential and the distant placing schemes become indistinguishable up to the mapping of thread identifiers to core identifiers if all cores of the compute nodes are in use. Control simulations show that the compute nodes reach even lower real-time factors if the threads are split into 2 MPI processes per node; binding one MPI process to each socket leads to an RTF of 0.7 on one node and of 0.53 on two nodes. The relative time spent in the update phase on a single node is decreased in the distant placing when compared with the sequential one and communication between the two nodes is not a limiting factor. This suggests that simulation time can be further reduced by increasing the number of nodes and alternatively using faster nodes.

We also assess the energy consumption of the simulation phase to investigate how the increased power uptake due to using more computational resources is counterbalanced by decreased simulation time (Figure 10.4(b)). For this we compare a configuration using all 128 cores of a node with two configurations using only half of the cores. The former sequentially fills the cores of one socket, the latter employs the distant placing scheme. During simulations of 100 s of model time we record the power consumption and obtain the energy consumed in the simulation phase by integrating over the power readings.

We observe that power consumption during the simulation phase is largest for the distant placing of 64 threads, amounting to 0.39 kW subtracting the baseline power of 0.2 kW. This is almost twice the power as in the sequential configuration (0.21 kW). Nevertheless the increase cannot be attributed to the use of the second socket. The 128 thread configuration consumes 0.33 kW which is close to the same power required per thread of the sequential case. The counterintuitively low power consumption in the 128 threads case may be explained by the potentially longer latencies resulting in the cores not working at full capacity. Measuring the number of cache misses confirms a relative frequency of 25% in distant as compared with 43% in sequential placing. Ultimately, the 128 thread configuration does not only exhibit the shortest time to solution but also requires the smallest amount of energy.

The energy per synaptic event for the two fastest configurations (128 and 256 threads with 2 MPI processes per node) are 0.33 µJ and 0.48 µJ, respectively.

Neuroscientific projects routinely require multiple runs of a model with different initial conditions to collect statistics of its dynamics or with different settings to map out a parameter space. The superlinear scaling of a compute node in the simulation of the microcircuit model raises the question whether it is more efficient to run simulations in sequence on the full machine instead of dividing up the computational resources between multiple independent instances of the model. Table 10.1 compares as an example the wall-clock time for

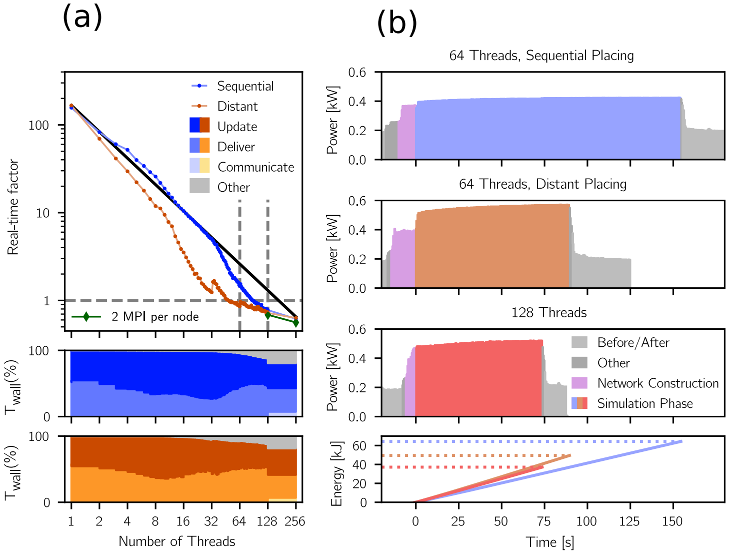


Figure 10.4: Strong scaling of a cortical microcircuit model on a conventional compute node. (a) Strong scaling for two placing schemes. Top graph shows realtime factor over total number of threads; dashed horizontal line indicates realtime; black solid line indicates linear scaling. The sequential scheme (blue) minimizes distance of threads on hardware, the distant scheme (brown) maximizes it both using 1 MPI process per node; pale colored line segments connect data points, dashed vertical lines indicate number of cores per processor (64) and node (128). On node hybrid parallelization by 2 MPI processes (1 per socket) for full nodes for comparison (green). Bottom graphs show fractions of wall-clock time consumed by different stages of the simulation cycle; update: integrates state of neurons, deliver: distributes spike events to target neurons, communicate: transfers spikes between MPI processes (for shared and distributed memory setups), other: not accounted for by timers. (b) Top three graphs: Power measurements of a compute node during 100 s of model time in three configurations. The measurements are aligned to the start of the simulation phase starting at $t = 0$ (legend: colors distinguish phases and baseline). Bottom graph: Cumulative energy consumption of the simulations.

the simultaneous simulation of 4 microcircuit models by independent instances of the simulation code with the total time required for 4 subsequent simulations of the model each using the full node. Indeed the super-linear reduction of simulation time by a higher degree of parallelization helps to earlier complete the task of carrying out a total of 4 simulations. The control of running a model on a quarter of the machine in isolation is already slower than the 4 subsequent simulations on the full node due to the super-linear scaling. The stress

T_{Wall} [s]	Configuration of compute node
72.64 ± 0.14	4 simultaneous simulations, one quarter of node each
49.43 ± 0.11	1 simulation, quarter node (control)
30.60 ± 0.16	4 subsequent simulations, full node each

Table 10.1: Wall-clock time T_{Wall} of alternative node configurations for multiple independent simulations of $T_{\text{Model}} = 10$ s of biological time. All configurations run a single process with sequential placement of threads (quarter: 32, full: 128). The standard error of the mean is estimated over 15 repetitions. The simultaneous configuration shows the mean over the largest of the 4 individual T_{Wall} measurements. The single and the subsequent case are the same configurations as in Figure 10.4 where for the subsequent simulations 4 times the mean and its standard error of a single run is displayed. Same network model and hardware as in Figure 10.4.

the 4 simultaneous jobs exert on the system slow the simulations by a further 50% relative to the control. As in the remainder of this chapter, T_{Wall} excludes the times required for starting and ending a simulation and for network instantiation constituting offsets independent of T_{Model}. Nevertheless, these contributions to the total run time of a simulation may be relevant, depending on the neuroscientific project at hand.

10.4 DISCUSSION

In this chapter we showed that a single compute node achieves subrealtime performance in the simulation of a natural density local cortical microcircuit model. To our best knowledge, for conventional system or the increasingly available GPUs[1], we report the lowest realtime factor so far —at a competitive energy consumption (Table 10.2). There are, however, studies (Heittmann et al., 2022; Kauth et al., 2023) reporting even smaller realtime factors for simulators using dedicated FPGA supercomputers. The simulation technologies presented there suffer from the decreased flexibility imposed by the hardware architecture. Heittmann et al. (2022), generate the connectivity procedurally, making simulations with synaptic plasticity impossible. For Kauth et al. (2023), the scalability of the system is limited by the topology of the wiring of the FPGA nodes in the compute cluster. While technologically impressive, the limited availability of the used hardware and the decreased flexibility make it unlikely that FPGAs will play a significant role in the simulations of spiking neural networks in the upcoming years.

A comparison with previous studies (Table 10.2) yields that conven-

[1] as enabled by modern super-computing facilities, see e.g. https://www.fz-juelich.de/de/ias/jsc/jupiter

RTF	$E_{\text{syn}-\text{event}}$ $[\mu\text{J}]$	Reference
6.29	4.39	2018, NEST(van Albada et al., 2018)
2.47	9.35	2018, NEST(van Albada et al., 2018)
26.08	0.30	2018, GeNN(Knight and Nowotny, 2018)
1.84	$0.47^{\dagger}$	2018, GeNN(Knight and Nowotny, 2018)
1.00	0.60	2019, SpiNNaker(Rhodes et al., 2019)
1.06	–	2021, NeuronGPU(Golosio et al., 2021)
0.70	–	2021, GeNN(Knight et al., 2021)
0.67	0.33	2022, NEST, AMD EPYC Rome, 1 node, 2 MPI
0.53	0.48	2022, NEST, AMD EPYC Rome, 2 nodes, 4 MPI
0.25	$0.78^{\dagger\dagger}$	2022, CsNN(Heittmann et al., 2022)
0.05	0.05	2023, neuroAIx, (Kauth et al., 2023)

Table 10.2: Realtime factor (RTF) and energy per synaptic event ($E_{\text{syn}-\text{event}}$) reported in the literature for simulations of the cortical microcircuit model (Potjans and Diesmann, 2014) using conventional hardware for NEST simulations, GPUs for GeNN and NeuronGPU, the dedicated neuromorphic hardware SpiNNaker and dedicated FPGA systems for CsNN and neuroAIx in historical sequence (top to bottom). The two values reported for NEST and GeNN in 2018 (corresponding to the most energy efficient and the fastest configuration) are obtained with a different number of employed cores and different GPUs, respectively. The values for CsNN and neuroAIx are obtained with different FPGAs. †Value estimated by Knight and Nowotny (2018) ††Value estimated by Kauth et al. (2023).

tional architectures keep pace regarding both: Realtime factor and energy efficiency.

The here employed generic simulation engine for spiking neuronal networks explicitly stores the connections between neurons with double floating point precision. Thus, although not exploited, plasticity and learning are possible in this representation.

We neither attempt to optimize the simulation code for the particular network model at hand, nor to find the optimal division of threads into MPI processes for the particular hardware. In comparison to prior work by van Albada et al. (2018), where an earlier version (2.8.0) of the code and older hardware is used, we observe a ten-fold improvement in performance. The older system suffers from the communication between nodes as a bottleneck. The newer hardware pushes the limits by integrating a larger number of computational cores into the nodes. The analysis shows that on a single node faster completion of the task comes with a lower energy consumption due to the substantial baseline power. The simulation time reduces if cores have a larger amount of cache available, and if all cores are in use, power consumption is lower than for half of the cores with optimal cache access. These observations indicate that threads suffer from cache misses and the resulting latencies in memory access. This does not only give practical guidance for the design of conventional hardware but also raises hope that methods of prefetching and latency hiding can further improve simulation code without restricting generality (Pronold et al., 2021).

Achieving realtime performance is a criterion for robotics. But for basic research and medical applications, also faster simulations are of use, because biological processes extending over long periods of time can be observed on a reduced time scale. Our results also give helpful advice for the practicing researcher: We find that on a given hardware sequential execution can be faster than simultaneous execution of the same number of independent simulations. While traditionally parallelization is employed to aggregate the total memory required for a simulation or to reduce the required wall-clock time of an individual simulation to a level enabling routine research, here the purpose is the efficient execution of algorithms with unpredictable memory access. Moreover, our results show that —possibly cache sensitive —binding of threads may increases performance. Armed with this knowledge, they can optimize the simulation time for multiple executions required for their research.

Our findings moreover confirms the view of Pronold et al. (2021) that for spiking neuronal networks parallelization overcomes the von Neumann bottleneck.

The constructive competition between neuromorphic hardware and conventional computer architectures led to two orders of magnitude improvement within just five years (Table 10.2).

11 BENCHMARKING NEURONAL NETWORK SIMULATIONS

The chapter is based on the following publication:

Albers, J., Pronold, J., Kurth, A. C., Vennemo, S. B., Haghighi Mood, K., Patronis, A., Terhorst, D., Jordan, J., Kunkel, S., Tetzlaff, T., Diesmann. M. & Senk, J. (2022). - A modular workflow for performance benchmarking of neuronal network simulations. Frontiers in Neuroinformatics, doi: 10.3389/fninf.2022.837549

Author contributions:
Under the supervision of MD, JS, and TT, the author, together with JA, JP, SV, JS and KM, generalized and unified the previous work and developed a standard for including new models. The author, DT, and KM worked on Builder and the author and KM worked on its integration into the benchmarking framework. AP, JJ, SK, JA, and JS provided code that was benchmarked using the framework. The author and JA worked on a generic visualization platform and its inclusion into the framework. The author, JA, and DT worked on metadata management. The benchmarks were performed by JA. All authors contributed to the writing of the manuscript.

11.1 INTRODUCTION

HPC benchmarking suffers from inherent difficulties. These difficulties need to be mitigated to systematically use benchmarks for assessing the reliability as well as efficiency of simulation technologies, and ultimately for facilitating their continuous development.

And indeed: Already in the comparatively simple setup used by the author of this thesis in Chapter 10, keeping track of different metadata relevant to the execution of the simulation code posed non-trivial problems. Among those were for example the used compute nodes as well as the software dependencies needed for the installation of the simulator. Especially the complex interplay between system and user-installed software is prone to introduce performance penalties on HPC systems and needs to be kept track of accurately. Additionally, a transparent, coherent way of displaying the results, highlighting —and not accidentally obscuring —the differences in performance was not straightforward to achieve.

Besides that, the ubiquity of benchmarks for different simulation technologies (see e.g. the LINPACK benchmarks Dongarra et al. (2003) initially released in 1979) clearly demonstrate their usefulness and relevance. This motivates a fundamental analysis of HPC benchmarking from an abstract point of view.

The aim of this chapter is to conduct this analysis and device a benchmarking workflow and subsequently framework for the reliable and reproducible use of benchmarks in the development of simulation technology. As a use case, the devised framework is then applied in the development of the NEST simulator.

Figure 11.1 illustrates the complexity of benchmarking experiments in simulation science and identifies five main dimensions: *Hardware configuration*, *Software configuration*, *Simulators*, *Models and parameters*, and *Researcher communication*.

We first set out to explore these dimensions with a view towards neuronal network simulations.

Hardware and software configurations —as already mentioned above —have a complex interplay. Moreover, both underlie updates and frequent releases. They can have difficult to predict influences on performance results. In addition, different laboratories may not have access to the same machines. Therefore, HPC benchmarks are performed on different contemporary compute clusters or supercomputers. For example, NEST benchmarks have been conducted on the systems located at Research Center Jülich in Germany but also on those at the RIKEN Advanced Institute for Computational Science in Japan e.g., Helias et al., 2012; Jordan et al., 2018. The wider availability of other technologies as GPUs only complicate the picture further. Meaningfully comparing results across different hardware and software configurations requires export knowledge of the compared technology.

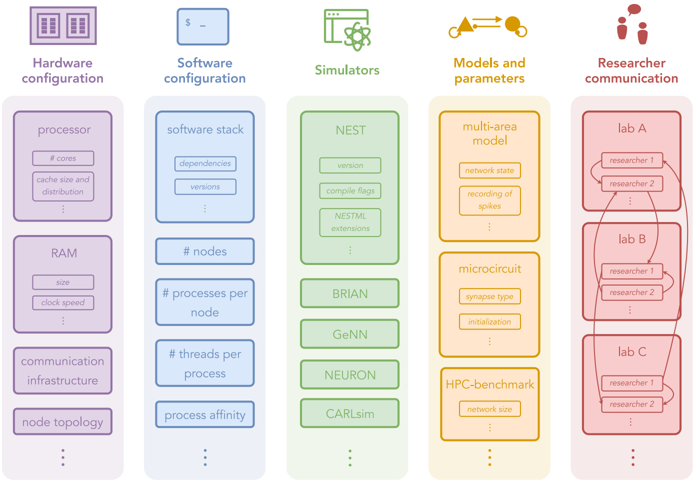


Figure 11.1: **Dimensions of HPC benchmarking experiments with examples from neuronal network simulations.** Hardware configuration: computing architectures and machine specifications. Software configuration: general software environments and instructions for using the hardware. Simulators: specific simulation technologies. Models and parameters: different models and their configurations. Researcher communication: knowledge exchange on running benchmarks.

A variety of **simulators** have been developed in the last decades that allow for large-scale simulations in Neuroscience (Brette et al., 2007). Examples are `NEST` (Gewaltig and Diesmann, 2007), `Brian` (Goodman and Brette, 2013), and `Neuron` (Carnevale, 2007) on CPUs and `CARLsim` (Beyeler et al., 2015), `GeNN` (Yavuz et al., 2016) and `NEST GPU` (Tiddia et al., 2022) on GPUs as well as the `SpiNNaker` neuromorphic system (Furber et al., 2014). Each simulator has a different scientific and technological focus. Yet, they also overlap in functionality. This motivates performance comparison between them.

Network models fall into two categories: *Functional models* come with an objective —for example finding an optimal solution to a task —that needs to be satisfied. Most commonly, these network models meet the objective by changing connection parameters via synaptic plasticity or learning mechanisms. On the other hand, *non-functional models* are devised to exhibit certain structural and dynamical features, which are meant to resemble corresponding properties in biological networks or physiological data. Also in this case the network structure might change during the simulation, again via e.g. synaptic plasticity or structural plasticity (Diaz-Pier et al., 2016). The interest in these networks is, however, primarily driven by studying the network model, or rather its activity, as a dynamical system.

While functional models, executed for example with different simulators, offer by their very construction a validation metric, non-functional models can usually only be compared on a statistical level. Spiking activity, for example, is typically evaluated based on distributions of quantities such as the average firing rate, rather than on precise spike times (Senk et al., 2017; van Albada et al., 2018). Reasons for that are inevitable differences between simulators such as different algorithms, number resolutions, or random number generators, combined with the fact that neuronal network dynamics is often chaotic, rapidly amplifying minimal deviations (Monteforte and Wolf, 2010; Sompolinsky et al., 1988; van Vreeswijk and Sompolinsky, 1998).

Testing and comparing new technologies only with a single network model is not sufficient for general-purpose simulators. Doing so might easily result in optimizations only improving certain aspects of the technology that are dominant in the used benchmarking model, and potentially penalizing others. Thus, in the subsequent sections, we employ three non-functional network models for benchmarking: The *HPC-benchmark model* (Jordan et al., 2018), the *microcircuit model* (Potjans and Diesmann, 2014) and the *multi-area model* (Schmidt et al., 2018b). The HPC-benchmark model employs leaky integrate-and-fire (LIF) neurons, alpha-shaped post-synaptic currents, and spike-timing-dependent plasticity (STDP) between excitatory neurons. The microcircuit model has already been described in Chapter 10. The multi-area model comprises of all 32 vision related areas in macaque monkey, where each area is modeled at the resolution of and akin to the microcircuit model. We note that the HPC benchmarking model can be easily scaled without changing the statistics of the activity, and thus allows for weak scaling experiments (c.f. Chapter 9).

Finally, problems in reproducing simulations do not arise from technical difficulties alone. **Researcher communication** is prone to errors due to a lack of documentation as well as implicit assumptions and localized knowledge on the side of the conducting researchers. Evidently, manual solutions for tracking hardware and software parameters, specifics of the simulator and the used models become unfeasible when the number of involved researchers increases. This issue is only amplified if multiple laboratories are involved.

Standardizing benchmarks can help to control the complexity but represents a challenge for the fast-moving and interdisciplinary field of Computational Neuroscience. While the field had some early success in the area of compartmental modeling (Bhalla et al., 1992) and Brette et al., 2007 made initial steps for spiking neuronal networks, neither a widely accepted set of benchmark models nor guidelines for performing benchmark simulations exist (but see the performance results for the microcircuit model across multiple simulation technologies presented in Chapter 10, Table 10.2). For the closely related field of deep learning, Dai and Berleant, 2019 summarize seven key properties that benchmarking metrics should fulfill: Relevance, representativeness, equity, repeatability, cost-effectiveness, scalability, and transparency. There exist standard benchmarks for machine learning and deep learning applications such as computer vision and natural language processing with standard data sets and a global performance ranking. The most prominent example is MLPerf[1] (Mattson et al., 2019). Ostrau et al., 2020 propose a benchmarking framework for deep spiking neural networks and they compare results obtained with the simulators Spikey (Pfeil et al., 2013), BrainScales (Schemmel et al., 2010), SpiNNaker, NEST, and GeNN.

For measuring and comparing the scaling performance of large-scale neuronal network model simulations, there exists, to our knowledge, no unifying approach yet. Here, our experience and the above analysis hint at the need of a shared benchmarking framework. The five dimensions outlined above suggest themselves to a modular framework integrating distinct components which can be updated, extended, or replaced independently. The framework needs to cover all steps of the benchmarking process from configuration, to execution, to handling of results. For enabling comparability and reproducibility, all relevant metadata and data need to be tracked.

In this chapter, we present a conceptual benchmarking workflow that meets these requirements. For a reference implementation of this workflow named beNNch, we employ the JUBE Benchmarking Environment (Lührs et al., 2016) (also used for the work presented in Chapter 10) and the simulator NEST in different versions (Gewaltig and Diesmann, 2007), on the contemporary supercomputer JURECA-DC (Thörnig and von St. Vieth, 2021). We additionally exemplify how to use the developed framework together with the reference implementation in simulator development. To this end, we execute the aforementioned three benchmarking models for different suggested improvements of the NEST simulator and compare them against ground truth versions. This ultimately justifies adding these improvements to NEST.

In the next section (Section 11.2) we briefly introduce NEST development as well as the suggested improvements mentioned above. Then we formalize the general concepts of the benchmarking workflow, implement these concept for the NEST simulator and apply the framework to generate benchmarking data for simulator development (Section 11.3). We end with a discussion in Section 11.4.

[1] https://mlcommons.org

11.2 METHODS

11.2.1 *NEST developments*

Our aim is to use the workflow proposed in Section 11.3 for assessing putative improvements for the NEST simulator. Here, we briefly introduce NEST and suggest changes to the code base the performance impact of which needs to be critically assessed with the developed benchmarking framework.

11.2.1.1 *Brief history of NEST*

The series of NEST 2.X releases includes enhancements, bug fixes, and contributions to maintenance with only marginal effects on the PyNEST user interface (Eppler et al., 2009). Performance-related updates to the simulation kernel are accomplished under the hood. The 3g kernel (Helias et al., 2012; Kunkel et al., 2012) is in use from NEST 2.2.0 (van Albada et al., 2015b). NEST 2.12.0 (Kunkel et al., 2017) introduces the 4g kernel (Kunkel et al., 2014) which implements novel data structures allowing for an efficient and flexible representation of sparse network connectivity on highly distributed computing systems such as supercomputers. The 5g kernel (Jordan et al., 2018) in NEST 2.16.0 (Linssen et al., 2018) continues this direction of development toward an optimal usage of HPC systems for large-scale simulations by disentangling the memory usage per compute node from the total network size. The transition from NEST 2 to NEST 3 corresponds to a refurbishment of the simulator code which also breaks the backwards compatibility of the user interface. While improved high-level functionality and parameter handling are the primary goals of this transition, the 5g kernel is supposed to remain. In the past, performance changes due to kernel updates have been predominantly assessed using the HPC-benchmark model. The performance of the NEST 3.0 release candidate ("3.0rc"), however, is in addition evaluated with the microcircuit and multi-area model which exhibit a more complex connectivity structure and a different distribution of synaptic delays. In this way, so far undetected performance bottlenecks are discovered and subsequently resolved, leading to the official release NEST 3.0 (Hahne et al., 2021).

