SPATIAL CONNECTIVITY PATTERNS SHAPE DYNAMICS OF LARGE-SCALE SPIKING NEURAL NETWORKS OF VISUAL CORTEX

von der Fakultät für Mathematik, Informatik und Naturwissenschaften der RWTH Aachen University zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Jasper Albers, M.Sc.

aus

Hagen

Berichter: Prof. Dr. Markus Diesmann
Prof. Dr. Andreas Offenhäusser

Tag der mündlichen Prüfung: 17.04.2025

Diese Dissertation ist auf den Internetseiten der Universitätsbibliothek verfügbar.

Jasper Albers: *Spatial connectivity patterns shape dynamics of large-scale spiking neural networks of visual cortex*, October 2024

EIDESSTATTLICHE ERKLÄRUNG

Ich, Jasper Albers,
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.
- Teile dieser Arbeit wurden zuvor veröffentlicht, siehe *Publications*.

Aachen, Oktober 2024

Jasper Albers

Never underestimate the big importance of small things.

— *The Midnight Library*, Matt Haig

SUMMARY

Unraveling the inner workings of the brain has been one of the most fascinating research topics in human society for centuries. Evolving from the early discoveries of neurons by Golgi and Cajal, the field of neuroscience has made significant progress in examining the anatomy, physiology and functionality of the human brain. A driving factor for this research is the explosion in the availability of detailed anatomical and physiological data from experiments over the past decades, as well as advances in the availability of computational resources. Combined, these changes enable ever more precise studies of both brain connectivity and function conducted in the field of Computational Neuroscience.

This thesis aims to further our understanding of the link between structure and dynamics in the brain, which fundamentally underlies its structure-function relationship. Specifically, a focus is set on studying the impact of spatial connectivity, i.e., the distance-dependent wiring principles of neurons, on cortical activity of the visual regions. Choosing a computational approach, detailed anatomical knowledge is integrated into biologically constrained models of cortical circuits, allowing for an investigation of the relationship of structure and dynamics *in silico*.

First, the technological foundations are established that enable efficient simulations of large-scale neural networks at realistic neural and synaptic density. An open-source software platform is conceptualized and implemented that facilitates the automated execution and analysis of performance benchmarks of neural network simulators, which are the backbone of simulations in modern Computational Neuroscience. Thereby, development is aided that optimizes the time-to-solution in such simulations, ultimately allowing to integrate ever more anatomical detail into cortical network models.

Second, this thesis consults a broad range of experimental data to construct a comprehensive spatial neural network model of the primary visual cortex (V1) of macaque monkey. Simulations reveal that the model of V1 exhibits strongly pathological activity: it is unable to exist in a balanced dynamical state in which the statistics of the activity resemble experimentally observed states. Revisiting the underlying connectivity and linking it to the dynamics of a model of local cortex exposes that the inherent recurrent targeting patterns stemming from low-resolution data based on light microscopy (LM) necessarily lead to diverging activity. In contrast, the implausible dynamics are remedied by replacing the LM data set by a recent high-resolution data set based on electron microscopy (EM). Substituting the LM connectivity by

EM connectivity in the model of V1 resolves the unbalanced activity, yielding a model that exhibits biologically plausible activity without further fine-tuning. Thus, the derived model can act as a platform for future research to study the dynamical and functional implications of spatial connectivity, shedding light on optimal wiring strategies employed by the visual cortex in the brain.

Third, a novel predictive connectivity rule is developed that explicitly takes the spatial distributions of synapses relative to the cell bodies into account. To compare the predicted connectivity to an experimental ground truth, a general method for comparing the recurrent connection structure of networks is devised based on matrix decomposition. A subsequent analysis demonstrates that the precise spatial connectivity rule outperforms a classical approach based on the distance between neurons. Thereby, it paves the way for incorporating the intricacies of local spatial connectivity into the next generation of modeling.

Overall, this work provides a thorough account on spatial connectivity in cortical neural networks. It thus contributes to uncovering the structure-dynamics-function triad of the brain, ultimately leading to a deeper understanding of its working mechanisms.

ZUSAMMENFASSUNG

Seit Jahrhunderten ist die Entschlüsselung des menschlichen Gehirns eines der faszinierendsten Forschungsthemen unserer Gesellschaft. Während die Entdeckung der Neuronen und ihrer Struktur durch Golgi und Cajal den Anfang der modernen Neurowissenschaften darstellt, wurden seither bedeutende Fortschritte bei der Untersuchung der Anatomie, Physiologie und Funktionalität des menschlichen Gehirns gemacht. Diese Forschung wurde in den letzten Jahrzehnten entscheidend beschleunigt durch die zunehmende Verfügbarkeit von Rechenressourcen sowie detaillierten anatomischen und physiologischen Daten aus Experimenten. Zusammengenommen ermöglichen diese Veränderungen präzisere Studien der Konnektivität und Funktion des Gehirns, die im Bereich der *Computational Neuroscience* durchgeführt werden.

Ziel dieser Arbeit ist es, unser Verständnis für die Verbindung zwischen Struktur und Dynamik des Gehirn zu vertiefen, welche für den Struktur-Funktions-Zusammenhang entscheidend ist. Dabei wird ein Schwerpunkt auf die Untersuchung von räumlicher Konnektivität gelegt, welche die entfernungsabhängige Verschaltung zwischen Neuronen beschreibt, sowie ihren Einfluss auf die kortikale Aktivität in visuellen Gehirnregionen. Durch die Wahl eines computergestützten Ansatzes wird detailliertes anatomisches Wissen in biologisch motivierte Modelle kortikaler Schaltkreise integriert, was eine Untersuchung der Beziehung zwischen Struktur und Dynamik in silico ermöglicht.

Zunächst werden in dieser Arbeit die technologischen Grundlagen geschaffen, die eine effiziente Simulation großer neuronaler Netzwerke mit realistischer neuronaler und synaptischer Dichte ermöglichen. Es wird eine Open-Source-Softwareplattform konzipiert und umgesetzt, die die automatisierte Ausführung und Analyse von Benchmarks für verschiedene Simulatoren ermöglicht, welche das Rückgrat der modernen *Computational Neurosciences* bilden. Insbesondere unterstützt die Softwareplattform die Entwicklungsarbeit an Simulatoren, welche die Rechenzeit minimiert. Dies ist letztendlich entscheidend für die Integration von immer mehr anatomischen Details in kortikale Netzwerkmodelle.

Im zweiten Teil der Arbeit wird ein breites Spektrum an experimentellen Daten herangezogen, um ein umfassendes räumliches neuronales Netzwerkmodell des primären visuellen Kortex (V1) von Makakenaffen zu erstellen. Simulationen zeigen, dass das Modell von V1 eine starke pathologische Aktivität aufweist: Das Modell ist nicht in der Lage einen stabilen dynamischen Zustand zu erreichen, in dem die simulierte Aktivität experimentell beobachteten Zuständen

ähnelt. Die Überprüfung der zugrundeliegenden Konnektivität und ihre Verknüpfung mit der Dynamik eines Modells des lokalen Kortex zeigt, dass die inhärenten rekurrenten Zielmuster, die aus niedrig aufgelösten, auf Lichtmikroskopie (LM) basierenden Daten stammen, zwangsläufig zu divergierender Aktivität führen. Dies kann behoben werden, indem der LM-Datensatz durch einen neueren, hochauflösenden Datensatz auf Basis der Elektronenmikroskopie (EM) ersetzt wird. Das Ersetzen der LM-Konnektivität durch EM-Konnektivität im Modell von V1 führt zu einer Modellarchitektur, die ohne weiteres *fine tuning* eine biologisch plausible Aktivität zeigt. Somit kann das entwickelte Modell als Plattform für zukünftige Forschungsprojekte dienen, die die dynamischen und funktionellen Auswirkungen der räumlichen Konnektivität untersuchen. Dadurch können die optimalen Strategien der Verschaltung von Neuronen untersucht werden, die im visuellen Kortex des menschlichen Gehirn vorliegen.

Im dritten Teil der Arbeit wird eine neuartige Regel für die Vorhersage der Konnektivität in Modellen entwickelt, die ausdrücklich die räumliche Verteilung der Synapsen relativ zu den Zellkörpern berücksichtigt. Um die vorhergesagte Konnektivität mit einer experimentellen Grundwahrheit zu vergleichen, wird eine allgemeine Methode zum Vergleich der rekurrenten Verbindungsstruktur von Netzwerken entwickelt, die auf Matrixzerlegung basiert. Eine anschließende Analyse zeigt, dass die präzise räumliche Konnektivitätsregel die klassischen Ansätze übertrifft. Damit ebnet sie den Weg für eine neue Generation von Modellen, welche die Feinheiten der lokalen räumlichen Konnektivität des visuellen Kortex verstärkt einbezieht.

Insgesamt stellt diese Arbeit eine umfassende Untersuchung der räumlichen Konnektivität von neuronalen Netzwerken im Kortex dar. Sie trägt somit dazu bei, den Struktur-Dynamik-Funktions-Dreiklang des Gehirns aufzudecken, was letztlich zu einem tieferen Verständnis der grundlegenden Funktionsmechanismen führt.

PUBLICATIONS

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

A Modular Workflow for Performance Benchmarking of Neuronal Network Simulations, by
J. Albers, J. Pronold, A. C. Kurth, S. B. Vennemo, K. H. Mood, A. Patronis, D. Terhorst, J. Jordan, S. Kunkel, T. Tetzlaff, M. Diesmann, J. Senk.
Published in *Frontiers in Neuroinformatics* (Albers et al., 2022).
Results of this publication entered Chapter 3.

Fast simulation of a multi-area spiking network model of macaque cortex on an MPI-GPU cluster, by
G. Tiddia, B. Golosio, J. Albers, J. Senk, F. Simula, J. Pronold, V. Fanti, E. Pastorelli, P. S. Paolucci, S. J. van Albada.
Published in *Frontiers in Neuroinformatics* (Tiddia et al., 2022).

Phenomenological modeling of diverse and heterogeneous synaptic dynamics at natural density, by
A. Korcsak-Gorzo, C. Linssen, J. Albers, S. Dasbach, R. Duarte, S. Kunkel, A. Morrison, J. Senk, J. Stapmanns, T. Tetzlaff, M. Diesmann, S. J. van Albada.
Published as a book chapter in *New Aspects in Analyzing the Synaptic Organization of the Brain* by *Springer US* (Korcsak-Gorzo et al., 2024).

Assessing the similarity of real matrices with arbitrary shape, by
J. Albers⋆, A. C. Kurth⋆, R. Gutzen, A. Morales-Gregorio, M. Denker, S. Grün, S. J. van Albada, M. Diesmann; ⋆ indicates equal contributions.
Published on *arXiv* (Albers et al., 2024), under review in *PRX Life*.
Results of this publication entered Chapter 6.

Spurious self-feedback of mean-field predictions inflates infection curves, by
C. Merger, J. Albers, C. Honerkamp, M. Helias.
Published in *Physical Review E* (Merger et al., 2024).

Cell-type specific projection patterns promote balanced activity in cortical microcircuits, by
A. C. Kurth⋆, J. Albers⋆, M. Diesmann, S. J. van Albada; ⋆ indicates equal contributions.
Published on *bioRxiv* (Kurth et al., 2024).
Results of this publication entered Chapter 5.

ACKNOWLEDGMENTS

I would first like to extend my sincere gratitude to my supervisors, Prof. Dr. Markus Diesmann and Prof. Dr. Sacha J. van Albada, for their invaluable guidance, support, and encouragement throughout the course of my PhD research. Their expertise, knowledge, and insightful feedback have significantly contributed to the quality and depth of this work. I am deeply grateful for their unwavering commitment to my professional development and academic growth during our collaborative journey. Furthermore, I would like to thank Prof. Dr. Andreas Offenhäusser for devoting his time to be my second examiner, and Prof. Dr. Sonja Grün for creating a wonderful institute in which science can flourish. I would also like to thank Stefan Mihalas, Ph.D. and his group at the Allen Institute for hosting me during my research stay in Seattle. It was a truly special time where I learned so much about science, collaboration, and culture.

Additionally, I would like to express my appreciation to Anno Kurth, who has been an essential partner in my research. Our fruitful exchanges of ideas have led to numerous intellectual breakthroughs and a shared passion for expanding our understanding of complex scientific concepts, be it at the whiteboard, laptop, or during our walks together. I am grateful for your continuous support and friendship throughout this experience.

Furthermore, my sincere thanks goes out to all members of the Institute for Advanced Simulation (IAS-6). Our thoughtful discussions, lunch table debates included, have broadened my perspective on and so many topics, neuroscience and beyond. I have not only found amazing colleagues that contributed to my academic success, but also sincere friends in you.

A special thank you also goes to Laura, my family, and my close friends, who have provided me with unwavering support throughout this journey. Your love, encouragement, patience, and belief in my abilities have been essential sources of strength during the challenges and successes along the way.

I am deeply grateful for all these relationships and could not have completed my PhD journey without your support, guidance, and belief in me. Thank you!

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Part I

INTRODUCTION

1 BACKGROUND

For centuries, humans have been fascinated by the inner workings of the brain. Not only is it able to process an astounding amount of simultaneous sensory information with an energy efficiency orders of magnitude higher than that of modern computer hardware, it also brings about phenomena such as consciousness and self-awareness, which have been studied as much by natural scientists as by philosophers. It is natural to ask: what is the physical structure that underlies the brain and all its capabilities? Early pioneers in the field of neuroanatomy have been instrumental at shedding light on this very question. The silver staining method invented by Camillo Golgi in 1873 visualized what we now know as *neurons* for the first time (Golgi, 1873). Santiago Ramón y Cajal applied Golgi's staining method to produce highly detailed drawings revealing the key constituent elements of neurons: the input-receiving *dendrite*, the cell body (also called *soma*), and the output-sending *axon* (Ramón y Cajal, 1888). Following these observations, Cajal proposed the *neuron doctrine*, which argues for neurons as individual cells communicating through contact points, or *synapses*, in a directed fashion. These discoveries laid the groundwork for the vast field of neuroscience that has emerged since.

The following decades of neuroscientific investigations have produced a broad knowledge base about the structure of the brain. This thesis concerns itself with the outermost layer of the brain, known as the *cerebral cortex*. Approximately 2 mm thick in humans, it plays a pivotal role in higher-order information processing and cognitive functions such as perception, attention, memory, and language (Kandel et al., 2000). Perpendicular to its surface, the cortex exhibits a laminar structure—it can be subdivided into distinct layers based on the density and morphology of the present neurons. Figure 1.1 shows original drawings by Cajal where a layered structure becomes apparent.

Commonly, six layers are identified and aptly named L1 to L6, where L4 is also referred to as the granular layer due to its small, densely packed neurons. This layered organization allows for hierarchical information processing: certain layers predominantly receive sensory inputs while others generate motor outputs or engage in higher-level computations. In the direction parallel to its surface, the cortex can be categorized into *areas* that each fulfill different functions. Also here, a functional hierarchy emerges. For example, the primary visual cortex (V1) of mammalian species, which is the key area investigated in this thesis, is part of the visual hierarchy: in the *ventral stream*, which is

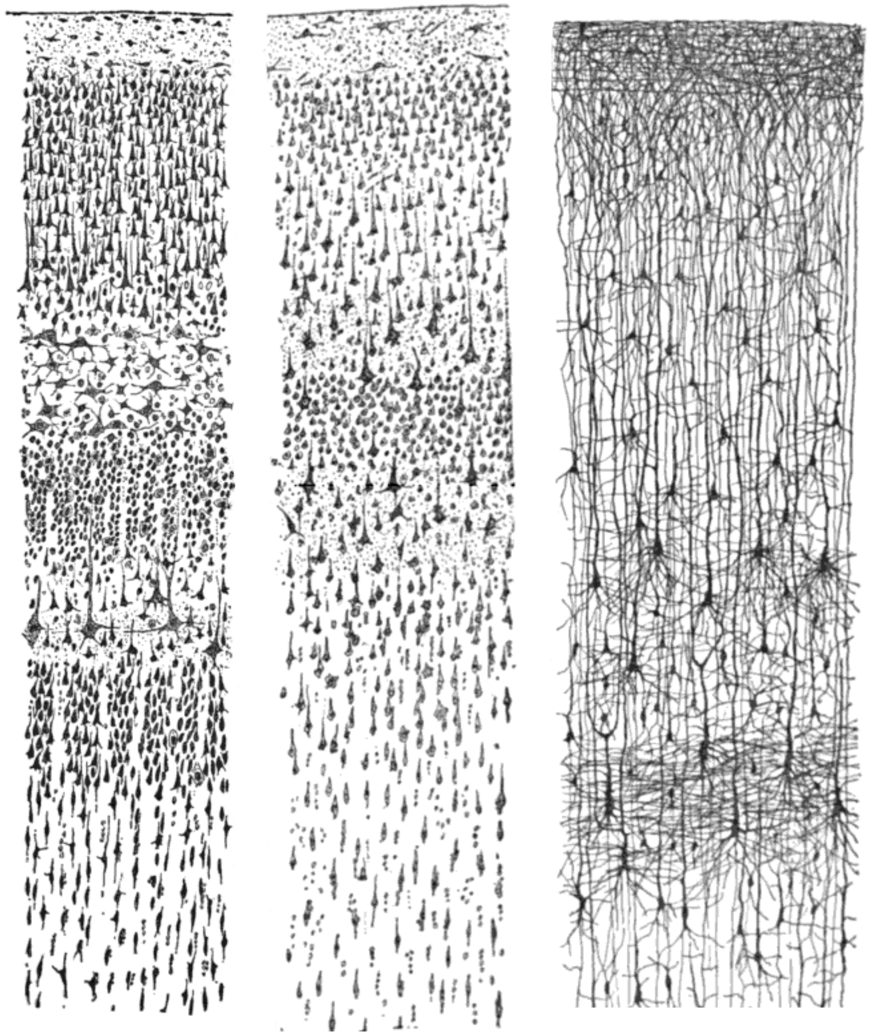

Figure 1.1: **Layered structure of the cerebral cortex.** Perpendicular to the cortical surface, shown here as the vertical direction, distinct layers emerge, each with its characteristic density of neural populations. Left: Nissl-stained visual cortex of a human adult. Middle: Nissl-stained motor cortex of a human adult. Right: Golgi-stained motor cortex of a human infant. Adapted from Ramon y Cajal (1899).

mainly responsible for object recognition in the visual input provided by the retina, V1 is the first area to receive sensory information. Each neuron in V1 has a *receptive field*: the collection of points in the visual scene from which a neuron receives information from the retina. After neuronal processing within V1, information is sent up the hierarchy to areas V2 and V4.

Brain areas also display characteristic organization principles within. Coming back to the example of V1, its surface is organized in a *retinotopic* manner: neurons close in cortical space have overlapping receptive fields—they receive sensory input from close by points in visual space. Thus, the visual scene is mapped continuously onto the cortical surface. The horizontal organization of V1 also induces *spatial connec-*

tivity patterns parallel to the cortical surface. For example, in certain species of primates, horizontal connections form clusters between neurons of similar tuning to input modalities. While conceptually useful, the functional categorization is not strict: since the brain is highly interconnected, also across areas, brain functions can be distributed among multiple areas.

Besides anatomical knowledge, physiological properties are of high relevance to understand the inner mechanisms of the brain. Even before the structure of the brain started to be revealed by Golgi and Cajal, Luigi Galvani found that electricity is the driving force in neural transmission (Galvani, 1791). Today, we recognize this as the electric communication between neurons. Essentially, neurons integrate the electrical stimulation induced by incoming signals on its dendrites, and once enough charge has accumulated—more precisely, its membrane is depolarized beyond a threshold—an *action potential* is generated: an all-or-nothing response that is characterized by a large momentary depolarization of the membrane potential. Subsequently, the action potential, also called *spike*, travels down the axon and acts, via synapses, as input to downstream neurons. Thus, the brain communicates in a digital fashion: the voltage trace of the action potential is always the same; the only information carried by it is its time of arrival. While the signal transmission along the presynaptic axon and postsynaptic dendrite are purely electrical, most types of synapses actually operate chemically. Upon arrival of an action potential, *neurotransmitters* are released that travel through the *synaptic cleft*—the gap in-between axon and dendrite—binding to receptors and evoking a fluctuation in the postsynaptic membrane potential that may lead to the formation of a new action potential.

The neurons themselves exhibit a variety of shapes and sizes, influencing their physiological properties and functional roles within the brain. Despite this diversity, neurons in the cerebral cortex of adult mammals can be broadly classified into two primary types based on the type of neurotransmitter that is released by their presynaptic terminals: *glutamatergic* and *GABAergic*. When they bind to receptors on the postsynaptic side of a synapse, glutamatergic neurotransmitters elicit a depolarization of the membrane potential, thereby acting towards the creation of an action potential. Conversely, GABAergic neurotransmitters elicit a hyperpolarization, driving the postsynaptic neuron further away from the threshold potential. Thus, these classes are also referred to as *excitatory* and *inhibitory*, indicating their electrophysiological properties. Since these kinds of neurotransmitter are characteristic for the presynaptic neuron, the elicited response is always of the same type—either of excitatory or of inhibitory nature. This feature is referred to as *Dale's law*.

Despite decades of work that followed the initial observations by Golgi, Cajal and Galvani, many aspects of the brain's working mechanisms remain poorly understood. Regarding clinical applications, a deep knowledge of the underlying processes that cause harmful conditions remains elusive to this day. As advances in modern medicine continue to raise life expectancy, such understanding is of particular relevance to treat diseases of the aging brain: both Alzheimer's and Parkinson's disease have seen marked increases in prevalence and incidence over the last 35 years (Li et al., 2022; Savica et al., 2016). These conditions put a significant burden not only on the affected individuals and families, but also the healthcare system at large: the worldwide annual societal cost of dementia, which is largely driven by Alzheimer's, is estimated at over one trillion US dollars (Wimo et al., 2023). Our lack of effective treatments for such diseases highlights the urgent need for deeper insights into brain function and dysfunction.

Advances in technology, however, open up avenues of research that were impossible just a few decades ago. The past years have seen an explosion in the availability of detailed anatomical and physiological data from experiments, enabling more precise studies of both brain connectivity and function. Anatomical studies have long been based on using *tracers*—visible substances injected into neural tissue that travel along axons, thereby revealing neural projections. In recent years, machine learning techniques have been applied to analyses of image-based electron microscopy reconstructions, uncovering precise connectivity information for ever larger volumes of cortex (e.g., MICrONS Consortium et al. (2021)). Electrophysiological studies have benefited from advanced electronics components as well as sophisticated signal processing pipelines. Using the Utah Array[1], the activity of neurons can be recorded *in vivo* at 100 sites across a surface of $4 \times 4\text{mm}^2$ simultaneously, and using the Neuropixels[2] probe, the activity orthogonal to the cortical surface can be observed at up to 5000 recording electrodes.

In parallel with experimental advances, the exponential growth in computational power has allowed researchers to create increasingly sophisticated models of the brain, incorporating ever more detailed data. Computational Neuroscience, the study of just such models of the brain or its constituent parts, aims to utilize the conceptual benefits that are inherent to simulations of physical systems to study its properties. First, comparing the behavior of a computational model to its physical counterpart allows to validate the set of assumptions made for its creation. Second, once a sufficiently close description to the physical system is found, alterations of its parameters—e.g., the

1 https://blackrockneurotech.com/products/utah-array
2 https://www.neuropixels.org

neuronal properties or the principles of connectivity—can uncover the working mechanisms of the system.

Crucial for a successful model-based investigation is the choice of abstraction—as George E.P. Box famously put it, "all models are wrong, but some are useful" (Box, 1976). Depending on the scientific hypothesis under investigation, studies in Computational Neuroscience may choose a *bottom-up* or *top-down* approach. The former focuses on how sensory information is processed starting from the simple, basic stimuli up to more complex representations. These models typically operate in a hierarchical fashion, where lower-level sensory inputs (e.g., edges) are combined to form higher-level representations (e.g., shapes). The processing in these models is largely stimulus-driven, meaning that the characteristics of the stimuli dictate the neural responses. In contrast, the top-down approach emphasizes the influence of higher-level cognitive processes on perception and interpretation of sensory information. Thus, it explicitly considers how expectations, prior knowledge, and context shape the brain's understanding of stimuli.

The level of description in these models ranges from including single ion channels in the membranes of morphologically detailed neuron models to representing whole brain areas by just a few differential equations. Since current technology does not allow for simulations that have arbitrary biological detail while at the same time spanning arbitrary network size, a balance between the two must be struck. Computational Neuroscience has seen the development of dedicated *simulators*—specialized software for executing biophysical simulations—that handle the technological difficulties of efficiently simulating neurons or neuronal networks (Einevoll et al., 2019). Thereby, the model complexity is separated from the technological complexity, and the performance of simulators can be optimized, enabling simulations of neural networks of ever greater complexity. In conjunction with the chosen level of description, also an appropriate choice of simulation technology is necessary, as different simulators are optimized for different purposes.

Given the developments on the experimental and computational side, high-resolution anatomical data can now be integrated to constrain biologically plausible neural networks. Such networks can be efficiently simulated, enabling larger and more complex models to be run for longer periods of biological time. Finally, the detailed electrophysiological data provides the ground truth activity against which the dynamics of the simulated networks can be validated. Thereby, computational models can probe the link between the structure and dynamics of neural networks, providing a crucial piece in the puzzle of the structure-function relationship of the brain.

2 SCOPE OF THIS THESIS

This thesis works towards an understanding of the spatial connectivity principles in the visual cortex of mammals. Thereby, it contributes to finding optimal principles of neural connectivity that facilitate the computations of processing visual scenes. A thorough understanding of such connectivity principles may lead to advancements in visual prostheses, aiding visually impaired humans and animals in perceiving their surroundings. Furthermore, it can guide implementations of image processing in artificial neural networks, bringing the speed and efficiency of these applications closer to the capabilities of the brain.

This thesis puts a focus on the early brain areas contributing to the ventral stream in the visual hierarchy of mammals, which mainly processes object recognition. A particular emphasis is put on studying the connectivity in the primary visual cortex (V1) of *Macaca mulatta*, also known as macaque monkey. This choice is motivated by three key reasons. First, the macaque monkey is a "visual" animal: in contrast to, for example, the mouse, it has a well-developed sense of vision that utilizes sophisticated connectivity principles that are of interest for this work. Second, compared to animals with a simpler cortical organization, its brain strikes a closer resemblance to the human brain. Third, compared to investigating the human brain directly, anatomical and electrophysiological data is much more broadly available for the macaque monkey, especially in the visual regions.

Since the inhomogeneities of spatial connectivity principles found in the visual system of macaque monkey make them difficult to be studied purely analytically, a computational approach is chosen: neural network models are built in a data-driven way mimicking biologically observed anatomy. Different spatial connectivity principles are integrated such that the distribution of synapses can be controlled via parameters, giving a handle on the model connectivity. Analyzing the activity of such models, this thesis sheds light on the structure-dynamics relationship, which is a key component of the structure-dynamics-function triad of the brain.

For an investigation of the spatial connectivity principles, models need to (a) represent single neurons between which the connectivity principles are enforced, and (b) represent neural tissue of sufficient size such that the spatial connectivity principles can be represented faithfully. The resulting networks consist of tens of millions of neurons with tens of billions of synapses, henceforth called *large-scale*. For these reasons, this thesis models the visual system on the level of *point neurons*, which allows for constructing networks at biologically realistic neural and synaptic densities even for comparatively large parts of cortex. In this paradigm, each neuron's internal state—e.g., its membrane potential—is governed by one or a few differential equations, while its concrete morphology is neglected. Whenever the membrane potential reaches a threshold, an action potential, or spike, is elicited and sent to the postsynaptic partners. Thus, statistical measures of the activity of cortex can be validated against experimental observations.

This thesis addresses three key aspects of using a computational approach to study spatial connectivity: the technological foundations for large-scale simulations of spiking neural networks, linking spatial connectivity to the dynamics of spiking neural networks of primary visual cortex, and deriving precise spatial connectivity for future modeling studies.