11.2.1.2 *Shrinking MPI buffers*

Motivated by reducing the memory footprint of the postsynaptic infrastructure—necessary to deliver spikes to their process-local targets—the 5g kernel of NEST 3.0rc prepares a separate part of the MPI send buffer for each target process and only includes the relevant spikes. Thus, each process is responsible for sending the spikes of its neurons to all target processes for each communication time step. NEST 3.0rc implements a homogeneous buffer size across processes to avoid overhead introduced by variable buffer sizes; in the latter case, each process would need to complete two rounds of communication, one for transmitting the size, and one for the actual spiking information. Similarly, transmitting a certain amount of information via sending MPI buffers is more efficient when fewer buffers—each carrying more information—are sent. NEST 3.0rc consequently aims to reduce the number of needed MPI buffers to only 1 by dynamically increasing the global buffer size whenever a process cannot fit all spikes into the buffer. Specifically, every time more than a single buffer needs to be sent by a process, NEST increases the buffer size of the following communication step by a factor of 1.5. In this scheme, a reduction of buffer sizes is not implemented, meaning that buffer sizes can only increase or stay constant. The kernel of NEST 3.0rc+ShrinkBuff addresses this by introducing the following algorithm for shrinking the global buffer size. In each communication round in which only a single send buffer is required, the buffer for the following round decreases by a factor of 1.1. Even though this implementation leads to an oscillation of buffer size for constant spiking activity, tests show that this simple mechanism only introduces negligible cost while being robust.

11.2.1.3 *Spike compression*

NEST's 5g kernel (Jordan et al., 2018) introduces a two-tier connection infrastructure for routing spikes. The connection infrastructure consists of data structures on the presynaptic side (the MPI process of the sending neuron) and the postsynaptic side (the MPI process of the receiving neuron), cf. Section 11.3.2.3. Communication of spikes is organized as follows: when a neuron becomes active, its targets are retrieved from the local presynaptic data structure. These targets represent indices of synapses in the "thread-local" postsynaptic data structure through which spikes are routed to the target neurons. The presynaptic side then creates MPI buffers containing collections of such indices which are subsequently communicated to the postsynaptic side via the MPI Alltoall function. To deliver spikes on the postsynaptic side, each thread uses the received spikes to index its local postsynaptic data structure and register a spike in the corresponding synapse (Figure 11.2, "original spike delivery"). If a presynaptic neuron has targets on multiple threads of a process, it hence has to send multiple spikes, i.e., indices in different thread-local data structures, to the target process.

Here, we adapt this infrastructure as follows. We introduce an additional data structure on the postsynaptic side which is shared across threads ("process local"). This data structure contains, arranged by source neuron, the indices of all process-local synapses. While the presynaptic part of communicating spikes remains essentially identi-

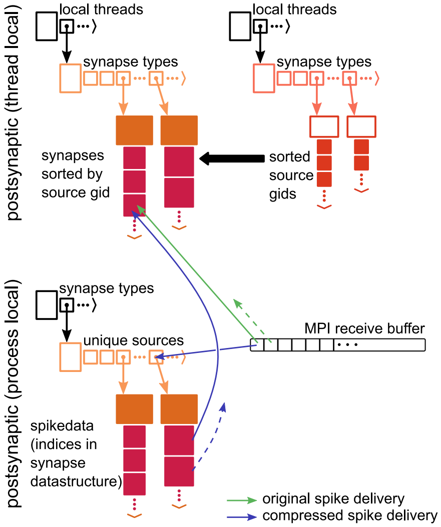


Figure 11.2: **Spike compression adds an additional indirection to postsynaptic spike routing.** Green arrow denotes original spike delivery introduced with the 5g kernel (Jordan et al., 2018, same display as their Figure 4A). Blue arrow illustrates additional indirection with compressed spike delivery. Dashed arrows indicate spikes from the same source neuron with target on a different thread.

cal, the postsynaptic part incurs an additional indirection: Each entry in the MPI receive buffer now represents an index in the new process-local postsynaptic data structure. Using this index, each thread can retrieve the indices of thread-local targets, to which it can then deliver spikes as previously (Figure 11.2, “compressed spike delivery”; note that the origin of the dashed arrow changes). In contrast to the previous implementation, each presynaptic neuron thus sends at most one spike to each process.

In `NEST 3.0`, spike compression is turned on by default, but the previous 5g behavior can be recovered by setting:

```
nest.SetKernelStatus({"use_compressed_spikes": False})
```

11.2.1.4 *Neuronal input buffers with multiple channels*

Simulation technology for spiking neuronal networks requires techniques to handle synaptic transmission delays. The reference simulation code (Section 11.3.2.2) follows a globally time-driven approach: spikes are constrained to a time grid and regularly exchanged between MPI processes using collective communication. The time grid defines the simulation time step for neuronal updates, whereas the minimum

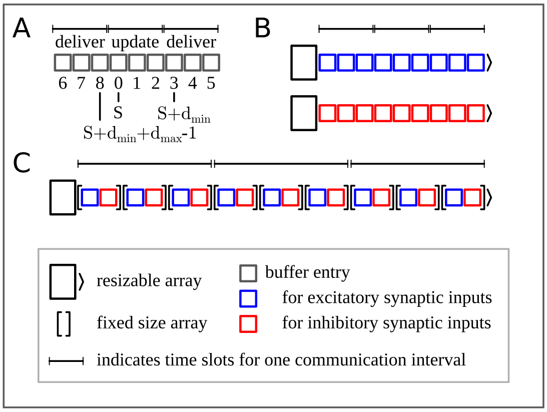

A
deliver update deliver
6 7 8 0 1 2 3 4 5
S
S+d_min
S+d_min+d_max-1
B
C
resizable array
fixed size array
buffer entry
for excitatory synaptic inputs
for inhibitory synaptic inputs
indicates time slots for one communication interval

Figure 11.3: **Neuronal input buffers accounting for synaptic delays in simulations of spiking neuronal networks. A** Structure of neuronal input buffers assuming a minimum synaptic delay d_{min} of three simulation time steps and a maximum delay $d_{\text{max}} = 2d_{\text{min}}$. To buffer upcoming inputs during simulation a total buffer size of $d_{\text{min}} + d_{\text{max}}$ time slots is required, which corresponds to three communication intervals of three simulation time steps each. After every spike communication and subsequent spike delivery to local targets, simulation time is advanced, meaning that the relative time origin S of the neuronal input buffers advances by d_{min} time slots with a wrap-around at the buffer end. A pre-calculated and continuously updated look-up table maps the index relative to S to the actual buffer index. Example: The relative time origin S is located at the fourth time slot. Synaptic delays of the inputs of the middle buffer segment elapse with the upcoming three simulation time steps; the neuron integrates these inputs updating its state. Spikes are then communicated and new inputs delivered to the neuron are added to the time slots in the last or first buffer segment depending on the delay, which is at least d_{min} and at most d_{max}. Relative time origin S then advances to the seventh buffer slot (not shown). **B** Original neuronal spike buffers for two input channels (e.g., excitatory and inhibitory synaptic inputs). For each channel a separate resizable array buffers the inputs for the upcoming time slots. **C** Multi-channel input buffer for two input channels. A single resizable array stores the inputs for the upcoming time slots, where for each time slot a fixed size array holds the inputs sorted by channel.

synaptic delay d_{min} in the network model defines the communication interval (Morrison et al., 2005b), which comprises at least one simulation time step. In the microcircuit model and the multi-area model used in this study the minimum delay is 0.1 ms (i.e., $d_{\text{min}} = 1$ simulation time step) and in the HPC-benchmark model it is 1.5 ms (i.e., $d_{\mathrm{min}} = 15$ simulation time steps). While communication and subsequent process-local delivery of spikes define interaction points between neurons, within a communication interval each neuron independently updates its state for all time steps without interruption. Hence, a simulation cycle of neuronal update, spike-communication, and spike-delivery phase propagates the network state by one communication interval, but within each update phase neurons propagate their state in potentially shorter simulation time steps. All spikes emitted by the process-local neurons during such an update are immediately transmitted during the subsequent communication and on the receiver side delivered to their target neurons. Hence, to account for synaptic delays, neurons cannot immediately integrate the incoming spikes into their dynamics, but they need to buffer the inputs until the corresponding delays elapse. To this end, neurons maintain input buffers of $d_{\mathrm{min}} + d_{\mathrm{max}}$ time slots, where d_{max} denotes the maximum synaptic delay in the network (Figure 11.3A). The relative time origin S defining the time slots from which to retrieve inputs during update and the time slots for adding inputs during spike delivery advances by d_{min} time slots at the end of every simulation cycle. In this way, the time slots that were read and reset during the update of the current cycle become available for adding new inputs during the spike delivery in the next cycle. For cases where the communication interval comprises multiple simulation time steps (e.g., HPC-benchmark model), input retrieval is most costly for the first step as the corresponding buffer entry needs to be loaded into cache, but then benefits from the already cached subsequent buffer entries in the subsequent steps of the communication interval. If, however, the communication interval consists of only one simulation step due to a very short minimal synaptic delay (e.g., microcircuit and multi-area model), input retrieval is costly for every simulation step as each step is handled in a separate simulation cycle, and hence caching of relevant input buffer entries is rendered ineffective during the spike communication and delivery that follows each neuronal update phase.

Most neuron models need to distinguish between input channels to treat the corresponding inputs dynamically differently, as for example, excitatory and inhibitory synaptic inputs causing different postsynaptic responses. The original input-buffer design required a separate resizable array per channel storing the channel's input values per time slot (Figure 11.3B). This entailed retrieval of the input values for a particular time step from separate locations in memory, which amplifies the cache inefficiency during update for network models with short minimum delays described above. To alleviate this issue, the newly introduced input buffer allows storing the input values for multiple channels per time slot contiguously in fixed size arrays in a single resizable array (Figure 11.3C). Thus, neurons now retrieve all input values for a particular time step by accessing subsequent locations in memory in one pass.

11.3 RESULTS

11.3.1 *Workflow Concepts*

Taking the analysis presented in Section 11.1 as a starting point, we develop a generic workflow for performance benchmarking, aiming at its applicability to simulations running on conventional HPC architectures. Figure 11.4 depicts a conceptual workflow that consists of four ordered segments each of which depends on the previous one. The segments are subdivided into different modules. These modules are related to specific realizations in our reference implementation of the workflow (see Section 11.3.2). Here, "workflow" refers to the concepts generally applicable to benchmarking while "framework" is used for the provided software implementation. Additionally, *internal* and *external* modules are distinguished. The former are essential component of the workflow while external the latter allow for a more rapid exchange. The following introduces each of the workflow's conceptual segments and expands on how the proposed solution addresses the identified problems (cf. Figure 11.1).

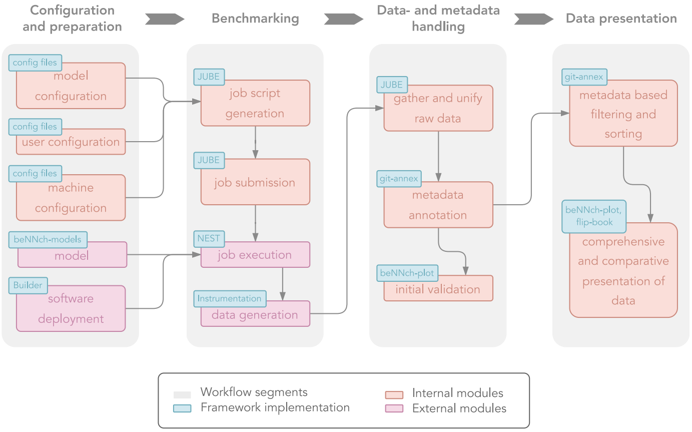


Figure 11.4: **Conceptual overview of the proposed benchmarking workflow.** Light gray boxes divide the workflow into four distinct segments, each consisting of multiple modules. Internal modules are shown in mint and external ones in cyan. Pink boxes indicate their respective realization in our reference implementation.

11.3.1.1 *Configuration and preparation*

The first four workflow segments comprise five distinct modules. Together, they provide the prerequisites for the simulations. First, "software deployment" takes care of the installation of the simulation software and its dependencies, while "machine configuration" provides parameters controlling the conditions of the simulation (e.g. the number of compute nodes that need to be reserved). Together, these two modules target the problem dimensions "hardware configuration", "software configuration", and "simulators". By providing the network implementation, the module "model" addresses "model and parameters". However, parameters of the used models remain unspecified. This is taken care of by "model configuration", including for example the biological model time to be simulated. As a consequence, the model is separated from its parameters. Finally, the "user configuration" module allows for the specification of user-specific data in a single location. Such data might include paths to store generated files or compute budgets.

11.3.1.2 *Benchmarking*

The second segment contains all modules relevant for running the benchmark simulation. On compute clusters workload is usually managed via queuing systems allowing for the definition of resource usage and specifying instructions needed for the simulation. In the workflow segment, this is handled by the module "job script generation". At this stage modules are linked for the first time: the workflow combines model, user and machine configuration to generate a job script for the simulation. Afterwards, this job script is submitted to the queuing system via the module "job submission". Since the deployment of the simulation software and the handling of the dependencies is already taken care of, "job execution" executes the simulation given the relevant parameters. While a simulation for neuroscientific research purposes would at this point focus on the output of the simulation (e.g. neuronal spike times or membrane potential traces), benchmarking is concerned with the performance results. These are recorded in the final benchmarking module called "data generation".

11.3.1.3 *Data- and metadata handling*

As already noted above, a major difficulty when conducting performance benchmarks is the handling the produced data and metadata. Here, data refers to the results of the performance measurements. Metadata, on the other hand, is used as an umbrella term for the description of the used settings according to the dimensions of benchmarking (Figure 11.1). Since executing multiple simulations using different configurations, software, hardware, and models is an integral part of benchmarking, data naturally accumulates. Thus, it is important to associate the metadata with the data. Standardized formats for both types of data make the results comparable for researchers working with the same underlying simulation technology. The workflow segment "Data- and metadata handling" proposes the following solution. First, the raw performance data, typically stemming from different units of the HPC system, are gathered and unified into a standardized format, while the corresponding metadata is automatically recorded. Next, the metadata is associated to the unified data files. This removes the need to manually keep track of parameters, experiment choices and software environment conditions. While there are different possible solution for this, attaching the relevant metadata directly to the performance-data files simplifies filtering and sorting of results. Finally, "initial validation" allows for a quick first assessment of the results. In this way, corrupted benchmarks can be swiftly identified.

11.3.1.4 *Data presentation*

The final workflow segment addresses the accessibility and comparability of benchmarking results. These are prerequisites for meaningful comparisons and consequently conclusions, also touching the complexity of "Researcher communication". In a first step, "metadata based filtering and sorting" allows dynamic choice of the results to be included in the comparison by the user. Here, dynamic means that arbitrary selection of axes in the high-dimensional metadata space can achieved so that the chosen results only differ in metadata fields of interest. Second, the data is presented in a format that allows for intuitive switching between benchmarks. This implies that key metadata is given alongside the results, and that the data representation is standardized. The presentation of data should be comprehensive, consistent, and comparative such that the benchmarking results are usable in the long term. Thereby, the risk of wasting resources through re-generation of results is reduced, which in turn allows for a more sustainable software development.

11.3.2 *beNNch: A reference implementation*

Building on the fundamental workflow concepts developed in Section 11.3.1, we introduce a reference implementation for modern computational neuroscience: `beNNch`[2]—a benchmarking framework for neuronal network simulations. Beyond being a mere proof-of-concept, the framework additionally provides a software tool useful for neuroscientists and simulator developers. In principle, `beNNch` is designed such that plug-ins for any neuronal network simulator can be added.

[2] https://github.com/INM-6/beNNch

Our implementation, however, is geared towards the NEST simulator (Gewaltig and Diesmann, 2007) designed for simulating large-scale spiking neuronal network models. In the following subsections, we detail software tools, templates, technologies, and user specifications needed to apply beNNch for benchmarking NEST simulations. Each of the conceptual modules of Figure 11.4 is here associated with a concrete reference.

11.3.2.1 *Builder*

Repeatability and comparability of benchmark simulations necessitates reproducible software deployment. On the other hand, usability of the framework dictates the abstraction from irrelevant information regarding the hardware architecture or software tool chain. The tool set is required to install software in a platform independent way and should not depend on a particular flavor of the operating system, the machine architecture or overly specific software dependencies. Additionally, it needs to be able to make use of system-provided tools and libraries, for example, to leverage machine specific MPI implementations. beNNch uses the tool Builder[3] for this purpose. Given a fixed software stack and hardware architecture, Builder provides identical executables by deriving the install instructions from so-called "plan files". By using the same environment module[4] system Builder achieves integration with other package management systems such as easy_build (Geimer et al., 2014) or Spack (Gamblin et al., 2015). This reduces the required user interaction, only configuration of installation parameters must be set. Given a specified variation of the software to be benchmarked, beNNch calls Builder to deploy the requested software. In doing so, Builder checks whether the software is already available and otherwise installs it according to the specifications in the plan file. The depth to which required dependencies need to be installed and which mechanisms are used depend on the conventions and prerequisites available at each execution site. For any installation, the used software stack—including library versions, compiler versions, compile flags, etc.—are recorded as metadata.

11.3.2.2 *NEST*

beNNch implements compatibility with the NEST simulator (Gewaltig and Diesmann, 2007), enabling the performance benchmarking of neuronal network simulations at the resolution of single-neurons. The NEST software is complex, and the consequences of code modifications for performance are often difficult to predict. NEST has an efficient C++ kernel, but network models and simulation experiments are defined via the user-friendly Python interface PyNEST (Eppler et al.,

[3] https://github.com/INM-6/Builder

[4] https://modules.readthedocs.io and http://lmod.readthedocs.io

2009; Zaytsev and Morrison, 2014). To parallelize simulations, NEST provides two methods: for distributed computing, NEST employs the Message Passing Interface (MPI, Message Passing Interface Forum, 2009), and for thread-parallel simulation, NEST uses OpenMP (OpenMP Architecture Review Board, 2008).

11.3.2.3 *Instrumentation*

The performance measurement mainly considered here is the time-to-solution. Acquiring accurate data on time consumption is of great importance for profiling and benchmarking. For this, two types of timers are employed to gather this data: Timers of the first type are built-in to NEST on the C++ level, or they are included on the Python level as part of the PyNEST network-model description. Timers of the second type are realized with explicit calls to the function `time.time()` of the Python Standard Library's `time`. Consistency throughout the framework is achieved by using standardized variables names for the various simulation phases. Typically, a NEST simulation consists of two stages, *network construction* and *state propagation*. In the former, neurons and auxiliary devices for stimulation and recording are created and subsequently connected according to the network-model description. In the latter, the network state is propagated in a globally time-driven manner. Additionally to the NEST timers already introduced in Chapter 10 (i.e. *update*, *deliver*, *communicate*), we here measure the time for the *collocation* of spike in MPI-communication buffers. NEST's built-in timers provide a detailed look into the contribution of all four phases of state propagation, while timers on the Python level measure network construction and state propagation.
In NEST, the postsynaptic connection infrastructure is established during the so called connection phase which is part of the network construction. However, the presynaptic counterpart is typically only set up at the beginning of the state propagation phase (see Jordan et al. (2018) for details). In this work, we trigger this step deliberately and include it in our measurement of network-construction time rather than state-propagation time. Moreover, a short pre-simulation before the actual start of the simulation allows the network to attain a dynamical state more reflective of the recurrent dynamics and less dependent on potential transients in the network activity due to initial conditions—such have decayed after this startup period. Thus, the state propagation phase is only recorded after the pre-simulation (Rhodes et al., 2019). The model time for pre-simulation can be configured via a parameter in `beNNch`.

11.3.2.4 *beNNch-models*

We instantiate the "model" module with the repository `beNNch-models`[5] which contains a collection of `PyNEST` neuronal network models, i.e., models that can be simulated using the Python interface of `NEST` (Eppler et al., 2009). In principle, any such model can be used in conjunction with `beNNch`; only a few adaptations are required concerning the interfacing. On the input side, the framework needs to be able to set relevant model parameters. For recording the performance data, the required Python timers (Section 11.3.2.3) must be incorporated. On the output side, the model description is required to include instructions to store the recorded performance data and metadata in a standardized format. Finally, if a network model is benchmarked with different `NEST` versions that require different syntax, as is the case for switching between `NEST 2.X` and `NEST 3.X`, the model description also needs to be adjusted accordingly. Which model version is used in a simulation can thereby be deduced from knowing which simulator version was tested. For fine-grained version tracking, we additionally record the commit hash of `beNNch-models` and attach it as metadata to the results. Instructions on how to adapt existing models are provided in the documentation of `beNNch-models`.
The current version of `beNNch` provides benchmark versions of three widely studied spiking neuronal network models: The two-population HPC-benchmark model[6], the microcircuit model[7] by Potjans and Diesmann, 2014 representing 1 mm^2 of cortical surface with realistic neuron and synapse densities, and the multi-area model[8] by Schmidt et al., 2018a,b consisting of 32 microcircuit-like interconnected networks representing different areas of visual cortex of macaque monkey. The model versions used for this study employ the required modifications described in Section 11.1.

11.3.2.5 *config files*

Executing benchmarks requires defining the characteristic parameters. This mainly defines the user interaction with `beNNch`. The definition of the parameters is separated from the executable code by providing `yaml`-based templates for "config files" that need to be customized by the user. Thereby, the information that defines a benchmark experiment is kept short and well arranged: the number of files a user has to touch is limited, the risk of user errors on the input side reduced. Additionally, this guarantees a concise definition of a given benchmark experiment and limits the number of files users needs to

[5] https://github.com/INM-6/beNNch-models
[6] original repository: https://github.com/nest/nest-simulator/blob/master/pynest/examples/hpc_benchmark.py
[7] original repository: https://github.com/nest/nest-simulator/tree/master/examples/nest/Potjans_2014
[8] original repository: https://github.com/INM-6/multi-area-model

interact with. Thus, the risk of user errors on the input side is reduced. Listing 1 presents an excerpt from such a config file which has distinct sections to specify model, machine, and software parameters. While some parameters are model specific, standardized variable names are defined for parameters that are shared between models.

Algorithm 1 Excerpt of a config file in `yaml`-format for setting model, machine, and software parameters for benchmarking the multi-area model. When giving a list (e.g., for `num_nodes`), a job for each entry of the list is created. **Model parameters**: `network_state` describes particular model choices that induce different dynamical fixed points; `model_time_sim` defines the total model simulation time in ms; `scaling_type` sets up the simulation for either a weak- or a strong-scaling experiment. The former scales the number of neurons linearly with the used resources which might be ill-defined for anatomically constrained models. **Machine parameters**: `num_nodes` defines the number of nodes over which the scaling experiment shall be performed; `tasks_per_node` and `threads_per_task` specify the number of MPI tasks per node and threads per MPI task respectively. **Software parameters**: `simulator` and `version` describe which version of which simulator to use (and to install if not yet available on the machine).

```
parameterset:

    - name: model_parameters
      parameter:
       # can be either "metastable" or "ground"
       - {name: network_state, type: string, _: "metastable"}
       # biological model time to be simulated in ms
       - {name: model_time_sim, type: float, _: "10000."}
       # "weak" or "strong" scaling
       - {name: scaling_type, _: "strong"}

    - name: machine_parameters
      parameter:
       # number of compute nodes
       - {name: num_nodes, type: int, _: "4,8,12,16,24,32"}
       # number of MPI tasks per node
       - {name: tasks_per_node, type: int, _: "8"}
       # number of OpenMP threads per task
       - {name: threads_per_task, type: int, _: "16"}

    - name: software_parameters
      parameter:
       # simulator used for executing benchmarks
       - {name: simulator, _: "nest-simulator"}
       # simulator version
       - {name: version, _: "3.0"}
```

11.3.2.6 *JUBE*

At this point, the first segment of the benchmarking workflow (Figure 11.4) is complete and hence all necessary requirements are set up: the software deployment provides the underlying simulator (here: NEST with built-in instrumentation), the models define the simulation, and the configuration specifies the benchmark parameters. This information is now processed by the core element of the framework: generating and submitting simulation jobs and gathering and unifying the obtained performance data. We construct this component of beNNch around the xml-based JUBE software tool using its yaml interface. Built around the idea of benchmarking, JUBE can fulfill the role of creating job scripts from the experiment, user and machine configuration, their subsequent submission, as well as gathering and unifying of the raw data output. Here, we focus on the prevalent scheduling software SLURM (Yoo et al., 2003), but extensions to allow for other workload managers would be straightforward to implement. Our approach aims at high code re-usability. Model specific code is kept to a minimum, and where necessary, written in a similar way across models. Adhering to a common interface between JUBE scripts and models facilitates the integration of new models, starting from existing ones as a reference. Since JUBE can execute arbitrary code, we use it to also record metadata in conjunction with each simulation. This includes specifications of the hardware architecture as well as parameters detailing the run and model configuration.

11.3.2.7 *git-annex*

Communication of benchmarking results poses a serious problems and thus requires a matured way of sharing data. Means of communication commonly employed in practice include attachments of emails, cloud-based storage options, or git repositories. It is not uncommon that all these means are used within one project be different, collaborating researchers. This complicates arriving at shared interpretation of the data, even more so when researchers from different labs are involved. Ideally, results would be stored in a decentralized fashion that allows for tracking the history of files while allowing on-demand access for collaborators. To this end, we use git-annex[9] as a versatile base technology; it synchronizes file information in a standard git repository while keeping the content of large files in a separate object store, thereby keeping the repository size at a minimum. git-annex is supported by the GIN platform[10] which we employ for organizing our benchmark results. In addition, it allows for metadata annotation: instead of relying on separate files that store the metadata, git-annex can directly attach them to the data files, thereby implementing the

[9] https://git-annex.branchable.com

[10] https://gin.g-node.org

"metadata annotation" module. Previously, as done for example in the work presented in Chapter 10, this needed to be cataloged by hand, whereas now the framework allows for an automatic annotation, reducing the workload on researchers and thus probability of human mistakes. A downside of following this approach is a limitation to command line-based interaction. Furthermore, `git-annex` is not supported by some of the more widely used `git` repository hosting services such as GitHub or GitLab in favor of `Git LFS`.
A difficult task when scaling up the usage of the framework and, by extension, handling large amounts of results, is providing an efficient way of dealing with user queries for specific benchmark results. This of course also applies to the framework developed here when scaling up the usage. Attaching the metadata directly to the performance data not only reduces the visible complexity of the repository, but also provides an efficient solution: `git-annex` implements a native way of selecting values for metadata keys via `git-annex` "views", automatically and flexibly reordering the results in folders and sub-folders accordingly. For example, consider the case of a user specifying the `NEST` version to be `3.0`, the `tasks_per_node` to be either 4 or 8, and the `network_state` to be either `metastable` or `ground`. First, `git-annex` filters out metadata keys for which only a single value is given; in our example, only benchmarks conducted with `NEST` version `3.0` remain. Second, a hierarchy of directories is constructed with a level for each metadata key for which multiple options are given. Here, the top level contains the folders "4" and "8", each containing sub-folders `metastable` and `ground` where the corresponding results reside. However, it may be difficult to judge exactly what metadata is important to collect; oftentimes, it is only visible in hindsight that certain metadata is relevant for the simulation performance. Therefore, recording as much metadata as possible would be ideal, allowing for retrospective investigations if certain metadata becomes relevant after run time. However, a balance needs to be attained between recording large amounts of metadata and keeping the volume of annotations manageable. In our implementation, this issue is resolved by recording detailed metadata about the system, software, and benchmarks. On the other hand, data from simulations is handled more parsimoniously: we only attach what we currently deem relevant to assess the performance. The remaining metadata is archived and stored alongside the data, thereby sacrificing ease of availability for a compact format. This way, if future studies discover that a certain hardware feature or software parameter is indeed relevant for performance, the information remains accessible also for previously simulated benchmarks while staying relatively hidden otherwise. This ensures accessibility of information if future studies discover that a certain hardware feature or software parameter is indeed relevant for performance—especially for previously conducted benchmarks. Furthermore, using `git` as a base technology

allows to collect data sets provided by different researchers in a curated fashion by using well established mechanisms like branches and merge-request reviews. This use of `git-annex` thereby implements the "metadata based filtering and sorting" module of Figure 11.4.

11.3.2.8 *beNNch-plot*

The comparison of plots of displaying benchmarking results along the dimensions illustrated in Figure 11.1 requires a unified plotting style. For this, an independent plotting package named `beNNch-plot`[11]) was developed using `matplotlib` (Hunter, 2007) as an backend. In this package we define a set of tools to create individual plot styles that can be combined by the user in a flexible manner. The standardized definitions of performance measures employed by `beNNch` directly plug into this package. In addition, `beNNch-plot` includes default plot styles that can be readily used, and provides a platform for creating and sharing new ones. `beNNch` utilizes the default plot styles of `beNNch-plot` for both initial validation—a preliminary plot offering a quick glance at the results, thereby enabling a swift judgement whether any problems occurred during simulation—and visualization of the final results.

11.3.2.9 *flip-book*

When devising a method of presenting benchmark results we found the following aspects to be of crucial relevance for our purposes. First, it should be possible to navigate the results such that plots are always at the same screen position and have the same dimensions, thereby minimizing the effort to visually compare results. To achieve such a format, we decided to create a flip-book in which each slide presents the results of one experiment. Second, relevant metadata should be displayed right next to the plots. This can include similarities across the runs, but more importantly should highlight the differences. As each user might be interested in different comparisons, we let the user decide which kind of metadata should be shown. Third, it should be easy to select only the benchmarks of interest in order to keep the number of plots small. This is already handled by the filtering provided by `git-annex` views as described in Section 11.3.2.7. As an underlying technology for programmatically creating `HTML` slides we use `jupyter notebooks`[12] in conjunction with the open source `HTML` presentation framework `reveal.js`[13]. An exemplary flip-book containing the `NEST` performance results described in this work is published alongside the `beNNch` repository[14]. By respecting these considerations,

[11] https://github.com/INM-6/beNNch-plot
[12] https://jupyter.org
[13] https://github.com/hakimel/reveal.js
[14] https://inm-6.github.io/beNNch

our proposed solution offers a way of sharing benchmarking insights between researchers that is both scalable and flexible.

11.3.2.10 *Exchanging external modules*

By its constructions, beNNch is inherently modular. This means that it is possible to exchange certain modules without compromising the functionality of the framework. In particular, the "external modules" (see Figure 11.4) are implemented such that an exchange is straightforward to implement. In the following we briefly present a step-by-step guide to exchange the "job execution" module, i.e., the simulator, along with necessary changes in "data generation" and "model" that follow:
First, the simulator the performance of which one wants to assess must be deployed. Builder—our implementation of the "software deployment" module—offers the flexibility to install any software as well as make it loadable via a module system. Consequently, a plan file specifying dependencies as well as source code location and installation flags needs to be created for the new simulator.
Second, appropriate models that allow for the performance assessment of the simulator need to be added. Regarding the framework one needs to adapt the execute commands. Required additions to the models are the same as for PyNEST models and are described in Section 11.3.2.4.
Third, the instrumentation recording the performance metrics needs to be added. As already noted above, the NEST simulator offers already built-in instrumentation, so that only some timing measurements must be conducted on the model level. The extent to which alterations have to be introduced depends on the existing ability of the new simulator. If different measurements than implemented are of interest, a simple addition to an existing list in beNNch suffices to add the recorded data to the csv-format result file.

11.3.3 *Using beNNch for Simulator Development*

The continuous improvement of the performance of simulation software requires the identification of those parts of the simulation loop that take up the most time. Benchmarking the performance can aid the detection of performance bottleneck. Additionally, it can highlight changes in performance after algorithmic alterations. The dimensions of benchmarking described Figure 11.1, however, highlight the difficulty: to guarantee that observed differences in performance are caused by changes in the simulator code, many variables need to be controlled for, such as hardware and software configurations as well as simulator versions. General-purpose simulators also need to be tested with respect to different settings and applications to ensure that a performance improvement in one case does not lead to a crucial decline in another case. Neuronal network simulators are one such

example as they should exhibit reasonable performance for a variety of different models with different resource demands. A systematic assessment of the scaling performance covering the relevant scenarios is therefore a substantial component of the iterative process of simulator development.
beNNch, as an implementation of the workflow outlined in Section 11.3.1, provides a platform to handle the complexity of benchmarking while staying configurable on a low level. In the following we suggest how beNNch can support the process of detecting and tackling performance issues of a simulator. First, the behavior of the simulator needs to be explored in various situations to identify performance bottleneck the current version. Here, the shared location of configuration parameters used by beNNch allows for keeping an overview over the performed experiment. Neuronal network simulations can usually be decomposed into separate stages, such as neuronal update and spike communication. For this beNNch stresses the importance of appropriate instrumentation and visualization pointing the researcher to the respective sections in the code. If a potential bottleneck for a certain model is identified, tests with other models provide the basis for deciding whether these are model- and scale-specific or are present across models, hinting at long-reaching issues of the simulator. beNNch's native support for handling the benchmarking of multiple models relieves the researchers of operating a different code base for every model. While improving the simulation engine, continuously performed benchmarks allow for a direct comparison between the current version and the putative improvements. The standardized visualization tools of beNNch support spotting differences in performance plots. In this way, adaptions that bear fruit can be swiftly identified. Finally, an ongoing development of a neuronal network simulator should respect the value of insights gained by resource-intensive benchmarks. The decentralized storage of standardized results employed by beNNch addresses this issue. In addition to preserving information for the long term, this also helps in communicating between researchers working on the simulator's development.

11.3.3.1 *Use case: NEST development*

This section illustrates the relevance of performance benchmarks for the development of neuronal network simulators with the example of recent changes to the NEST code base; for historical context see Section 11.2.1.1. We use beNNch to outline crucial steps of the development from the release candidate NEST 3.0rc to the final NEST 3.0 and also discuss improvements compared to the latest NEST 2 version (Fardet et al., 2021). Table 11.1 summarizes the NEST versions employed in this study. Regarding the dimensions of HPC benchmarking in Figure 11.1, this use case primarily addresses the "Simulators" dimension by testing different NEST versions and the "Models and parameters"

Shorthand notation of NEST version	Description
2.20.2	Official 2.20.2 release (Fardet et al., 2021)
3.0rc	Release candidate for 3.0
3.0rc+ShrinkBuff	3.0rc plus shrinking MPI buffers
3.0rc+ShrinkBuff+SpikeComp	3.0rc+ShrinkBuff plus spike compression
3.0	Official 3.0 release (Hahne et al., 2021) = all of the above plus neuronal input buffers with multiple channels

Table 11.1: Shorthand notation and description of NEST versions used in this work.

dimension by testing different network models; the approach can be extended similarly to the other dimensions. Our starting point is the weak-scaling experiments of the HPC-benchmark model (Jordan et al., 2018); the times for network construction and state propagation as well as the memory usage remain almost constant with the newly introduced 5g kernel (see their Figures 7 and 8). Figure 11.5 shows similar benchmarks of the same network model conducted with beNNch using the release candidate in Figure 11.5A and the final release in Figure 11.5B. The graph design used here corresponds to the one used in the flip-book format by the framework. For the realises candidate the state-propagation time increases with the number of nodes while the network construction times remain constant Figure 11.5A. In comparison to the with the former, the latter makes up less than 10% of the total simulation time for $T_{\mathrm{model}} = 1\,\mathrm{s}$. We additionally note that

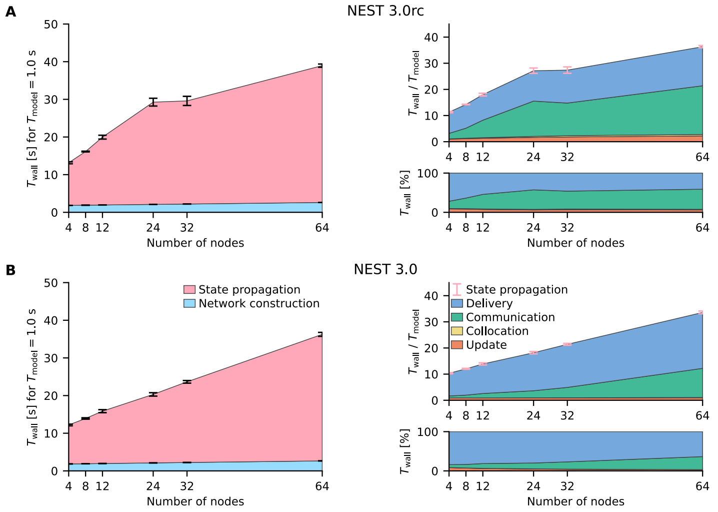


Figure 11.5: **Weak-scaling performance of the HPC-benchmark model on JURECA-DC. A** `NEST 3.0rc`. The left graph shows the absolute wall-clock time T_{wall} measured with Python-level timers for both network construction and state propagation (legend in panel **B**); the model time is $T_{\mathrm{model}} = 1\,\mathrm{s}$. Error bars indicate variability across three simulation repeats with different random seeds. The top right graph displays the real-time factor defined as wall-clock time normalized by the model time. Built-in timers resolve four different phases of the state propagation (legend in panel **B**): update, collocation, communication, and delivery. Pink error bars show the same variability of state propagation as the left graph. The lower right graph shows the relative contribution of these phases to the state-propagation time. **B** `NEST 3.0`. Same display as panel A.

the phases delivery and communication both contribute significantly to the state-propagation time. Jordan et al. (2018) report real-time factors of about 500 (e.g., their Figure 7C) in contrast to values smaller than 40 shown here and their simulations are by far dominated by the delivery phase (see their Figure 12). A comparison of our data and the data of Jordan et al., 2018 is not straightforward due to the inherent complexity of benchmarking and we here emphasize a few concurring aspects: First, Jordan et al., 2018 run their benchmarks on the dedicated supercomputers JUQUEEN (Jülich Supercomputing Centre, 2015) and K Computer (Miyazaki et al., 2012) while our benchmarks use the recent cluster JURECA-DC (Thörnig and von St. Vieth, 2021). Our data are, however, not directly comparable to the results of Jordan et al. (2018). This stems directly the aforementioned complexity of benchmarking, here, for example, differences in the used supercomputers: JUQUEEN (Jülich Supercomputing Centre, 2015) and K Computer (Miyazaki et al., 2012) in the case of Jordan et al. (2018), the recent cluster JURECA-DC (Thörnig and von St. Vieth, 2021) used here. Each compute node of the BlueGene/Q system JUQUEEN is equipped with a 16-core IBM PowerPC A2 processor running at 1.6 GHz and each node of the K Computer has an 8-core SPARC64 VIIIfx processor operating at 2 GHz; both systems provide 16 GB RAM per node. In contrast, the JURECA-DC cluster employs compute nodes consisting of two sockets, each housing a 64-core AMD EPYC Rome 7742 processors clocked at 2.2 GHz, that are equipped with 512 GB of DDR4 RAM. Here, nodes are connected via an InfiniBand HDR100/HDR network. Moreover, Jordan et al., 2018 use 1 MPI process per node and 8 threads per process while for our simulations we use 8 MPI processes per node and 16 threads per process. Third, Jordan et al., 2018 simulate 18, 000 neurons per MPI process while we only simulate 11, 250 neurons per process. This list of differences is not complete and only aims to illustrate that potential discrepancies in benchmarking results may be explained by differences in hardware, software, simulation and model configuration, and other aspects exemplified in Figure 11.1. After having conducted weak-scaling experiments of the HPC-benchmark model, next the performance of the simulator is assessed in strong-scaling benchmarks of the multi-area model (Schmidt et al., 2018b). At least three compute nodes of JURECA-DC are needed to satisfy the memory requirements of the model; here, we choose to demonstrate the scaling on four to 32 nodes. Initially, we compare the latest `NEST 2` version (Figure 11.6A) with the release candidate for `NEST 3.0` (Figure 11.6B). The improved parameter handling implemented in `NEST 3.0rc` reduces the network-construction time. In both `NEST` versions the communication phase by far dominates the state propagation. Previous simulations of the HPC-benchmark model have not identified the communication phase as a bottleneck Jordan et al., 2018, Figure 12. Only when the smallest delay in the network is close to the computation time step communication becomes an issue This is due to `NEST` using the smallest delay as the communication interval for MPI. In comparison to the HPC-benchmark model, using a 1.5 ms delay for all connections (a good estimate for inter-area communication), the multi-area and microcircuit model use distributed delays with a lower bound of 0.1 ms. This leads to a fifteen-fold increase in the MPI communication steps in the latter two models.

Next we introduce the first putative performance improvement to be applied to `NEST 3.0rc`. Cross-node communication, handled in `NEST` by MPI, needs to strike a balance between the amount of messages to transfer and the size of each message. The size of the MPI buffer limits the amount of data that fits into a single message, and is therefore the main parameter controlling this balance. In an ideal scenario, each buffer fits exactly the right amount of information by storing

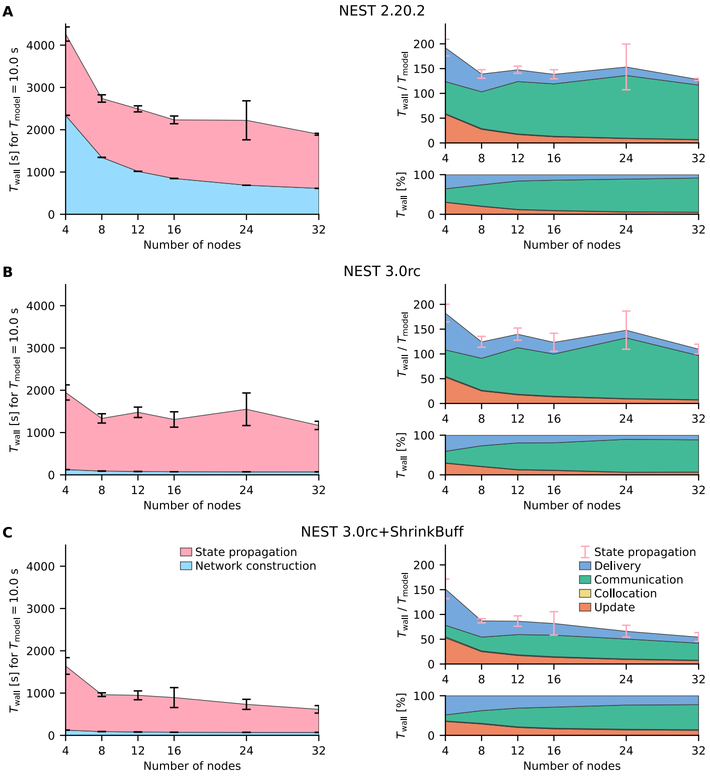


Figure 11.6: **Strong-scaling performance of the multi-area model on JURECA-DC.** Same display as in Figure 11.5. The multi-area model is simulated in its meta-stable state leading to a high amount of spikes that are communicated. The model time is $T_{\text{model}} = 10\,\text{s}$. Simulations are repeated for ten different random seeds. **A** `NEST 2.20.2` (latest `NEST 2` release). **B** `NEST 3.0` release candidate. **C** `NEST 3.0` release candidate with shrinking MPI buffers.

all spikes of the process relevant for the respective communication step. However, a scheme in which the buffer size adapts precisely for each MPI process for each communication step can be highly inefficient due to overhead attached to operating on additional vectors. In `NEST 3.0rc` it is assumed that in the usual case communications relatively homogeneous. In such a scenario, it is advantageous to fix the size of the exchanged buffers between all processes. While buffer sizes are constant across processes, `NEST` does adapt them over time to minimize the number of MPI communications. Concretely, whenever the spike information that a process needs to send exceeds what fits into one buffer, the buffer size for the next communication step is increased. However, the original 5g kernel of NEST does not shrink buffer sizes. In networks such as the multi-area model, the firing is not stationary over time; transients of high activity propagate through the network (Schmidt et al., 2018b). In general, startup transients may cause high spike rates only in the beginning of a simulation unless the network is carefully initialized (Rhodes et al., 2019). If the rates decrease, the spiking information becomes much smaller than the available space in the MPI buffer. Consequently, the original 5g kernel preserves unnecessarily large buffer sizes which results in the communication of useless data. To address this issue, a mechanism for automatically shrinking the buffer sizes has been introduced. For details see Section 11.2.1.2. The release candidate with the implementation of shrinking MPI buffers (`NEST 3.0rc+ShrinkBuff`) approximately halves the time spent in the communication phase compared to the original implementation (compare Figure 11.6C, Figure 11.6B). Next, we employ the microcircuit model (Potjans and Diesmann, 2014) to asses the strong-scaling performance of the NEST simulator (c.f. Chapter 10). The model size is similar to the size of one of the 32 areas of the multi-area model, and thus simulation of this model require less resources. Using `NEST 3.0rc`, the microcircuit is simulated faster than the HPC-benchmark and the multi-area models and achieves approximately real time ($T_{\mathrm{wall}}/T_{\mathrm{model}} \approx 1$, Figure 11.7A). In contrast, the sub-real time performance on a single node presented in Chapter 10 was achieved with `NEST 2.14.1` (hardware-wise the compute nodes are comparable). The finer resolution of the vertical axis of the top-right graph reveals a small gap between the state propagation measured with Python timers and the sum of the phases timed on the C++ level which is not visible for the other models. The state-propagation time of the microcircuit is also dominated by the communication phase similarly to the respective benchmarks with the multi-area model (Figure 11.6B) and even increases with the number of nodes used. However, shrinking MPI buffers does not reduce communication significantly (data not shown), indicating that we face a different bottleneck with the microcircuit model. With on the order of 10^3 outgoing connections per neuron, a single-neuron of this model has multiple targets on each MPI process and, in particular, on multiple threads of a given process. Since the 5g kernel is designed to send out a separate copy of a neuron's spiking information to each target thread, multiple copies of identical information about the activity of a presynaptic neuron may be sent to the same process, causing unnecessary communication load. To tackle this, we devise a spike compression algorithm which only requires transmitting the spiking information once to each MPI process where it is locally routed to the receiving threads. For details see Section 11.2.1.3. This algorithm leads to a significant reduction in communication time for the microcircuit

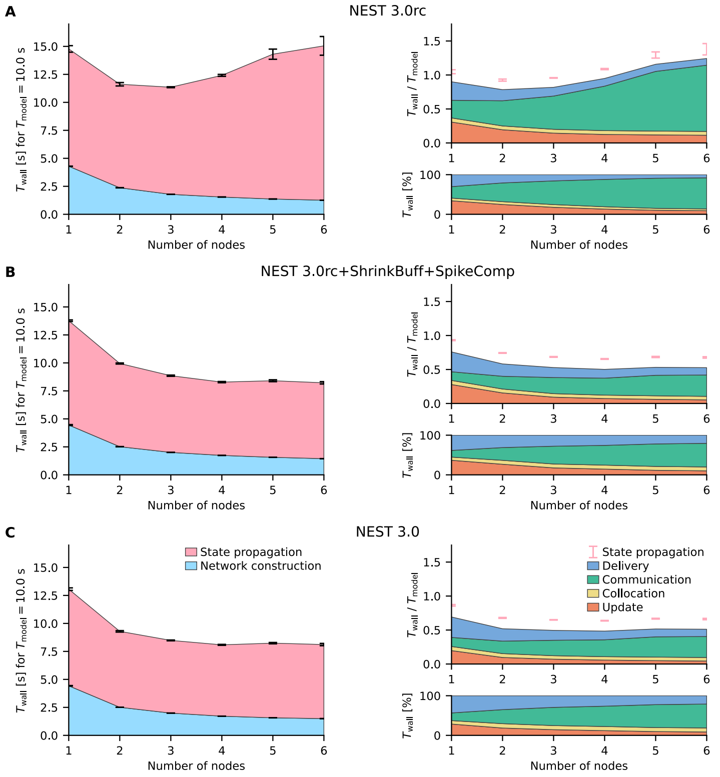


Figure 11.7: **Strong-scaling performance of the microcircuit model on JURECA-DC**. Same display as in Figure 11.5. The model time is $T_{\mathrm{model}} = 10\,\mathrm{s}$. Simulations are repeated for ten different random seeds. **A** `NEST 3.0` release candidate. **B** `NEST 3.0` release candidate with spike compression and shrinking MPI buffers. **C** `NEST 3.0`.

model (compare Figure 11.7A, Figure 11.7B). Yet, the real-time factor 0.52 measured on two nodes in Chapter 10 is still not achieved with the suggested improvement.