TECHNOLOGICAL FOUNDATIONS FOR LARGE-SCALE SIMULATIONS OF BIOLOGICALLY REALISTIC SPIKING NEURAL NETWORKS

Simulating biologically plausible cortical network models requires substantial computational resources. Fortunately, Computational Neuroscience has seen a separation between simulators which provide the technological infrastructure, and models that implement biological constraints such as the connectivity. Especially for the large-scale networks considered in this thesis, efficient implementations using optimized simulators are imperative. Continually optimizing the performance of these simulators for a wide range of models, software, and hardware environments is aided by *benchmarks*—experimental evidence of performance metrics with associated metadata. However, collecting, storing, and displaying benchmark results such that they can be integrated into the development workflow proves challenging. Therefore, a strategy is needed for benchmarking neural network simulators sustainably, i.e., making the data and metadata easy to collect and access, thereby avoiding costly reruns and providing valuable insight for future performance adaptations in the underlying simulators.

These technological foundations are considered in Part ii. In Chapter 3, a software solution for sustainable performance development of neural network simulators is conceptualized and implemented. Further, concrete performance implications of simulator adaptations are demonstrated for three neural networks which are characteristic of modern, biologically inspired models.

linking spatial connectivity to the dynamics of spiking neural networks of primary visual cortex

Models of neural networks necessarily come with many free parameters. For example, even the abstract *integrate-and-fire* neuron model, the state of which is governed only by a single variable, has over ten parameters. Further, the precise network structure is, by and large, unknown. Thus, models of neural networks, especially when created at a large physical scale, are notoriously under-constrained. To create models resembling the structure of cortex found in experiments, detailed connectivity data can help to better constrain the parameter space. Therefore, a thorough account of the available connectivity data and an implementation appropriate for efficient instantiation and simulation of such models is critical. A key benefit of the computational approach lies in the fact that neural network models allow to study how the simulated activity changes under alterations of the underlying network connectivity. Experimental recordings provide target dynamics, usually given in terms of the statistics of the observed activity. For example, one can infer neuronal spike rates or the distribution of inter-spike intervals from electrophysiological recordings. In a first step, the assumptions of the model building process can thereby be validated. Further, models can yield concrete predictions for connectivity parameters that are yet to be experimentally determined, which are testable in future experiments. Thereby, such computational models can shed light onto the intricate connectivity structure found in the cortex, and inform future functional studies about possible computational benefits.

Part iii details the construction and simulation of anatomically constrained spiking neural network models of visual cortex, linking the observed dynamics to the underlying anatomy. In Chapter 4, a model of macaque V1 is constructed that integrates a variety of distinct spatial connectivity principles operating at different spatial scales based on recent connectivity data. Simulations of this model uncover pathological states of activity that, after a critical evaluation of the underlying structure, is hypothesized to originate in the targeting patterns of excitatory neurons in the network. Chapter 5 thoroughly investigates this hypothesis by comparing the dynamical implications of different local cortical architectures.

deriving precise spatial connectivity for future modeling

To inform future computational studies of spatial connectivity, a precise account of the biological connectivity is imperative. Of course, neural network models necessarily make abstractions and assumptions. Therefore, the intricate details of local cortical connectivity need to be distilled into a form such that they are applicable to constrain computational models. Here, probabilistic connectivity rules that predict the number of synapses between neurons given, for example, their cell type and relative positions, are a powerful tool that can straight-forwardly be used to connect neurons in a network model.

Part iv aims to advance our understanding of cortical connectivity principles with a view towards computational modeling. As a prerequisite, Chapter 6 discusses how the connectivity structure of different neural networks can be compared in a systematic fashion. Based on their adjacency matrices, which represent the number of connections between all pairs of neurons, this chapter derives a novel measure of similarity between networks. Then, in Chapter 7, a novel probabilistic rule of connectivity is developed which takes the characteristic spatial distributions of synapses of neurons into account. Applying the measure of matrix similarity demonstrates that this rule captures the intricate details of spatial synaptic targeting patters to a greater degree than established rules. Thereby, Part iv paves the way for future computational investigations to include precise spatial connectivity information in the creation of ever more biologically realistic network models.

Finally, Part v summarizes the technological and neuroscientific insights gained throughout this thesis and gives an outlook on how future modeling studies can build on the presented results. It thereby paints a picture of how future computational studies, created in the spirit of models used in this thesis, can deepen our understanding of the functional implications of spatial connectivity.

Part II

TECHNOLOGICAL FOUNDATIONS FOR LARGE-SCALE SIMULATIONS OF BIOLOGICALLY REALISTIC SPIKING NEURAL NETWORKS

PREFACE

The networks studied in this work share one fundamental property: they are models that represent neurons and synapses at realistic density while, at the same time, spanning areas of cortex on the order of square millimeters. Consequently, the number of neurons range from tens of thousands to millions of neurons, and the number of synapses range from tens of millions to billions. As such, efficient model implementations on modern high-performance computing (HPC) infrastructure are imperative for scientific investigations, which require simulations over multi-dimensional parameter spaces to study differences in network activity. Furthermore, if models aim to interact with the outside world, e.g., by controlling an artificial agent, models need to run in real-time—the time it takes to perform the simulation needs to match the intrinsic biological time of the model.

Through decades of work, the community of neuroscientists has achieved a separation of model implementation from the underlying simulation technology (Einevoll et al., 2019). Thereby, new models are not required to implement fast solvers to the differential equations that govern the state variables of the neurons, or efficient communication strategies between neurons that are distributed on the compute cores of a machine or even across compute nodes in a multi-node setup. Instead, dedicated simulators, which are developed alongside novel neuroscientific models, can focus on optimal performance and efficiency, thereby enabling the simulation of models with ever greater detail and complexity. Currently available simulators specialize in different aspects of research in Computational Neuroscience. For example, `NEURON` (Carnevale and Hines, 2006; Lytton et al., 2016; Migliore et al., 2006) and `Arbor` (Akar et al., 2019) focus on small networks of morphologically detailed neurons, `NEST` (Gewaltig and Diesmann, 2007; Helias et al., 2012; Ippen et al., 2017; Jordan et al., 2018; Kunkel and Schenck, 2017; Kunkel et al., 2012, 2014; Morrison et al., 2005a; Plesser et al., 2007; Pronold et al., 2021) and `Brian` (Goodman and Brette, 2008; Stimberg et al., 2019) reduce the complexity of single neuron representation in favor of enabling simulations with a large number of neurons and synapses, and `GeNN` (Knight et al., 2021; Knight and Nowotny, 2018; Knight and Nowotny, 2021; Stimberg et al., 2020; Yavuz et al., 2016) and `NESTGPU` (Golosio et al., 2021) leverage the computational power of GPUs for neuronal network simulations.

To enable the next generation of network models which incorporate detailed anatomical as well as functional information, neural network simulators need to be continuously optimized for simulation performance. Over the past years, tremendous effort has been dedicated towards this philosophy (Jordan et al., 2018; Kunkel et al., 2014; Kurth et al., 2022; Pronold et al., 2021). A fundamental tool for analyzing the performance in all of these works is the concept of *benchmarking*, i.e., recording detailed information on simulation speed, memory usage, or other performance quantities, and comparing them across different versions of the underlying simulation code. Computer science routinely uses benchmarks for assessing the performance of novel hard- and software. For example, the latest version of the LINPACK benchmarks (Dongarra et al., 2003), originally released in 1979, quantifies the compute performance of HPC systems (TOP500 list). For applications in the field of Artificial Intelligence, standard benchmarks guide development to ever more powerful network architectures and training techniques, e.g., MLPerf[1] (Mattson et al., 2019) or the High Performance LINPACK for Accelerator Introspection (HPL-AI) benchmark[2]. Ostrau et al. (2020) developed a benchmarking framework for deep spiking neural networks, comparing results from Spikey (Pfeil et al., 2013), BrainScales (Schemmel et al., 2010), NEST, SpiNNaker, and GeNN.

This part of the thesis devises a benchmarking workflow for neuronal network simulators to continually evaluate the performance alongside development. Conceptual considerations are implemented in beNNch, a platform for executing and analyzing performance benchmarks for such simulators. Furthermore, beNNch is designed to store and display benchmarking results such that they can be readily shared between researchers. Thereby, reproducibility of benchmarking studies in the field of Computational Neuroscience is facilitated, ultimately paving the way for enhancements of the simulation technology that make it possible to study cortical networks of ever more biological realism.

1 https://mlcommons.org
2 https://www.icl.utk.edu/hpl-ai

3 A BENCHMARKING FRAMEWORK FOR NEURONAL NETWORK SIMULATORS

This chapter is based on the following publication:
J. Albers, J. Pronold, A. C. Kurth, S. B. Vennemo, K. H. Mood, A. Patronis, D. Terhorst, J. Jordan, S. Kunkel, T. Tetzlaff, M. Diesmann, and J. Senk: *A Modular Workflow for Performance Benchmarking of Neuronal Network Simulations* (Albers et al., 2022).

Author contributions:
The author jointly designed this study with Jari Pronold (JP), Anno Christopher Kurth (ACK), Stine Brekke Vennemo (SBV), Kaveh Haghighi Mood (KHM), Alexander Patronis (AP), Dennis Terhorst (DT), Jakob Jordan (JJ), Susanne Kunkel (SK), Tom Tetzlaff (TT), Markus Diesmann (MD) and Johanna Senk (JS). The author implemented the software framework together with JP, ACK, SBV, KHM, DT and JS. The author executed and analyzed all benchmarks. The author, together with JP, SK, and JJ produced the explanatory figures. The author and AP implemented the shrinking MPI buffers, while JJ and JS implemented spike compression, and SK implemented neuronal input buffers with multiple channels.

3.1 INTRODUCTION

In Computational Neuroscience, benchmarking poses a particular challenge, as the underlying neuroscientific simulations—which are often highly sensitive to numerical details—are already notoriously hard to reproduce (Crook et al., 2013; Gleeson et al., 2019; Gutzen et al., 2018; McDougal et al., 2016; Pauli et al., 2018; Rougier et al., 2017). Benchmarking adds another layer of complexity to this situation since the recorded performance metrics crucially depend on the computational environment. In fact, the complexity of comparing benchmarks in Computational Neuroscience can be separated in five independent *dimensions of complexity* (Figure 3.1).

First, the hardware configuration has an obvious impact on the performance metrics: different processors, RAM setups, and communication infrastructure inherently affect the benchmark outcome. Second, even if the hardware configuration is identical across benchmarks, the software configuration, such as the utilized software stack, the used number of nodes, processes per node, and threads per process can significantly alter performance. Third, different simulators,

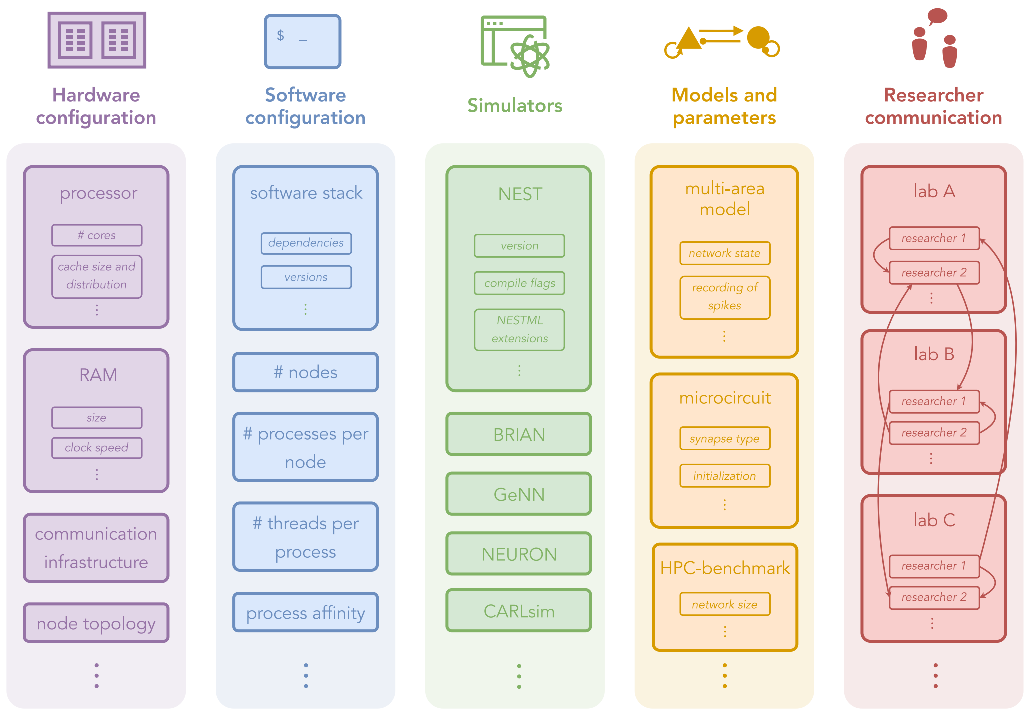

Figure 3.1: **Dimensions of complexity of HPC benchmarking experiments for neuronal network simulations.** Hardware configuration: computing architectures and machine specifications. Software configuration: general software environments and instructions for using the hardware. Simulators: specific simulation technologies. Models and parameters: different models and their configurations. Researcher communication: knowledge exchange on running benchmarks.

while producing the same results of network dynamics, can achieve those results with various degrees of efficiency. But even when using the same simulator, different versions or compile flags may alter performance significantly. The aforementioned three dimensions apply to nearly all software that is to be benchmarked. The fourth dimension is particular for Computational Neuroscience, where models and simulators are disentangled: different models may highlight various aspects of a simulators performance. For example, neuroscientific models may vary greatly in the number of connections between neurons, highly impacting how much simulation time is spent communicating between neurons. In addition, depending on the application, the neuron models employed may have various degrees of biological realism, and thereby differ in the computation time required to solve for their state variables. To avoid optimizations that only enhance the simulations of certain types of network models, simulators cannot rely on a single „representative" network model, but must instead be benchmarked for an array of models that incorporate different characteristic traits of models used in Computational Neuroscience. Finally, benchmarking is complex due to the required communication between researches within and across labs. Only when benchmarking results are displayed in a comparable fashion, and these results are easily sharable or accessible, benchmarking of neuronal network simulators can be achieved in a sustainable way.

To overcome the dimensions of complexity, we envision a platform for performing benchmarks of neuronal network simulators. In this work, we conceptualize a modular framework for executing such benchmarks across hardware and software configurations, simulator versions, and models, analyzing the performance results, and displaying them in a condensed and easily sharable manner. Further, we construct a particular implementation named `beNNch` that focuses on the NEST Simulator, and highlight how it can aid simulator development.

Section 3.2 conceptualizes the benchmarking workflow (Section 3.2.1), implements the concepts into a reference benchmarking framework for the `NEST` simulator (Section 3.2.2), and applies it on three frequently used network models, making a case for the relevance of performance benchmarking for simulator development (Section 3.2.3). Section 3.3 provides details on the specific performance optimizations investigated here. Finally, the use of this framework for further simulator development is discussed in Section 3.4.

3.2 RESULTS

3.2.1 *Workflow Concepts*

In this chapter, we devise a generic workflow for performance benchmarking that is applicable to simulations running on conventional HPC architectures. Figure 3.2 shows the conceptual workflow, which comprises four segments that sequentially depend on each other. The segments are further subdivided into modules, each one linked to a specific realization used in our reference implementation (Section 3.2.2). The notion "workflow" here describes abstract concepts of general applicability with regard to benchmarking efforts, and "framework" refers to the software implementation we have developed. Further, we differentiate between "internal" and "external" modules: the former we consider as essential building blocks of the workflow while the latter can be exchanged more readily. In the following, we introduce each of the workflow's conceptual segments and explain how our proposed solutions address the identified problems (cf. Figure 3.1).

3.2.1.1 *Configuration and preparation*

The first of the four workflow segments (Figure 3.2, left) comprises five distinct modules. Together, they set up all prerequisites necessary for the simulations. "Software deployment" handles the installation of the simulation software and its dependencies, and "machine configuration" specifies parameters that control the experiment conditions of the simulation such as the number of compute nodes. Thus, these two modules target the "hardware configuration", "software configuration", and "simulators" dimensions of the benchmarking problem (Figure 3.1). Both the "model" and the "model configuration" modules address the "models and parameters" dimension: the former provides the implementation of the network model, and the latter passes parameters to the model such as the biological model time. Thus, we achieve a separation of the model and its parameters. Finally, the "user configuration" module collects user-specific data such as custom file paths or specific compute budgets.

3.2.1.2 *Benchmarking*

All modules related to running the benchmark simulations are contained in the second segment (Figure 3.2, second from the left). Since compute clusters typically manage the workload via queuing systems, compute-intensive calculations are submitted as jobs via scripts which define the required resources and hold instructions for performing the simulation. In the workflow, the module aptly named "job script generation" handles this part. Here, we can see the first link between modules: model, user and machine configuration are combined to

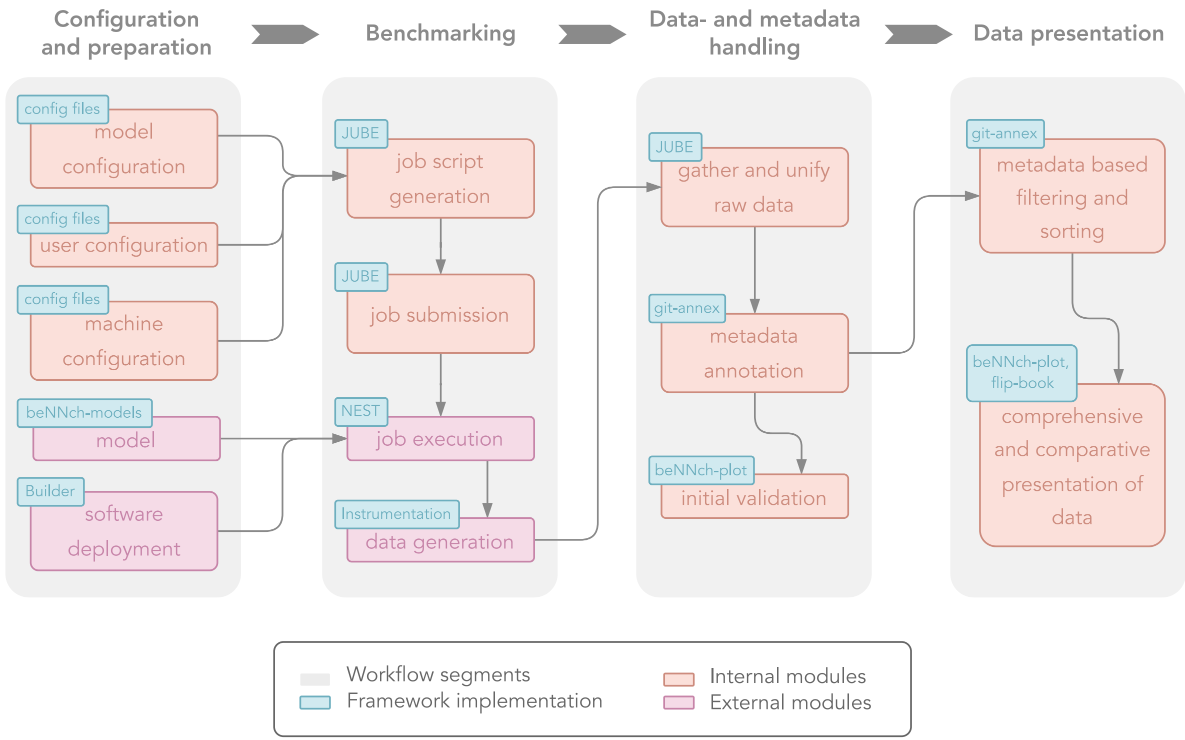

Figure 3.2: **Conceptual overview of the proposed benchmarking workflow.** Light gray boxes divide the workflow into four distinct segments, each consisting of multiple modules. Internal modules are shown in mint and external ones in cyan. Pink boxes indicate their respective realization in our reference implementation.

create a job script which is subsequently submitted to the job queue via the module "job submission". With the simulator software in place (handled by the "software deployment" module in the first segment), "job execution" performs the model simulation according to the job-submission parameters. Usually, simulations for neuroscientific research purposes would, at this point, focus on the output of the simulation—for example, neuronal spike times or voltage traces. However, benchmarking is mainly concerned with the performance results, which are recorded in the final benchmarking module called "data generation".

3.2.1.3 *Data- and metadata handling*

Handling all data and metadata produced by the simulations presents itself as a core challenge in conducting performance benchmarks. With data, we here refer to the results of the performance measurements, while we use metadata as an umbrella term describing the circumstances under which the data was recorded according to the dimensions of benchmarking (Figure 3.1). Since the execution of multiple simulations using different configurations, software, hardware, and models is an inherent part of benchmarking, data naturally accumulates. Recording the variations of performance across these dimensions leads to a breadth of metadata that needs to be associated to the measured data. Standardizing the underlying formats for both types of data is essential for making the results comparable for researchers working with the same simulation technology. The workflow segment "Data- and metadata handling" (Figure 3.2, third from the left) proposes the following solution: first, the raw performance data, which typically stems from different compute nodes of the HPC system, are gathered and transferred into a standardized format, while the simulation-specific metadata is automatically recorded. Next, the metadata is associated to the data files, thereby eliminating the need for manually keeping track of simulation parameters, software environment conditions, and experiment choices. We here choose to attach the relevant metadata directly to the performance-data files in order to simplify filtering and sorting of results. Finally, "initial validation" creates a swiftly accessible view on the results such that erroneous benchmarks can be quickly identified.

3.2.1.4 *Data presentation*

The last workflow segment (Figure 3.2, fourth from the left) addresses the challenges posed by "Researcher communication" . It proposes a solution to making the results of benchmarking accessible and comparable for the users, thereby facilitating the drawing of meaningful conclusions. First, "metadata based filtering and sorting" allows for a dynamic choice of which results to include in the comparison. Here, dynamic means that the user can choose arbitrary cuts through the hypercube of metadata dimensions such that the filtered results only differ in metadata fields of interest. Second, the data is presented in a format such that intuitive switching between benchmarks is possible, while key metadata is presented alongside the performance results. This includes a standardization of the data representation. The presentation of data should be comprehensive and consistent such that the benchmarking results are comparable in the long term. Thereby, the risk of wasting resources because of re-generations of results is severely reduced, making the process of software development more sustainable.

3.2.2 *beNNch: A Reference Implementation*

We build on the workflow concepts developed in Section 3.2.1 and here introduce a reference implementation for modern Computational Neuroscience: `beNNch`[1]—a benchmarking framework for neuronal network simulations. `beNNch` not only serves as a proof-of-concept, but also provides a software tool that can be readily used by neuroscientists and simulator developers. While `beNNch` is constructed in a way that plug-ins can be developed for any neuronal network simulator, we specifically implement compatibility for 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 3.2 is directly associated to a concrete implementation.

3.2.2.1 *Builder*

Reproducible software deployment is crucial for guaranteeing that the benchmarks are repeatable and comparable. To ensure usability of the benchmarking framework, however, we need to abstract information on the hardware architecture and the software tool chain that is not immediately relevant. An automated tool for installing software should not depend on the particular flavor of the operating system, the machine architecture or overly specific software dependencies. In addition, such a tool needs to make use of system-provided libraries. `beNNch` uses the tool `Builder`[2] for this purpose. Given a fixed software stack and hardware architecture, `Builder` provides identical executables by deriving the install instructions from "plan files". Integration with other package management systems such as `easy_build` (Geimer et al., 2014) or `Spack` (Gamblin et al., 2015) is achieved by

1 https://github.com/INM-6/beNNch
2 https://github.com/INM-6/Builder

using the same environment module systems[3]. Thereby, the required user interaction is minimized and, from a user perspective, installation reduces to the configuration of installation parameters. 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.

3.2.2.2 *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 hard 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., 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).

3.2.2.3 *Instrumentation*

We focus our performance measurements on the time-to-solution. Acquiring accurate data on time consumption is critical for profiling and benchmarking. To this end, we make use of two types of timers to collect this data: The timers are either built-in to `NEST` on the C++ level, or they are included on the Python level as part of the `PyNEST` network-model description. The latter type of timers are realized with explicit calls to the function `time.time()` of the Python Standard Library's `time` package. To achieve consistency throughout the framework, we use standardized variable names for the different phases of the simulation. Figure 3.3 shows the simulation flow of a typical `NEST` simulation. During "network construction", neurons and auxiliary devices for stimulation and recording are created and subsequently connected according to the network-model description. Afterwards, in the course of "state propagation", the network state is propagated in a globally time-driven manner. This comprises four main phases which are repeated until the entire model time has been simulated: update of

3 https://modules.readthedocs.io and http://lmod.readthedocs.io

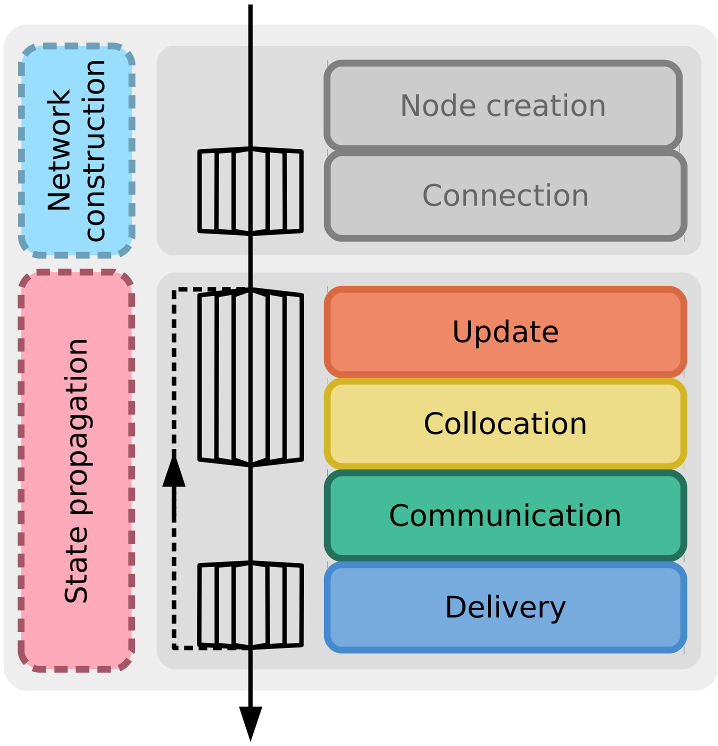

Figure 3.3: **Instrumentation to measure time-to-solution.** Successive phases of a NEST simulation; time is indicated by top-down arrow. Fanning arrows denote parallel operation of multiple threads. The main phases network construction (cyan) and state propagation (pink) are captured by external timers on the Python level. Built-in NEST timers on the C++ level measure sub-phases: node creation and connection (both grey, not used in benchmark plots); update (orange), collocation (yellow), communication (green), and delivery (blue). The sub-phases of the state propagation are repeated until the simulation is finished as shown by the dashed arrow connecting delivery and update.

neuronal states, collocation of spikes in MPI-communication buffers, communication of spikes, and delivery of the received spikes to their respective thread-local targets. 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 "connection" phase. 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. Besides, it is common practice to introduce a short pre-simulation before the actual simulation to give the network dynamics time to level out; the state propagation phase is only recorded when potential startup transients have decayed (Rhodes et al., 2019). The model time for pre-simulation can be configured via a parameter in beNNch. For simplicity, Figure 3.3 does not show this pre-simulation phase.

3.2.2.4 *beNNch-models*

We instantiate the "model" module with the repository beNNch-models[4] 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 3.2.2.3) 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[5], the microcircuit model[6] by Potjans and Diesmann (2014) representing 1 mm^2 of cortical surface with realistic neuron and synapse densities, and the multi-area model[7] 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 above.

3.2.2.5 *config files*

When executing benchmarks, the main user interaction with beNNch consists of defining the characteristic parameters. We separate this from the executable code by providing yaml-based templates for "config files". Thereby, the information that defines a benchmark experi-

---

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

ment is kept short and well arranged, limiting the number of files a user needs to touch. Code 3.1 presents an excerpt from such a config file which specifies model, machine, and software parameters.

```yaml
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"}
```

Code 3.1: **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 is created for each entry of the list. **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).

While some parameters are model-specific, standardized variable names are defined for parameters that are shared between models.

3.2.2.6 *JUBE*

At this point, the first segment of the benchmarking workflow (Figure 3.2) 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`[8] 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.

3.2.2.7 *git-annex*

Without a mature strategy for sharing benchmark results, communication can be a major obstacle. Typically, each researcher has their preferred workflow, thus results are shared over different means of communication, for example, via email attachments, cloud-based storage options, or `git` repositories. This makes it difficult to maintain an overview of all results, especially if researchers from different labs are involved. Ideally, results are 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

8 https://fz-juelich.de/en/ias/jsc/services/user-support/software-tools/jube
9 https://git-annex.branchable.com
10 https://gin.g-node.org

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 "metadata annotation" module. Previously 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. 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. Importantly, a balance needs to be struck between recording large amounts of metadata and keeping the volume of annotations manageable. In our implementation, we choose to solve this issue by recording detailed metadata about the system, software, and benchmarks, but only attaching what we currently deem relevant for performance to the data. 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. 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 3.2.

3.2.2.8 *beNNch-plot*

To enable a comparison between plots of benchmark results across the dimensions illustrated in Figure 3.1 it is paramount to use the same plotting style. To this end, we have developed the standalone plotting package `beNNch-plot`[11] based on `matplotlib` (Hunter, 2007). Here, we define a set of tools to create individual plot styles that can be combined flexibly by the user. 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.

3.2.2.9 *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 3.2.2.7. 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, our proposed solution offers a way of sharing benchmarking insights between researchers that is both scalable and flexible.

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

3.2.2.10 *Exchanging external modules*

beNNch is written in a modular fashion; as such, it is possible to exchange certain modules without compromising the functionality of the framework. In particular, the "external modules" (see Figure 3.2) are implemented such that an exchange is straight-forward to implement. This section presents a recipe to exchange the "job execution" module, i.e., the simulator, along with necessary changes in "data generation" and "model" that follow:

First, the simulator to be substituted instead of NEST needs to be properly installed. Builder—our implementation of the "software deployment" module—provides the flexibility to install any software as well as make it loadable via a module system. Thus, a plan file specifying dependencies as well as source code location and installation flags needs to be created for the new simulator.

Second, models compatible with the new simulator need to be added. On the framework side, the execute commands may need to be adapted. Required adaptations to the models are the same as for PyNEST models and are described in Section 3.2.2.4.

Third, the instrumentation needs to be changed. As NEST has built-in instrumentation, only superficial timing measurements are necessary on the model level. Depending on the new simulator's existing ability to measure performance, timing measurements might need to be implemented on the simulator or model level. 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.

For the development of simulation software with the goal to optimize its performance, it is vital to focus efforts on those parts of the simulation loop that take the most time to complete. Benchmarking can help in identifying performance bottlenecks and testing the effect of algorithmic adaptations. However, the dimensions of benchmarking identified in Figure 3.1 make this difficult: 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 simulator development.

beNNch, as an implementation of the workflow outlined in Section 3.2.1, provides a platform to handle the complexity of benchmarking while staying configurable on a low level. The following suggests how `beNNch` can support the process of detecting and tackling performance issues of a simulator. In a first step, exploration is necessary to identify the performance bottlenecks of the current version of the simulation engine. As many software and model parameters need to be explored, the centralized location of configuration parameters built into `beNNch` helps in maintaining an overview of conducted experiments. Neuronal network simulations can usually be decomposed into separate stages, such as neuronal update and spike communication. The instrumentation and visualization of these stages is part of `beNNch` and points 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 alleviates the researchers from operating a different code base for every model. In the process of solving the simulator issue, running further benchmarks and directly comparing new results can assist in deciding which adaptations bear fruit. The standardized visualization tools of `beNNch` support spotting differences in performance plots. Finally, an ongoing development of a neuronal network simulator should respect the value of insights gained by resource-intensive benchmarks. To this end, `beNNch` implements a decentralized storage of standardized performance results. In addition to preserving information for the long term, this also helps in communicating between researchers working on the simulator's development.

3.2.3 *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. 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 (`NEST 2.20.2`, Fardet et al., 2021). Table 3.1 summarizes the `NEST` versions employed in this study. Regarding the dimensions of HPC benchmarking in Figure 3.1, this use case primarily addresses the "Simulators" dimension by testing different NEST versions and the "Models and parameters" dimension by testing different network models; the approach can be extended similarly to the other dimensions.

Shorthand notation	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) = `3.0rc+ShrinkBuff+SpikeComp` plus neuronal input buffers with multiple channels

Table 3.1: Shorthand notation and description of NEST versions used in this work.

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 3.4 shows similar benchmarks of the same network model conducted with beNNch using the release candidate in Figure 3.4A and the final release in Figure 3.4B.

The graph design used here corresponds to the one used in the flip-book format by the framework. A flip-book version of the results shown in this work can be accessed via the GitHub Pages instance of the beNNch repository[14]. While the release candidate in Figure 3.4A exhibits growing state-propagation times when increasing the number of nodes, network-construction times stay constant and are, for $T_{\mathrm{model}} = 1\,\mathrm{s}$, small, making up less than 10% of the total simulation time. 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 will here emphasize a few concurring aspects: First, Jordan et al. (2018) run their benchmarks on the dedicated supercomputers JUQUEEN (Jülich

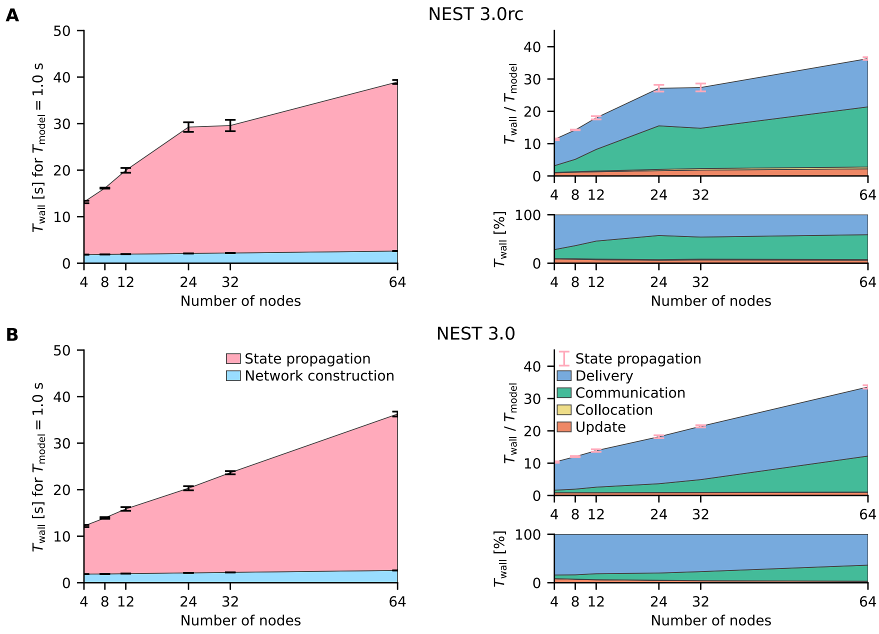

Figure 3.4: **Weak-scaling performance of the HPC-benchmark model on JURECA-DC. A** `NEST 3.0rc`. The left graph shows the absolute wall-clock time T_{wall} measured with Python-level timers for both network construction and state propagation (legend in panel **B**); the model time is $T_{\text{model}} = 1\,\text{s}$. 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. Same colors used for phases as in Figure 3.3. **B** `NEST 3.0`. Same display as panel A.

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). 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. Second, Jordan et al. (2018) use 1 MPI process per node and 8 threads per process while our simulations are performed throughout this study with 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 3.1.

Having demonstrated that beNNch can perform weak-scaling experiments of the HPC-benchmark model as done in previous publications, we next turn to strong-scaling benchmarks of the multi-area model (Schmidt et al., 2018b) shown in Figure 3.5.

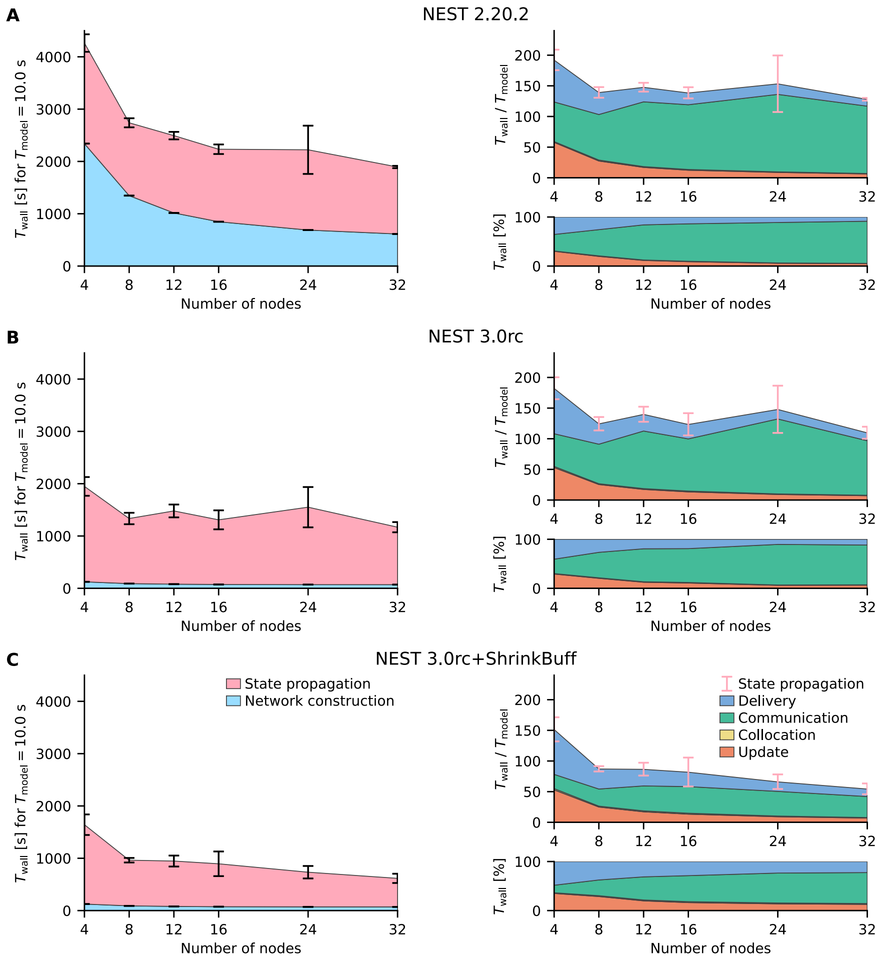

Figure 3.5: **Strong-scaling performance of the multi-area model on JURECA-DC.** Same display as in Figure 3.4. The multi-area model is simulated in its meta-stable state leading to a high amount of spikes that are communicated. The model time is $T_{\mathrm{model}} = 10\,\mathrm{s}$. Simulations are 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.

To fulfill the memory requirements of the model, at least three compute nodes of JURECA-DC are needed; here, we choose to demonstrate the scaling on four to 32 nodes. Initially, we compare the latest `NEST 2` version (Figure 3.5A) with the release candidate for `NEST 3.0` (Figure 3.5B). The improved parameter handling implemented in `NEST 3.0rc` reduces the network-construction time. However, the communication phase here largely dominates state propagation in both `NEST` versions shown; both use the original 5g kernel. Previous simulations of the HPC-benchmark model have not identified the communication phase as a bottleneck Jordan et al., 2018, Figure 12. Communication only becomes an issue when then smallest delay in the network is of the same order as the computation step size because `NEST` uses the smallest delay as the communication interval for MPI. While the HPC-benchmark model uses 1.5 ms for all connections—which is a good estimate for inter-area connections—the multi-area model and microcircuit use distributed delays with a lower bound of 0.1 ms leading to a fifteen-fold increase in the number MPI communication steps.

The following identifies and eliminates the main cause for the large communication time in case of the multi-area model, thus introducing the first out of three performance-improving developments 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. Ideally, each buffer would fit exactly the right amount of information by storing all spikes of the process relevant for the respective communication step. Due to overhead attached to operating on additional vectors, a scheme in which the buffer size adapts precisely for each MPI process for each communication step can be highly inefficient. Therefore, in cases where communication is roughly homogeneous, it is advantageous to keep the exchanged buffer between all processes the same size, as is implemented in `NEST 3.0rc`. 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 3.3.1. 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 3.5C, Figure 3.5B).

Next, we assess the strong-scaling performance of the microcircuit model (Potjans and Diesmann, 2014) in Figure 3.6.

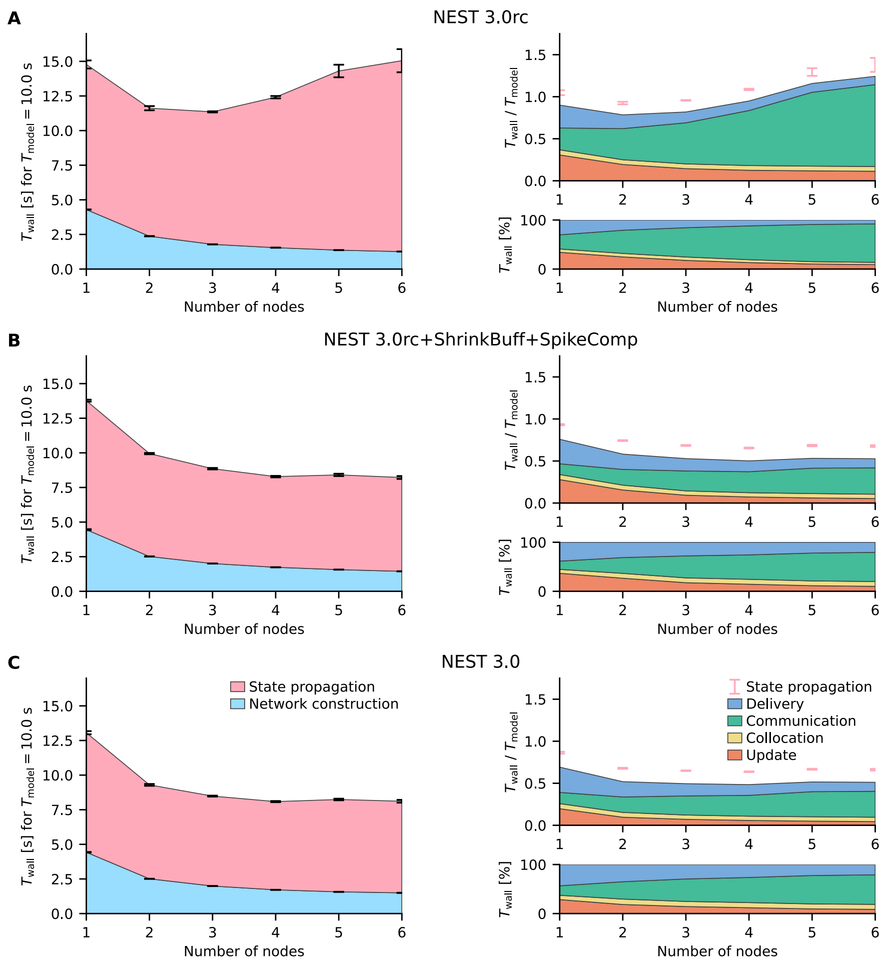

Figure 3.6: **Strong-scaling performance of the microcircuit model on JURECA-DC.** Same display as in Figure 3.4. The model time is $T_{\text{model}} = 10\,\text{s}$. Simulations are repeated for ten different random seeds. **A** `NEST 3.0` release candidate. **B** `NEST 3.0` release candidate with spike compression and shrinking MPI buffers. **C** `NEST 3.0`.

The model size is similar to the size of one of the 32 areas of the multi-area model. The microcircuit therefore requires fewer resources. We show results of the model run on one to six compute nodes; for a detailed analysis of NEST's thread scaling performance on the example of this model refer to Kurth et al. (2021). Using NEST 3.0rc, the microcircuit is simulated faster than the HPC-benchmark and the multi-area models and achieves approximately real time ($T_{\text{wall}}/T_{\text{model}} \approx 1$, Figure 3.6A). 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 3.5B) 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 3.3.2. This algorithm leads to a significant reduction in communication time for the microcircuit model (compare Figure 3.6A, Figure 3.6B).

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 sub-linear, 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 3.3.3. 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 3.6C shows the resulting decrease in update time for few compute nodes.

In summary, the analysis with `beNNch` exposes the communication phase as a major performance bottleneck in microcircuit and multi-area model simulations with the release candidate `NEST 3.0rc`. The underlying problem is, however, a different one for each of the two models, and they are rectified with different adaptations to the code: the shrinking MPI buffers (Section 3.3.1) improve the performance of the multi-area model while spike compression (Section 3.3.2) increases simulation speed of the microcircuit model. Notably, none of the adaptations introduce performance regressions for the respective other model (data not shown). In addition, the update phase is improved by introducing neuronal input buffers with multiple channels (Section 3.3.3). Returning to the HPC-benchmark model, Figure 3.4B shows that the kernel adaptations are not detrimental to the originally tested model; the overall state-propagation time is preserved with the final `NEST 3.0` release. However, the reduced communication and update times here come at the cost of increased delivery times 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.

3.3 METHODS

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).

3.3.1 *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.

3.3.2 *Spike compression*

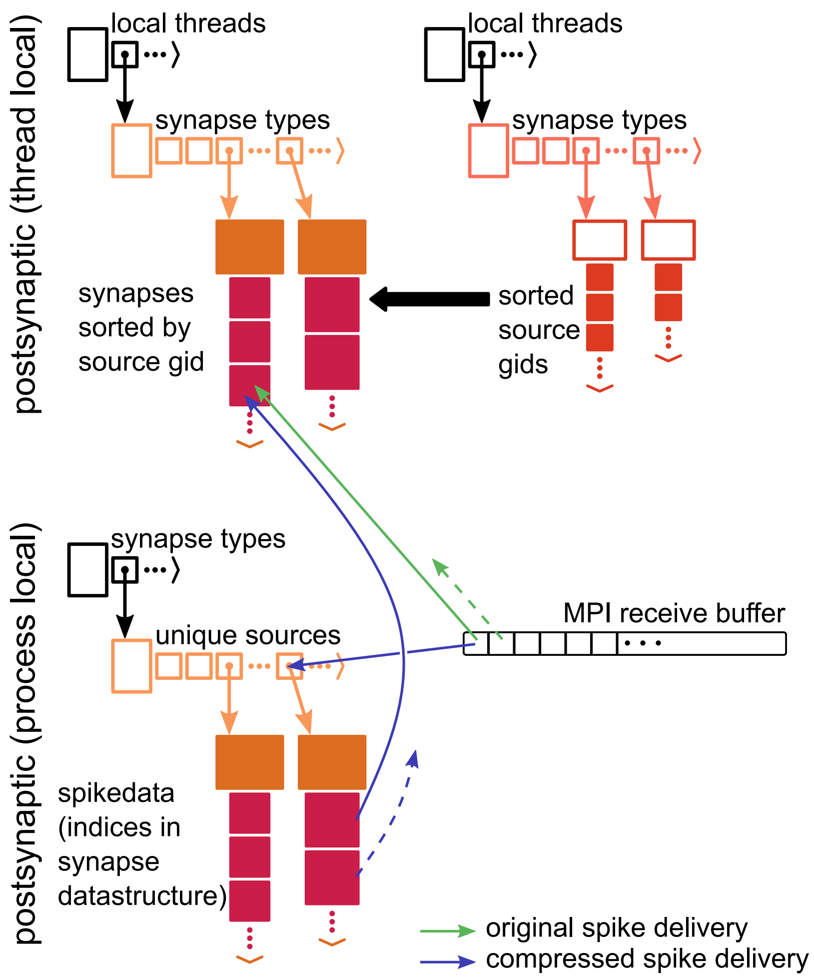

Figure 3.7: **Spike compression adds an additional indirection to postsynaptic spike routing.** Green arrow denotes original spike delivery introduced with the 5g kernel (Jordan et al., 2018, same display as their Figure 4A). Blue arrow illustrates additional indirection with compressed spike delivery. Dashed arrows indicate spikes from the same source neuron with target on a different thread.