The microcircuit model easily fits within the main memory of one compute node of JURECA-DC. Due to the simplicity of the employed model neurons and the absence of synaptic plasticity mechanisms, the network model causes little workload during update and delivery in a strong-scaling experiment—real-time simulation is already possible with a single compute node. Consequently, communication naturally starts to dominate the state-propagation time at a few compute nodes even with the spike-compression optimization described above. While increasing the number of compute nodes from one to two still results in a fair reduction of state-propagation time, scaling is already sublinear, and increasing the number of compute nodes further hardly results in further improvement. Therefore, simulation phases other than the so far discussed communication become important if the objective of the optimizations is, for example, achieving real-time performance with even fewer resources. In the following we highlight an algorithm adaptation that concentrates on the update phase. A redesign of the neuronal input buffers prevents neurons from retrieving the input values for different channels, for example, excitatory and inhibitory, from separate locations in memory. Thereby, the cache can be better utilized during neuronal updates. Instead of maintaining separate buffers for the input channels as in the original 5g kernel, neurons maintain a single buffer with all inputs for a particular simulation time step stored contiguously in memory. For details see Section 11.2.1.4. This adaptation is most effective for network models with short minimum synaptic delays; both the microcircuit and the multi-area model use 0.1 ms. Figure 11.7C shows the resulting decrease in update time for few compute nodes.

Taken together, the analysis performed using `beNNch` reveals that for the release candidate `NEST 3.0rc` the communication phase as a major performance bottleneck in microcircuit and multi-area model simulations. This similarity is, however, superficial, the underlying causes and their mitigations are different: while the shrinking MPI buffers (Section 11.2.1.2) increase the simulation speed of the multi-area model, the spike compression mechanism (Section 11.2.1.3) improves the performance of the microcircuit model. None of the adaptations introduce a performance regressions for the respective other model. In addition, the update phase is improved by introducing neuronal input buffers with multiple channels (Section 11.2.1.4). Also for the HPC-benchmark model, the kernel adaptations are not detrimental to the originally tested model, see Figure 11.5B). Indeed, the overall state-propagation time is preserved with the final `NEST 3.0` release. Reducing communication and update times comes here, however, at the cost of increased delivery periods. This is due to an additional indirection introduced with spike compression. Ongoing work targets the delivery phase (Pronold et al., 2021) and gives a perspective for performance improvements in future `NEST` releases.

11.4 DISCUSSION

Benchmarking HPC simulation technologies is a surprisingly difficult endeavor. Here, we proposed unified and modular workflow for defining, running and analyzing benchmark simulation based on five distinct dimensions. The five dimensions were identified together with the causes of their corresponding difficulties, namely hardware configuration, software configuration, simulators, models and parameters, as well as researcher communication. To mitigate the arising problems we developed reference implementation of the workflow named BeNNch. There, the issues stemming from the aforementioned dimension are addressed in systematic way. To highlight the practical relevance of the developed framework, we employed it to assess the performance of different versions and modifications of the NEST simulator. The reference implementation goes beyond existing benchmarking environment software such as JUBE: It adds an interface to models, installs and deploys simulation software, automates data and metadata annotation, and implements storage and presentation of results. The framework is provided as a tool to the community not only for the development of simulation technology, but can also be used for finding optimal performance configurations in other models. While the devised workflow is in principle generic, the reference implementation is tailored to the NEST simulator and still faces limitations and open problems:

A priori, it is unclear what parameters, configurations or external influences may possibly contribute to differences in the performance of complex software systems such as simulation engines. beNNch addresses this problem by employing a metadata archive which—in addition to the selection of metadata directly attached to the performance results—tracks further metadata that are seemingly insignificant at the time of simulation but may become relevant in future investigations. However, completeness of all possible metadata is not achieved.

The high-level language used for network model specification —Python for NEST when the models are expressed in PyNEST —require modifications to seamlessly work with the benchmarking framework. At the moment, this is a manual task requiring continuous efforts in keeping the benchmarking models synchronized with the original models. We use rigorous version control of the code, automatic checking for errors (via exceptions), and continuous testing for correct simulation outcome to reduce the risk of errors. This strategy could be automatized further in the future by finding methods to automatically inject respective instrumentation into the executable model descriptions. To mitigate the additional overhead, we keep the necessary changes as minimal as possible, thereby lowering the entry barrier for new models.

The reference implementation makes concrete choices on the employed tools. Alternatives, however, may be viable. For example, the required software for the simulations is installed with Builder which can be integrated with other package management systems or replaced by a different solution. Our strategy exploits the native software environment available on a compute cluster which is typically specifically configured for the underlying hardware. An alternative is to use containerized systems such as Docker[15] or Singularity[16]. Replacing NEST by a different simulator requires adapting the model implementations. Expressing the models in the simulator-independent language PyNN (Davison et al., 2009) instead of PyNEST would avoid this. However, additional layers of complexity such as PyNN may have an impact on performance, making it more challenging to pinpoint bottlenecks in the simulator backend. JUBE as an environment to manage jobs on compute clusters could be substituted by tools such as ecFlow[17], AiiDA[18] Further, one could replace git-annex with, e.g., DataLad[19] which is based on the same technology but extends its functionality and provides slightly different metadata handling. The flip-book-style presentation of results could also be replaced or supplemented with other approaches, for example an automatically generated overview figure showing results from multiple benchmarking runs together, similar to Figures 11.5–11.7 in this paper.

In its current form, beNNch focuses on a single performance measure: the time-to-solution. Different performance metrics potentially of interest include energy-to-solution (see Chapter 10) as well as memory consumption. Memory consumption determines for example the smallest system that can simulate a given benchmarking model. Reducing memory requirements was a major driving force behind the improvements to the NEST kernel (Helias et al., 2012; Jordan et al., 2018; Kunkel et al., 2012, 2014) in the past decade. The spike compression introduced here reduces the time-to-solution (communication phase, Figures 11.5 and 11.7). However, this code also changes the memory consumption. Assuming that the number of postsynaptic targets per neuron is fixed, the memory overhead is negligible if the number of MPI processes is small. But in the limit of a large number of MPI processes, i.e., when each neuron has at most one target on each process, the effective size of each synapse is increased by 8 byte. In this limit, users thus are encouraged to actively deactivate the "spike compression" feature.

Continuous development is an integral part for long term sustainability. To facilitate this, beNNch is developed as an open source software project. This allows to make use of a public issue tracker, suggestions via pull requests, public code reviews, and detailed documentation. The chosen approach allows for constructive exchange between the users and developers ultimately allowing for enhancements actually needed by the users. While the concrete application of NEST benchmarks of neuronal network models shaped our specific implementations, the modular structure allows for adaptation to other use cases. In certain domains of software development, it is already common practice to verify each code change on the basis of syntax, results, and

[15] https://www.docker.com
[16] https://sylabs.io
[17] https://confluence.ecmwf.int/display/ECFLOW
[18] https://www.aiida.net
[19] https://www.datalad.org

other unit tests. The proposed automated approach to execute performance benchmarks creates the opportunity to integrate an aspect of validation directly into the development cycle. This way, performance regressions of algorithm adaptations are immediately exposed, while positive effects can readily be demonstrated. For high-performance software, however, comprehensive checks for scaling performance are particularly costly because they require compute time on state-of-the-art clusters and supercomputers. Therefore, it is important to conduct the performance benchmarks purposefully and with care. By organizing benchmarking results and keeping track of metadata, `beNNch` helps to avoid redundant benchmark repeats and instead encourages a direct comparison with previous results.

As already noted at the beginning of Part iv, reliable, efficient and continuously development software is of great importance for simulation sciences. Albeit all the remaining difficulties, the proposed framework enables HPC benchmarking of research software to evolve from one-off tasks of individual researchers to a collaborative routine effort, thereby increasing the benchmarking capacity and reducing its susceptibility to errors. Thus, `beNNch` facilitates the continuous development of HPC simulation technology and puts it onto a solid foundation.

Part V

DISCUSSION

12 DISCUSSION AND OUTLOOK

In this thesis, digital twins were used as a tool for neuroscientific research. As such, digital twins help integrating the large body of neuroscientific findings into coherent platforms. Additionally, they allow for investigating neural mechanism with *in silico* experiments where all observables (given a certain level of description) can be recorded and everything can be altered—the physiologist's dream. These platforms should be designed so that they can be extended in a cumulative way leading to improvements and refinements of models over time. In this way, digital twins can be used by different researchers over multiple scientific projects. To fully harness their potential, the construction and use of digital twins needs to be embedded into a wider neuroscientific research program. Such a program needs to go beyond the mere representation of the "physical twin" in a computer model. As already stated in Chapter 2, we here suggest to at least combine three perspectives: identification of neural phenomena that lack quantitative explanation by an underlying mechanism implemented in the nervous system, construction of appropriately designed computer models that allow to investigate the phenomena under consideration, and the continuous development and improvement of technologies that enable efficient instantiations and simulations of the constructed digital twins. In this thesis, all of these perspectives were treated.
This chapter summarizes the results, relates it to other work and suggests potential future lines of research.

ANALYSIS OF NEURAL DATA In Part ii, we turned to the first perspective of the formulated program. There, the resting-state activity of macaque primary visual cortex was investigated (Figure 4.1). We observed two distinct high-dimensional clusters—also referred to as neural manifolds—in the neural activity (Figure 4.2). These clusters were found to be correlated with the eye-closure and were shown to differ in their dimensionality (Figure 4.5). Our findings agree with analysis of the dimensionality of visual cortical activity (Gao and Ganguli, 2015; Stringer et al., 2019a). A spectral Granger causality analysis of the local field potential revealed a top-down modulation from V4 to V1 in the β-band during eyes-open periods (Figure 4.6). This agrees with the findings of Bastos et al. (2015) and Vezoli et al. (2021) who also associated top-down inputs with spectral Granger causality in the β-band. Additionally, we confirmed previous work by Semedo et al. (2022) regarding top-down modulation in the absence of visual stimulation. So far, only theoretical studies investigated to what extent neural manifolds could be modulated by top-down inputs (Naumann et al., 2022). Our findings suggest for the first time that this also may happen *in vivo* (Figure 4.8). However, given the experimental data at hand, this hypothesis could neither be confirmed nor disproved. Future work could pay greater attention to the transition between manifolds at high temporal resolution. This might help to further investigate the role of the top-down pathway in the change in neural manifold.

CONSTRUCTION AND SIMULATION OF SPIKING NEURAL NETWORK MODELS In Part iii, we turned to the construction and simulation of spiking neural network models of cortical tissue. First, we aimed at validating the hypothesis proposed in Chapter 4, i.e. that top-down modulations can induce a separation of neural manifolds. To this end, in Chapter 6 we modified a standard model of local cortical circuits (Brunel, 2000) and stimulated the network with inputs mimicking top-down modulations. We found that such additional inputs could indeed explain the emergence of two separate neural manifolds in the spiking activity (Figure 6.1). This is in agreement with the experimental data, see Figure 4.4. A more detailed analysis revealed that the split is due to a change in the most active neurons during the stimulated periods in comparison to the non-stimulated state. However, the change in dimensionality also observed in the spiking neural activity cannot be reproduced with the spiking network model employed here. This is due to the uniform recurrent connectivity and synaptic weights in the network studied here (Tetzlaff et al., 2012). Introducing an additional structure in the connectivity akin to Rostami et al. (2022) might help to overcome this problem. Other mechanisms that potentially contribute to or even control decorrelation of the network activity (and thus an increase in dimensionality) in the presence of top-down modulations include N-methyl-D-aspartate (NMDA) spikes (Herrero et al., 2013; Major et al., 2013; Schiller et al., 2000). This seems particularly likely since NMDA receptors have been suggested to control the efficacy of top-down interactions from V4 to V1 (Self et al., 2013; Van Loon et al., 2016). However, to settle this question further experimental and computational studies are needed.

The model presented in Chapter 6 is only plausible if the simulated network represents rather local cortical circuits. While suitable to address the separation of neural manifolds in the resting-state, it is not a biologically plausible model of the primary visual cortex in a more general sense. To construct digital twins more faithful to the fine structural details of the recurrent connectivity in this area, the laminar as well as horizontal organization of cortex needs to be taken into account. In Chapter 7, we constructed such a biologically plausible, large-scale spiking neural network model of macaque V1. For this, we devised an analysis workflow that allowed for a systematic reconciliation of anatomical findings across multiple studies. We integrated a large body of anatomical data (Binzegger et al., 2004; Vanni et al., 2020) and combined connectivity across multiple length scales (Angelucci et al., 2002; Perin et al., 2011; Sincich and Blasdel, 2001) (see Figure 7.2, Figure 7.4, Figure 7.5). Instantiating and simulating the model resulted in physiologically implausibly firing rates of the neurons in the form of run-away activity (Figure 7.6). A closer examination revealed that the derived connectivity exhibits excitatory feedback loop leading to the observed diverging firing rates (Figure 7.7). The digital twin constructed here, while integrating realistic number of neurons and synapses with plausible connectivity data across multiple length scales, does not allow for a reconciliation of cortical architecture and realistic activity.

This observation motivated a more fundamental analysis of the local cortical connectivity in Chapter 8. There, we analyzed two recent large-scale reconstructions of the local cortical circuits obtained via electron microscopy (MICrONS Consortium et al., 2021; Shapson-Coe et al., 2021). The data are the most complete estimates of local connectivity to date. From this, we derived new connectivity maps and compared them with an identically constructed model based on the older data by Binzegger et al. (2004), which were also the basis for the derivations in Chapter 7. While general features are conserved (see Figure 8.1), there are prominent differences in the target specificity of especially the excitatory neural populations (Figure 8.2). The results suggest a fundamental bias in the older data, which is prone to excitatory feedback loops. The more recent data suggests strong innervation of inhibitory populations by excitatory neurons. Similar innervations from inhibitory to excitatory populations have already been introduced by Potjans and Diesmann (2014) on the grounds of electrophysiological connectivity data. We here showed that the modifications introduced in this study, which altered the recurrent connectivity in order to obtain a dynamically more stable network, are indeed consistent with the local anatomy of cortical circuits.

Embedding these new findings into the model of macaque primary visual cortex is the next goal: the aforementioned workflow for processing and combining anatomical data to derive a network model allows for straightforward extensions integrating the new insights regarding the local cortical connectivity. After achieving physiologically plausible spiking activity, the model can be used to investigate cortical dynamics in macaque V1. For this, a suitable model for providing visual input to the network has to be devised. Here, one can build on the large body of modeling approaches for the retina and the lateral geniculate nucleus (Li, 2014). Next, the model needs to be validated against experimental findings. For the primary visual cortex, Antolík et al. (2018) suggests a list of functional properties that can serve as a start for such validations. Additionally, extracellular quantities that can be readily obtained in standard electrophysiological experiments can be used as benchmarks for the model activity. Senk et al. (2023) show this for a spatially organized spiking neural network model using the local field potential. There, the prediction is generated with a hybrid scheme separating the spiking activity and the generation of the extracellular potentials (Hagen et al., 2016). This method is computationally costly and cannot be scaled easily to larger models. More recent approaches based on linear predictors, however, promise a swifter and cheaper estimation (Hagen et al., 2022).

Ultimately, we aim for using the constructed digital twin to investigate cortical dynamics in a precision currently not accessible by experimental methods. Restrictions are imposed by technological limitations and ethical considerations in the treatment of laboratory animals. Here, especially the role of patchy connections for cortical dynamics (Chavane et al., 2011) or for contextual modulations comes to mind (Liang et al., 2017). The presented model of macaque primary visual cortex might help bridging complex functional phenomena as for example described in the more abstract modeling work by Li (1999a,b) with fundamental physiological and anatomical constraints.

Finally, the model can be extended to include multiple cortical areas of the visual hierarchy. Here, canonical choices for future additions would be the areas V2 and V4. V1, V2 and V4 are the first three stages of the ventral stream classically associated with object identification (Kandel et al., 2013). To incorporate the suggested multi-stage functional integration of information in the visual cortex (Zeki and Shipp, 1988), convergent and divergent connections between different cortical areas must be integrated. This requires to model all areas as large-scale, spatially organized networks. Such a model of the first three stages of the ventral stream would combine the separated efforts by Schmidt et al. (2018b) regarding multi-area and Senk et al. (2023) regarding spatial modeling in a joint framework. With this, a finer investigation of contextual modulations in V1 and a distinction between internal (i.e. withing V1) and top-down effects (Liang et al., 2017) is possible. Additionally, the separation of neural manifolds as observed in Chapter 4 can be studied in a more detailed way.

SIMULATION TECHNOLOGY In order to perform neuroscientific research using digital twins, simulation technologies are needed that allow for efficient instantiations and executions of the devised models. Such simulation technologies must to be improved continuously: among others, features desired by users have to be added, correctness has to be assured, the code has to be maintained and the performance has to be monitored to identify bottlenecks as well as potential problems of added components. In Part iv, we focused on assessing the performance of the `NEST` simulator (Gewaltig and Diesmann, 2007). We first measured the performance (estimated by the simulation speed

and consumed energy) of simulations of a full-density model of the local cortical circuit in Chapter 10. The simulations ran faster than real-time on a single compute node (Figure 10.4). In comparison with other technologies, we showed that the NEST simulator can keep pace in simulation speed as well as consumed energy with more specialized, neuromorphic approaches (Table 10.2).
As it turned out, benchmarking simulation technologies in itself poses major difficulties, beyond the mere comparison of the results (Gutzen et al., 2018): keeping track of metadata, the interplay between user- as well as system-software and hardware, the choice of models representing a specific problem class, and a transparent way of displaying and comparing benchmarking results is non-trivial. In Chapter 11 we addressed this issue by first classifying the different dimensions of HPC benchmarking (Figure 11.1). We then devised a workflow that met the requirements of the different dimensions with a view to spiking neural network simulations and provided a reference implementation (Figure 11.4). With this, we provide a framework to reproducibly and systematically benchmark simulation technologies. We subsequently applied the derived framework using NEST to assess the performance of low-level adjustment to the communication infrastructure: the suggested changes indeed improve the performance for large models (Figure 11.6) while keeping it identical for smaller networks (Figure 11.7). The developed workflow and framework thus was successfully employed in the development of the NEST simulator. Future extensions of the framework might include additional benchmarking models assessing the performance of other features of the NEST simulator (as e.g. structural plasticity) and a graphical user interface allowing for an easier usage.

Taking together, in this thesis a holistic view on the usage of digital twins in Neuroscience has been proposed, and work according to this view has been conducted. As such, this thesis is—beyond the presented scientific results in themselves—an attempt towards a broader, more systematic usage of digital twins in neuroscientific research.

Part VI

APPENDIX

13 SUPPLEMENTARY FIGURES: NEURAL MANIFOLDS IN V1 CHANGE WITH ...

13.1 ADDITIONAL DATA FIGURES

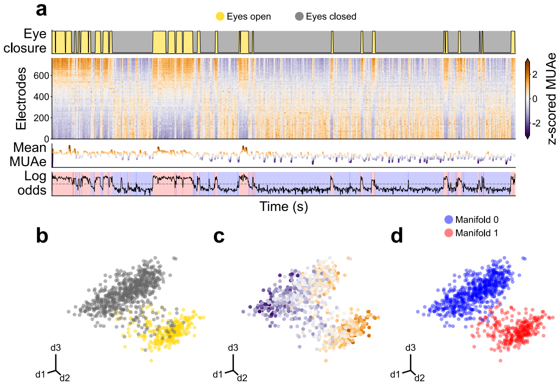


Figure S1: Overview of the experimental data from session L_RS_090817. **a** Time evolution of signals. **b, c, d** First three principal components of the MUAe neural manifold.

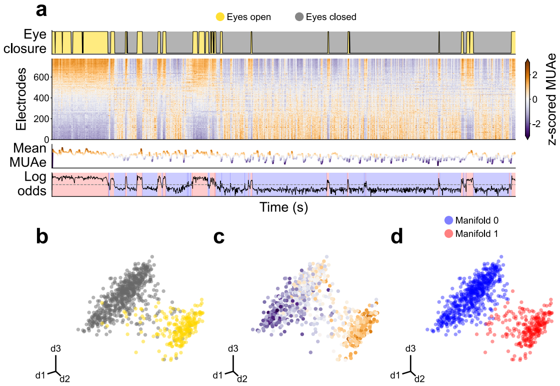


Figure S2: Overview of the experimental data from session L_RS_100817. **a** Time evolution of signals. **b, c, d** First three principal components of the MUAe neural manifold.

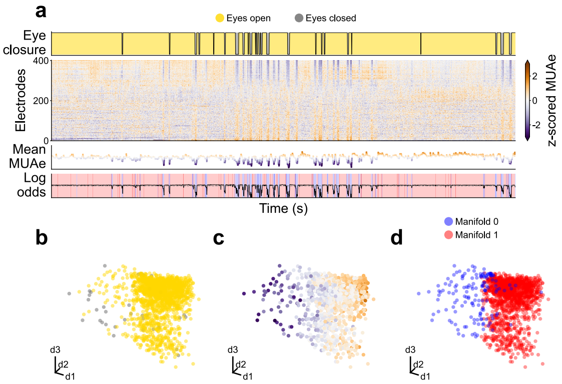


Figure S3: Overview of the experimental data from session A_RS_150819. **a** Time evolution of signals. **b, c, d** First three principal components of the MUAe neural manifold.

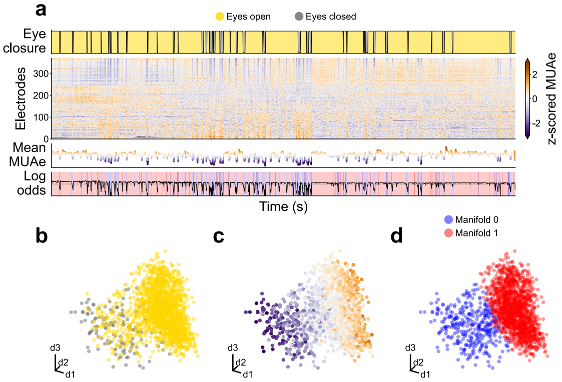


Figure S4: Overview of the experimental data from session A_RS_160819. **a** Time evolution of signals. **b, c, d** First three principal components of the MUAe neural manifold.

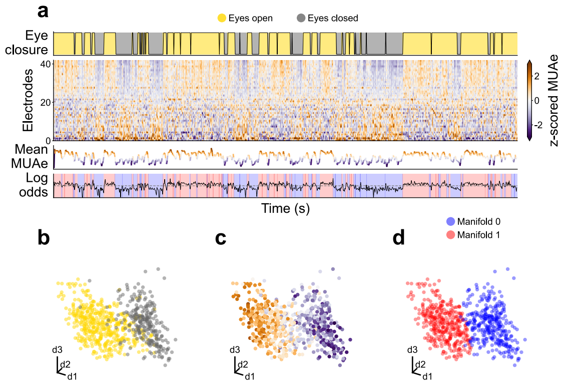


Figure S5: Overview of the experimental data from session Y_RS_180122. **a** Time evolution of signals. **b, c, d** First three principal components of the MUAe neural manifold.

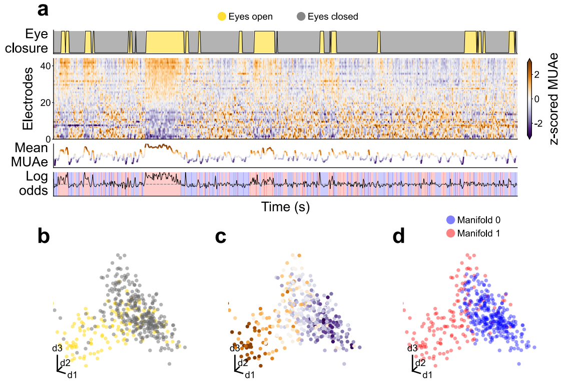


Figure S6: Overview of the experimental data from session Y_RS_180122. **a** Time evolution of signals. **b, c, d** First three principal components of the MUAe neural manifold.

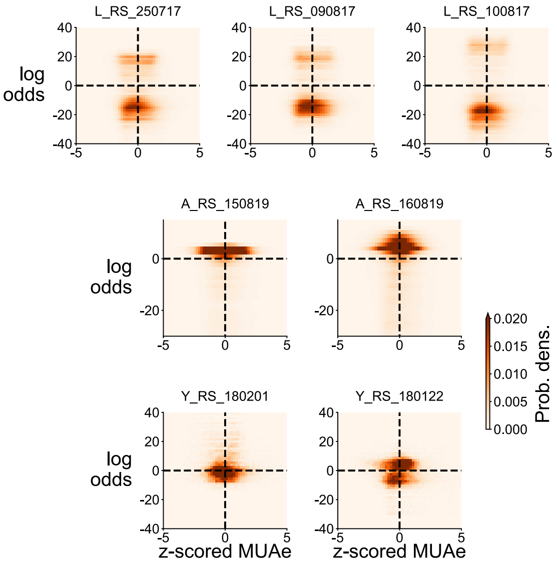


Figure S7: Heatmap of z-scored MUAe and log odds. Darker colour indicates higher occurrence. If the cluster in the neural activity were completely explained by the change (higher in eyes-open periods) activity, the heatmap would reveal a diagonal structure. Instead, a spread over multiple quadrants can be observed.

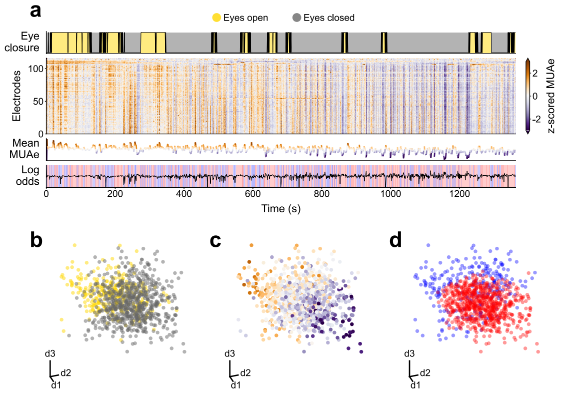


Figure S8: V4 activity from session L_RS_250717 does not show distinct clusters in its neural manifold. **a** Time evolution of signals. **b, c, d** Three dimensional PCA of the MUAe neural manifold.

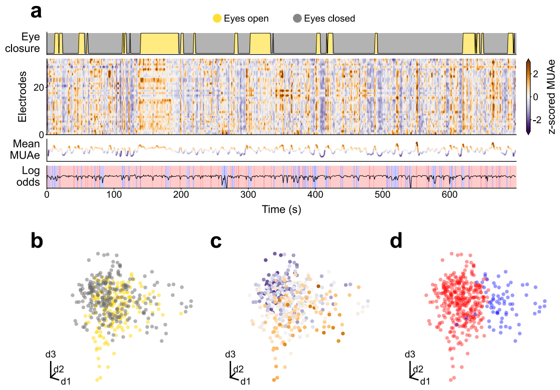


Figure S9: DP activity from session Y_RS_180201 does not show distinct clusters in its neural manifold. **a** Time evolution of signals. **b, c, d** Three dimensional PCA of the MUAe neural manifold.

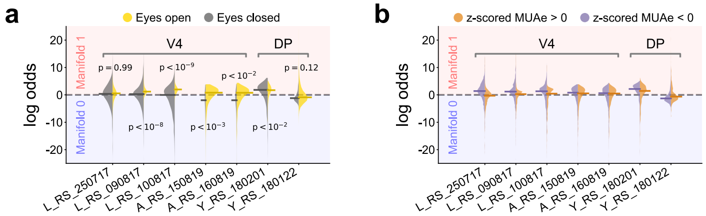


Figure S10: V4 and DP manifold log odds are not strongly correlated with eye closure nor with MUAe. **a, b** Violin plots of V4 and DP for eye closure (**a**) and MUAe activity (**b**). Neither show a clear separation along different neural manifolds.

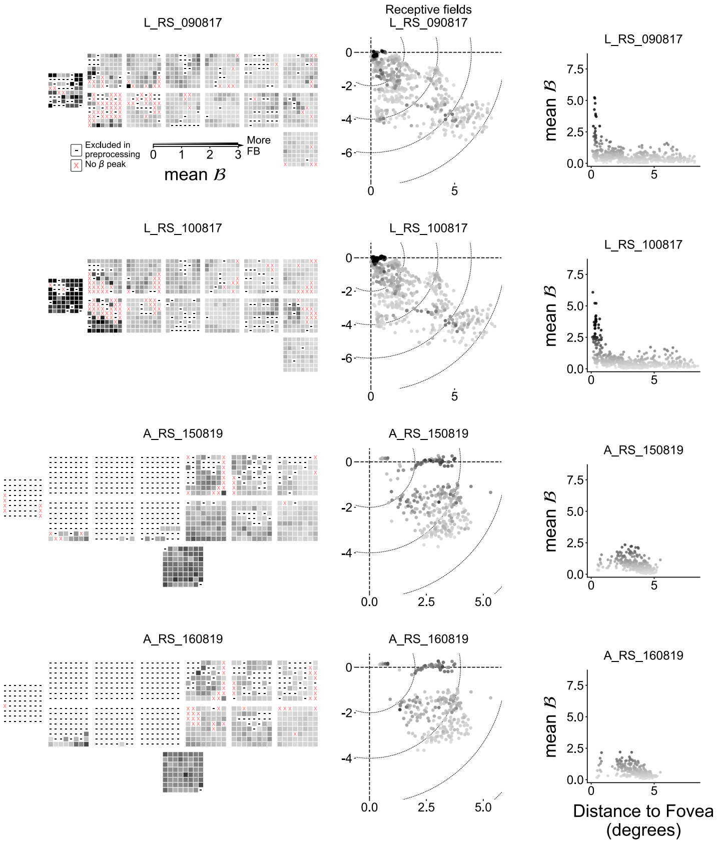


Figure S11: Spatial distribution of Granger causality strength per electrode for all relevant sessions. *(Left column)* Schematic representation of the electrode locations overlaid with the mean top-down signal strength $\mathcal{B}$ per electrode (see Section 4.2.3 for a description of $\mathcal{B}$). *(Center column)* Receptive field (RF) map overlaid with the mean $\mathcal{B}$ per electrode. Stronger $\mathcal{B}$ is found around the foveal region of V1. *(Right column)* Mean $\mathcal{B}$ displayed against the distance from the fovea.

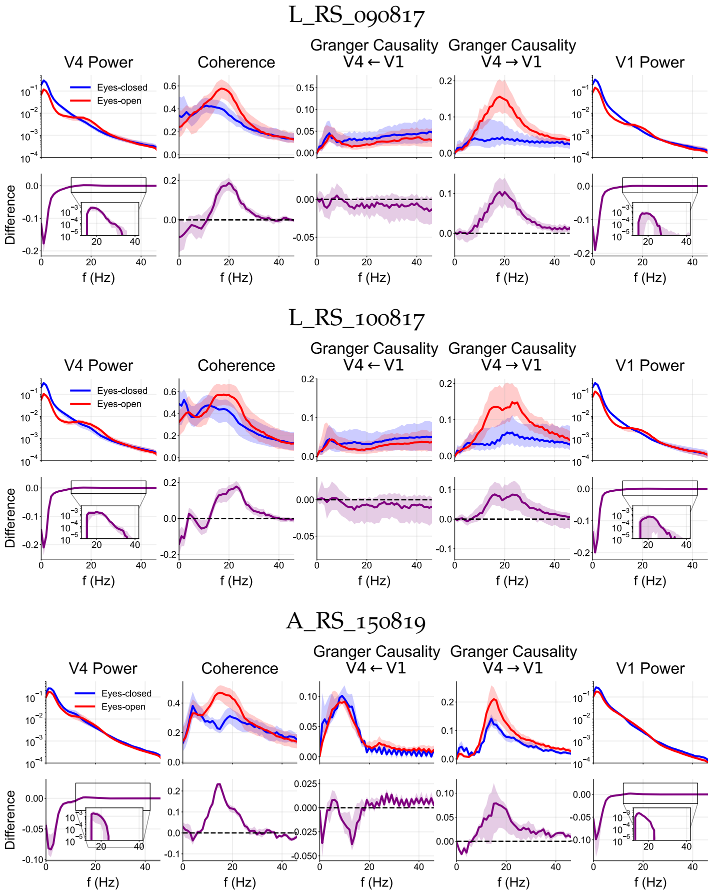


Figure S12: Spectral power, coherence, and Granger causality for the electrodes with high causality strength ($\mathcal{B} > 10$) in sessions L_RS_090817, L_RS_100817, and A_RS_150819. The data for each behavioural condition (eyes-open/closed) were concatenated and their metrics reported separately. Thick line shows median across electrodes (or pairs of electrodes) and shading indicates the 25th to 75th percentile (top row for each session). The difference between eyes-open and eyes-closed was calculated for each electrode or pair of electrodes (bottom row for each session).

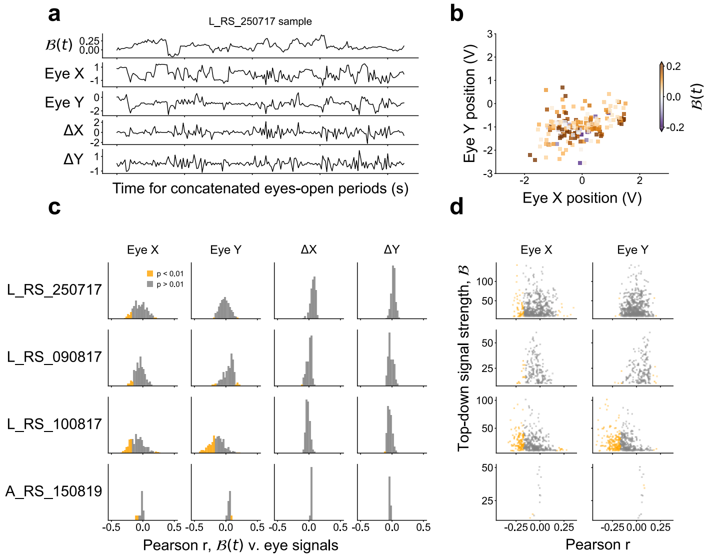
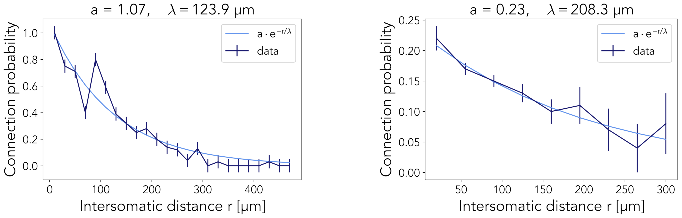


Figure S13: Top-down signals are not correlated with gaze direction. **a** Sample traces of the β-band Granger causality difference, gaze direction (Eye X, Eye Y), and gaze direction derivative (ΔX, ΔY). **b** Sample β-band causality difference over the gaze locations. Higher top-down causality is not concentrated in particular regions. **c** Histograms of Pearson correlation coefficients between time-dependent causality difference and gaze signals, computed for all electrode pairs in all sessions. Significant ($p < 0.01$ two-tailed) part of histograms shown in orange. Gaze direction derivatives show no significant correlations. Note that we did not correct for multiple testing, since reducing the p-value threshold would simply reinforce our finding that no strong correlation was present between the gaze and the top-down signals; i.e., no multiple testing correction was the more conservative approach in this case. **d** Scatter plot of the summed time-independent causality difference against the correlation with gaze direction. There is no clear relation between $\mathcal{B}(t)$-gaze correlation and causality strength.

14

SUPPLEMENTARY MATERIAL: A SPATIAL MODEL OF THE PRIMARY ...

14.1 SPATIAL STRUCTURE OF INTRA-AREAL CONNECTIVITY



(a) Layer 2/3 interneurons.

(b) Layer 5 pyramidal neurons.

Figure S1: Distance-dependent connection probability. Navy points are data from Packer and Yuste (2011) (left, mouse) and Perin et al. (2011) (right, rat), light blue curves are exponential fits to extract λ_{vX}.

Source layer	**Spatial spread** [mm]		
	Supragranular site	**Granular sites**	**Infragranular site**
L2	0.6	0.3	0.25
L3	0.6	0.3	0.3
L4AB	0.5	0.3	0.5
L4C	0.25	0.3	0.5
L5	0.3	0.3	0.6
L6	0.25	0.3	0.6

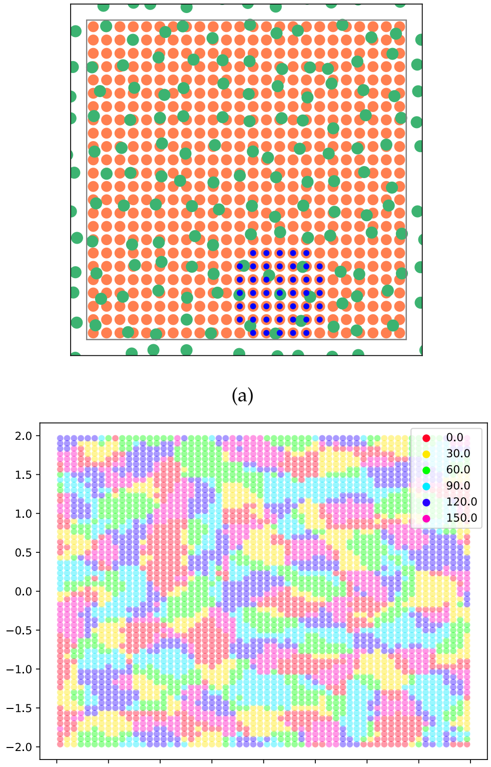
Table S1: Estimated spread of lateral connectivity of inhibitory neurons. The table was compiled by combining numeric values as well as information regarding the overall connectivity scheme provided by Kritzer et al. (1992) *Source layer* indicates the layer of the projecting neurons, *Spatial spread* is resolved by the respective injection sites.

Target layer	$\lambda_{v,\mathrm{L3E}}$ **[mm]**
L2	0.204
L3	0.282
L4B	0.241
L4Cα	0.171
L4Cβ	0.112
L5	0.226

Table S2: Target specific characteristic lengths for layer 3 excitatory neurons derived from anterograde tracing data from Sincich and Blasdel (2001).

Target Layer	**Injection Site**					
	L2/3A	**L3B/L4A**	**L4B**	**L5A**	**L5B**	**L6**
L2/3A	long	med.	short	short	long	short
L3B	med.	long	med.	med.	med.	med.
L4AB	short	long	long	short	med.	long
L4C	N.A.	N.A.	N.A.	short	med.	med.
L5	long	med.	long	long	long	N.A.
L6	med.	short	long	med.	long	long

Table S3: Qualitative assessment of lateral axonal spreading after injection into different cortical layers (Blasdel et al., 1985). N.A. indicates that no axonal processes are reported. “Med.” stands for medium.



(a)

(b)

Figure S2: Instantiations of thalamic and cortical dummy neurons used in orientation map generation a) Green: Thalamic neurons Orange: Cortical dummy neurons (downsampled to 15% for better visibility). Blue Target cortical dummy neurons of one thalamic neuron. b) Raw orientation map used to generated orientation map for model of macaque V1.

$$
\begin{aligned}
p^{\mathrm{E}}_{\mathrm{ON,ON}}(x,y) &= p(x,y,\lambda,0,\gamma,\sqrt{2}\cdot\sigma)\\
p^{\mathrm{E}}_{\mathrm{ON,OFF}}(x,y) &= p(x,y,\lambda,\pi,\gamma,\sqrt{2}\cdot\sigma)\\
p^{\mathrm{I}}_{\mathrm{ON,ON}}(x,y) &= p(x,y,\lambda,\pi,\gamma,\sqrt{2}\cdot\sigma)\\
p^{\mathrm{I}}_{\mathrm{ON,OFF}}(x,y) &= p(x,y,\lambda,0,\gamma,\sqrt{2}\cdot\sigma)\\
p^{\mathrm{E}}_{\mathrm{OFF,OFF}}(x,y) &= p(x,y,\lambda,0,\gamma,\sqrt{2}\cdot\sigma)\\
p^{\mathrm{E}}_{\mathrm{OFF,ON}}(x,y) &= p(x,y,\lambda,\pi,\gamma,\sqrt{2}\cdot\sigma)\\
p^{\mathrm{I}}_{\mathrm{OFF,OFF}}(x,y) &= p(x,y,\lambda,\pi,\gamma,\sqrt{2}\cdot\sigma)\\
p^{\mathrm{I}}_{\mathrm{OFF,ON}}(x,y) &= p(x,y,\lambda,0,\gamma,\sqrt{2}\cdot\sigma)
\end{aligned}
\tag{14.1}
$$

Radius R	$\alpha(R)$
λ_{vX}	3.78
$2\lambda_{vX}$	1.68
$3\lambda_{vX}$	1.25
$4\lambda_{vX}$	1.10
$5\lambda_{vX}$	1.04

Table S4: Radius dependent scaling factor for peak zero-distance connection probability

BIBLIOGRAPHY

Abeles, Moshe (1982). *Local Cortical Circuits: An Electrophysiological Study*. Studies of Brain Function. Berlin, Heidelberg, New York: Springer-Verlag (cit. on p. 10).

– (1991). *Corticonics: Neural Circuits of the Cerebral Cortex*. 1st. Cambridge: Cambridge University Press. DOI: 10.1017/CBO9780511574566 (cit. on pp. 3, 62).

Adrian, Edgar D (1926). “The impulses produced by sensory nerve endings: Part I.” In: *J. Physiol.* 61.1, pp. 49–72. DOI: 10.1113/jphysiol.1926.sp002273 (cit. on p. 4).

Aertsen, Ad and Hubert Preißl (1990). “Dynamics of Activity and Connectivity in Physiological Neuronal Networks.” In: *Nonlinear Dynamics and Neuronal Networks*. Ed. by H. G. Schuster. Proceedings of the 63rd W. E. Heraeus Seminar Friedrichsdorf 1990. VCH, pp. 281–301 (cit. on pp. 5, 19).

Albers, Jasper et al. (2022). “A Modular Workflow for Performance Benchmarking of Neuronal Network Simulations.” In: *Front. Neuroinform.* 16, p. 837549. DOI: 10.3389/fninf.2022.837549 (cit. on p. vii).

Amit, Daniel J., Hanoch Gutfreund, and H. Sompolinsky (1985). “Storing Infinite Numbers of Patterns in a Spin-Glass Model of Neural Networks.” In: *Phys. Rev. Lett.* 55.14, pp. 1530–1533. DOI: 10.1103/physrevlett.55.1530 (cit. on p. 6).

Angelucci, A. et al. (2002). “Circuits for Local and Global Signal Integration in Primary Visual Cortex.” In: *J. Neurosci.* 22.19, pp. 8633–8646 (cit. on pp. 62, 80, 97, 169).

Antolík, Ján et al. (2018). “A comprehensive data-driven model of cat primary visual cortex.” In: *BioRxiv*. DOI: 10.1101/416156 (cit. on pp. 6, 64, 65, 85, 169).

Avitan, Lilach and Carsen Stringer (July 2022). “Not so spontaneous: Multi-dimensional representations of behaviors and context in sensory areas.” en. In: *Neuron*, S0896627322005888. ISSN: 08966273. DOI: 10.1016/j.neuron.2022.06.019 (cit. on p. 19).

Bastos, Andre Moraes et al. (2015). “Visual areas exert feedforward and feedback influences through distinct frequency channels.” In: *Neuron* 85.2, pp. 390–401 (cit. on pp. 18, 34, 41, 167).

Baxter, Rob et al. (Sept. 2012). “The research software engineer.” In: *Digital Research 2012*. Oxford, United Kingdom, pp. 1–3 (cit. on p. 118).

Beaulieu, C. and M. Colonnier (1985). “A laminar analysis of the number of round-asymmetrical and flat-symmetrical synapses on spines, dendritic trunks, and cell bodies in area 17 of the cat." In: *J. Comp. Neurol.* 231, pp. 180–189 (cit. on pp. 67, 109, 113).