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 3.2.2.3. 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 3.7, "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 identical, 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 3.7, "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})
```

3.3.3 *Neuronal input buffers with multiple channels*

Simulation technology for spiking neuronal networks requires techniques to handle synaptic transmission delays. The reference simulation code 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 synaptic delay d_{min} in the network model defines the communication interval (Morrison et al., 2005b), which comprises at least one

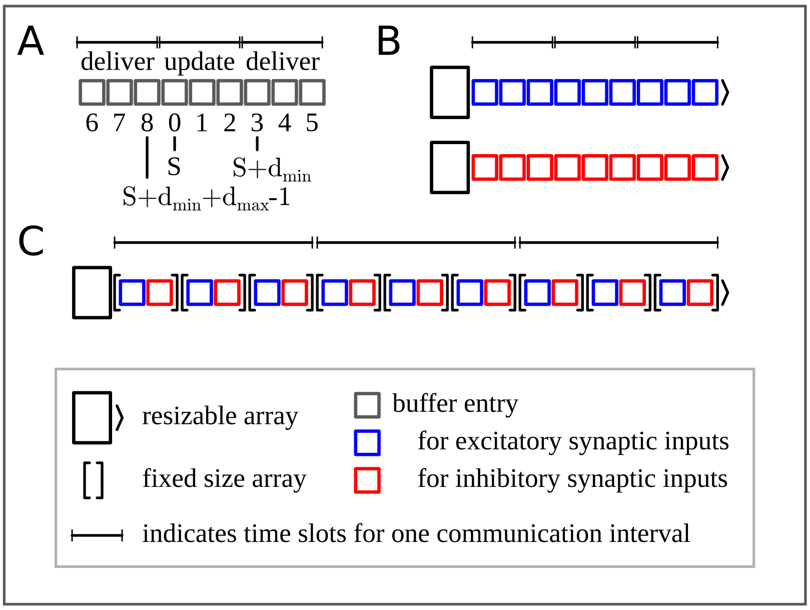

Figure 3.8: **Neuronal input buffers accounting for synaptic delays in simulations of spiking neuronal networks. A** Structure of neuronal input buffers assuming a minimum synaptic delay d_{min} of three simulation time steps and a maximum delay $d_{\text{max}} = 2d_{\text{min}}$. To buffer upcoming 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.

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., $d_{\text{min}} = 15$ 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 $d_{\text{min}} + d_{\text{max}}$ time slots, where d_{max} denotes the maximum synaptic delay in the network (Figure 3.8A). 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 3.8B). 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 3.8C). Thus, neurons now retrieve all input values for a particular time step by accessing subsequent locations in memory in one pass.

3.4 DISCUSSION

We identify five fundamental dimensions of complexity of neuronal network benchmarks: hardware configuration, software configuration, simulators, models and parameters, and researcher communication. This chapter proposes a solution that spans all five dimensions: a generic workflow for conducting benchmarking studies of arbitrary simulator software for neuronal networks. We envision a modular design where different workflow elements interact in a standardized way, while the modules themselves can have various realizations. The workflow starts at the user input for configuring the benchmark, and ends at displaying the results such that performance differences between simulator versions, software, or hardware configurations are easy to spot. As such, it transforms the labor-intensive task of performing neuronal network benchmarks into a routine effort that can be done in a collaborative way. Automatized execution reduces the potential for errors, and the tracing of hardware and software metadata ensures comparability of benchmark results over time.

To bring the theoretical considerations towards an application, a concrete software implementation called `beNNch` is developed that specializes in performance benchmarks of the NEST simulator. Since simulation performance crucially depends on properties of the neuroscientific network model, such as the number of neurons, number of synapses, and communication delays, `beNNch` includes three well-known network models that have been adapted for automatized benchmarking: a balanced random network model consisting of an excitatory and an inhibitory population of neurons, a microcircuit model constrained by local cortical connectivity, and a multi-area model with slow, long-range connectivity between areas. Thereby, `beNNch` can aid in ensuring that future adaptations to the underlying simulator benefit a wide rage of network models.

To demonstrate how `beNNch` can be directly integrated into the loop of simulator development, we investigate the effect of three particular performance adaptations: shrinking MPI buffers for optimized communication across compute nodes, spike compression for efficient routing of spikes to the relevant neurons, and modified neuronal input buffers for optimal cache usage. Testing the code adaptations on all three models, we show that each adaptation speeds up particular stages of the simulation for at least one model, while avoiding negative performance impacts on the other models. Thanks to these adaptations, future models can leverage the computational performance of NEST to include ever more biological detail into the simulations. This enables, for example, detailed parameter scans of microcircuit-like models (see Chapter 5), as well as the simulation of larger scale models that take spatial connectivity explicitly into account (see Chapter 4).

Since its inception, the proposed reference implementation has been implemented into the development workflow of the NEST simulator. Still, future versions of `beNNch` could overcome some limitations of this proof-of-concept implementation. First, the interface for executing and analyzing benchmarks could be streamlined, for example using a graphical user interface. This would not only simplify the process of regularly executing benchmarks, but also enable neuroscientists with less proficiency in high-performance computing to find optimal simulation setups for their own models. Further, the execution of benchmarks could be directly integrated into the development loop of simulators such as NEST: triggering a reference benchmark whenever a performance-relevant modification is made to the source code could ensure that the performance never degrades on accident. Finally, `beNNch` can be extended in a straight forward way. For example, new reference network models can be added which highlight different performance bottlenecks, and new performance metrics can be integrated into both the recording and the plotting process.

Part III

LINKING SPATIAL CONNECTIVITY TO THE DYNAMICS OF SPIKING NEURAL NETWORKS OF PRIMARY VISUAL CORTEX

PREFACE

In this part of the thesis, we develop novel neural network models that are constrained in their connectivity principles by neuroanatomical findings. We thereby aim to gain insights into computational benefits of the distinct wiring structure that can be observed in cortex. In particular, we focus on the primary visual cortex of mammalian species, often referred to as V1. Hubel and Wiesel pioneered the study of this brain area in modern neuroscience (Hubel and Wiesel, 1959, 1962, 1968), revealing a columnar organization and the existence of simple and complex cells that differ in their response properties to external visual stimuli. Since then, the primary visual cortex has been under intense investigation in the field of neuroscience, resulting in a large body of existing anatomical and electrophysiological data. These studies have revealed particular spatial organizational principles of the neural substrate: vertically, the classical distinction into layers of cortex applies, while horizontally, neurons follow a retinotopic organization where neurons that are close in cortical space have receptive fields that are close in visual space. Therefore, it is natural to ask how spatial connectivity principles can efficiently aid computation in such neural circuits.

When the connectivity is sufficiently homogeneous, mean-field theory can be employed to obtain analytical estimates of dynamical state variables (e.g., van Vreeswijk and Sompolinsky (1996), Amit and Brunel (1997), Brunel (2000)). However, the inhomogeneous spatial connectivity schemes observed in V1 call for a different approach. In this work, we follow a simulation-based ansatz that allows us to integrate such complex connectivity principles in our models. Because of the breadth of experimental studies concerning mammal V1, this brain area lends itself well to simulation-based investigations as (a) the parameters of connectivity are reasonably well constrained, and (b) the dynamical behavior of simulated models can readily be validated by comparing to the spiking activity of biological networks.

We here choose to work in the paradigm of neural networks that consist of point-like neurons with simplified dynamical variables. This formulation allows for constructing networks at biologically realistic densities of neurons and synapses even for comparatively large parts of cortex. Avoiding downscaling or subsampling is crucial for accurately comparing the resulting system activity to experimental data since otherwise, either the first-order or second-order statistics of the system dynamics are necessarily altered (van Albada et al., 2015a). Since the length scale of spatial connectivity investigated here—reaching up to 4 mm for long-range connectivity—necessitates network sizes on the same order, the resulting neural networks consist of tens of millions of neurons with tens of billions of synapses. Simulations of this magnitude can only be routinely performed thanks to the continued development of dedicated simulation technology, which is aided by benchmarking tools such as the framework developed in Part ii.

This part is split into two chapters. First, in Chapter 4, we construct a large-scale model of macaque primary visual cortex using various anatomical data sources. We focus on a derivation of spatial synaptic connectivity according to three distinct organizing principles: medium-range *isotropic connectivity* that depends on the inter-soma distance of neurons, long-range *patchy connectivity* that links neurons of similar tuning preference across receptive fields, and short-range *push-pull connectivity* that depends on the input correlations of neurons that predominantly receive sensory input. After deriving the connectome and its crucial parameters, we turn to instantiating spiking neural network simulations of the model. This reveals a fundamentally unbalanced dynamical state in which neurons are either completely silent or fire at frequencies well beyond what is biologically plausible.

Hypothesizing that the instability is inherent in the derived connectivity, we set out to systematically investigate the underlying connectivity structures that leads to such pathological behavior. In particular, we suspect the lack of data quality in an anatomical data set to be the root cause, leading to positive feedback loops in the derived connectivity. In Chapter 5, we investigate a recently released alternative data set that exchanges light microscopy (LM) for electron microscopy (EM) to reconstruct not only neural processes but also single synapses. Deriving a population-based connectivity map for the EM-based data set provides an updated view on the local cortical connectivity which, beyond our own work, may also be relevant for the community at large, which has relied on LM-based connectivity for numerous computational studies (Antolík et al., 2024; Potjans and Diesmann, 2014; Schmidt et al., 2018b). Instantiating spiking neural network simulations of both models allows to directly compare the influence of neural targeting patterns, revealing that the putative excitatory feedback loops are indeed a key factor underlying the unstable network dynamics observed in Chapter 4.

Finally, we turn back to the model of macaque V1 and replace the connectivity derived from the LM data set by the newly derived, EM-based connectivity. Here, we observe that a ground state of biologically realistic firing rates is reached without the need for any fine tuning of input parameters or connectivity strengths. Thereby, we provide a comprehensive model of macaque visual cortex that takes numerous spatial connectivity schemes into account, is arbitrarily scalable, and lends itself to being incorporated into a network of multiple areas.

The resulting model establishes a platform for investigating the computational benefits of spatial connectivity rules in visual processing.

4

A SPATIALLY ORGANIZED MODEL OF MACAQUE PRIMARY VISUAL CORTEX

Author contributions:
Under the supervision of Prof. Dr. Sacha van Albada (SvA) and Prof. Dr. Markus Diesmann (MD), the author conceived the project and conceptualized the work presented in this chapter together with Anno Christopher Kurth (ACK). The author and ACK jointly constructed the model based on anatomical data and theoretical considerations. Both jointly analyzed and compared the resulting connectivity maps. The analyses were refined in ongoing discussions of the author with SvA and MD.

4.1 INTRODUCTION

In this chapter we aim to construct a model of primary visual cortex (V1) of macaque monkey that is constrained by biologically informed spatial connectivity rules. We aim to build a platform with which computational questions about the nature of these connectivity rules can be answered systematically. We choose the macaque monkey as our target animal because the connectivity found in its visual system exhibits intricate patterns not found in, for example, mice or rat: beyond simple spatial connectivity where the connection probability is only a function of distance between the neurons, Livingstone and Hubel (1984) showed already 40 years ago that long-range connectivity forms distinct patches. Further, functional connectivity schemes have been theoretically proposed, e.g., push-pull connections between neurons in the thalamic input layer 4C that aid in producing contrast-invariant responses of cortical neurons (Troyer et al., 1998). With a spatial model of macaque V1 in place, connectivity schemes observed experimentally or proposed theoretically can be investigated in a single framework.

Previous studies have already investigated spatial connectivity using different approaches. Senk et al. (2023) extend the classical local microcircuit (Potjans and Diesmann, 2014) by adding isotropic spatial connectivity to obtain a scalable network of macaque V1. In this study, however, more intricate spatial connectivity patterns are missing. Antolík et al. (2024) successfully compile a large body of experimental data to create a model of cat V1 that includes patchy and push-pull connectivity. They create an extensive list of dynamical properties that can be used to validate such models against experimental recordings, and show that their model exhibits these characteristics. There, the authors only consider a subset of cortical layers (2/3 and 4), and further downscale the number of neurons and synapses in the model, thereby necessarily altering the statistics of the activity (van Albada et al., 2015a). Billeh et al. (2020) create a biologically constrained, full-scale model of mouse V1, exploiting a vast library of anatomical and electrophysiological data recorded by the Allen Institute for Brain Science. In contrast to macaque, mice do not exhibit more intricate spatial connectivity principles such as patchy connectivity. Therefore, this model does not allow for a dynamical evaluation of the effects of such connectivity schemes.

In our work, we aim to create a model that combines the strengths of the previous approaches by implementing a full-scale neuronal network of all cortical layers of macaque V1, including spatial organizational principles beyond isotropic distance-based connectivity. Investigating the dynamical property of this model enables an investigation of the structure-dynamics link, paving the way for future studies of the structure-dynamics-function triad.

4.2 RESULTS

In this chapter, we develop and implement a spatially organized spiking neural network model of macaque primary visual cortex. While our derivation focuses on V1—the earliest of the visual areas in the cortical hierarchy—we construct our model in a way such that it is amendable to include areas further up in the hierarchy as a natural extension. To constrain our model, we incorporate a large body of experimental data: Vanni et al. (2020) review over 350 published studies on connectivity data, and condense these information into connectivity profiles. We further supplement these data with additional connectivity data from other mammal species where necessary.

In our modeling, we make the following key assumptions. We represent neurons as leaky integrate-and-fire (LIF) model neurons that are connected using current-based synapses. Neurons are grouped into excitatory (E) and inhibitory (I) cell types in the cortical layers L2/3A, L3B, L4A/B, L4C, L5, and L6. L1 is disregarded due to its small density of neurons. L2/3 is split in accordance with the counterstream hypothesis which suggests that L2/3A processes feedback information from areas higher in the cortical hierarchy, while L3B processes feedforward information (Markov et al., 2013). Further, we split L4 due to their difference in anatomy: L4C, the layer which is primarily targeted by the thalamus, exhibits spatial connectivity on a shorter length scale. Thus, we model 12 populations in total: one excitatory and one inhibitory population per layer.

We start by deriving a cell type and layer-resolved synaptic connectome that specifies the number of local synapses between the populations of our model. We then distribute these synapses according to three spatial organization principles: *isotropic connectivity*, where the connection probability decays with distance; *patchy connectivity*, which links neurons with similar input-tuning preferences; and *push-pull connectivity* based on input correlations from thalamic projections. Thereby, we make the size of our model arbitrarily scalable. For all further investigations, we choose a size of $4 \times 4\text{mm}^2$. After constructing our model, we instantiate it using the NEST simulator (Gewaltig and Diesmann, 2007) to run dynamical simulations in order to investigate the effect of distributing the synapses according to the different spatial organization principles.

4.2.1 *Cell type and layer-resolved synaptic connectome*

Our main data source is the comprehensive overview of connectivity data in macaque visual cortex by Vanni et al. (2020). Given a presynaptic population, they report relative connection strengths that indicate preferential termination layers of outgoing synapses of the population's neurons. To obtain a full connectome between presynaptic and postsynaptic populations, we have to supplement information on the targeting patterns of these synapses on postsynaptic populations. This additional data is not available for macaque monkey in a sufficient resolution. Resting on the observation that certain structural features are shared across species (Douglas and Martin, 2004), we derive this data from morphological reconstructions in cat visual cortex (Binzegger et al., 2004). This procedure allows us to split the incoming synapses onto a neuron in our model into three categories: intra-areal, i.e., coming from other neurons in V1; thalamo-cortical, i.e., originating in the thalamus and carrying sensory information; or cortico-cortical, i.e., stemming from different cortical regions such as V2, V4, or non-visual areas. For connections of the first kind, we derive the total number of recurrent synapses between the different populations of our model, S^{ia}_{AB}, shown in Figure 4.1 (see Section 4.3.1.1 for details of the derivation).

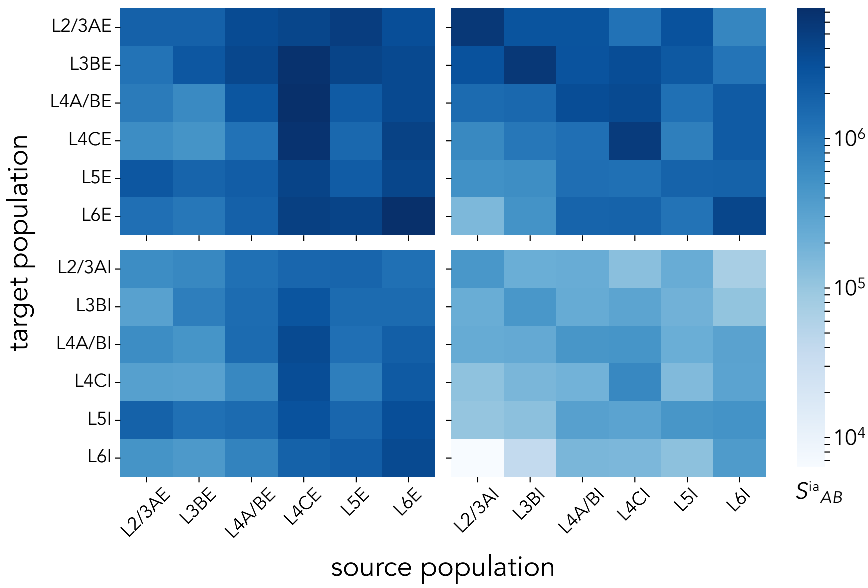

Figure 4.1: **Population-resolved synaptic connectome of macaque V1.** Number of synapses S_{AB} between presynaptic source populations B and postsynaptic target populations A in logarithmic scale.

For inhibitory source populations (right column), we observe a pronounced diagonal in the number of synapses, indicating a preference of neurons to connect to to other neurons in the same layer. For excitatory source populations (left column), the targeting patterns are more diverse, see e.g. L4CE strongly connecting to populations in L3B, L4A/B, and L4C. This pattern is in line with the hypothesis of a bottom-up flow of information: layer 4C, the main input layer for visual sensory information, acts a relay and transmits its inputs to the supragranular layers for further computation (see, e.g., Yabuta and Callaway (1998)).

Next to the recurrent connectivity, we further derive the number of thalamo-cortical synapses S_A^{th}, and cortico-cortical synapses S_A^{cc} from other brain areas, for all populations A from the same data sets (for details of the derivation, see Section 4.3.1.2 and Section 4.3.1.3, respectively). This yields the mean in-degrees per neuron given in Table 4.1.

Population A	S_A^{th}	S_A^{cc}
2/3AE	0	169
2/3AI	0	117
3BE	0	169
3BI	0	108
4A/BE	51	253
4A/BI	30	136
4CE	238	197
4CI	187	139
5E	11	184
5I	0	44
6E	139	699
6I	54	578

Table 4.1: **Thalamo-cortical and cortico-cortical mean in-degrees.** Population-specific mean number of incoming connections of thalamo-cortical and cortico-cortical type per neuron.

In agreement with previous studies on thalamic projections (Blasdel and Fitzpatrick, 1984; Garcia-Marin et al., 2017; Yabuta and Callaway, 1998), the main input layer of sensory information from the thalamus is L4C. The cortico-cortical input mainly targets the excitatory populations across all layers, with a preference of connecting to neurons in L6.

4.2.2 *Spatial connectivity*

Having derived the total number of synapses in our model, we next seek to distribute the intra-areal synapses according to spatial organization principles observed experimentally.

4.2.2.1 *Medium-range isotropic connectivity*

As an underlying model, we assume an exponential decay of connection probability P with horizontal inter-somatic distance r:

$$P(r) = P^0 e^{-\frac{r}{\lambda}} \tag{4.1}$$

To obtain numerical estimates for the decay parameter λ, which in principle may depend on the cell types and layers of the populations under consideration, we fit spatial connection profiles to experimental data as exemplified in Figure 4.2.

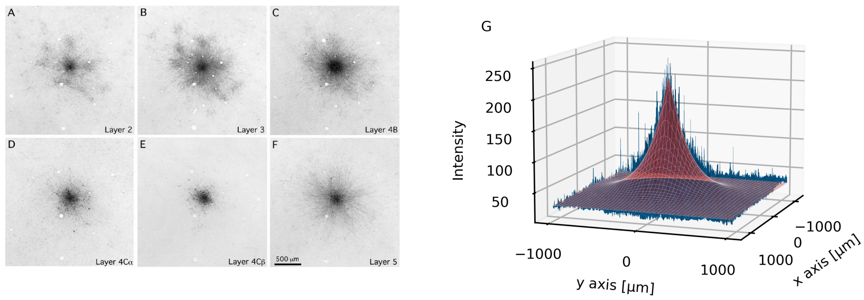

Figure 4.2: **Lateral connectivity of excitatory presynaptic neurons. A-F** Bidirectional tracing of axons in different layers of primary visual cortex of adult squirrel and owl monkeys from injections into layer 3, from Sincich and Blasdel (2001). **G** Fitting an exponential decay with horizontal distance to the inverse luminosity of the tracing image in layer 5.

Estimates for λ between different populations are summarized in Table 4.2 and Table 4.3 (for a detailed derivation, see Section 4.3.2.1).

	Source Population					
Target 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	0.216	0.121
L6	0.215	0.141	0.212	0.141	0.249	0.261

Table 4.2: **Estimates of characteristic lengths of connectivity λ_{vB} in** mm **for excitatory presynaptic neurons.**

We find that excitatory presynaptic populations have a consistently larger spatial spread than inhibitory ones. Further, one can observe that the spatial spread is generally largest for connections where both

	Source Population					
Target 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 4.3: **Estimates of characteristic lengths of connectivity λ_{vB} in mm for inhibitory presynaptic neurons.**

the pre- and postsynaptic neurons are in the same layer. Finally, we note that neurons in L4C have a particularly short range of isotropic connectivity. This might be related to its function as the main input layer for sensory information from the thalamus, rather playing a role of relay to the different cortical layers than performing computations within itself.

4.2.2.2 *Long-range patchy connectivity*

Neurons in V1 exhibit an orientation tuning preference: their neural response is strongest if a bar stimulus that is presented in their receptive field is of a particular orientation (Hubel and Wiesel, 1959). In macaque monkey, neurons that are close to each other, and therefore have overlapping receptive fields due to the retinotopic arrangement of V1, also have similar orientation tuning preference (Hubel and Wiesel, 1977). Thereby, they form a spatially clustered *orientation tuning map* along the dimensions parallel to the cortical surface. This map is intricately linked to experimentally observed patchy connectivity: long-range connections link neurons of similar orientation tuning preference (Stettler et al., 2002). Spanning different receptive fields, these connections may facilitate the spatial integration of information.

Since experimental evidence on the fraction of patchy synapses with respect to isotropically distributed synapses is lacking, we introduce the patchiness parameter $\psi \in [0, 1]$ that controls this fraction while keeping the total number of synapses S^{ia}_{AB} constant (Equation 4.73). Outgoing patchy connections are constrained to an encompassing ellipse which is oriented at the same angle as the orientation preference of the neuron under investigation (Bosking et al., 1997). From Angelucci et al. (2002), we derive the layer-specific spatial extents of these ellipses as shown in Table 4.4.

First, we implement an orientation tuning map for the neurons in our model according to an input-driven scheme following Sadeh

Layer of source population	Major axis [mm]	Minor axis [mm]
2/3A	3.0	1.9
3B	3.0	1.9
4A/B	3.3	2.2
5	4.0	2.3
6	4.0	2.3

Table 4.4: **Layer-dependent extent of patchy connectivity.** Major and minor axis lengths of the ellipse that encompasses the patchy connectivity profile, centered on the source neuron.

and Rotter (2013) (see Section 4.3.2.2 for details). Figure 4.3 shows a top-down view of the V1 model with the imposed orientation tuning map.

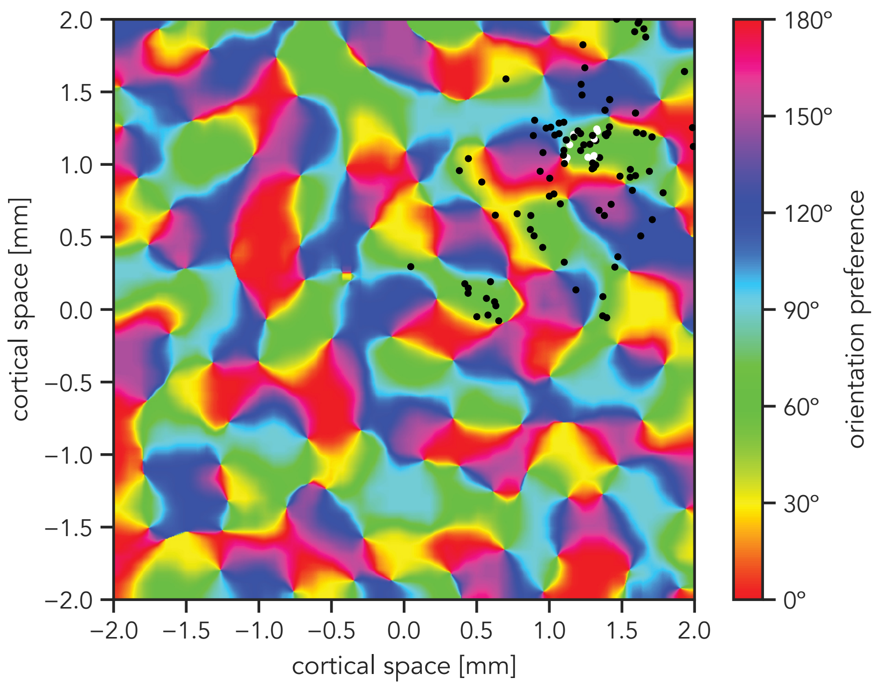

Figure 4.3: **Patchy connectivity links neurons of similar orientation preference.** Colors represent the spatially clustered orientation tuning preference of cortical neurons in our model. White circles represent randomly chosen source neurons with an orientation preference of 60°, black circles represent target neurons for which connections are established in the model. Connections are a superposition of isotropic connections for short distances ($\lesssim 500\,\mu m$) and patchy connections at large distances ($\gtrsim 500\,\mu m$). The latter are encompassed by an ellipse oriented at the same angle as the orientation preference of the source neurons (here, 60°).

Here, the color codes for the orientation preference of the neurons in space. Next, we connect neurons of the same orientation preference with a uniform connection probability within the elliptical bounds given by Table 4.4. Due to the arrangement of iso-orientation domains, this leads to a patchy pattern in the connectivity. In Figure 4.3, we see examples of such patchy connections superimposed with isotropic connectivity as established by a randomly chosen subset of neurons in L2/3E : at short distances ($\lesssim 500\,\mu m$), connections are isotropic and independent of orientation preference. At larger distances ($\gtrsim 500\,\mu m$), connections are preferentially made to neurons of the same orientation preference. Since the preferred orientation of the source neurons under consideration here is $60°$, the encompassing ellipse is also oriented at $60°$.

4.2.2.3 *Short-range push-pull connectivity*

In the sensory input layer 4C, we implement an additional connectivity scheme called *push-pull connectivity* following Troyer et al. (1998) and Martinez et al. (2005). Depending on the input correlations of any two cortical neurons, which here stem from upstream connections that originate in the thalamus, their joint connection probability is either increased or decreased. Functionally, this organization principle is thought to support contrast-invariant responses of cortical neurons in the visual areas.

First, we create a thalamic input model consisting of two excitatory populations: ON neurons fire most strongly when a bright stimulus is centered in their receptive fields, and OFF neurons fire most strongly when a dark stimulus is centered in their receptive fields. The cortical neurons in V1 receive input connections according to a Gabor profile (Equation 4.76) with either ON or OFF thalamic neurons in the center of their receptive field, where the orientation of the Gabor distribution matches the orientation preference of the neuron under consideration (see Figure 4.4).

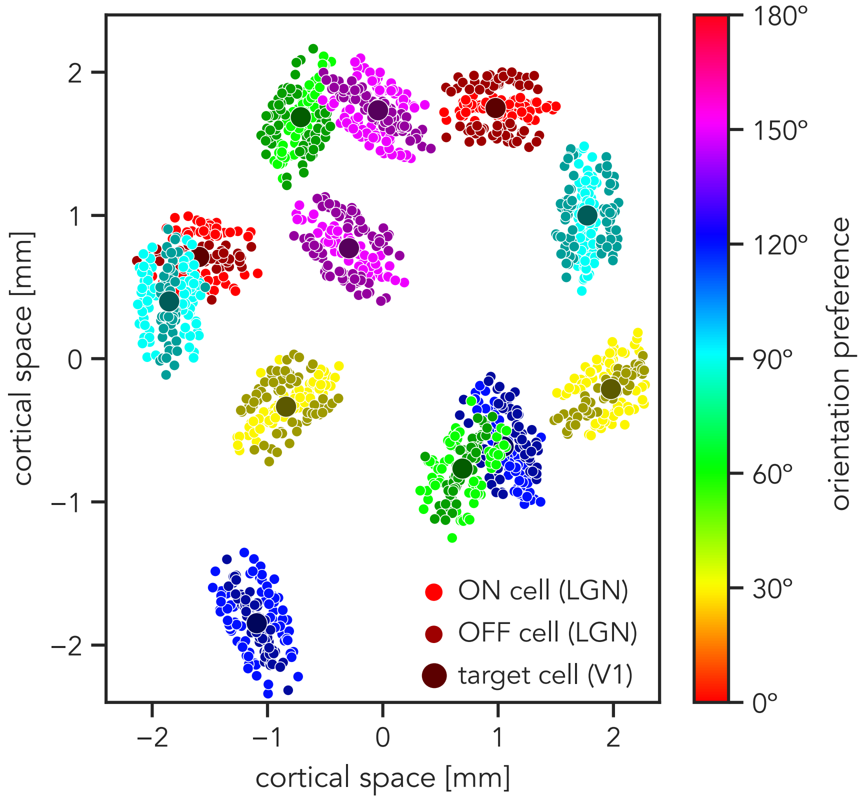

Figure 4.4: **LGN projection patterns to V1 neurons.** Large circles mark V1 neurons, small circles mark ON LGN neurons (bright) or OFF LGN neurons (dark). V1 neurons get input connections from LGN neurons according to a Gabor filter centered on their position, and rotated according to their orientation preference.

Based on these projection patterns, push-pull connections are created between neurons of the same orientation preference. We denote by ζ the fraction of push-pull connections to the total number of recurrent connections, which is a free parameter of the model. For a detailed derivation of the push-pull connectivity scheme, see Section 4.3.2.3.

4.2.3 *Dynamical simulations*

Having defined how neurons of different cell types and in different layers connect with each other, we instantiate a spiking neural network simulation to investigate the dynamical properties of the network. We sample neuron positions randomly inside a $4 \times 4\text{mm}^2$ area matching the densities reported in O'Kusky and Colonnier (1982). Next, we recurrently connect neurons according to the spatial connectivity principles outlined in Section 4.2.2. Finally, we provide a model-external stimulus via cortico-cortical connections which act as a background drive. For all simulations, we use the high-performance neural network simulator NEST.

To study the dynamical behavior, we first aim to find a biologically plausible ground state of activity when no thalamic input is present. To first order, we define such a state by populations exhibiting mean firing rates on the order of 0.1 Hz to 10 Hz. We consider how the network behaves under change of two main parameters: the firing rate of the external drive, ν_{ext}, and the weight of inhibitory recurrent connections relative to the strength of excitatory ones, g.

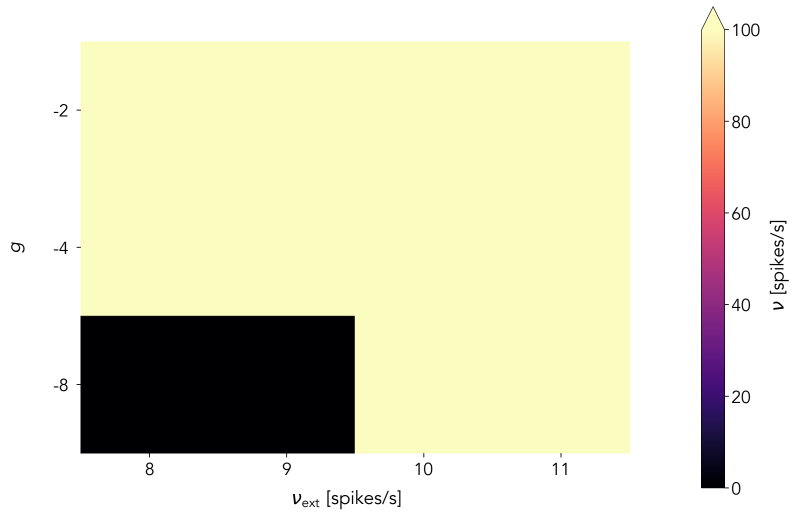

Figure 4.5: **Activity of spiking network of macaque V1.** Mean firing rate ν across all neurons depending on g, the weight of inhibitory recurrent connections relative to the strength of excitatory ones, and ν_{ext}, the rate of external drive.

Independent of the patchiness parameter ψ and the push-pull parameter ζ, Figure 4.5 shows that the model does not settle into a biologically plausible ground state for realistic parameter ranges of the external input ν_{ext} and strength of inhibition g. Indeed, there is a sharp transition when increasing the external input or decreasing the strength of inhibition. Once the threshold is reached, the network activity transitions from a quiescent state to a diverging one in which neurons fire at pathological frequencies of over 100 Hz. To test whether the inhomogeneity in the external input, i.e., in the derived number of cortico-cortical synapses S_A^{cc}, causes this pathological dynamical state, we perform the same simulations with a fixed external in-degree of $S_A^{cc} = 600$ for excitatory populations, and an in-degree of $S_A^{cc} = 500$ for inhibitory connections (Figure 4.6).

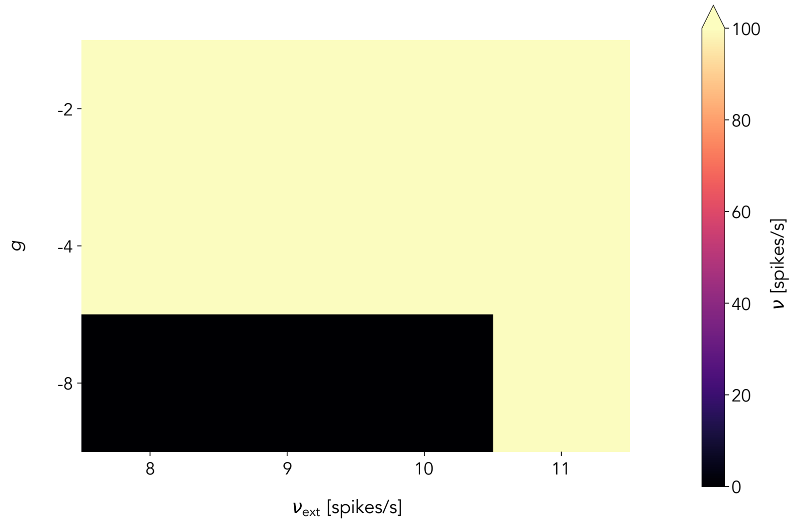

Figure 4.6: **Activity of spiking network of macaque V1 for homogeneous external input degrees.** Indegree of cortico-cortical external connections set to $S_A^{cc} = 600$ for excitatory and $S_A^{cc} = 500$ for inhibitory neurons. Mean firing rate ν across all neurons depending on g, the weight of inhibitory recurrent connections relative to the strength of excitatory ones, and ν_{ext}, the rate of external drive.

Also in the case of homogeneous input, the network dynamics exhibit the same sharp transition between completely quiescent and pathologically active. This suggests that the recurrent cortical connectivity in the model is responsible for the biologically implausible dynamics. We next investigate this hypothesis by considering the targeting patterns of recurrent connections.

TARGET SPECIFICITY We define the target specificity as the preference of a neuron of population B to target excitatory postsynaptic neurons relative to its preference of targeting inhibitory neurons:

$$\mathrm{TS}_{v,B} = \frac{p_{v\mathrm{E},B} - p_{v\mathrm{I},B}}{p_{v\mathrm{E},B} + p_{v\mathrm{I},B}} \in [-1, 1] \, . \tag{4.2}$$

where $p_{v\mathrm{X}}$ is the probability of a presynaptic neuron of population B to make a connection to a postsynaptic neuron in layer v of efficacy $X \in \{\mathrm{E}, \mathrm{I}\}$. Figure 4.7 shows the target specificity of all populations.

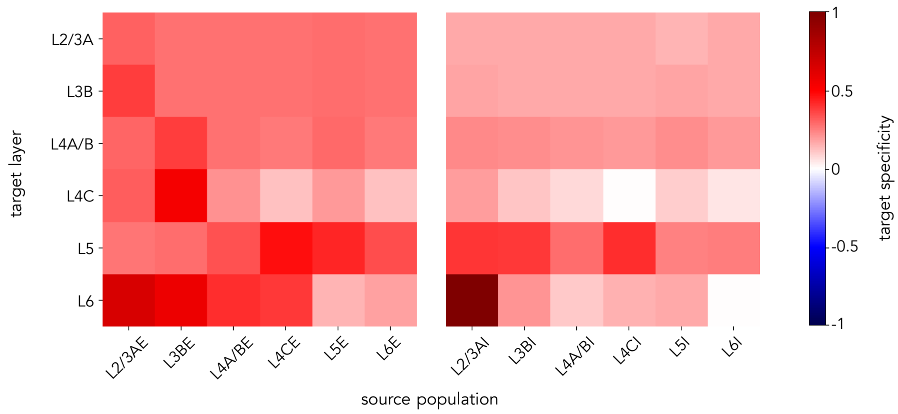

Figure 4.7: **Target specificity of neurons in the model of V1.** A positive (red) TS implies that the neurons in the presynaptic population preferably target excitatory postsynaptic neurons in the target layer; a negative (blue) TS implies that the neurons in the presynaptic population preferably target inhibitory postsynaptic neurons in the target layer.

Strikingly, we find that all populations prefer to target excitatory neurons. This is especially problematic for excitatory presynaptic populations: we suspect that the implied positive feedback loops provide a possible explanation for the observed pathological dynamical behavior of the model. This conclusion is in agreement with Potjans and Diesmann (2014) who amend the connectivity of their cortical microcircuit mode, which they also derived from connectivity data by Binzegger et al. (2004), with additional physiological connectivity data motivated by a discrepancy in target specificity between data sources (see their Figures 4 and S3). Their proposed reorganization precisely leads to a reduction of target specificity, i.e., a stronger targeting of inhibitory neurons by excitatory neurons. The authors found that such a reorganization is essential for asynchronous and irregular activity with realistic firing rates in their model.

4.3 METHODS

Given a layer u in V1, Vanni et al. (2020) provide the number of synapses S^u in that layer. The data by Binzegger et al. (2004) can be used to split these synapses into three categories that depend on the location of the presynaptic neuron:

$$S^u = S^{\mathrm{ia},u} + S^{\mathrm{tha},u} + S^{\mathrm{cc},u} \ , \tag{4.3}$$

where $S^{\mathrm{ia},u}$, $S^{\mathrm{tha},u}$, and $S^{\mathrm{cc},u}$ denote the number of intra-areal, thalamo-cortical, and cortico-cortical synapses, respectively.

First, we focus on the recurrent connectivity of the network model, which is governed by the intra-areal connections. Combining the data by Vanni et al. (2020), which specifies connectivity from presynaptic neurons to synapses, and the data by Binzegger et al. (2004), which specifies connectivity from synapses to postsynaptic neurons, we first derive the total number of synapses between a presynaptic population B and a postsynaptic population A, called S_{AB}. Following the same recipe, we next consider thalamo-cortical connections originating from other brain areas, before turning to cortico-cortical connections. After completing an initial connectome, which specifies all incoming synapses for our model neurons, we introduce the notion of space to our model. Specifically, this means distributing the synapses according to experimentally observed spatial connectivity schemes dominant at different spatial scales.

4.3.1 *Synaptic connectome*

4.3.1.1 *Intra-areal synapses*

SYNAPSES TO POSTSYNAPTIC POPULATIONS Binzegger et al. (2004) provide three main quantities, which we denote by the following for this work:

- $p|^u_{|ij}$, the probability for an existing connection onto a neuron of type i with synapse in layer u to come from a neuron of type j. Note that this probability is conditional on i and u, which we mark by pipes in front of the respective indices. Subsequently, the normalization condition is given by $\sum_j p|^u_{|ij} = 1 \ \forall \, i, u$.

- s_i^u, the total number of synapses onto a neuron of type i that reside in layer u. One can calculate the total in-degree of such a neuron by summing over all layers in which a connection is possible: $\sum_u s_i^u = K_i$.

- c_i, the relative occurrence of neurons of type i by. The normalization condition here reads $\sum_i c_i = 1$.

Investigating the precise numerical values of $p_{|ij}^{|u}$ reveals that given a postsynaptic neuron resides in layer 6 with its synapse in the same layer, the fraction of cortico-cortical presynaptic partners is unrealistically large (0.557). Unfortunately, alternative data for this type of connectivity does not exist in the literature. Therefore, we resort to manually adapting this value, and introduce $\gamma < 0.557$ as a replacement. Other connection probabilities are scaled such that the normalization is fulfilled.

First, we define the probability for an intra-areal synapse (i.e., a synapse for which both the pre- and postsynaptic neurons reside in V1) of efficacy X in layer u to establish a connection that targets population A as

$$p_A^{|u|X} = \frac{\text{\# intra-areal syn. of efficacy } X \text{ in layer } u \text{ targeting pop. } A}{\text{\# intra-areal synapses of efficacy } X \text{ in layer } u} . \tag{4.4}$$

The conditionality on efficacy and layer implies a normalization condition $\sum_A p_A^{|u|X} = 1 \; \forall \; u, X$. This probability can be calculated from the data by Binzegger et al. (2004).

We first note that

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

Thus, $p_A^{|u|X}$ can be calculated as

$$p_A^{|u|X} = \frac{\sum\limits_{i \in A} \sum\limits_{j \in X \cap \mathrm{ia}} N \, c_i \, s_i^u \, p_{|ij}^{|u}}{\sum\limits_{i} \sum\limits_{j \in X \cap \mathrm{ia}} N \, c_i \, s_i^u \, p_{|ij}^{|u}} = \frac{\sum\limits_{i \in A} \sum\limits_{j \in X \cap \mathrm{ia}} c_i \, s_i^u \, p_{|ij}^{|u}}{\sum\limits_{i} \sum\limits_{j \in X \cap \mathrm{ia}} c_i \, s_i^u \, p_{|ij}^{|u}} . \tag{4.5}$$

Here, $j \in X \cap \mathrm{ia}$ is a symbolic notation for all neuron types j that are of efficacy X that reside in V1. The total number of intra-areal synapses in layer u with efficacy X can be read from the denominator:

$$S_X^{\mathrm{ia},u} = \sum_i \sum_{j \in X \cap \mathrm{ia}} N \, c_i \, s_i^u \, p_{|ij}^{|u} \tag{4.6}$$

From this, we calculate the fraction of intra-areal synapses in layer u as

$$f^{\mathrm{ia},u} = \frac{\sum\limits_{i} \sum\limits_{j \in \mathrm{ia}} N \, c_i \, s_i^u \, p_{|ij}^{|u}}{\sum\limits_{i} \sum\limits_{j} N \, c_i \, s_i^u p_{|ij}^{|u}} = \frac{\sum\limits_{i} \sum\limits_{j \in \mathrm{ia}} N \, c_i \, s_i^u \, p_{|ij}^{|u}}{\sum\limits_{i} N \, c_i \, s_i^u} . \tag{4.7}$$

In a similar fashion, we can calculate $f^{\mathrm{cc},u}$, the fraction of cortico-cortical synapses in layer u, as well as $f^{\mathrm{tha},u}$, the fraction of thalamo-cortical synapses in layer u. For these calculations, we first define the reduced fraction of intra-areal and cortico-cortical synapses in layer u as

$$\bar{f}^{\mathrm{ia},u} = \frac{f^{\mathrm{ia},u}}{f^{\mathrm{ia},u} + f^{\mathrm{cc},u}} , \qquad \bar{f}^{\mathrm{cc},u} = \frac{f^{\mathrm{cc},u}}{f^{\mathrm{ia},u} + f^{\mathrm{cc},u}} . \tag{4.8}$$

We can derive the relative fraction of intra-areal excitatory and inhibitory synapses in a given layer u via

$$\not{s}_E^{|u} = \frac{S_E^{\mathrm{ia},u}}{S_E^{\mathrm{ia},u} + S_I^{\mathrm{ia},u}} \qquad \text{and} \qquad \not{s}_I^{|u} = 1 - \not{s}_E^{|u} . \tag{4.9}$$

Note here that

$$S_E^{\mathrm{ia},u} + S_I^{\mathrm{ia},u} = \sum_i \sum_{j \in \mathrm{ia}} N\, c_i\, s_i^u\, p_{|ij}^{|u} . \tag{4.10}$$

With these prerequisites in place, we can finally define

$$\begin{aligned}
\bar{p}_A^{|uX} &:= \not{s}_X^{|u} \cdot p_A^{|u|X} \qquad\qquad (4.11)\\
&= \frac{\sum\limits_i \sum\limits_{j \in X \cap \mathrm{ia}} N\, c_i\, s_i^u\, p_{|ij}^{|u}}{\sum\limits_i \sum\limits_{j \in \mathrm{ia}} N\, c_i\, s_i^u\, p_{|ij}^{|u}} \cdot \frac{\sum\limits_{i \in A} \sum\limits_{j \in X \cap \mathrm{ia}} c_i\, s_i^u\, p_{|ij}^{|u}}{\sum\limits_i \sum\limits_{j \in X \cap \mathrm{ia}} c_i\, s_i^u\, p_{|ij}^{|u}} \\
&= \frac{\sum\limits_{i \in A} \sum\limits_{j \in X \cap \mathrm{ia}} c_i\, s_i^u\, p_{|ij}^{|u}}{\sum\limits_i \sum\limits_{j \in \mathrm{ia}} c_i\, s_i^u\, p_{|ij}^{|u}} .
\end{aligned}$$

We interpret this quantity as the probability that an intra-areal synapse in layer u targets population A and has efficacy X. Here, we have to take into account that the layer resolution provided by Binzegger et al. (2004) is coarser than the one chosen for this model of V1. In particular, layer 2/3 and layer 4 are not divided into the sub-layers 2/3A, 3B, and 4A/B, 4C. Therefore, we need to adapt the formulation of $\bar{p}_A^{|uX}$. First, we define

$$A = A_1 \uplus A_2 \tag{4.12}$$

$$u = u_1 \uplus u_2 \tag{4.13}$$

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, and v is some different layer. For the case where an intra-areal synapse of efficacy X resides in layer v and makes a connection to population A, we split the probability as

$$\bar{p}_A^{|vX} = \bar{p}_{A_1}^{|vX} + \bar{p}_{A_2}^{|vX} \tag{4.14}$$

where

$$\bar{p}_{A_1}^{|vX} = \frac{N_1}{N_1 + N_2}\,\bar{p}_A^{|vX}, \qquad \bar{p}_{A_2}^{|vX} = \frac{N_2}{N_1 + N_2}\,\bar{p}_A^{|vX} \tag{4.15}$$

with N_1 and N_2 being the number of neurons of the respective sub-populations. We here assume that a synapse of efficacy X that resides in layer v uniformly targets neurons of population A. In the case where 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.

$$\bar{p}_A^{|v_1X} = \bar{p}_A^{|v_2X} = \bar{p}_A^{|vX}\,. \tag{4.16}$$

Therefore, Equation 4.14 becomes

$$\bar{p}_A^{|vX} = \underbrace{\bar{p}_{A_1}^{|v_1X}}_{\frac{N_1}{N_1+N_2}\bar{p}_A^{|vX}} + \underbrace{\bar{p}_{A_2}^{|v_1X}}_{\frac{N_2}{N_1+N_2}\bar{p}_A^{|vX}} = \bar{p}_{A_1}^{|v_2X} + \bar{p}_{A_2}^{|v_2X} \tag{4.17}$$

Finally, we need to consider the case in which the synapse lies in the same (sub-)layer u as the postsynaptic neuron. If we take the approach outlined above, the split into sub-layers yields

$$\bar{p}_{A_1}^{|u_1X} > \bar{p}_{A_1}^{|u_2X} \text{ and } \bar{p}_{A_2}^{|u_1X} > \bar{p}_{A_2}^{|u_2X} \tag{4.18}$$

or

$$\bar{p}_{A_1}^{|u_1X} < \bar{p}_{A_1}^{|u_2X} \text{ and } \bar{p}_{A_2}^{|u_1X} < \bar{p}_{A_2}^{|u_2X} \tag{4.19}$$

since

$$\bar{p}_{A_j}^{|u_iX} = \frac{N_j}{N_1 + N_2}\bar{p}_A^{|u_iX} = \frac{N_j}{N_1 + N_2}\bar{p}_A^{|uX} \tag{4.20}$$

is independent of i and thus

$$\bar{p}_{A_1}^{|u_1X} = \bar{p}_{A_1}^{|u_2X}, \quad \bar{p}_{A_2}^{|u_1X} = \bar{p}_{A_2}^{|u_2X}. \tag{4.21}$$

Here, A_i denotes a sub-population in layer u_i. The second inequality of Equation 4.18 and the first inequality of Equation 4.19 are problematic: we expect that the connection probability is highest if the target population is in the same layer as the synapse. To circumvent this issue, we introduce an ad-hoc factor $\alpha = 0.8$ that replaces the relative frequency of number of neurons in the sub-layers. Using α, we define

$$\bar{p}_{A_1}^{|u_1X} = \alpha \cdot \bar{p}_A^{|vX}, \quad \bar{p}_{A_2}^{|u_1X} = (1-\alpha) \cdot \bar{p}_A^{|vX} \tag{4.22}$$

and

$$\bar{p}_{A_1}^{|u_2X} = (1-\alpha) \cdot \bar{p}_A^{|vX}, \quad \bar{p}_{A_2}^{|u_2X} = \alpha \cdot \bar{p}_A^{|vX}. \tag{4.23}$$

PRESYNAPTIC NEURONS TO SYNAPSES From Vanni et al. (2020), we directly obtain the following quantities:

- η^X_{uv} = the relative anatomical connection strength for connections of efficacy X from layer $v \to u$, where v is the layer of the presynaptic neuron and u is the layer in which the intra-areal synapse resides
- S^u = the total number of synapses in layer u

In contrast to the data from Binzegger et al. (2004), the data from Vanni et al. (2020) has a finer layer resolution than our model of V1. Therefore, we can reduce the information to our desired layer resolution by summing the finer relative connection strengths over the sub-layers u and v, respectively, and normalize the expression by dividing by the product of the number of sub-layers of u and v:

$$\eta^X_{uv} = \frac{\sum_{u' \in u, v' \in v} \eta^{\text{fine},X}_{u'v'}}{\#u \# v}. \tag{4.24}$$

From this, we can determine the probability for a synapse of efficacy X terminating in u to be originating in v, $q^{|X}_{|uv}$. As before, the condition on X and u implies the normalization $\sum_v q^{|X}_{|uv} = 1 \,\forall\, u, X$, which can be guaranteed by calculating

$$q^{|X}_{|uv} = \frac{\eta^X_{uv}}{\sum\limits_v \eta^X_{uv}} \,. \tag{4.25}$$

The relative anatomical connection strengths derived in Vanni et al. (2020) were obtained by combining a large body of studies based retrograde tracer experiments, where a dye is injected into brain tissue, which is subsequently taken up by synapses and transported along the axon towards the cell body. Since the injection site itself still contains the dye even after update of the surrounding synapses, neurons at the injection sites cannot be identified. Consequently, this method underestimates the number of neurons within the uptake zone of the dye, which leads to an underestimation of intra-layer connections in our model. We therefore devise a scheme to correct for this underestimation. Following the findings of Markov et al. (2011), we assume an exponential decrease in density of labeled neurons with distance from the injection site:

$$l^X_{uv}(r) = l^X_{uv,0}\, e^{-\frac{r}{\lambda_{AB}}} \,. \tag{4.26}$$

Here, $l^X_{uv}(r)$ denotes the density of labeled neurons in layer u after injecting a hypothetical tracer that is taken up by neurons of efficacy X in layer v, $l^X_{uv,0}$ marks the zero-distance density, and λ_{AB} denotes the characteristic length scale of spatial decay. For the anatomical connection strength, this implies

$$\eta_{uu}^{X} \propto e^{-\frac{r}{\lambda_{AB}}} . \tag{4.27}$$

We make the following simplifying assumptions:

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

A graphical depiction of these assumptions is shown in the top panel of Figure 4.8; the bottom panel of Figure 4.8 defines the coordinate system used in the following.

We formally set $r_l = \infty$. Given these assumptions, the number of synapses of efficacy X in layer u as measured by tracing studies is

$$S_u^X \sim \underbrace{\lim_{r_l \to \infty} \int_{Z(r_l)} e^{-\frac{r}{\lambda_{AB}}} \mathrm{d}V - \int_{S(r_{\text{blob}})} e^{-\frac{r}{\lambda_{AB}}} \mathrm{d}V}_{=:\, I(r_{\text{blob}})} , \tag{4.28}$$

where $Z(r_l)$ denotes the cylinder with radius r_l that approximates layer u, and $S(r_{\text{blob}})$ denotes the sphere with radius r_{blob} which approximates the uptake zone of the dye. We now account for the missing connections inside the blob of dye, calculating a corrected number of connections:

$$S_u^{X,\text{corr}} = S_u^X \cdot \frac{I(0)}{I(r_{\text{blob}})} . \tag{4.29}$$

Consequently, the intra-layer relative anatomical connection strengths get corrected:

$$\eta_{uu}^{X,\text{corr}} = \eta_{uu}^X \cdot \frac{I(0)}{I(r_{\text{blob}})} . \tag{4.30}$$

This leaves the issue of evaluating the integral $I(r_{\text{blob}})$. First, we consider the spherical blob of dye which is centered on the origin of the coordinate system. Since we use spherical coordinates, excluding the dye simply imposes the constraint $r > r_{\text{blob}}$. Next, we consider the layer boundaries. If $r < L_u/2$, the integration is not constrained by the layer boundaries. In the case of $r > L_u/2$, the height of the integration is constrained: $|z| \overset{!}{\leq} \frac{L_u}{2}$. In spherical coordinates, z is expressed as $z = r\cos\theta$, thereby implying a constraint on the azimuth angle θ:

$$r|\cos\theta| \leq \frac{L_u}{2} \quad \Rightarrow \quad \arccos\left(\frac{L_u}{2r}\right) \leq \theta \leq \pi/2 , \tag{4.31}$$

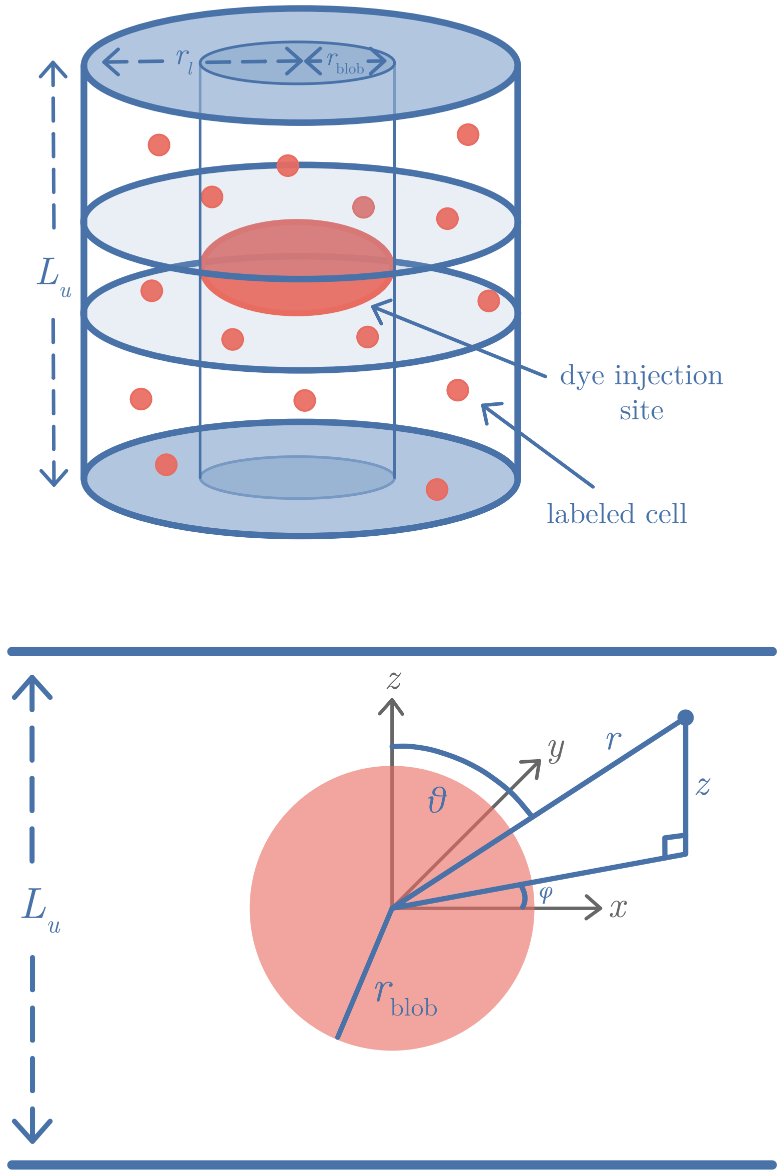

Figure 4.8: **Amending estimated strengths of connectivity for short distances. Top** Model assumptions: a sphere of dye in the center of cylindrical layers. **Bottom** Spherical coordinate system centered on the blob.

where we use $0 \leq \theta \leq \pi/2$. For the integration, we need to consider two distinct cases. If the radius of the blob is smaller than or equal to half the laminar thickness ($r_{\text{blob}} \leq \frac{L_u}{2}$), the entire blob is contained inside the layer. We term this case the *spherical case*, as the blob boundary within the layer boundaries describes a sphere. If, however, the radius of the blob exceeds this limit ($r_{\text{blob}} > \frac{L_u}{2}$), only part of the sphere will be contained in the layer boundaries as the top and bottom are cut off. We aptly name this the *marshmallow case*.

Before solving the integrations for the two cases, let us first derive the solution to two generic integrals, $I_1(r_1)$ and $I_2(r_1, r_2)$.

$$\begin{aligned} I_1(r_1) := \int_{r_1}^{\infty} r e^{-r/\lambda}\,\mathrm{d}r &= -\lambda r e^{-r/\lambda}\Big|_{r_1}^{\infty} + \lambda \int_{r_1}^{\infty} e^{-r/\lambda}\,\mathrm{d}r \\ &= \left[\lambda r_1 + \lambda^2\right] e^{-r_1/\lambda} \end{aligned} \tag{4.32}$$

$$\begin{aligned} I_2(r_1, r_2) := \int_{r_1}^{r_2} r^2 e^{-r/\lambda}\,\mathrm{d}r &= -\lambda r^2 e^{-r/\lambda}\Big|_{r_1}^{r_2} + 2\lambda \int_{r_1}^{r_2} r e^{-r/\lambda}\,\mathrm{d}r \\ &= \left[\lambda r^2 + 2\lambda^2 r\right] e^{-r/\lambda}\Big|_{r_2}^{r_1} + 2\lambda^2 \int_{r_1}^{r_2} e^{-r/\lambda}\,\mathrm{d}r \\ &= \left[\lambda r^2 + 2\lambda^2 r + 2\lambda^3\right] e^{-r/\lambda}\Big|_{r_2}^{r_1} \end{aligned} \tag{4.33}$$

For the spherical case, we first note that we can split the integral into two regions. For $r_{\text{blob}} \le r \le \frac{L_u}{2}$, we integrate over the whole sphere, and θ is unconstrained. For $r > \frac{L_u}{2}$, we need to enforce Equation 4.31. Adding the two parts of the integral, exploiting the symmetry of the system, and performing some simplifications, we obtain

$$\begin{aligned} I_{\text{sphere}}(r_{\text{blob}}) = &\int_{r_{\text{blob}}}^{\frac{L_u}{2}} \int_0^{2\pi} \int_0^{\pi} e^{-\frac{r}{\lambda}} r^2 \sin(\theta)\,\mathrm{d}\theta\mathrm{d}\varphi\mathrm{d}r \\ &+ 2\int_{\frac{L_u}{2}}^{\infty} \int_0^{2\pi} \int_{\arccos\left(\frac{L_u}{2r}\right)}^{\pi/2} e^{-\frac{r}{\lambda}} r^2 \sin(\theta)\,\mathrm{d}\theta\mathrm{d}\varphi\mathrm{d}r \\ = &\,4\pi \int_{r_{\text{blob}}}^{\frac{L_u}{2}} e^{-\frac{r}{\lambda}} r^2 \mathrm{d}r \\ &+ 2\cdot 2\pi \int_{\frac{L_u}{2}}^{\infty} e^{-\frac{r}{\lambda}} r^2 \Bigg[-\cos(\pi/2) \\ &\qquad + \cos\left(\arccos\left(\frac{L_u}{2r}\right)\right)\Bigg]\,\mathrm{d}r \\ = &\,4\pi \left\{ \underbrace{\int_{r_{\text{blob}}}^{\frac{L_u}{2}} e^{-\frac{r}{\lambda}} r^2 \mathrm{d}r}_{I_2(r_{\text{blob}}, \frac{L_u}{2})} + \frac{L_u}{2} \underbrace{\int_{\frac{L_u}{2}}^{\infty} e^{-r/\lambda} r\,\mathrm{d}r}_{I_1(\frac{L_u}{2})} \right\} \\ = &\,4\pi \Bigg\{ \left[\lambda r^2 + 2\lambda^2 r + 2\lambda^3\right] e^{-r/\lambda}\Big|_{r_{\text{blob}}}^{\frac{L_u}{2}} \\ &\qquad + \frac{L_u}{2}\left[\lambda \frac{L_u}{2} + \lambda^2\right] e^{\frac{L_u}{2\lambda}} \Bigg\}\,. \end{aligned} \tag{4.34}$$

In this derivation, we write λ instead of λ_u^X to enhance readability.

In the marshmallow case, $r \geq r_{\text{blob}} > \frac{L_u}{2}$ holds. Therefore, we do not need to split the integral as in the spherical case. Thus, it becomes

$$\begin{aligned} I_{\text{marshmallow}}(r_{\text{blob}}) &= 2\int_{r_{\text{blob}}}^{\infty}\int_{0}^{2\pi}\int_{\arccos\left(\frac{L_u}{2r}\right)}^{\pi/2} e^{-\frac{r}{\lambda}} r^2 \sin(\theta)\,\mathrm{d}\theta\mathrm{d}\varphi\mathrm{d}r \\ &= 4\pi\frac{L_u}{2}\underbrace{\int_{r_{\text{blob}}}^{\infty} re^{-r/\lambda}\,\mathrm{d}r}_{I_1(r_{\text{blob}})} \\ &= 4\pi\frac{L_u}{2}\left[\lambda r_{\text{blob}} + \lambda^2\right] e^{r_{\text{blob}}/\lambda}\ . \end{aligned} \tag{4.35}$$

$I_{\text{case}}(0)$ can be trivially obtained for both cases via

$$I_{\text{case}}(0) = \lim_{r_{\text{blob}}\to 0} I_{\text{case}}(r_{\text{blob}}) \tag{4.36}$$

COMBINING THE DATA At this point, we have all the necessary ingredients for deriving the recurrent connectome in our model. Putting them together yield the total number of synapses from population B to A:

$$S^{\text{ia}}_{AB} = \sum_u f^{\text{ia},u}\, S^u\, \cancel{s}^{|u}_{X}\, q^{|X}_{|uv}\, p^{|u|X}_{A}\ , \tag{4.37}$$

where v is the layer that the source population B resides in, and X is B's efficacy. Using the definition $\bar{p}^{|uX}_{A}$ (Equation 4.12), this expression reduces to

$$S^{\text{ia}}_{AB} = \sum_u f^{\text{ia},u}\, S^u\, q^{|X}_{|uv}\, \bar{p}^{|uX}_{A}\ . \tag{4.38}$$

When calculating the number of synapses from population B to A in area V1, we need to replace

$$f^{\text{ia},u}\, S^u \quad \text{with} \quad \bar{f}^{\text{ia},u}\cdot(S^u - S^u_{\text{tha}})\ . \tag{4.39}$$

4.3.1.2 *Thalamo-cortical synapses*

Next, we consider synaptic inputs to model neurons that stem from the thalamus, i.e., the input region from which sensory data from the retina is relayed. We follow the recipe laid our for intra-areal synapses, first estimating the distribution of synapses given a presynaptic population, then estimating the strength of connectivity from a given synapse to the postsynaptic populations, and finally combining the two.

PRESYNAPTIC POPULATIONS TO SYNAPSES We use an analogous notation as in Section 4.3.1.1 and denote the probability for a synapse with its presynaptic partner in the thalamus that resides in layer u to establish a connection with a cortical population A as

$$p^{|u,\text{tha}}_{A} = \frac{\#\ \text{thalamo-cortical synapses in layer } u \text{ targeting population } A}{\#\ \text{thalamo-cortical synapses in layer } u}.$$

(4.40)

Similarly to before, we calculate

$$p_A^{|u,\mathrm{tha}} = \frac{\sum_{i\in A}\sum_{j\in\mathrm{tha}} N\, c_i\, s_i^u\, p_{|ij}^{|u}}{\sum_i \sum_{j\in\mathrm{tha}} N\, c_i\, s_i^u\, p_{|ij}^{|u}} = \frac{\sum_{i\in A}\sum_{j\in\mathrm{tha}} c_i\, s_i^u\, p_{|ij}^{|u}}{\sum_i \sum_{j\in\mathrm{tha}} c_i\, s_i^u\, p_{|ij}^{|u}}\,, \tag{4.41}$$

where $j \in$ tha encompasses all thalamic neuron types identified in Binzegger et al. (2004). We can again read off the total number of thalamo-cortical synapses from the denominator:

$$S^{\mathrm{tha},u} = \sum_i \sum_{j\in\mathrm{tha}} N\, c_i\, s_i^u\, p_{|ij}^{|u}\,, \tag{4.42}$$

with which the fraction of thalamo-cortical synapses that terminate in layer u with respect to the total number of synapses terminating in this layer can be computed as

$$f^{\mathrm{tha},u} = \frac{\sum\limits_i \sum\limits_{j\in\mathrm{tha}} N\, c_i\, s_i^u\, p_{|ij}^{|u}}{\sum\limits_i N\, c_i\, s_i^u}\,. \tag{4.43}$$

Remembering that Binzegger et al. (2004) uses a coarser layer resolution as our model, we need to modify the connection probabilities $p_A^{|u,\mathrm{tha}}$ as in Equation 4.14 and Equation 4.16 except for synapses terminating in layer 4. Here, we assume that synapses which terminate in a sub-layer of L4 also only make connections within that sub-layer, and accordingly set

$$p_{\mathrm{L4A/BX}}^{|\mathrm{L4A/B}} = p_{\mathrm{L4X}}^{|\mathrm{L4}}, \quad p_{\mathrm{L4CX}}^{|\mathrm{L4A/B}} = 0 \tag{4.44}$$

as well as

$$p_{\mathrm{L4CX}}^{|\mathrm{L4C}} = p_{\mathrm{L4X}}^{|\mathrm{L4}}, \quad p_{\mathrm{L4A/BX}}^{|\mathrm{L4C}} = 0. \tag{4.45}$$

SYNAPSES TO POSTSYNAPTIC POPULATIONS Due to the finer layer resolution of the data by Vanni et al. (2020), we collect the termination data of the sublayers of L2/3, L3B, L4A/B, and L4C, which are the main termination layers for thalamo-cortical projections in our model. First, we define

$$g^{\mathrm{tha,L2/3A}} = \frac{\sum_v \sum_{u\in\mathrm{L2/3A}} \eta_{uv}^{\mathrm{tha}}}{\sum_v \sum_{u\in\mathrm{L2/3}} \eta_{uv}^{\mathrm{tha}}}, \quad g^{\mathrm{tha,L3B}} = \frac{\sum_v \sum_{u\in\mathrm{L3B}} \eta_{uv}^{\mathrm{tha}}}{\sum_v \sum_{u\in\mathrm{L2/3}} \eta_{uv}^{\mathrm{tha}}} \tag{4.46}$$

and $g^{\mathrm{tha,L4A/B}}, g^{\mathrm{tha,L4C}}$ analogously. Using these relative connection strengths, we calculate the number of thalamo-cortical synapses that terminate L4A/B as

$$S^{\mathrm{tha,L4A/B}} = g^{\mathrm{tha,L4A/B}}\, S^{\mathrm{tha,L4}} = \frac{g^{\mathrm{tha,L4A/B}}}{f^{\mathrm{tha,L4A/B}}}\, S^{\mathrm{tha,L4}}\, f^{\mathrm{tha,L4A/B}}\,, \tag{4.47}$$

and analogously for the other model layers L2/3A, L3B, and L4C. For convenience, we define

$$\bar{g}^{\text{tha},u} := \frac{g^{\text{tha},u}}{f^{\text{tha},u}} \quad \forall u \in \{\text{L2/3A, L3B, L4A/B, L4C}\} \tag{4.48}$$

where we set $\bar{g}^{\text{tha},u} = 1$ for $u \in \{\text{L1, L5, L6}\}$. This allows us to generally write the number of synapses in layer u as

$$S^{\text{tha},u} = \bar{g}^{\text{tha},u} f^{\text{tha},u} S^u \,. \tag{4.49}$$

Analogous to the intra-areal connectivity, we finally compute the probability that a thalamo-cortical synapse that terminates in layer u originates from the thalamic sub-population v:

$$q^{\text{tha}}_{|uv} = \frac{\eta^{\text{tha}}_{uv}}{\sum_v \eta^{\text{tha}}_{uv}} \,. \tag{4.50}$$

COMBINING THE DATA Similar to Equation 4.37, we combine the data to obtain the number of thalamo-cortical synapses that target population A which originate in sub-population v as

$$S^{\text{tha}}_{A,v} = \sum_u \bar{g}^{\text{tha},u} f^{\text{tha},u} S^u q^{\text{tha}}_{|uv} p^{|u,\text{tha}}_A \,. \tag{4.51}$$

Since our model does not distinguish between the different thalamic sub-populations, we marginalize over v using that $\sum_v q^{\text{tha}}_{|uv} = 1$:

$$S^{\text{tha}}_{A} = \sum_u \bar{g}^{\text{tha},u} f^{\text{tha},u} S^u p^{|u,\text{tha}}_A \,. \tag{4.52}$$

4.3.1.3 *Cortico-cortical synapses*

Finally, we seek to estimate the cortico-cortical input that neurons in our model of V1 get. Since we are here only interested in the input summed over all areas of origin, we only need to consider the data set by Binzegger et al. (2004) describing the connectivity from synapses to postsynaptic populations combined with the cortico-cortical synapse density in the different layers. Analogous to the derivations for intra-areal and thalamo-cortical connectivity, we start by defining the probability for a cortico-cortical synapse in layer u to establish a connection with population A as

$$p^{|u,\text{cc}}_A = \frac{\#\text{ cortico-cortical synapses in layer } u \text{ targeting population } A}{\#\text{ cortico-cortical synapses in layer } u}\,, \tag{4.53}$$

which can be evaluated via

$$p^{|u,\text{cc}}_A = \frac{\sum_{i\in A} N\, c_i\, s^u_i\, p^{|u}_{|i,\text{cc}}}{\sum_i N\, c_i\, s^u_i\, p^{|u}_{|i,\text{cc}}} = \frac{\sum_{i\in A} c_i\, s^u_i\, p^{|u}_{|i,\text{cc}}}{\sum_i c_i\, s^u_i\, p^{|u}_{|i,\text{cc}}} \,. \tag{4.54}$$

Here, $p^{|u}_{|i,\mathrm{cc}}$ is the probability of a synapse in layer u that targets population i has a presynaptic neuron of cortico-cortical origin. We read off the total number of cortico-cortical synapses in layer u as

$$S^{\mathrm{cc},u} = \sum_i N\, c_i\, s_i^u\, p^{|u}_{|i,\mathrm{cc}} \,, \tag{4.55}$$

and calculate the fraction of cortico-cortical synapses that terminate in layer u as

$$f^{\mathrm{cc},u} = \frac{\sum_i N\, c_i\, s_i^u\, p^{|u}_{|i,\mathrm{cc}}}{\sum_i N\, c_i\, s_i^u} \,. \tag{4.56}$$

As before, the coarser layer resolution demands splitting the probabilities $p_A^{|u,\mathrm{cc}}$ between the sublayers akin to Equation 4.14 and Equation 4.16.

Putting the ingredients together, we arrive at the number of cortico-cortical synapses targeting population A:

$$S_A^{\mathrm{cc}} = \sum_u f^{\mathrm{cc},u}\, S^u\, p_A^{|u,\mathrm{cc}} \,. \tag{4.57}$$

4.3.1.4 *Redistributing synapses*

Looking closer at the derived number of synapses that target the modeled populations, we observe that the fraction of intra-areal, thalamocortical, and cortico-cortical synapses averaged over all populations yield

$$P^{\mathrm{ia}}_{\mathrm{model}} = 75\%, \quad P^{\mathrm{cc}}_{\mathrm{model}} = 21.5\% \quad P^{\mathrm{th}}_{\mathrm{model}} = 3.5\% \,. \tag{4.58}$$

We can compare these numerical results to Markov et al. (2011) who determine the same fractions experimentally using retrograde tracer injections in V1. They report the following values:

$$P^{\mathrm{ia}}_{\mathrm{Markov}} = 85\%, \quad P^{\mathrm{cc}}_{\mathrm{Markov}} = 13\% \quad P^{\mathrm{th}}_{\mathrm{Markov}} = 2\% \,. \tag{4.59}$$

To match the experimentally obtained fractions, we redistribute synapses from cortico-cortical to intra-areal.

4.3.2 *Spatial connectivity*

We consider three types of recurrent spatial connectivity in our model. First, we implement medium-range isotropic connectivity that decreases in strength with distance between somata. Next, we add long-range patchy connectivity which establishes synapses between neurons with similar tuning preferences. Finally, we introduce short-range push-pull connectivity based on input correlations of cortical neurons.

4.3.2.1 *Medium-range isotropic connectivity*

The aim of this section is to derive the parameters of the isotropic intra-areal connectivity. We assume an exponential decay of connection probability with distance:

$$P_{AB}(r) = P^0_{AB}\, e^{-\frac{r}{\lambda_{AB}}} \tag{4.60}$$

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.

We first note that, given the quantities derived in the previous section and assuming a characteristic length λ_{AB}, we can calculate P^0_{AB} with the following ansatz:

$$S^{\text{ia}}_{AB} = \rho_A N_B \int_0^{2\pi} \int_0^{\infty} P^0_{AB}\, r e^{-\frac{r}{\lambda_{AB}}}\, \mathrm{d}r \mathrm{d}\varphi \tag{4.61}$$

$$= 2\pi \rho_A N_B {\lambda_{AB}}^2 P^0_{AB}\,, \tag{4.62}$$

where ρ_A is the density of postsynaptic neurons and N_B is the number of presynaptic neurons. Solving Equation 4.62 for P^0_{AB} yields the peak connection probability at zero distance. Thus, the task left is determining the numeric values for λ_{AB}.

We usually interpret Equation 4.60 along the lines suggest by Equation 4.62: given an arbitrary presynaptic (source) neuron, $P_{AB}(r;\lambda_{AB})$ describes the probability to establish a connection with a neuron of the postsynaptic (target) population. Note that this relation also holds in the reverse direction: the distance-dependent connection probability between a given target neuron in population A and a putative source neuron in population B follows Equation 4.60 as well. In this case, Equation 4.62 needs to be modified, replacing ρ_A by ρ_B (the density of presynaptic neurons), and replacing N_B by N_A (the number of postsynaptic neurons). The two equations are mathematically equivalent since

$$\rho_A N_B = \rho_A \rho_B \cdot \text{Area} = N_A \rho_B. \tag{4.63}$$

Here, Area denotes the area under consideration.

In principle, the characteristic lengths are specific to all pairs of populations A, B. However, the full matrix $(\lambda_{AB})_{A,B}$ cannot be determined since the data on some pairs of populations is lacking. Thus, we need to introduce some simplifying assumptions.

INHIBITORY PRESYNAPTIC NEURONS We start by considering connections in which the presynaptic neuron is inhibitory. For lateral connections of these neurons, precise data is generally sparse. Packer and Yuste (2011) measured the distance-dependent connection probability between paravalbumin-positive neurons and pyramidal cells in mouse cortex. Fitting an exponential decay with distance to this data

(Figure 4.9) reveals a good agreement with our modeling assumption (Equation 4.60).

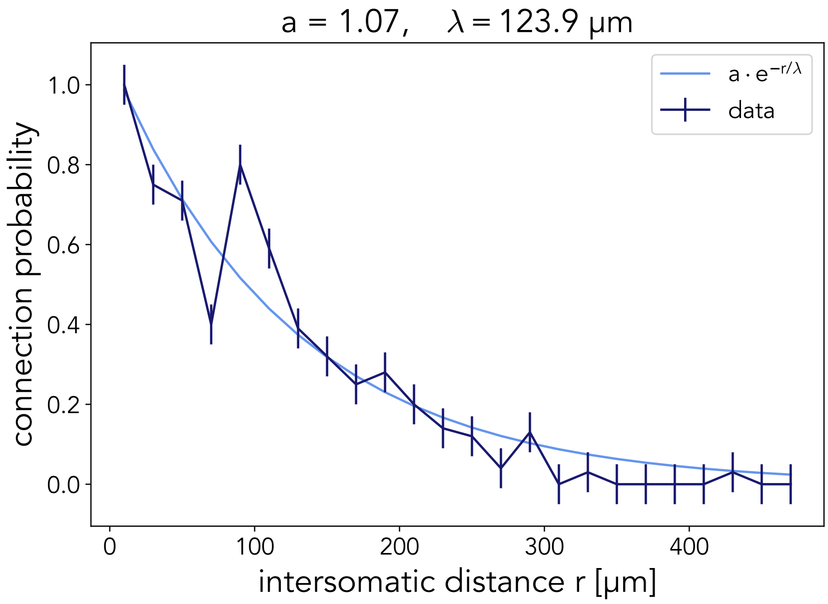

Figure 4.9: **Distance-dependent connection probability.** Navy points are experimental data for inhibitory presynaptic neurons in layer 2/3 (Packer and Yuste, 2011), light blue curves show exponential fits.

Unfortunately, data of this quality are not available for macaque V1. Kritzer et al. (1992) studied inter- and intra-laminar connections established by GABAergic neurons in macaque V1 by retrogradely tracing these neurons, and comparing the spread of labeling from injections into supragranular, granular and infragranular layers. The authors observe that the labeled neurons spread in a circle centered on the injection site, irrespective of its layer. Further, most of the heavily labeled neurons cluster in a smaller circle, also centered on the injection site, consistent with our picture of decreasing connection probability with distance. While this spatial spread is of equal size across all layers for injection sites in the granular layer, the spread gets increasingly smaller with laminar distance for injection sites in supra- and infragranular layers. The data on circle radii are summarized in Table 4.5.

These findings give a good *qualitative* understanding of the projection of inhibitory neurons. To obtain *quantitative* estimates, we combine these insights with the absolute value of λ for inhibitory connections originating in L2/3 and terminating in the same layer.

	Injection site		
Source layer	**Supragranular**	**Granular**	**Infragranular**
L2	0.6	0.3	0.25
L3	0.6	0.3	0.3
L4A/B	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 4.5: **Estimated spread of lateral connectivity of inhibitory presynaptic neurons in** mm **from Kritzer et al. (1992).** The columns of spatial spread refer to the different injection sites under consideration.

Denoting the number of intra-areal synapses within a radius R between two populations A, B by $S_{AB}^{\mathrm{ia},R}$, we can calculate the fraction of all connections that are established within a circle of radius r as

$$P_{AB}^{R} = \frac{S_{AB}^{\mathrm{ia},R}}{S^{\mathrm{ia}_{AB}}} \; . \tag{4.64}$$

where v denotes the layer in which the population A resides. Plugging in

$$S_{AB}^{\mathrm{ia},R} = \rho_A N_B \int_0^{2\pi} \int_0^R P_{AB}^0 \, r e^{-\frac{r}{\lambda_{vB}}} \, \mathrm{d}r \mathrm{d}\varphi \tag{4.65}$$