Beaulieu, C et al. (1992). "Quantitative distribution of GABA-immunopositive and-immunonegative neurons and synapses in the monkey striate cortex (area 17)." In: *Cereb. Cortex* 2.4, pp. 295–309 (cit. on p. 66).

Beaulieu, Clermont and Marc Colonnier (July 1983). "The number of neurons in the different laminae of the binocular and monocular regions of area 17 in the cat." In: *J. Comp. Neurol.* 217.3, pp. 337–344. DOI: 10.1002/cne.902170308 (cit. on pp. 67, 109, 113).

Berger, Hans (1929). "Über das Elektrenkephalogramm des Menschen." In: *Arch. Psychiatr. Nervenkr.* 87, pp. 527–570 (cit. on pp. 37, 40).

Bernstein, Julius (1868). "Ueber den zeitlichen Verlauf der negativen Schwankung des Nervenstroms." In: *Archiv für die gesamte Physiologie des Menschen und der Tiere* 1.1, pp. 173–207 (cit. on p. 3).

Beyeler, Michael et al. (July 2015). "CARLsim 3: A user-friendly and highly optimized library for the creation of neurobiologically detailed spiking neural networks." In: *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE. DOI: 10.1109/ijcnn.2015.7280424 (cit. on p. 133).

Bhalla, U., D. Bilitch, and J. M. Bower (1992). "Rallpacks: A set of benchmarks for neuronal simulators." In: *Trends Neurosci.* 15.11, pp. 453–458. DOI: 10.1016/0166-2236(92)90009-W (cit. on p. 134).

Billeh, Yazan N et al. (2020). "Systematic integration of structural and functional data into multi-scale models of mouse primary visual cortex." In: *Neuron* 106.3, 388–403.e18. DOI: https://doi.org/10.1016/j.neuron.2020.01.040 (cit. on pp. 6, 7, 9, 64, 65).

Binzegger, Tom, Rodney J. Douglas, and Kevan A. C. Martin (2004). "A Quantitative Map of the Circuit of Cat Primary Visual Cortex." In: *J. Neurosci.* 39.24, pp. 8441–8453. DOI: 10.1523/JNEUROSCI.1400-04.2004 (cit. on pp. 7, 66, 67, 71, 93, 103, 106, 107, 113, 114, 169).

Blasdel, Gary G, Jennifer S Lund, and D Fitzpatrick (1985). "Intrinsic connections of macaque striate cortex: axonal projections of cells outside lamina 4C." In: *J. Neurosci.* 5.12, pp. 3350–3369. DOI: 10.1523/JNEUROSCI.05-12-03350.1985 (cit. on pp. 80, 94, 184).

Blasdel, Gary G and Guy Salama (1986). "Voltage-sensitive dyes reveal a modular organization in monkey striate cortex." In: *Nature* 321.6070, pp. 579–585 (cit. on p. 63).

Bosking, William H. et al. (1997). "Orientation selectivity and the arrangement of horizontal connections in tree shrew striate cortex." In: *J. Neurosci.* 17.6, pp. 2112–2127. DOI: 10.1523/JNEUROSCI.17-06-02112.1997 (cit. on p. 80).

Braitenberg, Valentin and Almut Schüz (1991). *Anatomy of the Cortex: Statistics and Geometry*. Berlin, Heidelberg, New York: Springer-Verlag. ISBN: 3-540-53233-1 (cit. on pp. 3, 5, 67, 78).

Brette, R. et al. (July 2007). “Simulation of networks of spiking neurons: A review of tools and strategies.” In: *J. Comput. Neurosci.* 23.3, pp. 349–398. DOI: 10.1007/s10827-007-0038-6 (cit. on pp. 117, 133, 134).

Brodmann, Korbinian (1909). *Vergleichende Lokalisationslehre der Großhirnrinde in ihren Prinzipien dargestellt auf Grund des Zellenbaues*. Leipzig: Johann Ambrosius Barth (cit. on p. 3).

Brüderle, Daniel et al. (2009). “Establishing a Novel Modeling Tool: A Python-based Interface for a Neuromorphic Hardware System.” In: *Front. Neuroinform.* 3.17 (cit. on p. 117).

Brunel, Nicolas (2000). “Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons.” In: *J. Comput. Neurosci.* 8.3, pp. 183–208. DOI: 10.1023/a:1008925309027 (cit. on pp. 6, 45, 50, 51, 59, 168).

Budd, Julian M. and Zoltán F. Kisvárday (July 2001). “Local lateral connectivity of inhibitory clutch cells in layer 4 of cat visual cortex (area 17).” In: *Exp. Brain Res.* 140.2, pp. 245–250. DOI: 10.1007/s002210100817 (cit. on p. 93).

Carnevale, T. (2007). “Neuron simulation environment.” In: *Scholarpedia* 2.6, p. 1378. DOI: 10.4249/scholarpedia.1378 (cit. on p. 133).

Chaudhuri, Rishidev et al. (2019). “The intrinsic attractor manifold and population dynamics of a canonical cognitive circuit across waking and sleep.” en. In: *Nature Neuroscience* 22.9, pp. 1512–1520. ISSN: 1546-1726. DOI: 10.1038/s41593-019-0460-x (cit. on pp. 18, 19, 23).

Chavane, Frédéric et al. (2011). “Lateral spread of orientation selectivity in V1 is controlled by intracortical cooperativity.” In: *Front. Syst. Neurosci.* 5, p. 4. DOI: 10.3389/fnsys.2011.00004 (cit. on p. 170).

Chen, Xing et al. (Dec. 2022). “1024-Channel Electrophysiological Recordings in Macaque V1 and V4 during Resting State.” In: *Sci. Data* 9.1, p. 77. ISSN: 2052-4463. DOI: 10.1038/s41597-022-01180-1 (cit. on pp. 19, 20, 40).

Churchland, Mark M et al. (2012). “Neural population dynamics during reaching.” In: *Nature* 487.7405, p. 51 (cit. on p. 18).

Cunningham, John P. and M. Yu Byron (Aug. 2014). “Dimensionality reduction for large-scale neural recordings.” In: *Nat. Neurosci.* 17.11, pp. 1500–1509. DOI: 10.1038/nn.3776 (cit. on p. 19).

Dąbrowska, Paulina Anna et al. (May 2021). “On the Complexity of Resting State Spiking Activity in Monkey Motor Cortex.” In: *Cereb. Cortex Commun.* 2.3. tgab033. ISSN: 2632-7376. DOI: 10.1093/texcom/tgab033 (cit. on pp. 11, 15, 40).

Dahmen, David et al. (2020). “Strong coupling and local control of dimensionality across brain areas.” In: *BioRxiv*. DOI: 10.1101/2020.11.02.365072 (cit. on pp. 18, 24, 59).

Dai, Wei and Daniel Berleant (Dec. 2019). “Benchmarking Contemporary Deep Learning Hardware and Frameworks: A Survey of Qualitative Metrics." In: IEEE. DOI: 10.1109/cogmi48466.2019.00029 (cit. on p. 135).

Davison, Andrew et al. (2009). "PyNN: a common interface for neuronal network simulators." In: *Front. Neuroinform.* 2.11, p. 10. DOI: 10.3389/neuro.11.011.2008 (cit. on p. 162).

De Haan, Marcel Jan et al. (2018). "Real-time visuomotor behavior and electrophysiology recording setup for use with humans and monkeys." en. In: *J. Neurophysiol.* 120.2, pp. 539–552. ISSN: 0022-3077 and 1522-1598. DOI: 10.1152/jn.00262.2017 (cit. on pp. 19, 21).

De No, Rafael Lorente (1933). "Studies on the structure of the cerebral cortex I The area entorhinalis." In: *J. Psychol. Neurol.* 45, pp. 381–438 (cit. on p. 4).

Deco, Gustavo, Viktor K. Jirsa, and Anthony R. McIntosh (2011). "Emerging concepts for the dynamical organization of resting-state activity in the brain." In: *Nat. Rev. Neurosci.* 12, pp. 43–56. DOI: 10.1038/nrn2961 (cit. on p. 15).

Dehay, Colette et al. (1988). "Absence of interhemispheric connections of area 17 during development in the monkey." In: *Nature* 331.6154, pp. 348–350 (cit. on p. 77).

Denker, Michael et al. (Apr. 2022). *Elephant 0.11.1*. DOI: 10.5281/ZENODO.6470226 (cit. on p. 26).

Dhamala, Mukeshwar, Govindan Rangarajan, and Mingzhou Ding (2008). "Estimating Granger Causality from Fourier and Wavelet Transforms of Time Series Data." en. In: *Phys. Rev. Lett.* 100.1. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.100.018701 (cit. on pp. 24, 25).

Diaz-Pier, Sandra et al. (2016). "Automatic Generation of Connectivity for Large-Scale Neuronal Network Models through Structural Plasticity." In: *Front. Neuroanat.* 10, p. 57. DOI: 10.3389/fnana.2016.00057 (cit. on p. 133).

Diesmann, Markus, Marc-Oliver Gewaltig, and Ad Aertsen (1999). "Stable propagation of synchronous spiking in cortical neural networks." In: *Nature* 402.6761, pp. 529–533. DOI: 10.1038/990101 (cit. on p. 10).

Doiron, Brent et al. (2016). "The mechanics of state-dependent neural correlations." In: *Nat. Neurosci.* 19, pp. 383–393. DOI: 10.1038/nn.4242 (cit. on p. 59).

Dongarra, Jack J., Piotr Luszczek, and Antoine Petitet (2003). "The LINPACK Benchmark: past, present and future." In: *Concurr. Comput.* 15.9, pp. 803–820. DOI: 10.1002/cpe.728 (cit. on p. 132).

Du, P., W. A. Kibbe, and S. M. Lin (Sept. 2006). "Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching." en. In: *Bioinformatics* 22.17, pp. 2059–2065. ISSN: 1367-4803, 1460-2059. DOI: 10.1093/bioinformatics/btl355 (cit. on p. 26).

Dura-Bernal, Salvador et al. (2023). "Multiscale model of primary motor cortex circuits predicts in vivo cell-type-specific, behavioral state-dependent dynamics." In: *Cell Rep.* 42.6. DOI: 10.1016/j.celrep.2023.112574 (cit. on p. 6).

Edelsbrunner, Herbert and John L Harer (2022). *Computational topology: an introduction*. American Mathematical Society (cit. on p. 23).

Einevoll, Gaute T. et al. (2019). "The Scientific Case for Brain Simulations." In: *Neuron* 102.4, pp. 735–744. ISSN: 0896-6273. DOI: 10.1016/j.neuron.2019.03.027 (cit. on pp. 10, 117).

El Boustani, Sami and Alain Destexhe (2009). "A Master Equation Formalism for MacroscopicModeling of Asynchronous Irregular Activity States." In: *Neural Comput.* 21, pp. 46–100 (cit. on p. 45).

Eliasmith, C. et al. (2012). "A Large-Scale Model of the Functioning Brain." In: *Science* 338.6111, pp. 1202–1205. DOI: 10.1126/science.1225266 (cit. on p. 7).

Eliasmith, Chris and Charles H Anderson (2004). *Neural engineering: Computation, representation, and dynamics in neurobiological systems*. MIT press (cit. on p. 7).

Eliasmith, Chris and Oliver Trujillo (2014). "The use and abuse of large-scale brain models." In: *Curr. Opin. Neurobiol.* 25, pp. 1–6. DOI: 10.1016/j.conb.2013.09.009 (cit. on p. 7).

Eppler, J. M. et al. (2009). "PyNEST: a convenient interface to the NEST simulator." In: *Front. Neuroinform.* 2, p. 12. DOI: 10.3389/neuro.11.012.2008 (cit. on pp. 136, 144, 146).

Evans, Jason (2006). "A Scalable Concurrent malloc(3) Implementation for FreeBSD." In: *Proceedings of the BSDCan Conference*. Ottawa, Canada (cit. on p. 121).

Fardet, Tanguy et al. (Aug. 2021). *NEST 2.20.2*. DOI: 10.5281/zenodo.5242954 (cit. on pp. 153, 154).

Felleman, Daniel J and David C Van Essen (1991). "Distributed hierarchical processing in the primate cerebral cortex." In: *Cereb. Cortex* 1, pp. 1–47. DOI: 10.1093/cercor/1.1.1-a (cit. on p. 63).

Festa, Dylan et al. (2021). "Neuronal variability reflects probabilistic inference tuned to natural image statistics." In: *Nat. Commun.* 12.1, p. 3635. DOI: 10.1038/s41467-021-23838-x (cit. on p. 4).

Feulner, Barbara and Claudia Clopath (Feb. 2021). "Neural manifold under plasticity in a goal driven learning behaviour." en. In: *PLOS Comput. Biol.* 17.2. Ed. by Abigail Morrison, e1008621. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1008621 (cit. on p. 18).

Fitzpatrick, D et al. (1987). "Distribution of GABAergic neurons and axon terminals in the macaque striate cortex." In: *J. Comp. Neurol.* 264.1, pp. 73–91. DOI: 10.1002/cne.902640107 (cit. on p. 66).

Fitzpatrick, David, Jennifer S Lund, and Gary G Blasdel (1985). "Intrinsic connections of macaque striate cortex: afferent and efferent connections of lamina 4C." In: *J. Neurosci.* 5.12, pp. 3329–3349. DOI: 10.1523/JNEUROSCI.05-12-03329.1985 (cit. on pp. 80, 94).

Frégnac, Yves (2017). "Big data and the industrialization of neuroscience: A safe roadmap for understanding the brain?" In: *Science* 358.6362, pp. 470–477. DOI: 10.1126/science.aan8866 (cit. on p. 7).

Froudarakis, Emmanouil et al. (2020). "Object manifold geometry across the mouse cortical visual hierarchy." en. In: *BioRxiv*. DOI: 10.1101/2020.08.20.258798 (cit. on p. 18).

Furber, S.B. et al. (2014). "The SpiNNaker Project." In: *Proc. IEEE* 102.5, pp. 652–665. DOI: 10.1109/JPROC.2014.2304638 (cit. on p. 133).

Furber, Steve (Aug. 2016). "Large-scale neuromorphic computing systems." In: *J. Neural Eng.* 13.5, p. 051001. DOI: 10.1088/1741-2560/13/5/051001 (cit. on p. 120).

Gabriel, Edgar et al. (2004). "Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation." In: *Recent Advances in Parallel Virtual Machine and Message Passing Interface*. Ed. by Dieter Kranzlmüller, Péter Kacsuk, and Jack Dongarra. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 97–104. DOI: 10.1007/978-3-540-30218-6_19 (cit. on p. 122).

Gallego, Juan A et al. (2017). "Neural manifolds for the control of movement." In: *Neuron* 94.5, pp. 978–984 (cit. on p. 18).

Gallego, Juan A et al. (2018). "Cortical population activity within a preserved neural manifold underlies multiple motor behaviors." In: *Nat. Commun.* 9.1, pp. 1–13 (cit. on p. 18).

Gamblin, Todd et al. (Nov. 2015). "The Spack package manager." In: ACM, pp. 1–12. DOI: 10.1145/2807591.2807623 (cit. on p. 144).

Gancarz, Gregory and Stephen Grossberg (1998). "A neural model of the saccade generator in the reticular formation." In: *IEEE Trans. Neural Netw.* 11.7, pp. 1159–1174 (cit. on p. 6).

Gao, Peiran and Surya Ganguli (2015). "On simplicity and complexity in the brave new world of large-scale neuroscience." en. In: *Curr. Opin. Neurobiol.* 32, pp. 148–155. ISSN: 09594388. DOI: 10.1016/j.conb.2015.04.003 (cit. on pp. 19, 40, 167).

Gao, Peiran et al. (2017). "A theory of multineuronal dimensionality, dynamics and measurement." In: *BioRxiv*, p. 214262. DOI: 10.1101/214262. arXiv: bioRxiv214262 (cit. on pp. 18, 19).

Gao, Richard et al. (2020). "Neuronal timescales are functionally dynamic and shaped by cortical microarchitecture." In: *eLife* 9, e61277 (cit. on p. 114).

Garcia-Marin, Virginia, Jenna G Kelly, and Michael J Hawken (Nov. 2017). "Major Feedforward Thalamic Input Into Layer 4C of Primary Visual Cortex in Primate." In: *Cereb. Cortex* 29.1, pp. 134–149. ISSN: 1047-3211. DOI: 10.1093/cercor/bhx311. eprint: https://academic.oup.com/cercor/article-pdf/29/1/134/28546658/bhx311.pdf (cit. on pp. 65, 76).

Gardner, Richard J. et al. (2022). "Toroidal topology of population activity in grid cells." en. In: *Nature* 602.7895, pp. 123–128. ISSN:

0028-0836, 1476-4687. DOI: 10.1038/s41586-021-04268-7 (cit. on p. 18).

Geimer, Markus, Kenneth Hoste, and Robert McLay (Nov. 2014). "Modern Scientific Software Management Using EasyBuild and Lmod." In: IEEE. DOI: 10.1109/hust.2014.8 (cit. on p. 144).

Gewaltig, Marc-Oliver and Markus Diesmann (2007). "NEST (NEural Simulation Tool)." In: *Scholarpedia J.* 2.4, p. 1430. DOI: 10.4249/scholarpedia.1430 (cit. on pp. 6, 10, 86, 118, 121, 133, 135, 144, 170).

Ghrist, Robert (2008). "Barcodes: The persistent topology of data." en. In: *Bulletin of the American Mathematical Society* 45.1, pp. 61–75. ISSN: 0273-0979, 1088-9485. DOI: 10.1090/S0273-0979-07-01191-3 (cit. on p. 19).

Giannaris, Eustathia Lela and Douglas L Rosene (2012). "A stereological study of the numbers of neurons and glia in the primary visual cortex across the lifespan of male and female rhesus monkeys." In: *J. Comp. Neurol.* 520.15, pp. 3492–3508 (cit. on p. 66).

Gilbert, C. D. and T. N. Wiesel (1983). "Clustered intrinsic connections in cat visual cortex." In: *J. Neurosci.* 5, pp. 1116–33 (cit. on pp. 81, 82).

– (1989). "Columnar Specificity of Intrinsic Horizontal and Corticocortical Connections in Cat Visual Cortex." In: *J. Neurosci.* 9.7, pp. 2432–2442. DOI: 10.1523/jneurosci.09-07-02432.1989 (cit. on p. 62).

Golosio, Bruno et al. (Feb. 2021). "Fast Simulations of Highly-Connected Spiking Cortical Models Using GPUs." In: *Front. Comput. Neurosci.* 15, p. 627620. DOI: 10.3389/fncom.2021.627620 (cit. on pp. 117, 128).

Goodman, D.F.M. and R. Brette (2013). "Brian Simulator." In: *Scholarpedia J.* 8.1, p. 10883. DOI: 10.4249/scholarpedia.10883 (cit. on pp. 6, 133).

Griffith, J. S. (1963). "On the Stability of Brain-Like Structures." In: *Biomed. Pharmacol. J.* 3, pp. 299–308 (cit. on p. 10).

Grün, Sonja and Stefan Rotter, eds. (2010). *Analysis of Parallel Spike Trains*. Springer (cit. on p. 6).

Gutzen, Robin et al. (2018). "Reproducible Neural Network Simulations: Statistical Methods for Model Validation on the Level of Network Activity Data." In: *Front. Neuroinform.* 12, p. 90. ISSN: 1662-5196. DOI: 10.3389/fninf.2018.00090 (cit. on pp. 117, 171).

Hagen, Espen et al. (Oct. 2016). "Hybrid scheme for modeling local field potentials from point-neuron networks." In: *Cereb. Cortex* 26.12, pp. 4461–4496. DOI: 10.1093/cercor/bhw237 (cit. on pp. 64, 170).

Hagen, Espen et al. (Aug. 2022). "Brain signal predictions from multi-scale networks using a linearized framework." In: *PLOS Comput. Biol.* 18.8. Ed. by Michele Migliore, e1010353. DOI: 10.1371/journal.pcbi.1010353 (cit. on p. 170).

Hager, Georg and Gerhard Wellein (2010). *Introduction to High Performance Computing for Scientists and Engineers*. 1st Edition. New York: CRC Press. ISBN: 978-1439811924 (cit. on p. 117).

Hahne, Jan et al. (June 2021). *NEST 3.0*. Version 3.0. DOI: 10.5281/zenodo.4739103 (cit. on pp. 136, 154).

Hanslmayr, Simon et al. (2005). "Visual discrimination performance is related to decreased alpha amplitude but increased phase locking." en. In: *Neurosci. Lett.* 375.1, pp. 64–68. ISSN: 03043940. DOI: 10.1016/j.neulet.2004.10.092 (cit. on p. 41).

Harris, Kenneth D. and Alexander Thiele (2011). "Cortical state and attention." In: *Nat. Rev. Neurosci.* 12, pp. 509–523. DOI: 10.1038/nrn3084 (cit. on p. 18).

Hartline, HK (1941). "The neural mechanisms of vision." In: vol. 37. Harvey Lectures, pp. 39–68 (cit. on p. 4).

Hebb, D. O. (1949). *The organization of behavior: A neuropsychological theory*. New York: John Wiley & Sons. DOI: 10.1002/sce.37303405110 (cit. on p. 5).

Heittmann, Arne et al. (2022). "Simulating the Cortical Microcircuit Significantly Faster Than Real Time on the IBM INC-3000 Neural Supercomputer." In: *Front. Neurosci.* 15. ISSN: 1662-453X. DOI: 10.3389/fnins.2021.728460 (cit. on pp. 127, 128).

Helias, Moritz and David Dahmen (2020). *Statistical Field Theory for Neural Networks*. Springer International Publishing, p. 203. DOI: 10.1007/978-3-030-46444-8 (cit. on p. 6).

Helias, Moritz et al. (2012). "Supercomputers ready for use as discovery machines for neuroscience." In: *Front. Neuroinform.* 6, p. 26. DOI: 10.3389/fninf.2012.00026 (cit. on pp. 132, 136, 162).

Hendry, Stewart H et al. (1987). "Numbers and proportions of GABA-immunoreactive neurons in different areas of monkey cerebral cortex." In: *J. Neurosci.* 7.5, pp. 1503–1519. DOI: 10.1523/JNEUROSCI.07-05-01503.1987 (cit. on p. 66).

Herrero, Jose L. et al. (May 2013). "Attention-Induced Variance and Noise Correlation Reduction in Macaque V1 Is Mediated by NMDA Receptors." en. In: *Neuron* 78.4, pp. 729–739. ISSN: 08966273. DOI: 10.1016/j.neuron.2013.03.029 (cit. on pp. 59, 168).

Hodgkin, A. L. and A. F. Huxley (1952). "A Quantitative Description of Membrane Current and Its Application to Conduction and Excitation in Nerve." In: *J. Physiol.* 117, pp. 500–544. DOI: 10.1113/jphysiol.1952.sp004764 (cit. on pp. 3, 6).

Hopfield, J. J. (1982). "Neural networks and physical systems with emergent collective computational abilities." In: *Proc. Natl. Acad. Sci. USA* 79, pp. 2554–2558 (cit. on p. 6).