$$= 2\pi \rho_A N_B \left({\lambda_{vB}}^2 - \lambda_{vB}(\lambda_{vB} + R) e^{-\frac{R}{\lambda_{vB}}} \right) , \tag{4.66}$$

and Equation 4.62, we obtain

$$P_{AB}^{R} = 1 - \left(\frac{R}{\lambda_{vB}} + 1\right) . \exp\left(-\frac{R}{\lambda_{vB}}\right) . \tag{4.67}$$

Taking $\lambda_{\mathrm{L2/3,L2/3I}}$ =123.9 µm from fitting to the data by Packer and Yuste (2011) and plugging in R =600 µm from Table 4.5, we obtain

$$P_{\mathrm{L2/3X,L2/3}}^{600\,\mu\mathrm{m}} \approx 0.95 \; . \tag{4.68}$$

In the context of our model, we interpret this as follows: only neurons falling within the radius in which 95% of target neurons reside are identified as labeled by the tracer. Assuming that the same relation also holds for identifying labeled cells in the other layers, we can determine the lacking characteristic lengths by solving

$$1 - \left(\frac{R_{vB}}{\lambda_{vB}} + 1\right) \exp\left(\frac{R_{vB}}{\lambda_{vB}}\right) = 0.95 \tag{4.69}$$

for λ_{vB}, where R_{vB} is given by Table 4.5. Thereby, we obtain the characteristic lengths of connectivity from inhibitory presynaptic neurons for all layers (Table 4.3).

EXCITATORY PRESYNAPTIC NEURONS Next, we turn to connections in which the presynaptic neurons are excitatory. Sincich and Blasdel (2001) provide image-based data on the spread lateral connectivity in adult squirrel and owl monkeys. The authors use the bidirectional (i.e., retro- and anterograde) tracer biocytin injected into layer 3 of primary visual cortex to label axons and cells (left panels of Figure 4.2). Their images thus describe the termination pattern of lateral projections with post and presynaptic neurons in L3. Assuming that the target neurons are uniformly distributed in the cortical tissue, we employ Peter's rule, which states that the proximity of axons to dendrites is a proxy for connections between neurons. Thus, the darkness (i.e., inverse luminosity) of the staining in the left panels of Figure 4.2 are a proxy for the connection probability. We fit the isotropic function

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

to the inverse luminosity to obtain estimates of the characteristic length of connectivity. Alongside λ, the parameters c (a pre-factor), x_0 and y_0 (the coordinated of the point of highest connection probability), and d (an offset) are optimized. This yields all characteristic lengths for excitatory neurons residing in layer 3 (Table 4.6).

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 4.6: **Characteristic length scales for lateral connections of excitatory presynaptic neurons in layer 3 derived from data by Sincich and Blasdel (2001).**

To estimate the horizontal connectivity from excitatory neurons outside of layer 3, we have to turn to additional retro- and anterograde tracing studies. In the following, we consider Blasdel et al. (1985) for neurons outside of layer 4C, and Fitzpatrick et al. (1985) for neurons inside layer 4C. Unfortunately, these data do not describe the detailed decay of connectivity with distance, only providing estimates of the maximal range of connections. First, we qualitatively group the range of connections into the categories *short*, *medium*, and *long*, depending on whether their maximal range is below 300 µm, below 600 µm, or above 600 µm, respectively (Table 4.7).

	Source layer					
Target population	**L2/3A**	**L3B/L4A**	**L4B**	**L5A**	**L5B**	**L6**
L2/3A	long	med.	short	short	long	short
L3B	med.	long	med.	med.	med.	med.
L4A/B	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 4.7: **Qualitative spread of horizontal connectivity from excitatory neurons.** Maximal extent of axons smaller than 300 µm (short), smaller than 600 µm (med.), or above 600 µm (long) as reported by Blasdel et al. (1985) and Fitzpatrick et al. (1985).

Then, we assign a representative characteristic length scale derived from the data by Sincich and Blasdel (2001) to each group: 141 µm for short-range, 215 µm for medium-range, and 261 µm for long-range connections. The characteristic lengths for populations of excitatory presynaptic neurons are summarized in Table 4.2

4.3.2.2 *Long-range patchy connectivity*

In addition to isotropic connectivity governed by a uni-directional exponential decay with distance, we connect neurons in our model with long-range patchy connectivity. We first define a scheme with which to assign preferred stimulus orientations to neurons in our model, thereby forming a spatially clustered *orientation tuning map*, where neurons of similar orientation preference are close to each other. Building on this foundation, we introduce long-range connections between iso-orientation domains in the orientation tuning map, i.e., patchy connectivity.

ORIENTATION TUNING MAP To generate an orientation tuning map, we follow the approach outlined in Sadeh and Rotter (2013). There, an orientation tuning map emerges naturally from the targeting patterns of thalamic input to visual cortex. We start by creating a grid of N cortical neurons in a square of size $L \times L$. Next, we create a similar grid of $N_{\text{th}} < N$ thalamic neurons in a square of size $L' \times L'$, and jitter both the horizontal and vertical coordinates with displacements drawn uniformly from the range $[-a, a]$. Choosing $L' > L$ avoids boundary effects. Thalamic neurons then connect to cortical neurons with a spatial weight distribution w given by

$$w = w_0 \cdot e^{-\frac{d^2}{2\sigma^2}} , \tag{4.70}$$

where d denotes the horizontal distance between a thalamic and cortical neuron, w_0 represents a parameter governing the peak connection weight, and the standard deviation σ controls the spatial spread (Figure 4.10).

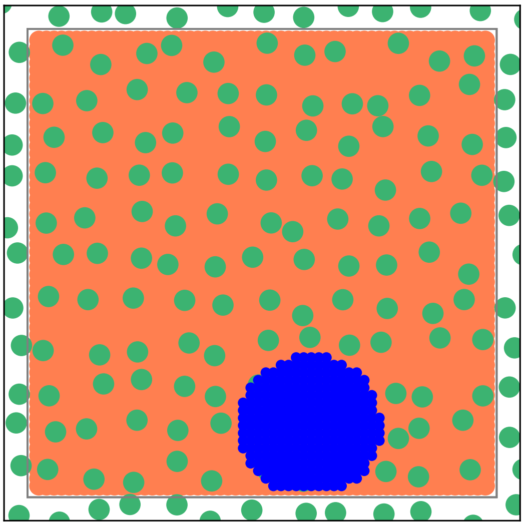

Figure 4.10: **Connectivity between thalamic and cortical neurons which induces an orientation tuning map.** Cortical neurons in orange, thalamic neurons in green. Connections of one exemplary thalamic neuron to all cortical neurons within 2σ around its location in blue.

The thalamic neurons provide a sinusoidal current $I(t)$ to all connected cortical neurons following

$$I(t) = \hat{I} \sin(2\pi \cdot f_t + \phi_o) \ , \tag{4.71}$$

where $\hat{I}$ is the amplitude, f_t is the temporal frequency, and ϕ_o is the phase. To emulate a moving bar stimulus, the phase ϕ_o can be adjusted depending on the desired orientation $\theta \in \{0°, 30°, 60°, 90°, 120°, 150°\}$ as follows:

$$\phi(\theta) = \frac{360}{2\pi} \cdot f_s \cdot \langle \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix}, \begin{pmatrix} x \\ y \end{pmatrix} \rangle \ . \tag{4.72}$$

Here, f_s denotes the spatial frequency, and (x, y) mark the coordinates of the thalamic neuron in question. The orientation tuning map is generated by determining the preferred orientation, i.e., the orientation θ under which a given neuron exhibits the largest increase in membrane potential V_m, for all neurons. For the grid of cortical neurons defined above, this yields the orientation tuning map displayed in the left panel of Figure 4.11.

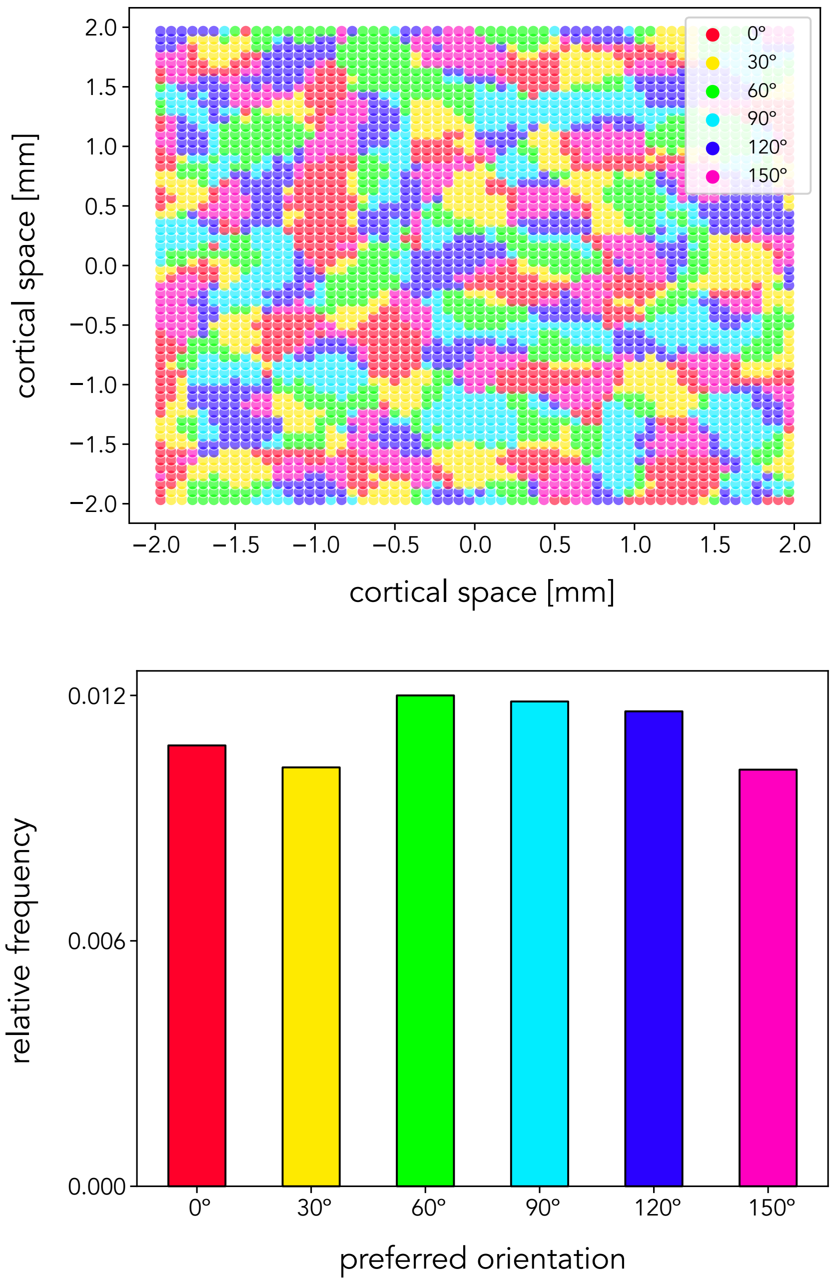

Figure 4.11: **Modeled orientation tuning map. Top** Preferred orientation of a bar stimulus of cortical neurons placed on a regular grid that receive input from thalamic neurons. **Bottom** Relative frequency of occurrence of orientation preference across cortical neurons.

We note that no orientation is strongly over-represented (right panel of Figure 4.11), in line with experimental evidence in V1 (Ju et al., 2021). Further, we can adjust σ to obtain iso-orientation domains of biologically realistic sizes: Amir et al. (1993) report patch widths, which are a proxy for the iso-orientation domain size, between 0.1 mm and 0.6 mm.

For the model, we can use this map to assign randomly sampled neurons an orientation preference by determining the orientation preference of the closest point in this grid. However, for comparing this map to experimentally obtained ones, we process it in a similar way to what is done in experimental studies by smoothing it spatially. For this, we first generate a much finer grid of $K \times K$ points that represents a discretized version of the cortical sheath, where to each point we assign the orientation preference of the closest grid neuron in the $L \times L$ square. Then, we define a square kernel window with a side length of $k \approx \frac{L}{20}$, which is moved over the fine grid of points and calculates the circular mean (i.e., where 0° is equal to 180°) over the kernel window. This results in an orientation tuning map as shown in Figure 4.12 exhibiting similar properties, such as right-handed and left-handed pinwheels, as experimentally observed ones (cf. Blasdel (1992), Malach et al. (1993), and Yacoub et al. (2008)).

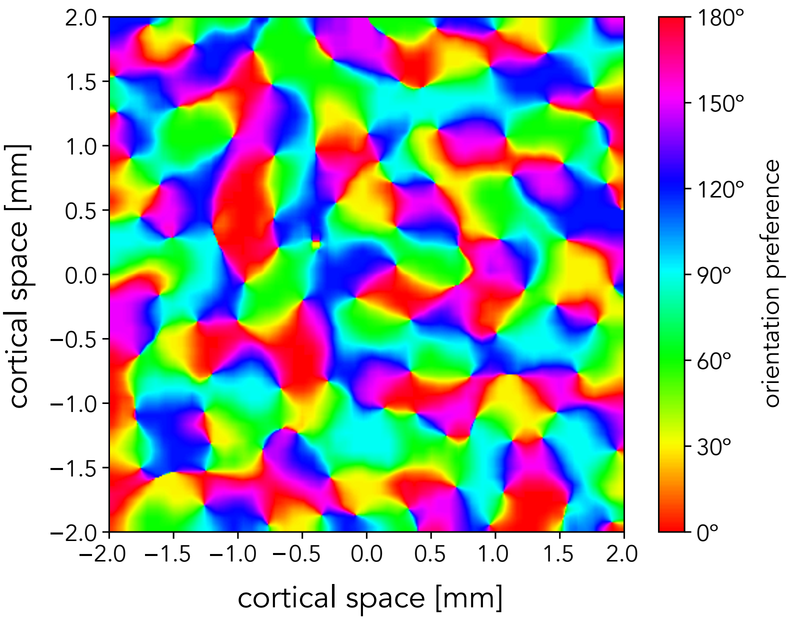

Figure 4.12: **Processed orientation tuning map.** Smoothed with a square kernel containing $k \times k$ neurons.

The parameters used for creating the orientation maps are summarized in Table 4.8.

PATCHY CONNECTIVITY With the orientation tuning map in place, we turn to establish patchy connections between neurons of the same

Parameter	**Value**	**Description**
L	4 mm	Side length of cortical sheath
L'	4.8 mm	Side length of thalamic sheath
N	3600	Number of cortical neurons
N_{th}	196	Number of thalamic neurons
a	0.1 mm	Spatial jitter for thalamic neurons
w_0	10 pA	Peak connection weight
σ	0.25 mm	Standard deviation of spatial connectivity
$\hat{I}$	30 pA	Peak current amplitude
f_t	1 Hz	Temporal input frequency
f_s	4 mm^{-1}	Spatial input frequency
K	750	Number of points in fine grid
k	35	Number of fine grid points in kernel

Table 4.8: **Parameters for creating and processing the orientation tuning map.**

orientation preference. We introduce the patchiness parameter $\psi \in [0, 1]$ controlling the fraction of patchy to isotropic connections:

$$S_{AB}^{\text{ia}} = \underbrace{\psi S_{AB}^{\text{ia}}}_{\text{synapses of patchy conn.}} + \underbrace{(1-\psi) S_{AB}^{\text{ia}}}_{\text{synapses of isotropic conn.}} \tag{4.73}$$

Following Bosking et al. (1997) and Sincich and Blasdel (2001), we model patchy connectivity such that a source neuron connects equally likely (i.e., with probability P_{AB}^{patchy}) to all other neurons that are of the same orientation preference and that lie inside of an ellipse, the orientation of which depends on the stimulus orientation preference of the source neuron. Here, the semi-minor and semi-major axes of the encapsulating ellipse are typically much larger than the diameter of the iso-orientation domains, such that a patchy pattern connecting various iso-orientation domains emerges. We can calculate the expected number of synapses of both connection schemes and equate them to the two terms of Equation 4.73:

$$\begin{aligned}
& S_{AB}^{\text{ia,patchy}} = \frac{1}{N_O} \int_{\text{ellipse}} \rho_A N_B P_{AB}^{\text{patchy}} \mathrm{d}\Omega \stackrel{!}{=} \psi S_{AB}^{\text{ia}} \\
\Rightarrow \quad & \frac{ab\pi}{N_O} \rho_A N_B P_{AB}^{\text{patchy}} = \psi S_{AB}^{\text{ia}} \\
\Rightarrow \quad & P_{AB}^{\text{patchy}} = \frac{\psi S_{AB}^{\text{ia}} N_O}{ab\pi \rho_A N_B}
\end{aligned} \tag{4.74}$$

and

$$
\begin{aligned}
& S_{AB}^{\text{ia,iso}} = \int_0^{2\pi} \int_0^{\infty} \rho_A N_B P_{AB}^{0,\text{iso}} r e^{-r/\lambda_{vX}} \mathrm{d}r \mathrm{d}\varphi \stackrel{!}{=} (1-\psi) S_{AB}^{\text{ia}} \\
\Rightarrow \quad & \frac{P_{AB}^{0,\text{iso}}}{P_{AB}^{0}} \underbrace{\int_0^{2\pi} \int_0^{\infty} \rho_A N_B P_{AB}^{0} r e^{-r/\lambda_{vX}} \mathrm{d}r \mathrm{d}\varphi}_{=S_{AB}^{\text{ia}}} = (1-\psi) S_{AB}^{\text{ia}} \\
\Rightarrow \quad & P_{AB}^{0,\text{iso}} = (1-\psi) P_{AB}^{0}
\end{aligned}
\tag{4.75}
$$

In this derivation, we use that ρ_A is 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 encapsulating ellipse, respectively, and N_O is the number of orientations used in the model. Patchy connectivity is present between all populations except connections from or to L4C, and from L6 to L5.

4.3.2.3 *Short-range push-pull connectivity*

Following Troyer et al. (1998) we implement a spatial projection scheme from LGN neurons to V1. This creates a natural way of feeding visual input to the model, and simultaneously establishes a basis for a push-pull connectivity scheme that connects neurons based on the correlation of their receptive fields.

LGN INPUT MODEL We represent the LGN by two populations of uniformly distributed neurons in a sheath with size $L \times L$. The ON population contains thalamic neurons with an ON receptive field, i.e., neurons that most strongly fire when a light stimulus is centered in their receptive fields while the surround is dark. The OFF population contains thalamic neurons with the opposite OFF property, i.e., neurons that most strongly fire when the stimulus centered on the receptive field is dark with a bright surround. Connections from these ON and OFF neurons to the cortical neurons in V1 are established as follows. Given a cortical target neuron in L4C with an orientation tuning preference of θ, where $\theta \in \{0°, 30°, 60°, 90°, 120°, 150°\}$, and a random phase $\psi \in \{0°, 180°\}$, we connect presynaptic ON/OFF thalamic neurons according to a probability distribution shaped like a Gabor filter:

$$
G(x, y; \lambda, \psi, \gamma, \sigma) = \left[\cos\left(2\pi \cdot \frac{y'}{\lambda} + \psi\right)\right]^{+/-} \exp\left(-\frac{\gamma^2 x'^2 + y'^2}{2\sigma^2}\right) \tag{4.76}
$$

where $\left[\,\cdot\,\right]^{+/-}$ denotes the absolute value of the positive/negative part of a function, and

$$
x' = x\cos(\theta) + y\sin(\theta), \quad y' = -x\sin(\theta) + y\cos(\theta)\,. \tag{4.77}
$$

Following Jones and Palmer (1987), we choose $\gamma = 0.6$ and $\lambda = 4\sigma$. We constrain the spatial width σ of the Gabor filter by considering the size of receptive fields in macaque V1 (Hubel and Wiesel, 1968; Tootell et al., 1982) to $\sigma = 0.165$ mm. This procedure yields the connectivity pattern shown in Figure 4.13.

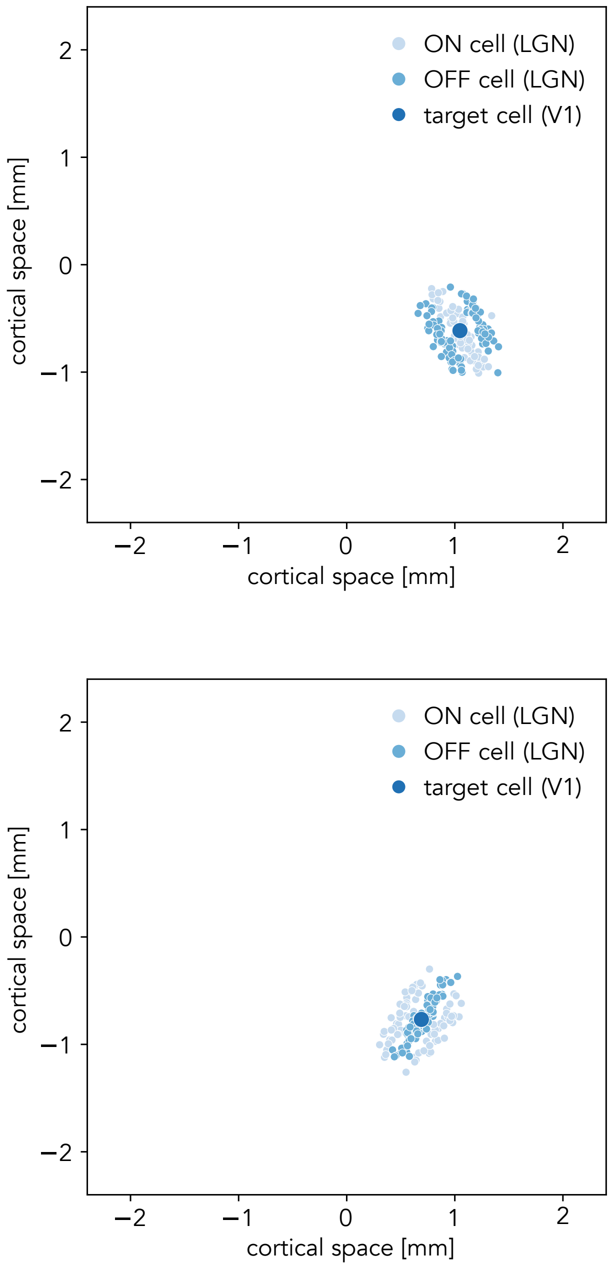

Figure 4.13: **Spatial connectivity patterns between thalamic LGN neurons (small circles) and cortical V1 neurons (big circles). Top** Source neurons of a randomly chosen cortical neuron with an orientation preference of 120° and ψ =0°. **Bottom** Source neurons of a randomly chosen cortical neuron with an orientation preference of 60° and ψ =180°.

PUSH-PULL CONNECTIVITY Following Troyer et al. (1998), we implement intra-areal push-pull connections between cortical neurons in L4C depending on their input correlations. If the presynaptic neuron is excitatory (inhibitory), positive correlation implies stronger (weaker) connectivity, whereas negative correlation implies weaker (stronger) connectivity. We here use the presynaptic connectivity patterns described by Equation 4.76 to determine input correlations: overlapping areas of ON and ON or OFF and OFF thalamic neurons imply a correlated response in the neurons, whereas overlapping areas of ON and OFF imply an anti-correlated one. Since the overlap of ON and OFF areas, and thereby the absolute value of this correlation, is highest for neurons with similar orientation preference, we neglect push-pull connections between neurons that have different orientation preferences. In this simplified case of Gabor filters with the same orientation θ, the described correlation can be approximated by a convolution of the respective single-neuron Gabor filters, which results in a Gabor filter with a widened spatial spread of $\sigma' = \sqrt{2}\sigma$ and a phase ψ that is the sum of the phases of the single-neuron Gabor filters. Thus, writing $p^X_{Y,Z}$ as the connection probability of a neuron of type Z of efficacy X to a neuron of type Y, where $X \in \{\mathrm{E}, \mathrm{I}\}$ and $Y, Z \in \{\mathrm{ON}, \mathrm{OFF}\}$, we obtain:

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

4.3.3 *Implementation of spatial connectivity*

In the isotropic connectivity scheme (Section 4.3.2.1), the naive NEST implementation of exponentially decaying connection probability with distance leads to performance issues when scaling up the size of the considered network since each presynaptic neuron evaluates whether to create a connection with *all* potential postsynaptic partners. Therefore, the connection algorithm scales with the number of neurons N as $\mathcal{O}(N^2)$. To allow for an arbitrary scaling of network size, we implement a modification to this algorithm that preserves the number of synapses while keeping the time for establishing connections constant. We introduce a parameter R which defines the radius of a cylinder in which connections are considered, effectively setting the connection probability to zero outside of this range. To keep the number of synapses constant, we calculate a scaling factor $\alpha(R)$ that linearly increases the zero-distance connection probability P^0_{AB}.

The number of intra-areal synapses between two populations A, B is given by

$$S^{\mathrm{ia},R}_{AB} = \rho_A N_B \int_0^{2\pi} \int_0^R P^0_{AB}\, r e^{-\frac{r}{\lambda_{vX}}} \mathrm{d}r \mathrm{d}\varphi \tag{4.78}$$

$$= 2\pi \rho_A N_B \left({\lambda_{vX}}^2 - \lambda_{vX}(\lambda_{vX} + R) e^{-\frac{R}{\lambda_{vX}}} \right) . \tag{4.79}$$

With the total number of synapses given by

$$S^{\mathrm{ia}}_{AB} = S^{\mathrm{ia},\infty}_{AB} \tag{4.80}$$

we can calculate the scaling factor as

$$\alpha(R) = \frac{S^{\mathrm{ia}}_{AB}}{S^{\mathrm{ia},R}_{AB}} = \frac{{\lambda_{vX}}^2}{{\lambda_{vX}}^2 - \lambda_{vX}(\lambda_{vX} + R) e^{-\frac{R}{\lambda_{vX}}}} \tag{4.81}$$

$$= \frac{1}{1 - \left(1 + \frac{R}{\lambda_{vX}}\right) e^{-\frac{R}{\lambda_{vX}}}} . \tag{4.82}$$

If we choose $R \propto \lambda_{vX}$ we eliminate the dependency of α on the layer under consideration, obtaining the scaling factors listed in Table 4.9.

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 4.9: **Radius-depend scaling factor α for peak zero-distance connection probability.** The proposed scaling procedure preserves the number of synapses while limiting the radius around a presynaptic neuron in which connections are established.

For our model, we choose $R = 4\lambda_{vX}$ to strike a balance between minimizing adaptations of the peak zero-distance connection probability and impacts on performance.

4.3.4 *Simulation parameters*

Table 4.10 gives a summary of the model specifications for the conducted spiking neural network simulations of the derived model of macaque V1. We further list all chosen parameters of the model in Table 4.11.

Neuron and synapse models	
Populations	Excitatory (E) and inhibitory (I) neurons in layers 2/3A, 3B, 4AB, 4C, 5, 6
Subthreshold dynamics	$\dot{V}(t) = -\frac{V(t)-E_L}{\tau_m} + \frac{I_{\mathrm{syn}}(t)}{C_m}$
Synaptic current	$I_{\mathrm{syn}}(t) = w \cdot e^{-\frac{t-t^*-d}{\tau_{\mathrm{syn}}}} \Theta(t-t^*-d)$, where Θ denotes the Heaviside function
Spiking	If $V(t-\delta t) < V_{\mathrm{th}}$ and $V(t) \geq V_{\mathrm{th}}$, where δt is the simulation time step: • $t^* = t$ • $V(t^*) = V_{\mathrm{reset}}$ for $t \in [t^*, t^* + \tau_{\mathrm{ref}}]$
Delay	Time d between emission and reception of spikes is given by $d = d_0 + rv_c + \mathcal{N}(0, \sigma_d)$, where r is the horizontal inter-soma distance
Spatial connectivity	
Medium-range	Isotropic connection probability $p \propto e^{-r/\lambda_{AB}}$, where r is the horizontal inter-soma distance and λ_{AB} depends on the presynaptic population B and postsynaptic population A.
Long-range	Patchy connectivity between iso-orientation domains. Proportion of patchy connections governed by $\psi \in [0,1]$. Present between all populations except connections to or from L4C, and from L6 to L5.
Short-range	Push-pull connection probability depends on neural input correlations which are estimated by thalamic projection patters. Proportion of push-pull connections governed by $\zeta \in [0,1]$. Present only in L4C.
Input	
Input drive	Cortico-cortical (cc) synapses as model-external input; cc neurons fire according to a Poisson process with rate ν_{ext}.

Table 4.10: **Model summary.** Neuron and synapse model, spatial connectivity, and input.

	Neuron parameters	
E_L	$-65\,\mathrm{mV}$	Reversal potential
τ_m	$10\,\mathrm{ms}$	Membrane time constant
C_m	$250\,\mathrm{pF}$	Membrane capacity
V_{th}	$-50\,\mathrm{mV}$	Threshold membrane potential
V_{reset}	$-65\,\mathrm{mV}$	Reset membrane potential
τ_{ref}	$2\,\mathrm{ms}$	Refractory period
	Synapse parameters	
w_{XE}	$87.8\,\mathrm{pA}$	Synaptic weight of excitatory neurons
w_{XI}	$-g \cdot w_{XE}$	Synaptic weight of inhibitory neurons
τ_{syn}	$0.5\,\mathrm{ms}$	Time constant of postsynaptic current
	Delay parameters	
d_0	$0.5\,\mathrm{ms}$	Delay offset
v_c	$0.3\,\mathrm{mm\,ms^{-1}}$	Conduction velocity
σ_d	$0.05\,\mathrm{ms}$	Standard deviation
	Spatial connectivity parameters	
ψ	0.3	Fraction of patchy connections
ζ	0.5	Fraction of push-pull connections

Table 4.11: **Model parameters.** Descriptions and numerical values used for simulations of the macaque V1 model.

4.4 DISCUSSION