Hubel, D. H. and T. N. Wiesel (1959). "Receptive fields of single neurones in the cat's striate cortex." In: *J. Physiol.* 148, pp. 574–591 (cit. on pp. 4, 63).

– (1962). “Receptive Fields, Binocular Interaction, and Functional Architecture in the Cat’s Visual Cortex.” In: *J. Physiol.* 160, pp. 106–154 (cit. on p. 4).

Hunter, John D. (2007). “Matplotlib: A 2D Graphics Environment.” In: *Comput. Sci. Eng.* 9.3, pp. 90–95. ISSN: 1521-9615. DOI: 10.1109/MCSE.2007.55 (cit. on p. 151).

Ippen, Tammo et al. (2017). “Constructing neuronal network models in massively parallel environments.” In: *Front. Neuroinform.* 11, p. 30. ISSN: 1662-5196. DOI: 10.3389/fninf.2017.00030 (cit. on p. 122).

Ito, J et al. (2011). “Saccade-related modulations of neuronal excitability support synchrony of visually elicited spikes.” In: *Cereb. Cortex* 21.11, pp. 2482–2497 (cit. on p. 45).

Izhikevich, Eugene M (2007). *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting*. MIT Press. DOI: 10.7551/mitpress/2526.001.0001 (cit. on p. 6).

Izhikevich, Eugene M. and Gerald M. Edelman (2008). “Large-scale model of mammalian thalamocortical systems.” In: *Proc. Natl. Acad. Sci. USA* 105.9, pp. 3593–3598. DOI: 10.1073/pnas.0712231105 (cit. on pp. 6, 67).

Jiang, Xiaolong et al. (2015). “Principles of connectivity among morphologically defined cell types in adult neocortex.” In: *Science* 350.6264, aac9462. DOI: 10.1126/science.aac946 (cit. on p. 110).

Jones, Judson P and Larry A Palmer (1987). “An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex.” In: *J. Neurophysiol.* 58.6, pp. 1233–1258. DOI: 10.1152/jn.1987.58.6.1233 (cit. on p. 84).

Jordan, Jakob et al. (Feb. 2018). “Extremely Scalable Spiking Neuronal Network Simulation Code: From Laptops to Exascale Computers.” In: *Front. Neuroinform.* 12, p. 2. DOI: 10.3389/fninf.2018.00002 (cit. on pp. 10, 132, 134, 136–138, 145, 154–156, 162).

Ju, Nian-Sheng et al. (2021). “Orientation tuning and end-stopping in macaque V1 studied with two-photon calcium imaging.” In: *Cereb. Cortex* 31.4, pp. 2085–2097. DOI: 10.1093/cercor/bhaa346 (cit. on p. 63).

Jülich Supercomputing Centre (2015). *JUQUEEN: IBM Blue Gene/Q® Supercomputer System at the Jülich Supercomputing Centre*. DOI: 10.17815/jlsrf-1-18 (cit. on pp. 155, 156).

Kandel, Eric R. et al. (2013). *Principles of Neural Science*. 5th ed. New York: McGraw-Hill. ISBN: 978-0-071-39011-8 (cit. on pp. 3, 62, 104, 170).

Kauth, Kevin et al. (Apr. 2023). “neuroAIx-Framework: design of future neuroscience simulation systems exhibiting execution of the cortical microcircuit model 20× faster than biological real-time.” In: *Front. Comput. Neurosci.* 17. DOI: 10.3389/fncom.2023.1144143 (cit. on pp. 127, 128).

Klimesch, Wolfgang, Paul Sauseng, and Simon Hanslmayr (2007). "EEG alpha oscillations: The inhibition–timing hypothesis." In: *Brain Res. Rev.* 53.1, pp. 63–88. ISSN: 01650173. DOI: 10.1016/j.brainresrev.2006.06.003 (cit. on pp. 37, 41).

Klink, P. Christiaan et al. (2017). "Distinct Feedforward and Feedback Effects of Microstimulation in Visual Cortex Reveal Neural Mechanisms of Texture Segregation." en. In: *Neuron* 95.1, 209–220.e3. ISSN: 08966273. DOI: 10.1016/j.neuron.2017.05.033 (cit. on p. 18).

Knight, James C., Anton Komissarov, and Thomas Nowotny (Apr. 2021). "PyGeNN: A Python Library for GPU-Enhanced Neural Networks." In: *Front. Neuroinform.* 15, p. 659005. DOI: 10.3389/fninf.2021.659005 (cit. on pp. 117, 128).

Knight, James C. and Thomas Nowotny (Dec. 2018). "GPUs Outperform Current HPC and Neuromorphic Solutions in Terms of Speed and Energy When Simulating a Highly-Connected Cortical Model." In: *Front. Neurosci.* 12, pp. 1–19. DOI: 10.3389/fnins.2018.00941 (cit. on pp. 120, 128).

Korcsak-Gorzo, Agnes et al. (2022). "Phenomenological modeling of diverse and heterogeneous synaptic dynamics at natural density." In: *arXiv:2212.05354*. DOI: 10.48550/arXiv.2212.05354 (cit. on p. 94).

Kristensson, Krister and Yngve Olsson (1971). "Retrograde axonal transport of protein." In: *Brain Res.* 29.2, pp. 363–365 (cit. on p. 5).

Kritzer, MF, A Cowey, and P Somogyi (1992). "Patterns of inter-and intralaminar GABAergic connections distinguish striate (V1) and extrastriate (V2, V4) visual cortices and their functionally specialized subdivisions in the rhesus monkey." In: *J. Neurosci.* 12.11, pp. 4545–4564. DOI: 10.1523/JNEUROSCI.12-11-04545.1992 (cit. on pp. 79, 95, 183).

Kubota, Yoshiyuki, Jaerin Sohn, and Yasuo Kawaguchi (2018). "Large volume electron microscopy and neural microcircuit analysis." In: *Front. Neural Circuits* 12, p. 98. DOI: https://doi.org/10.3389/fncir.2018.00098 (cit. on pp. 5, 107).

Kunkel, Susanne et al. (2012). "Meeting the memory challenges of brain-scale simulation." In: *Front. Neuroinform.* 5, p. 35. DOI: 10.3389/fninf.2011.00035 (cit. on pp. 136, 162).

Kunkel, Susanne et al. (2014). "Spiking network simulation code for petascale computers." In: *Front. Neuroinform.* 8, p. 78. DOI: 10.3389/fninf.2014.00078 (cit. on pp. 136, 162).

Kunkel, Susanne et al. (Mar. 2017). *NEST 2.12.0*. DOI: 10.5281/zenodo.259534 (cit. on p. 136).

Kurth, Anno C et al. (Mar. 2022). "Sub-realtime simulation of a neuronal network of natural density." In: *Neuromorphic Comput. Eng.* 2.2, p. 021001. DOI: 10.1088/2634-4386/ac55fc (cit. on p. vii).

Li, Zhaoping (1999a). "Contextual influences in V1 as a basis for pop out and asymmetry in visual search." In: *Proc. Natl. Acad. Sci. USA*

96.18, pp. 10530–10535. DOI: 10.1073/pnas.96.18.10530 (cit. on p. 170).

– (1999b). "Visual segmentation by contextual influences via intracortical interactions in the primary visual cortex." In: *Netw. Comput. Neural Syst.* 10.2, pp. 187–212. DOI: 10.1088/0954-898X_10_2_305 (cit. on p. 170).

– (2014). *Understanding vision: theory, models, and data*. Oxford University Press. DOI: 10.1093/acprof:oso/9780199564668.001.0001 (cit. on p. 169).

Liang, Hualou et al. (July 2017). "Interactions between feedback and lateral connections in the primary visual cortex." In: *Proc. Natl. Acad. Sci. USA* 114.32, pp. 8637–8642. DOI: 10.1073/pnas.1706183114 (cit. on pp. 18, 170).

Liley, David T.J. and Suresh D. Muthukumaraswamy (Mar. 2020). "Evidence that alpha blocking is due to increases in system-level oscillatory damping not neuronal population desynchronisation." In: *Neuroimage* 208, pp. 116–408. ISSN: 10538119. DOI: 10.1016/j.neuroimage.2019.116408 (cit. on p. 38).

Linssen, Charl et al. (Aug. 2018). *NEST 2.16.0*. DOI: 10.5281/zenodo.1400175 (cit. on p. 136).

Löwel, Siegrid and Wolf Singer (1992). "Selection of intrinsic horizontal connections in the visual cortex by correlated neuronal activity." In: *Science* 255.5041, pp. 209–212. DOI: 10.1126/science.1372754 (cit. on p. 5).

Lührs, Sebastian et al. (2016). "Flexible and Generic Workflow Management." In: *Parallel Computing: On the Road to Exascale*. Vol. 27. Advances in parallel computing. Amsterdam: IOS Press, pp. 431–438. DOI: 10.3233/978-1-61499-621-7-431 (cit. on pp. 124, 135).

Lund, J.S., T. Yoshioka, and J.B. Levitt (1993). "Comparison of Intrinsic Connectivity in Different Areas of Macaque Monkey Cerebral Cortex." In: *Cereb. Cortex* 3.2, pp. 148–162 (cit. on p. 63).

MICrONS Consortium et al. (2021). "Functional connectomics spanning multiple areas of mouse visual cortex." In: *BioRxiv*, pp. 2021–07. DOI: 10.1101/2021.07.28.454025 (cit. on pp. 103, 107, 109, 113, 169).

Major, G, ME Larkum, and J Schiller (2013). "Active Properties of Neocortical Pyramidal Neuron Dendrites." In: *Annu. Rev. Neurosci.* 36, pp. 1–24 (cit. on p. 168).

Maksimov, A., M. Diesmann, and S. J. van Albada (2018). "Criteria on balance, stability, and excitability in cortical networks for constraining computational models." In: *Front. Comput. Neurosci.* 12, p. 44. DOI: 10.3389/fncom.2018.00044 (cit. on pp. 45, 50, 114).

Mante, Valerio et al. (2013). "Context-dependent computation by recurrent dynamics in prefrontal cortex." In: *Nature* 503.7474, pp. 78–84 (cit. on p. 18).

Markov, N. T. et al. (2011). "Weight Consistency Specifies Regularities of Macaque Cortical Networks." In: *Cereb. Cortex* 21.6, pp. 1254–1272. DOI: 10.1093/cercor/bhq201 (cit. on pp. 62, 72, 77, 93, 103, 109).

Markov, N. T. et al. (2014a). "A Weighted and Directed Interareal Connectivity Matrix for Macaque Cerebral Cortex." In: *Cereb. Cortex* 24.1, pp. 17–36. DOI: 10.1093/cercor/bhs270 (cit. on p. 3).

Markov, Nikola T and Henry Kennedy (2013). "The importance of being hierarchical." In: *Curr. Opin. Neurobiol.* 23.2, pp. 187–194. ISSN: 0959-4388. DOI: 10.1016/j.conb.2012.12.008 (cit. on p. 104).

Markov, Nikola T. et al. (2013). "Cortical High-Density Counterstream Architectures." In: *Science* 342.6158. DOI: 10.1126/science.1238406. eprint: http://www.sciencemag.org/content/342/6158/1238406.full.pdf (cit. on p. 65).

Markov, Nikola T. et al. (2014b). "Anatomy of hierarchy: Feedforward and feedback pathways in macaque visual cortex." In: *J. Comp. Neurol.* 522.1, pp. 225–259. ISSN: 1096-9861. DOI: 10.1002/cne.23458 (cit. on p. 41).

Markram, Henry et al. (Oct. 2015). "Reconstruction and simulation of neocortical microcircuitry." In: *Cell* 163.2, pp. 456–492. DOI: 10.1016/j.cell.2015.09.029 (cit. on pp. 6, 9).

Martinez, Luis M et al. (2005). "Receptive field structure varies with layer in the primary visual cortex." In: *Nat. Neurosci.* 8.3, pp. 372–379. DOI: 10.1038/nn1404 (cit. on p. 98).

Mattson, Peter et al. (2019). *MLPerf Training Benchmark*. arXiv: 1910.01500 [cs.LG] (cit. on p. 135).

Maunsell, John HR and William T Newsome (1987). "Visual processing in monkey extrastriate cortex." In: *Annu. Rev. Neurosci.* 10.1, pp. 363–401 (cit. on p. 63).

Mazzucato, Luca, Alfredo Fontanini, and Giancarlo La Camera (2016). "Stimuli reduce the dimensionality of cortical activity." In: *Front. Syst. Neurosci.* 10, p. 11. DOI: 10.3389/fnsys.2016.00011 (cit. on pp. 19, 24).

Mejias, Jorge F et al. (2016). "Feedforward and feedback frequency-dependent interactions in a large-scale laminar network of the primate cortex." In: *Sci. Adv.* 2.11, e1601335. DOI: 10.1126/sciadv.1601335 (cit. on p. 6).

Mendoza-Halliday, Diego et al. (Oct. 2022). "A ubiquitous spectrolaminar motif of local field potential power across the primate cortex." en. In: *BioRxiv*. DOI: 10.1101/2022.09.30.510398 (cit. on p. 41).

Message Passing Interface Forum (2009). *MPI: A Message-Passing Interface Standard, Version 2.2*. Tech. rep. Knoxville, Tennessee (cit. on p. 145).

Miyazaki, Hiroyuki et al. (2012). "Overview of the K computer System." In: *Fujitsu Sci. Tech. J.* 48.3, pp. 255–265 (cit. on pp. 155, 156).

Monteforte, Michael and Fred Wolf (2010). "Dynamical entropy production in spiking neuron networks in the balanced state." In: *Phys.*

Rev. Lett. 105.26, p. 268104. DOI: 10.1103/PhysRevLett.105.268104 (cit. on p. 134).

– (2012). "Dynamic Flux Tubes Form Reservoirs of Stability in Neuronal Circuits." In: *Phys. Rev. X* 2.4, p. 041007. ISSN: 2160-3308. DOI: 10.1103/PhysRevX.2.041007 (cit. on p. 50).

Morales-Gregorio, Aitor et al. (2023). "Neural manifolds in V1 change with top-down signals from V4 targeting the foveal region." In: *BioRxiv*, pp. 2023–06. DOI: https://doi.org/10.1101/2023.06.14.544966 (cit. on p. vii).

Morrison, Abigail et al. (2003). "Distributed Simulation of Large Biological Neural Networks." In: *Proceedings of the 29th Göttingen Neurobiology Conference*, p. 590 (cit. on p. 10).

Morrison, Abigail et al. (2005a). "Advancing the boundaries of high connectivity network simulation with distributed computing." In: *Neural Comput.* 17.8, pp. 1776–1801. DOI: 10.1162/0899766054026648 (cit. on p. 124).

Morrison, Abigail et al. (2005b). "Precise spike timing with exact subthreshold integration in discrete time network simulations." In: *Proceedings of the 30th Göttingen Neurobiology Conference*, 205B (cit. on p. 139).

Mountcastle, Vernon B. (July 1957). "Modality and topographic properties of single neurons in cat's somatic sensory cortex." In: *J. Neurophysiol.* 20.4, pp. 408–434. DOI: 10.1152/jn.1957.20.4.408 (cit. on p. 62).

Naumann, Laura B, Joram Keijser, and Henning Sprekeler (2022). "Invariant neural subspaces maintained by feedback modulation." en. In: *eLife* 11, e76096. ISSN: 2050-084X. DOI: 10.7554/eLife.76096 (cit. on pp. 18, 168).

Nordlie, Eilen, Marc-Oliver Gewaltig, and Hans Ekkehard Plesser (Aug. 2009a). "Towards Reproducible Descriptions of Neuronal Network Models." In: *PLOS Comput. Biol.* 5.8, e1000456. DOI: 10.1371/journal.pcbi.1000456 (cit. on p. 51).

Nordlie, Eilen, Hans Ekkehard Plesser, and Marc-Oliver Gewaltig (2009b). "Towards reproducible descriptions of neuronal network models." In: doi:10.3389/conf.neuro.11.2008.01.086 (cit. on p. 87).

O'Kusky, J. and M. Colonnier (1982). "A laminar analysis of the number of neurons, glia, and synapses in the visual cortex (area 17) of adult macaque monkeys." In: *J. Comp. Neurol.* 210.3, pp. 278–290 (cit. on p. 66).

Ohki, Kenichi and R. Clay Reid (2007). "Specificity and randomness in the visual cortex." In: *Curr. Opin. Neurobiol.* 17.4, pp. 401–407. DOI: 10.1016/j.conb.2007.07.007 (cit. on p. 65).

OpenMP Architecture Review Board (2008). *OpenMP Application Program Interface*. http://www.openmp.org/mp-documents/spec30.pdf. Specification. Accessed: 2016-09-27 (cit. on pp. 122, 145).

Ostojic, Srdjan (Feb. 2014). “Two types of asynchronous activity in networks of excitatory and inhibitory spiking neurons.” In: *Nat. Neurosci.* 17, pp. 594–600. DOI: 10.1038/nn.3658 (cit. on p. 45).

Ostrau, Christoph et al. (Mar. 2020). “Benchmarking of Neuromorphic Hardware Systems.” In: *Proceedings of the Neuro-Inspired Computational Elements Workshop*. Vol. 16. NICE ’20. Heidelberg, Germany: Association for Computing Machinery, p. 4. ISBN: 9781450377188. DOI: 10.1145/3381755.3381772 (cit. on p. 135).

Packer, Adam M. and Rafael Yuste (Sept. 2011). “Dense, Unspecific Connectivity of Neocortical Parvalbumin-Positive Interneurons: A Canonical Microcircuit for Inhibition?” In: *J. Neurosci.* 31.37, pp. 13260–13271. DOI: 10.1523/JNEUROSCI.3131-11.2011. eprint: http://www.jneurosci.org/content/31/37/13260.full.pdf+html (cit. on pp. 78, 79, 95, 183).

Pang, Rich, Benjamin J. Lansdell, and Adrienne L. Fairhall (July 2016). “Dimensionality reduction in neuroscience.” en. In: *Curr. Biol.* 26.14, R656–R660. ISSN: 09609822. DOI: 10.1016/j.cub.2016.05.029 (cit. on p. 19).

Pauli, Robin et al. (2018). “Reproducing polychronization: a guide to maximizing the reproducibility of spiking network models.” In: *Front. Neuroinform.* 12.46. DOI: 10.3389/fninf.2018.00046 (cit. on p. 117).

Peng, Yangfan et al. (2021). “Spatially structured inhibition defined by polarized parvalbumin interneuron axons promotes head direction tuning.” In: *Sci. Adv.* 7.25, eabg4693. DOI: 10.1126/sciadv.abg4693 (cit. on p. 62).

Perin, Rodrigo, Thomas K. Berger, and Henry Markram (Mar. 2011). “A synaptic organizing principle for cortical neuronal groups.” In: *Proc. Natl. Acad. Sci. USA* 108.13, pp. 5419–5424. DOI: 10.1073/pnas.1016051108 (cit. on pp. 62, 169, 183).

Peyser, Alexander et al. (2021). *NEST 2.14.1*. DOI: 10.5281/ZENODO.4018724 (cit. on p. 121).

Pfeil, Thomas et al. (2013). “Six networks on a universal neuromorphic computing substrate.” In: *Front. Neurosci.* 7.11, p. 11. DOI: 10.3389/fnins.2013.00011 (cit. on p. 135).

Poort, Jasper et al. (July 2012). “The Role of Attention in Figure-Ground Segregation in Areas V1 and V4 of the Visual Cortex.” en. In: *Neuron* 75.1, pp. 143–156. ISSN: 08966273. DOI: 10.1016/j.neuron.2012.04.032 (cit. on p. 18).

Potjans, Tobias C. and Markus Diesmann (Mar. 2014). “The Cell-Type Specific Cortical Microcircuit: Relating Structure and Activity in a Full-Scale Spiking Network Model.” In: *Cereb. Cortex* 24.3, pp. 785–806. DOI: 10.1093/cercor/bhs358 (cit. on pp. 6, 9, 64, 65, 86, 87, 91, 92, 99, 100, 103, 106, 110, 112, 113, 120, 128, 134, 146, 158, 169).

Pronold, Jari et al. (2021). *Routing brain traffic through the von Neumann bottleneck: Efficient cache usage in spiking neural network simulation* *code on general purpose computers*. arXiv: 2109.12855 [cs.DC] (cit. on pp. 129, 160).

Quaglio, Pietro et al. (2017). "Detection and evaluation of spatio-temporal spike patterns in massively parallel spike train data with spade." In: *Front. Comput. Neurosci.* 11, p. 41 (cit. on p. 6).

Ramón y Cajal, Santiago (1888). "Estructura de los centros nerviosos de las aves." In: *Rev. Trim. Histol. Norm. Pat.* 1, pp. 1–10 (cit. on p. 3).

Rhodes, Oliver et al. (2018). "sPyNNaker: A Software Package for Running PyNN Simulations on SpiNNaker." In: *Front. Neurosci.* 12. DOI: 10.3389/fnins.2018.00816 (cit. on p. 117).

Rhodes, Oliver et al. (Dec. 2019). "Real-time cortical simulation on neuromorphic hardware." In: *Philos. Trans. R. Soc. A* 378.2164, p. 20190160. DOI: 10.1098/rsta.2019.0160 (cit. on pp. 120, 122, 128, 145, 158).

Riehle, Alexa et al. (1997). "Spike Synchronization and Rate Modulation Differentially Involved in Motor Cortical Function." In: *Science* 278, pp. 1950–1953 (cit. on p. 6).

Rieke, Fred et al. (1997). *Spikes: Exploring the Neural Code*. Cambridge, MA: The MIT Press. ISBN: 0-262-18174-6 (cit. on p. 6).

Rockland, K. S. and J. S. Lund (1982). "Widespread periodic intrinsic connections in the tree shrew visual cortex." In: *Science* 215, pp. 1532–1534 (cit. on p. 62).

Romo, Ranulfo et al. (1999). "Neuronal correlates of parametric working memory in the prefrontal cortex." In: *Nature* 399, pp. 470–473 (cit. on p. 45).

Rosenbaum, Robert and Brent Doiron (2014). "Balanced Networks of Spiking Neurons with Spatially Dependent Recurrent Connections." In: *Phys. Rev. X* 4.2, p. 021039. DOI: 10.1103/PhysRevX.4.021039 (cit. on p. 50).

Rostami, Vahid et al. (2020). "Spiking neural network model of motor cortex with joint excitatory and inhibitory clusters reflects task uncertainty, reaction times, and variability dynamics." In: *BioRxiv* (cit. on p. 104).

Rostami, Vahid et al. (2022). "Excitatory and inhibitory motor cortical clusters account for balance, variability, and task performance." In: *BioRxiv*. DOI: 10.1101/2020.02.27.968339. eprint: https://www.biorxiv.org/content/early/2022/02/08/2020.02.27.968339.full.pdf (cit. on pp. 59, 168).

Sadeh, Sadra and Stefan Rotter (Nov. 2013). "Statistics and geometry of orientation selectivity in primary visual cortex." In: *Biol. Cybern.* 108.5, pp. 631–653. DOI: 10.1007/s00422-013-0576-0 (cit. on p. 81).

Schaub, Michael T. et al. (2015). "Emergence of slow-switching assemblies in structured neuronal networks." In: *PLOS Comput. Biol.* 11.7, e1004196. DOI: 10.1371/journal.pcbi.1004196 (cit. on pp. 50, 59).

Schemmel, Johannes et al. (May 2010). "A Wafer-Scale Neuromorphic Hardware System for Large-Scale Neural Modeling." In: *Proceedings of the 2010 International Symposium on Circuits and Systems (ISCAS)*.

Paris: IEEE Press, pp. 1947–1950. DOI: 10.1109/iscas.2010.5536970 (cit. on p. 135).

Schiller, J. et al. (2000). "NMDA spikes in basal dendrites of cortical pyramidal neurons." eng. In: *Nature* 404.6775, pp. 285–289. DOI: 10.1038/35005094 (cit. on p. 168).

Schmidt, Maximilian et al. (2018a). "A multi-scale layer-resolved spiking network model of resting-state dynamics in macaque visual cortical areas." In: *PLOS Comput. Biol.* 14.10, e1006359. DOI: 10.1371/journal.pcbi.1006359 (cit. on pp. 6, 146).

Schmidt, Maximilian et al. (Apr. 2018b). "Multi-scale account of the network structure of macaque visual cortex." In: *Brain Struct. Funct.* 223.3, pp. 1409–1435. ISSN: 1863-2661. DOI: 10.1007/s00429-017-1554-4 (cit. on pp. 7, 9, 64–66, 86, 91, 92, 100, 103, 104, 134, 146, 156, 158, 170).

Schnepel, Philipp et al. (Oct. 2015). "Physiology and Impact of Horizontal Connections in Rat Neocortex." In: *Cereb. Cortex* 25.10, pp. 3818–3835. DOI: 10.1093/cercor/bhu265 (cit. on p. 103).

Schulz, David PA, Maneesh Sahani, and Matteo Carandini (2015). "Five key factors determining pairwise correlations in visual cortex." In: *J. Neurophysiol.* 114.2, pp. 1022–1033. DOI: 10.1152/jn.00094.2015 (cit. on p. 45).

Sclar, Gary and RD Freeman (1982). "Orientation selectivity in the cat's striate cortex is invariant with stimulus contrast." In: *Exp. Brain Res.* 46.3, pp. 457–461. DOI: 10.1007/BF00238641 (cit. on p. 83).

Segundo, JP et al. (1963). "Sensitivity of neurones in Aplysia to temporal pattern of arriving impulses." In: *Journal of Experimental Biology* 40.4, pp. 643–667. DOI: 10.1242/jeb.40.4.643 (cit. on p. 4).

Self, Matthew W. et al. (2013). "Distinct Roles of the Cortical Layers of Area V1 in Figure-Ground Segregation." en. In: *Curr. Biol.* 23.21, pp. 2121–2129. ISSN: 09609822. DOI: 10.1016/j.cub.2013.09.013 (cit. on pp. 18, 59, 168).

Semedo, João D. et al. (Dec. 2022). "Feedforward and feedback interactions between visual cortical areas use different population activity patterns." en. In: *Nat. Commun.* 13.1, p. 1099. ISSN: 2041-1723. DOI: 10.1038/s41467-022-28552-w (cit. on pp. 41, 167).

Senk, Johanna et al. (2017). "A Collaborative Simulation-Analysis Workflow for Computational Neuroscience Using HPC." In: *High-Performance Scientific Computing. JHPCS 2016.* Ed. by Edoardo Di Napoli et al. Vol. 10164. Lecture Notes in Computer Science. Springer, Cham, pp. 243–256. DOI: 10.1007/978-3-319-53862-4_21 (cit. on p. 134).

Senk, Johanna et al. (2022). "Connectivity Concepts in Neuronal Network Modeling." In: *PLOS Comput. Biol.* 18.9, e1010086. DOI: 10.1371/journal.pcbi.1010086 (cit. on p. 77).

Senk, Johanna et al. (2023). "Reconciliation of weak pairwise spike-train correlations and highly coherent local field potentials across space." In: *ArXiv*, 1805.10235v1 [q–bio.NC]. DOI: 10.48550/arXiv.1805.10235 (cit. on pp. 6, 9, 64, 65, 87, 99, 100, 102–104, 170).

Shapson-Coe, Alexander et al. (2021). "A connectomic study of a petascale fragment of human cerebral cortex." In: *BioRxiv*. DOI: 10.1101/2021.05.29.446289 (cit. on pp. 7, 103, 106–108, 111, 113, 169).

Sherrington, C. S. (1906). "Observations on the scratch-reflex in the spinal dog." In: *J. Physiol.* 34.1-2, pp. 1–50. DOI: 10.1113/jphysiol.1906.sp001139 (cit. on p. 4).

Siegle, Joshua H. et al. (2021). "Survey of spiking in the mouse visual system reveals functional hierarchy." In: *Nature* 592.7852, pp. 86–92. DOI: 10.1038/s41586-020-03171-x (cit. on p. 114).

Sincich, L. C. and G. G. Blasdel (2001). "Oriented axon projections in primary visual cortex of the monkey." In: *J. Neurosci.* 21.12, pp. 4416–4426 (cit. on pp. 79, 80, 94, 169, 184).

Sincich, Lawrence C and Jonathan C Horton (2005). "The circuitry of V1 and V2: integration of color, form, and motion." In: *Annu. Rev. Neurosci.* 28, pp. 303–326. DOI: 10.1146/annurev.neuro.28.061604.135731 (cit. on p. 63).

Singh, Gurjeet et al. (2008). "Topological analysis of population activity in visual cortex." en. In: *Journal of Vision* 8.8. ISSN: 1534-7362. DOI: 10.1167/8.8.11 (cit. on pp. 18, 19).

Skottun, Bernt C et al. (1987). "The effects of contrast on visual orientation and spatial frequency discrimination: a comparison of single cells and behavior." In: *J. Neurophysiol.* 57.3, pp. 773–786 (cit. on p. 83).

Snyder, Abraham Z. and Marcus E. Raichle (Aug. 2012). "A brief history of the resting state: The Washington University perspective." In: *Neuroimage* 62.2, pp. 902–910. DOI: 10.1016/j.neuroimage.2012.01.044 (cit. on p. 15).

Sompolinsky, H., A. Crisanti, and H. J. Sommers (1988). "Chaos in Random Neural Networks." In: *Phys. Rev. Lett.* 61 (3), pp. 259–262. DOI: 10.1103/PhysRevLett.61.259 (cit. on p. 134).

Spreizer, Sebastian et al. (2022). *NEST 3.3*. DOI: 10.5281/ZENODO.6368023 (cit. on p. 52).

Stepanyants, A. et al. (2008). "Local potential connectivity in cat primary visual cortex." In: *Cereb. Cortex* 18.1, pp. 13–28 (cit. on p. 114).

Stepanyants, Armen et al. (Apr. 2007). "Local Potential Connectivity in Cat Primary Visual Cortex." In: *Cereb. Cortex* 18.1, pp. 13–28. DOI: 10.1093/cercor/bhm027 (cit. on pp. 78, 93).

Stettler, D.D. et al. (2002). "Lateral Connectivity and Contextual Interactions in Macaque Primary Visual Cortex." In: *Neuron* 36, pp. 739–750 (cit. on pp. 63, 80, 82).

Stringer, Carsen et al. (2019a). "High-dimensional geometry of population responses in visual cortex." In: *Nature* 571.7765, pp. 361–365 (cit. on pp. 18, 19, 32, 40, 167).

Stringer, Carsen et al. (2019b). "Spontaneous behaviors drive multidimensional, brainwide activity." In: *Science* 364.6437, eaav7893. DOI: 10.1126/science.aav7893 (cit. on p. 40).

Strubell, Emma, Ananya Ganesh, and Andrew McCallum (2019). "Energy and Policy Considerations for Deep Learning in NLP." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 3645–3650. DOI: 10.18653/v1/p19-1355 (cit. on p. 120).

Supèr, Hans and Pieter R. Roelfsema (2005). "Chronic multiunit recordings in behaving animals: Advantages and limitations." In: *Progress in Brain Research*. Vol. 147. SPEC. ISS. Elsevier, pp. 263–282. DOI: 10.1016/S0079-6123(04)47020-4 (cit. on pp. 20, 27).

Talluri, Bharath C. et al. (2022). "Activity in primate visual cortex is minimally driven by spontaneous movements." In: *BioRxiv*. DOI: 10.1101/2022.09.08.507006 (cit. on pp. 27, 40).

Tanigawa, Hisashi, QuanXin Wang, and Ichiro Fujita (Dec. 2005). "Organization of horizontal axons in the inferior temporal cortex and primary visual cortex of the macaque monkey." In: *Cereb. Cortex* 15.12, pp. 1887–1899. DOI: 10.1093/cercor/bhi067 (cit. on pp. 63, 114).

Tartaglia, Elisa M and Nicolas Brunel (2017). "Bistability and up/down state alternations in inhibition-dominated randomly connected networks of LIF neurons." In: *Sci. Rep.* 7.1, p. 11916 (cit. on p. 45).

Tenenbaum, Joshua B., Vin de Silva, and John C. Langford (2000). "A Global Geometric Framework for Nonlinear Dimensionality Reduction." In: *Science* 290.5500, pp. 2319–2323 (cit. on p. 23).

Tetzlaff, Tom et al. (2012). "Decorrelation of Neural-Network Activity by Inhibitory Feedback." In: *PLOS Comput. Biol.* 8.8. Ed. by Nicolas Brunel, e1002596. DOI: 10.1371/journal.pcbi.1002596 (cit. on pp. 59, 168).

Thomson, David J. (1982). "Spectrum estimation and harmonic analysis." In: *Proc. IEEE* 70.9, pp. 1055–1096. DOI: 10.1109/proc.1982.12433 (cit. on p. 25).

Thörnig, Philipp and Benedikt von St. Vieth (2021). "JURECA: Data Centric and Booster Modules implementing the Modular Supercomputing Architecture at Jülich Supercomputing Centre." In: *JLSRF* 7, A182. ISSN: 2364-091X. DOI: 10.17815/jlsrf-7-182 (cit. on pp. 99, 135, 155, 156).

Tiddia, Gianmarco et al. (July 2022). "Fast Simulation of a Multi-Area Spiking Network Model of Macaque Cortex on an MPI-GPU Cluster." In: *Front. Neuroinform.* 16, p. 883333. DOI: 10.3389/fninf.2022.883333 (cit. on p. 133).

Tootell, Roger BH et al. (1982). "Deoxyglucose analysis of retinotopic organization in primate striate cortex." In: *Science* 218.4575, pp. 902–904. DOI: 10.1126/science.713498 (cit. on p. 63).

Traub, Roger D et al. (2005). “Single-column thalamocortical network model exhibiting gamma oscillations, sleep spindles, and epileptogenic bursts.” In: *J. Neurophysiol.* 93.4, pp. 2194–2232. DOI: 10.1152/jn.00983.2004 (cit. on pp. 6, 46).

Trepka, Ethan B et al. (2022). “Functional interactions among neurons within single columns of macaque V1.” In: *eLife* 11, e79322. DOI: 10.7554/eLife.79322 (cit. on p. 63).

Troyer, Todd W et al. (1998). “Contrast-invariant orientation tuning in cat visual cortex: thalamocortical input tuning and correlation-based intracortical connectivity.” In: *J. Neurosci.* 18.15, pp. 5908–5927. DOI: 10.1523/JNEUROSCI.18-15-05908.1998 (cit. on pp. 84, 98).

Van Hooser, S. D. et al. (2006). “Lack of patchy horizontal connectivity in primary visual cortex of a mammal without orientation maps.” In: *J. Neurosci.* 26.29, pp. 7680–7692 (cit. on p. 63).

Van Loon, Anouk M. et al. (May 2016). “NMDA Receptor Antagonist Ketamine Distorts Object Recognition by Reducing Feedback to Early Visual Cortex.” en. In: *Cereb. Cortex* 26.5, pp. 1986–1996. ISSN: 1047-3211, 1460-2199. DOI: 10.1093/cercor/bhv018 (cit. on pp. 59, 168).

Van Albada, Sacha J. et al. (May 2018). “Performance Comparison of the Digital Neuromorphic Hardware SpiNNaker and the Neural Network Simulation Software NEST for a Full-Scale Cortical Microcircuit Model.” In: *Front. Neurosci.* 12, p. 291. DOI: 10.3389/fnins.2018.00291 (cit. on pp. 120, 128, 129, 134).

Van Albada, Sacha J. et al. (2021). “Usage and Scaling of an Open-Source Spiking Multi-Area Model of Monkey Cortex.” In: *Lecture Notes in Computer Science*. Springer International Publishing, pp. 47–59. DOI: 10.1007/978-3-030-82427-3_4 (cit. on p. 120).

Van Albada, Sacha Jennifer, Moritz Helias, and Markus Diesmann (2015a). “Scalability of Asynchronous Networks Is Limited by One-to-One Mapping between Effective Connectivity and Correlations.” In: *PLOS Comput. Biol.* 11.9. Ed. by Peter E. Latham, e1004490. DOI: 10.1371/journal.pcbi.1004490 (cit. on pp. 8, 45, 120).

Van Albada, Sacha et al. (2015b). *NEST 2.2.0*. DOI: 10.5281/zenodo.5772624 (cit. on p. 136).

Van Kerkoerle, Timo, Matthew W. Self, and Pieter R. Roelfsema (2017). “Layer-specificity in the effects of attention and working memory on activity in primary visual cortex.” en. In: *Nat. Commun.* 8.1. ISSN: 2041-1723. DOI: 10.1038/ncomms13804 (cit. on p. 18).

Van Kerkoerle, Timo et al. (Sept. 2014). “Alpha and gamma oscillations characterize feedback and feedforward processing in monkey visual cortex.” In: *Proc. Natl. Acad. Sci. USA* 111.40, pp. 14332–14341. DOI: 10.1073/pnas.1402773111 (cit. on pp. 18, 41).

Van Meegen, Alexander and Sacha J. van Albada (2021). “Microscopic theory of intrinsic timescales in spiking neural networks.” In: *Phys.*

Rev. Res. 3 (4), p. 043077. DOI: 10.1103/PhysRevResearch.3.043077 (cit. on p. 45).

Van Vreeswijk, C. and H. Sompolinsky (1998). "Chaotic Balanced State in a Model of Cortical Circuits." In: *Neural Comput.* 10.6, pp. 1321–1371. DOI: 10.1162/089976698300017214 (cit. on pp. 99, 134).

Van Vreeswijk, Carl and Haim Sompolinsky (Dec. 1996). "Chaos in Neuronal Networks with Balanced Excitatory and Inhibitory Activity." In: *Science* 274, pp. 1724–1726. DOI: 10.1126/science.274.5293.1724 (cit. on p. 50).

Vanni, Simo et al. (2020). "Anatomy and Physiology of Macaque Visual Cortical Areas V1, V2, and V5/MT: Bases for Biologically Realistic Models." In: *Cereb. Cortex* 30.6, pp. 3483–3517. DOI: 10.1093/cercor/bhz322 (cit. on pp. 63, 66, 71, 72, 75, 103, 106, 169).

Vezoli, Julien et al. (Dec. 2021). "Brain rhythms define distinct interaction networks with differential dependence on anatomy." en. In: *Neuron* 109.23, pp. 3862–3878. ISSN: 08966273. DOI: 10.1016/j.neuron.2021.09.052 (cit. on pp. 34, 41, 167).

Vogels, T.P. et al. (2011). "Inhibitory plasticity balances excitation and inhibition in sensory pathways and memory networks." In: *Science* 334.6062, pp. 1569–1573. DOI: 10.1126/science.1211095 (cit. on p. 103).

Voges, Nicole et al. (Nov. 2010). "A modeler's view on the spatial structure of intrinsic horizontal connectivity in the neocortex." In: *Prog. Neurobiol.* 92.3, pp. 277–292. DOI: 10.1016/j.pneurobio.2010.05.001 (cit. on pp. 62, 63).

Wang, Mingli et al. (2022). "Retinotopic organization of feedback projections in primate early visual cortex: implications for active vision." In: *BioRxiv*. DOI: 10.1101/2022.04.27.489651 (cit. on pp. 34, 35, 41).

Wang, Tian et al. (2020). "Laminar subnetworks of response suppression in macaque primary visual cortex." In: *J. Neurosci.* 40.39, pp. 7436–7450. DOI: 10.1523/JNEUROSCI.1129-20.2020 (cit. on p. 91).

Wildenberg, Gregg A et al. (2021). "Primate neuronal connections are sparse in cortex as compared to mouse." In: *Cell Rep.* 36.11. DOI: https://doi.org/10.1016/j.celrep.2021.109709 (cit. on p. 110).

Williamson, Ryan C. et al. (Dec. 2016). "Scaling Properties of Dimensionality Reduction for Neural Populations and Network Models." en. In: *PLOS Comput. Biol.* 12.12. Ed. by Jonathan W. Pillow, e1005141. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1005141 (cit. on p. 19).

Wilson, G. Tunnicliffe (1972). "The Factorization of Matricial Spectral Densities." en. In: *SIAM J. Appl. Math.* 23.4, pp. 420–426. ISSN: 0036-1399, 1095-712X. DOI: 10.1137/0123044 (cit. on p. 25).

Xue, Mingshan, Bassam V Atallah, and Massimo Scanziani (2014). "Equalizing excitation-inhibition ratios across visual cortical neurons." eng. In: *Nature* 511.7511, pp. 596–600. DOI: 10.1038/nature13321 (cit. on p. 103).

Yavuz, Esin, James Turner, and Thomas Nowotny (Jan. 2016). "GeNN: a code generation framework for accelerated brain simulations." In: *Sci. Rep.* 6.1. DOI: 10.1038/srep18854 (cit. on p. 133).

Yger, Pierre et al. (2018). "A spike sorting toolbox for up to thousands of electrodes validated with ground truth recordings in vitro and in vivo." In: *eLife* 7, e34518. DOI: 10.7554/eLife.34518 (cit. on p. 21).

Yoo, Andy B., Morris A. Jette, and Mark Grondona (2003). "SLURM: Simple Linux Utility for Resource Management." In: *Job Scheduling Strategies for Parallel Processing*. Ed. by Dror Feitelson, Larry Rudolph, and Uwe Schwiegelshohn. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 44–60. ISBN: 978-3-540-39727-4. DOI: 10.1007/10968987_3 (cit. on p. 149).

Yu, Byron M. et al. (2009). "Gaussian-Process Factor Analysis for Low-Dimensional Single-Trial Analysis of Neural Population Activity." In: *J. Neurophysiol.* 102.1, pp. 614–635. DOI: 10.1152/jn.90941.2008 (cit. on p. 6).

Yuste, Rafael (2015). "From the neuron doctrine to neural networks." In: *Nat. Rev. Neurosci.* 16.8, pp. 487–497 (cit. on p. 3).

Zaytsev, Y. V. and A. Morrison (2014). "CyNEST: a maintainable Cython-based interface for the NEST simulator." In: *Front. Neuroinform.* 8.23. DOI: 10.3389/fninf.2014.00023 (cit. on p. 145).

Zeki, Semir and Stewart Shipp (1988). "The functional logic of cortical connections." In: *Nature* 335.6188, pp. 311–317 (cit. on pp. 104, 170).

Znamenskiy, Petr et al. (2018). "Functional selectivity and specific connectivity of inhibitory neurons in primary visual cortex." In: *BioRxiv*. DOI: 10.1101/294835. eprint: https://www.biorxiv.org/content/early/2018/04/04/294835.1.full.pdf (cit. on p. 103).

Schriften des Forschungszentrums Jülich
Reihe Information

Band / Volume 93
DC and RF characterization of bulk CMOS and FD-SOI devices at cryogenic temperatures with respect to quantum computing applications
A. Artanov (2023), xv, 80, xvii-liii pp
ISBN: 978-3-95806-687-8

Band / Volume 94
HAXPES study of interface and bulk chemistry of ferroelectric HfO_2 capacitors
T. Szyjka (2023), viii, 120 pp
ISBN: 978-3-95806-692-2

Band / Volume 95
A brain inspired sequence learning algorithm and foundations of a memristive hardware implementation
Y. Bouhadjar (2023), xii, 149 pp
ISBN: 978-3-95806-693-9

Band / Volume 96
Characterization and modeling of primate cortical anatomy and activity
A. Morales-Gregorio (2023), ca. 260 pp.
ISBN: 978-3-95806-698-4

Band / Volume 97
Hafnium oxide based memristive devices as functional elements of neuromorphic circuits
F. J. Cüppers (2023), vi, ii, 214 pp
ISBN: 978-3-95806-702-8

Band / Volume 98
Simulation and theory of large-scale cortical networks
A. van Meegen (2023), ca. 250 pp
ISBN: 978-3-95806-708-0

Band / Volume 99
Structure of two-dimensional multilayers and topological superconductors: surfactant mediated growth, intercalation, and doping
Y.-R. Lin (2023), x, 111 pp
ISBN: 978-3-95806-716-5

Band / Volume 100
Frequency mixing magnetic detection for characterization and multiplex detection of superparamagnetic nanoparticles
A. M. Pourshahidi (2023), X, 149 pp
ISBN: 978-3-95806-727-1

Band / Volume 101
Unveiling the relaxation dynamics of Ag/HfO2 based diffusive memristors for use in neuromorphic computing
S. A. Chekol (2023), x, 185 pp
ISBN: 978-3-95806-729-5

Band / Volume 102
Analysis and quantitative comparison of neural network dynamics on a neuron-wise and population level
R. Gutzen (2024), xii, 252 pp
ISBN: 978-3-95806-738-7

Band / Volume 103
3D Scaffolds with Integrated Electrodes for Neuronal Cell Culture
J. Abu Shihada (2024), vii, 163 pp
ISBN: 978-3-95806-756-1

Band / Volume 104
Advances in Photoemission Orbital Tomography
A. Haags (2024), ix, 254 pp
ISBN: 978-3-95806-766-0

Band / Volume 105
Quantitative investigation of point defects and their dynamics in focused ion beam-prepared group III-nitride lamellas by off-axis electron holography
K. Ji (2024), 164 pp
ISBN: 978-3-95806-782-0

Band / Volume 106
NeuCoNS and Stacked-Net: Facilitating the Communication for Accelerated Neuroscientific Simulations
R. Kleijnen (2024), xx, 110, xxi-xxxiv pp
ISBN: 978-3-95806-788-2

Band / Volume 107
Construction of a Spiking Network Model of Macaque Primary Visual Cortex: Towards Digital Twins
A. Kurth (2024), xvi, 207 pp
ISBN: 978-3-95806-800-1

Weitere ***Schriften des Verlags im Forschungszentrum Jülich*** unter
http://wwwzb1.fz-juelich.de/verlagextern1/index.asp

Information
Band / Volume 107
ISBN 978-3-95806-800-1

Mitglied der Helmholtz-Gemeinschaft

JÜLICH
Forschungszentrum