In this chapter, we construct a detailed model of macaque V1 at the resolution of single neurons and synapses. The model is split into six cortical layers along the axis perpendicular to the cortical surface: L2/3A, L3B, L4A/B, L4C, L5, and L6, where the number of neurons and synapses varies between the layers. Further, each layer houses one excitatory and one inhibitory population, bringing the total number of populations in the model to twelve. We use a broad set of anatomical data to constrain the connectivity between the populations in the model, combining estimates of connection probability from presynaptic neurons to terminating synapse locations, and from synapses to postsynaptic populations. Based on the calculated number of synapses S_{AB} between populations B and A, we further derive spatial connection principles on three different scales: medium-range isotropic connectivity, long-range patchy connectivity, and short-range push-pull connectivity. In the process, we develop a thalamic input model that supplies sensory information to the neurons of V1. Based on the model of the thalamus, we also endow the model with an orientation tuning map parallel to the cortical surface which imposes spatially clustered iso-orientation domains where nearby neurons respond most strongly to the similarity oriented bar stimuli. By introducing the spatial connectivity principles, we elevate the model to a model that is arbitrary scalable in size. Since spatial convergence and divergence of connectivity between areas can be straight-forwardly integrated in this framework, the model is amendable to future extensions such as the addition of cortical areas higher up in the hierarchy.

Implementing the biological constraints using the NEST simulator, we instantiate spiking neural network simulations to record and analyze the resulting activity. Probing the dynamical state of the model, we find that it is unable to exhibit a biologically realistic ground state of spontaneous activity: increasing the external drive ν_{ext} or reducing the inhibitory coupling strength parameter g beyond a certain threshold always leads to runaway excitation that pushes the model into a pathological state in which the firing rate of neurons averaged across populations exceeds 100 spikes/s, much higher than what is biologically plausible. The same holds true when replacing the heterogeneous external drive by a homogeneous one that is the same for all populations, indicating that the intra-areal connectivity is the root cause of the instability. In line with Potjans and Diesmann (2014), we suspect that the target specificity encoded in the anatomical data by Binzegger et al. (2004) is a main contributor to this unrealistic ground state: we identify potential positive feedback loops in the recurrent targeting pattern of excitatory populations. In Chapter 5, we therefore set out to systematically investigate this hypothesis by taking a closer look at the architecture of local cortical circuits using recent high-resolution experimental data.

5 REVISITING ARCHITECTURES OF LOCAL CIRCUITS IN PRIMARY VISUAL CORTEX

This chapter is based on the following publication:
A. C. Kurth*, J. Albers*, M. Diesmann, S. J. van Albada: *Cell-type specific projection patterns promote balanced activity in cortical microcircuits* (Kurth et al., 2024). * indicates equal contributions.

Author contributions:
Under the supervision of Prof. Dr. Sacha van Albada (SvA) and Prof. Dr. Markus Diesmann (MD), the author conceived the project together with Anno Christopher Kurth (ACK). Both the author and ACK conceptualized the work presented in this chapter. The author analyzed the MICrONS data set, ACK analyzed the Stepanyants data set, and the author and ACK jointly analyzed the Binzegger data set. Both jointly analyzed and compared the resulting connectivity maps, instantiated spiking neural network simulations, and analyzed the dynamical activity. The author amended the previously developed model of V1 with the connectivity from the MICrONS data set derived in this work, and analyzed the dynamical properties of the amended model. All analyses were refined in ongoing discussions of the author with SvA and MD.

5.1 INTRODUCTION

In this section, we aim to understand the role of recurrent targeting patterns in local cortical circuits for their dynamical states. In particular, we want to investigate the hypothesis posed in Section 4.4: excitatory feedback loops in the connectivity derived from Binzegger et al. (2004) lead to unstable network dynamics. Staying in the framework of data-driven modeling, we investigate the targeting patters in a recent data set of the local cortical connectivity in mouse V1 (MICrONS Consortium et al., 2021), and link them to the dynamical states of a network model based on this connectivity. Notably, this set is based on morphological reconstructions using electron microscopy (EM) instead of light microscopy (LM) as was the case for the data by Binzegger et al. (2004). While the basic technology of EM has been around for almost a century (Knoll and Ruska, 1932), only recent advances in machine learning on image-based data have enabled the analysis of volumes on the order of 1 mm^3 of cortical tissue, as this entails on the order of 1 PB of imaging data.

We start by deriving the spatial scale of connectivity from both EM and LM data in Section 5.2.1. Next, from each data set, we derive local cortical microcircuits, i.e., models without a notion of space, but constrained in their size by their spatial connections such that 75 % of all incoming connections originate from within the model (Section 5.2.2). After instantiating spiking neural network simulations and analyzing the resulting activity (Section 5.2.3), we link the observed dynamics to the underlying targeting patterns of the recurrent connectivity. We find that indeed, positive feedback loops are a main contributing factor for unstable dynamics in microcircuits based on the LM data set. Having derived an alternative connectivity based on EM, we return to our model of V1 derived in Chapter 4 and replace the LM-based connectivity (Section 5.2.5). Thereby, we are able to remedy the unstable dynamics and bring the spatial network of V1 into a state of biologically plausible resting state activity without the need of further fine-tuning.

5.2 RESULTS

We here compare the architectures of local cortical microcircuits derived from two data sets, and derive corresponding *microcircuit models*—spiking neural networks of point neurons akin to the model developed in Chapter 4 but on a smaller spatial scale ranging from 0.5 to $1\,\mathrm{mm}^2$.

Binzegger et al. (2004) estimate connectivity based on LM reconstructions of the morphology of 39 neurons from cat primary visual cortex, using a generalized version of Peters' rule (Braitenberg and Schüz, 1991; Peters and Feldman, 1976). Supplementing the data with previously obtained numbers of neurons and synapses (Beaulieu and Colonnier, 1985; Beaulieu and Colonnier, 1983), the authors provide estimates for the numbers of synapses between neurons of various classes. For inhibitory neurons, the authors mainly focus on basket cells, leaving a certain fraction of inhibitory synapses unexplained.

The MICrONS data set (MICrONS Consortium et al., 2021) contains an EM reconstruction of $\sim 1\,\mathrm{mm}^3$ of mouse visual cortex, consisting of about 75 thousand neurons and about 500 million synapses. For this work, we only consider connections originating from 266 presynaptic neurons with fully reconstructed and proofread axons.

While the MICrONS data give direct access to *actual* neural connections, the data of Binzegger et al. (2004) yield *potential* connectivity. In the following, we denote the model based on the MICrONS data set with $\mathcal{M}_{EM}$, and the model based on the Binzegger data set with $\mathcal{M}_{LM}$. In the two models, the cortical layers 2/3, 4, 5, and 6 are distinguished (henceforth, we refer to these as L2/3, L4, L5, and L6). Each layer contains one excitatory (E) and two inhibitory populations (Ib, Inb) corresponding to basket and non-basket cells. For the LM data set, we follow Izhikevich and Edelman (2008), assigning unexplained inhibitory synapses to presynaptic non-basket cells. In the EM data set, all neurons can be straightforwardly mapped to one of the populations.

5.2.1 *Spatial Connectivity*

We estimate the spatial organization of local cortical connections using both the distance-dependent mean number of synapses and the distance-dependent connection probability. Here, the connection probability is defined as the probability that two neurons establish at least one synapse. The connection probability consequently neglects the multiplicity of synapses between pairs of neurons.

We assume an exponential decay of the mean number of synapses $S_{AB}(d)$ between one presynaptic neuron of population B and one postsynaptic neuron of population A with horizontal somatic distance d:

$$S_{AB}(d) = S^0_{AB} \exp\left(-\frac{d}{\lambda^{\text{syn}}_{AB}}\right) \tag{5.1}$$

Here, S^0_{AB} denotes the peak number of synapses, and $\lambda^{\text{syn}}_{AB}$ denotes the characteristic length of cortical connectivity between populations B and A. Taking the density $\omega(d)$ of distances d between pre- and postsynaptic neurons into account, the density of the expected mean number of synapses $s_{AB}(d)$ between a single neuron in presynaptic population B and all neurons in postsynaptic population A at distance d is $\omega(d) \cdot S_{AB}(d)$.

Likewise, we assume an exponential decay of the connection probability with distance d between individual neurons in the presynaptic population B and postsynaptic population A:

$$p_{AB}(d) = p^0_{AB} \exp\left(-\frac{d}{\lambda^{\text{conn}}_{AB}}\right) \tag{5.2}$$

Here, p^0_{AB} denotes the connection probability at zero distance. $p_{AB}(d)$ is the conditional probability for a neuron in population B to form at least one synapse with a particular neuron in population A given that the horizontal distance of their somata is d. This assumption is in agreement with the isotropic connectivity rule assumed throughout Chapter 4.

For $\mathcal{M}_{EM}$, $S_{AB}(d)$ and $p_{AB}(d)$ can straightforwardly be extracted from the actual connectivity reported in the EM data. Figure 5.1B shows example fits of the density of the mean number of synapses, with resulting spatial decay constants $\lambda^{\text{syn}}_{AB}$ displayed in Figure 5.1A. We here assume a uniform distribution of neurons to approximate the true distribution of distances in the reconstructed volume, avoiding problems arising due to sparse data for some population pairs. (cf. Section 5.3.1). Further, we show example fits for the connection probability in Figure 5.2B and spatial decay constants $\lambda^{\text{conn}}_{AB}$ in Figure 5.2A.

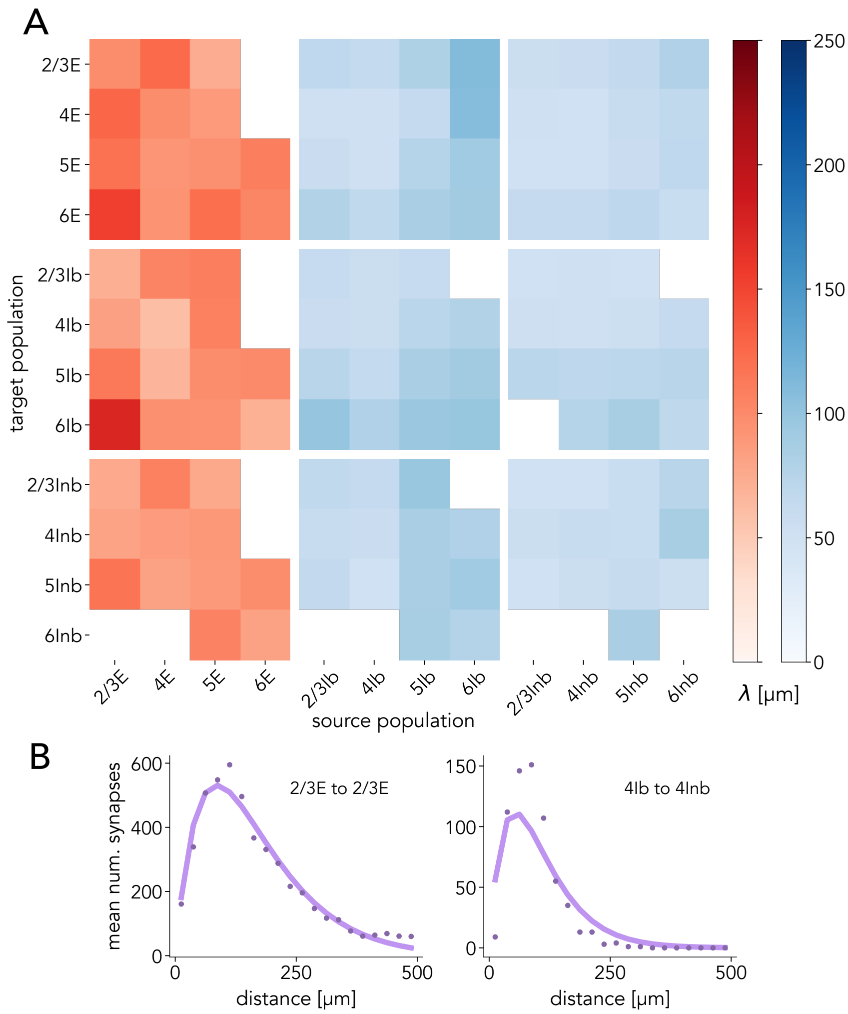

Figure 5.1: **Spatial scale of the mean number of synapses from EM data. A** Spatial scale of the mean number of synapses estimated from MICrONS data for excitatory (E), inhibitory basket (Ib), and inhibitory non-basket (Inb) cells with somata in cortical layers 2/3, 4, 5, or 6. White tiles indicate $<$ 50 synapses. Tiles colored blue (red) for connections in which the source population is inhibitory (excitatory). **B** Example fits showcasing high-quality (left) and low-quality (right) curve fits (see Section A.1.1 for overview of fits).

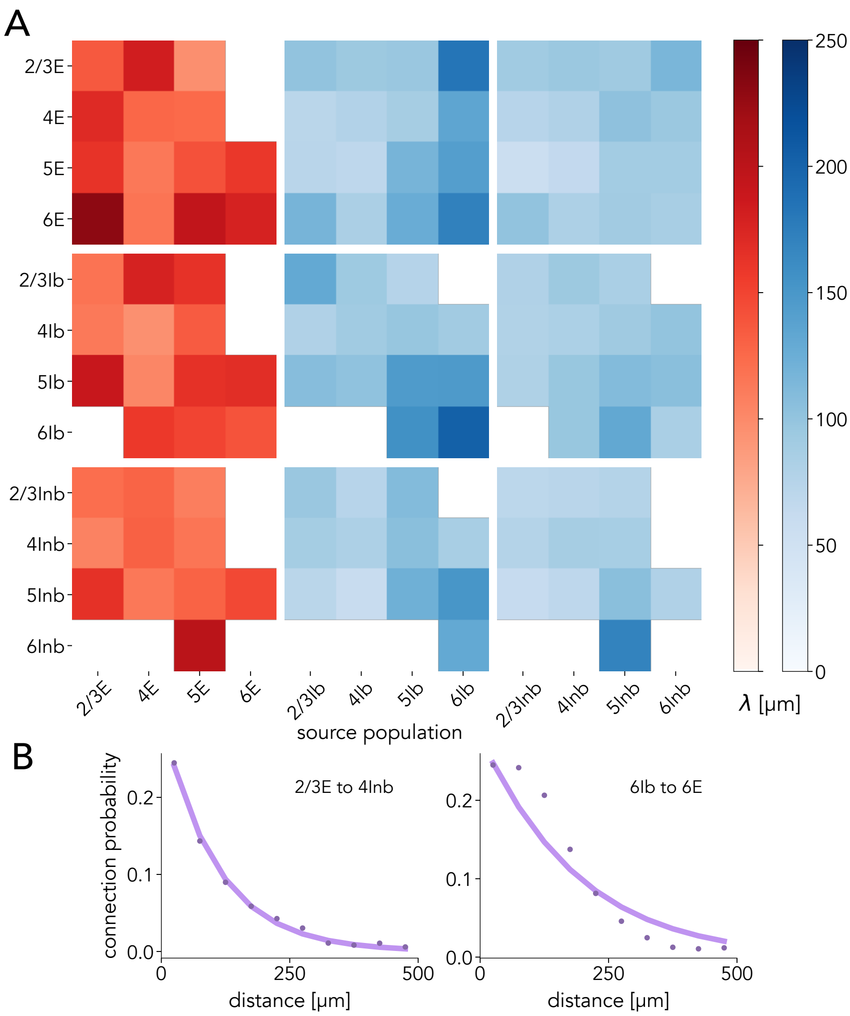

Figure 5.2: **Spatial scale of the connection probability from EM data. A** Spatial scale of the connection probability estimated from MICrONS data for excitatory (E), inhibitory basket (Ib), and inhibitory non-basket (Inb) cells with somata in cortical layers 2/3, 4, 5, or 6. White tiles indicate < 50 synapses. Tiles colored blue (red) for connections in which the source population is inhibitory (excitatory). **B** Example fits showcasing high-quality (left) and low-quality (right) curve fits (see Section A.1.1 for overview of fits).

In agreement with the literature, we find that the spatial scales of local excitatory and inhibitory connectivity at the studied range are generally comparable, with a tendency for excitatory connection lengths to exceed the inhibitory ones (Levy and Reyes, 2012; Packer and Yuste, 2011; Perin et al., 2011; Reimann et al., 2017). Furthermore, we observe that $S_{AB}(d)$ consistently exhibits smaller characteristic lengths than $p_{AB}(d)$. Calculating the fraction

$$c_X = \frac{\lambda^{\mathrm{syn}}_{A,vX}}{\lambda^{\mathrm{conn}}_{A,vX}} \quad , \quad X \in \{\mathrm{E}, \mathrm{Ib}, \mathrm{Inb}\} \tag{5.3}$$

for all pairs of populations, where v is the layer of the presynaptic population and X is that population's cell type, we observe a consistent average $c^\star = 0.7$ (Figure 5.3). Thereby, we can estimate $\lambda^{\mathrm{syn}}_{AB}$ from $\lambda^{\mathrm{conn}}_{AB}$, and vice versa.

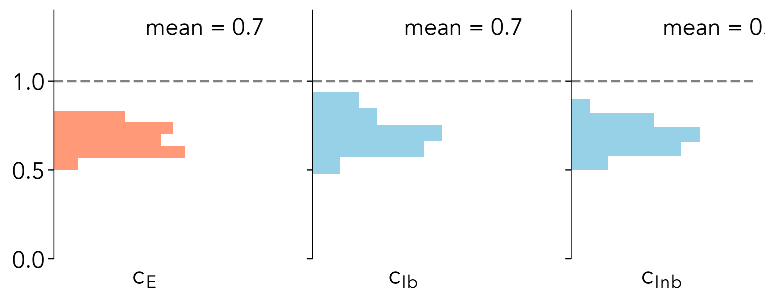

Figure 5.3: **Comparing the spatial scales of the mean number of synapses and connection probability.** Ratios between spatial scales of the mean number of synapses and the connection probability for actual connectivity from the EM data set, grouped by cell type of presynaptic population. Histogram over all pairs of populations.

For $\mathcal{M}_{LM}$, data on actual connectivity is not available. Instead, the spatial scale of the potential connectivity has been estimated solely based on the morphological reconstructions (Hellwig, 2000; Stepanyants et al., 2007): a potential synapse between a pre- and postsynaptic neuron is registered if their axon and dendrite are sufficiently close (Figure 5.4).

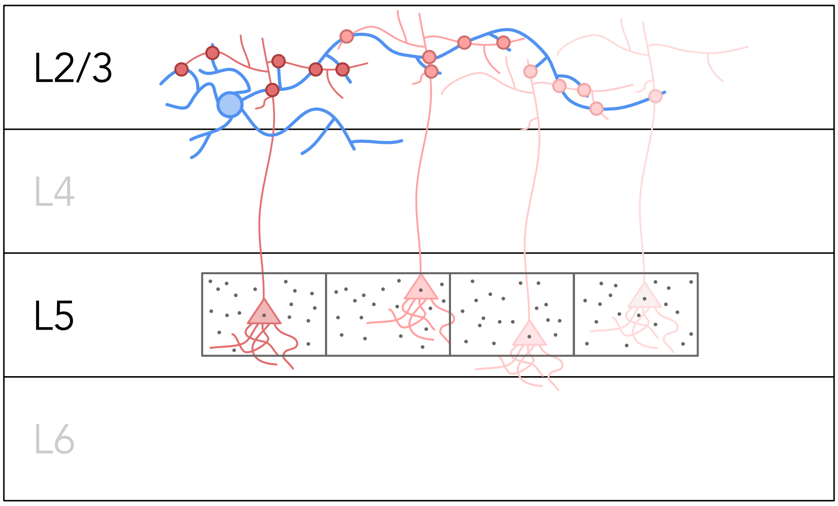

Figure 5.4: **Deriving potential connectivity from morphological reconstructions.** Illustrating the calculation of potential connectivity from morphologies based on Peters' rule. The location of the presynaptic neuron (blue) of population B is fixed. The postsynaptic neuron (red) of population A is randomly placed in a box with side length $\Delta = 25\,\mu\text{m}$ centered on the horizontal coordinates of the presynaptic neuron and the vertical coordinate of the postsynaptic neuron. The positions inside the box and rotations around the vertical axis are sampled $n = 100$ times (gray dots). The connection probability is defined as m/n where m is the number of samples for which the axon and dendrite are sufficiently close (red circles) at least once.

With this approach, the authors estimate both the mean number of synapses and the connection probability. First, we assess the consistency of potential and actual connectivity on the basis of the EM data. Using the provided morphologies, we reproduce the method (Figure 5.4) to obtain an estimate of the characteristic length scale of the connection probability $\lambda_{AB}^{\text{conn,pot}}$. Comparing to the previously derived estimates from actual connectivity, $\lambda_{AB}^{\text{conn}}$, we find a good agreement across most pairs of populations (Figure A.1.5).

Next, we calculate $\lambda_{AB}^{\text{syn,pot}}$ for all pairs of populations from the LM data set. In the methodology of potential connectivity, we expect that the estimated number of potential synapses is prone to noise due to multiple adjacent segments of reconstructed neural processes being counted as locations of potential synapses. Thus, we use estimates of the connection probability for the pairings of populations where this information is available in Stepanyants et al. (2007), and then convert $\lambda_{AB}^{\text{conn,pot}}$ to $\lambda_{AB}^{\text{syn,pot}}$ using $c^{\star}$. For all other pairs of populations, we directly estimate $S_{AB}(d)$. For pairs of populations with no data in Stepanyants et al. (2007), we make additional generalizing assumptions

(Section A.1.2). Example fits of this procedure are shown in Figure 5.5B, with the resulting $\lambda_{AB}^{\mathrm{syn,pot}}$ displayed in Figure 5.5A.

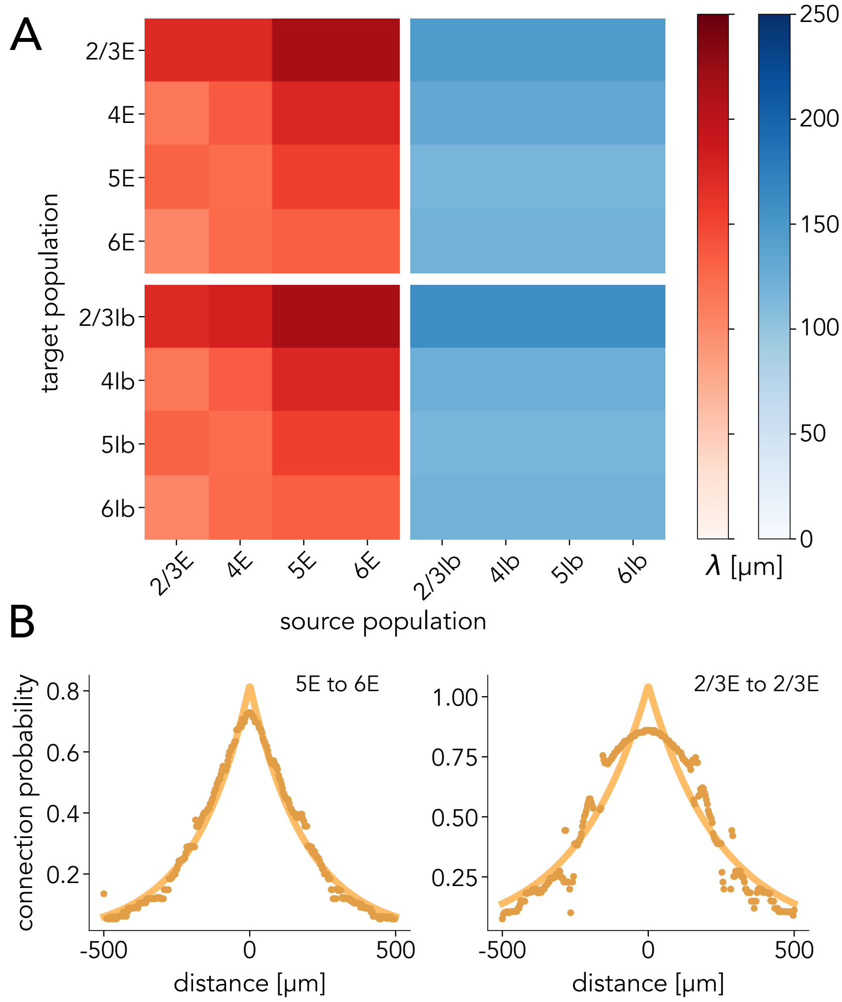

Figure 5.5: **Spatial scale of the mean number of synapses from LM data.** **A** Spatial scale of the mean number of synapses estimated from data by Stepanyants et al. (2007). Excitatory-to-excitatory characteristic lengths estimated from connection probabilities and converted using $c^{\star}$. Remaining length scales estimated from expected number of potential synapses. **B** Example fits showcasing high-quality (left) and low-quality (right) curve fits (see Section A.1.1 for overview of fits).

5.2.2 *Connectivity Maps*

Because our models $\mathcal{M}_{LM}$ and $\mathcal{M}_{EM}$ consider distance-independent mean connection probabilities p_{AB}, the size of the represented cortical tissue and consequently the number of neurons and synapses must be fixed to derive the corresponding connectivity maps. To this end, we distribute neurons with realistic densities for each model in a template space. Assuming the distance-dependence of the mean number of synapses Equation 5.1, we determine the fraction of model-internal synapses for each pair of pre- and postsynaptic populations for circular patches of cortical tissue (Figure 5.6).

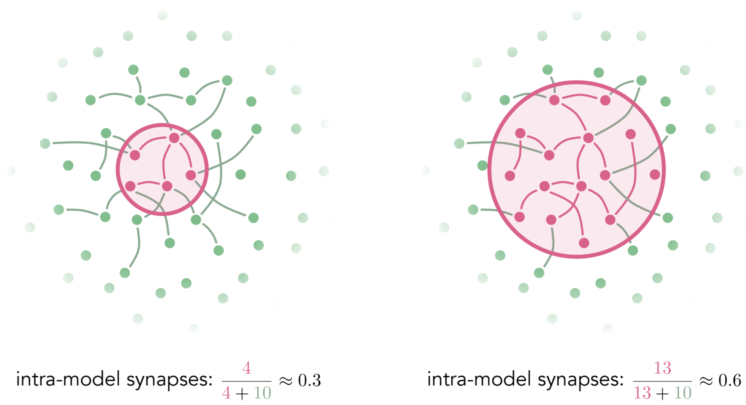

Figure 5.6: **Model construction.** Sketch illustrating the fraction of intra-model synapses depending on model size.

Here, model-internal synapses refer to incoming synapses originating from neurons in the model. The model radii $r^{\star}_{EM}$ and $r^{\star}_{LM}$ are then chosen such that the fractions of model-internal synapses averaged over pre- and postsynaptic populations are approximately 75 % (see Section 5.3.2), resulting in $r^{\star}_{EM} = 0.55\,\mathrm{mm}$ and $r^{\star}_{LM} = 0.75\,\mathrm{mm}$.

Figure 5.7 shows that $\mathcal{M}_{LM}$ has a higher number of neurons in most populations, which is expected due to the larger size of the model. Turning to connectivity, we observe that the populations of $\mathcal{M}_{LM}$ have a significantly higher in-degree, i.e., number of incoming connections per neuron, except for 6Ib. Consequently, the connection probabilities between populations are higher in $\mathcal{M}_{LM}$ than in $\mathcal{M}_{EM}$ (Figure 5.8).

Still, common patterns can be identified: both models show an excitatory sub-circuit between L2/3E and L5E, inhibitory basket cells project mainly within the same layer, and inhibitory non-basket cells preferentially target neurons in the same or higher layers. The main differences can be observed in the projection pattern of excitatory to inhibitory cells, as further discussed in Section 5.2.4.

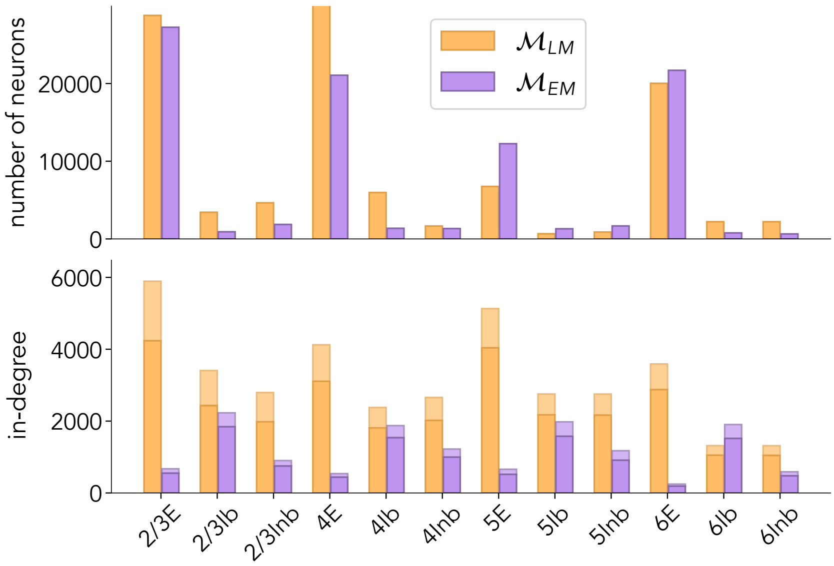

Figure 5.7: **Number of neurons and in-degrees of derived microcircuits.** Numbers of neurons (top) and in-degrees (bottom) for both models with radii $r^{\star}_{EM} = 0.55\,\mathrm{mm}$ and $r^{\star}_{LM} = 0.75\,\mathrm{mm}$. The model radii are chosen so that the fraction of model-internal synapses is approximately 75 % averaged over population pairs (Equation 5.5). For the in-degrees, solid bars indicate the number of intra-model synapses, while transparent bars (stacked) indicate synapses with presynaptic neurons not contained in the models. $\mathcal{M}_{EM}$ consists of ∼96 thousand neurons and ∼64 million synapses; $\mathcal{M}_{LM}$ consists of ∼108 thousand neurons and ∼455 million synapses.

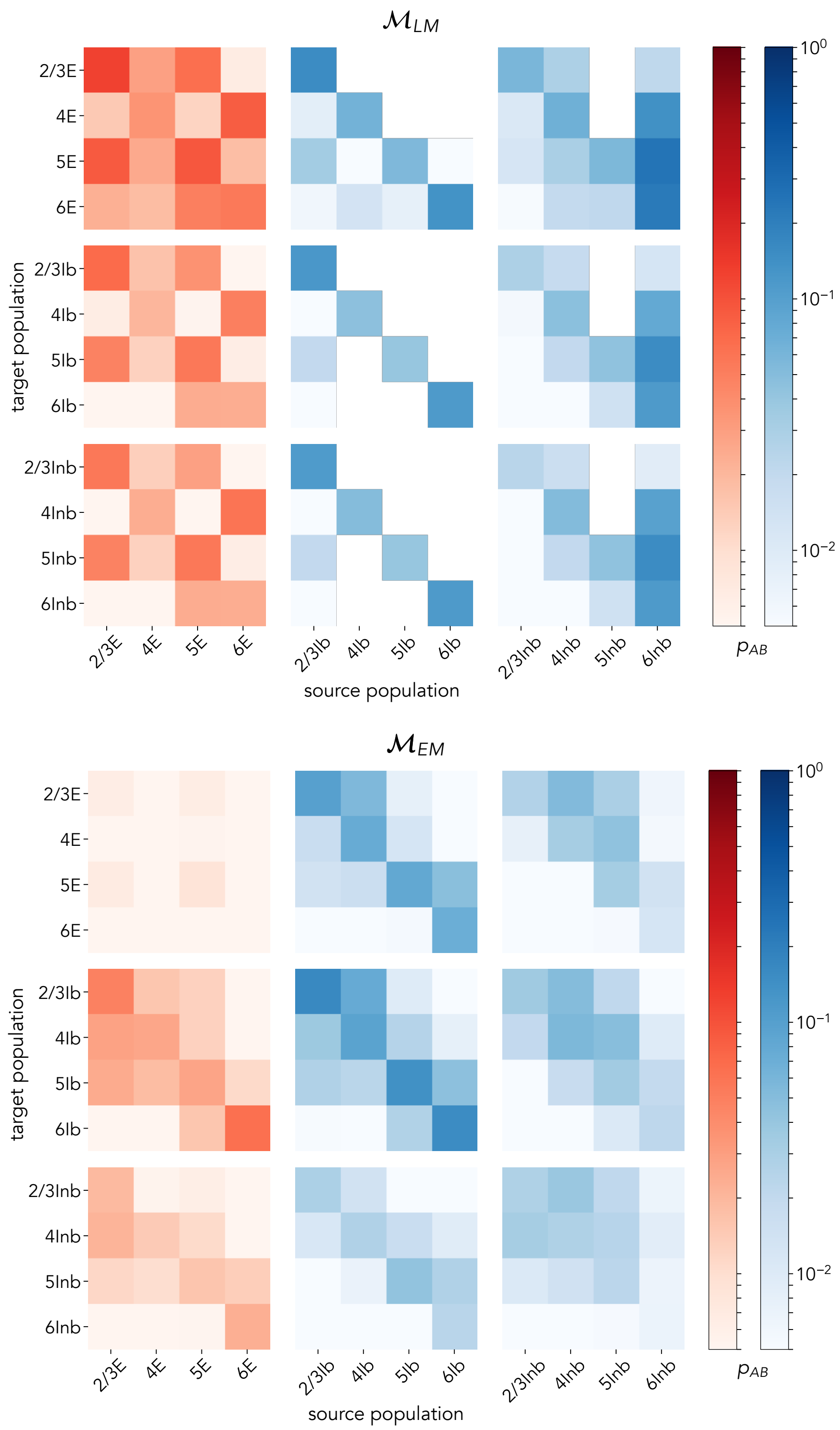

Figure 5.8: **Connectivity maps.** Mean connection probabilities between two arbitrary neurons of given source- and target populations for the two models.

5.2.3 *Model dynamics*

To investigate the dynamical properties of the constructed models $\mathcal{M}_{LM}$ and $\mathcal{M}_{EM}$, we instantiate and simulate spiking neural networks defined by the derived numbers of neurons and connectivity maps (see Section 5.3.3). Neurons are modeled as leaky integrate-and-fire units with conductance-based synapses using the neural simulation engine `NEST` (Gewaltig and Diesmann, 2007). Excitatory and inhibitory non-basket cells are modeled as regular-spiking neurons and include spike-frequency adaptation and relative refractoriness (Dayan and Abbott, 2001); inhibitory basket cells, which are Parvalbumin-expressing, are fast-spiking neurons and are modeled without adaptation (Feldmeyer et al., 2018) (see Section 5.3.3). We constrain the recurrent synaptic weights using the data of Avermann et al. (2012) for E, Ib, and Inb neurons of layer 2/3 in rat cortex. While this choice neglects layer- and species-specific diversity, it reduces model complexity. The weights are consistent with a recently discussed hierarchy of connection strengths in mouse visual cortex (Kraynyukova et al., 2022) (see Section A.1.3). Synaptic inputs from neurons not contained in the networks are modeled as excitatory conductance fluctuations in the form of Ornstein-Uhlenbeck processes with population-specific mean μ_A, variance σ_A, and time scale τ_A (Destexhe et al., 2001; Isbister et al., 2023). The extrinsic drive to each neuron is statistically independent of all other drives. For a detailed model specification see Section A.1.4.

The extrinsic drive to both network models is adjusted to approximately satisfy experimentally observed firing rates (Ringach et al., 2002; Wohrer et al., 2013) using the method suggested by Isbister et al. (2023) (Table A.1.7, Table A.1.8). Concretely, μ_A and σ_A are chosen for each population A such that $\sigma_A = \chi\mu_A$ with cell-type independent χ. The optimized inputs lead to the network activity displayed in Figure 5.9. In the resultant state, the spiking activity of both networks exhibits biologically plausible characteristics: asynchronous irregular activity (Cohen and Kohn, 2011; Ecker et al., 2010) (Figure 5.9A,B), low firing rates obeying $\nu_{\mathrm{Ib}} > \nu_{\mathrm{Inb}} > \nu_{\mathrm{E}}$ in most cases (Atallah et al., 2012; Gentet et al., 2012; Snyder et al., 2016) (Figure 5.9C,F), broadly distributed coefficients of variation of the inter-spike intervals (Softky and Koch, 1993) (Figure 5.9D,G), and low synchrony assessed by co-fluctuations of neural membrane potentials (Golomb and Rinzel, 1994) (Figure 5.9E,H). The dot displays only include neurons that spiked at least once in the observation interval, akin to experimental recordings where silent neurons remain unobserved.

To systematically investigate the dynamical repertoire of both models, we perform a parameter scan varying the mean drives μ_E and μ_I to the sets of excitatory and inhibitory populations (Figure 5.10).

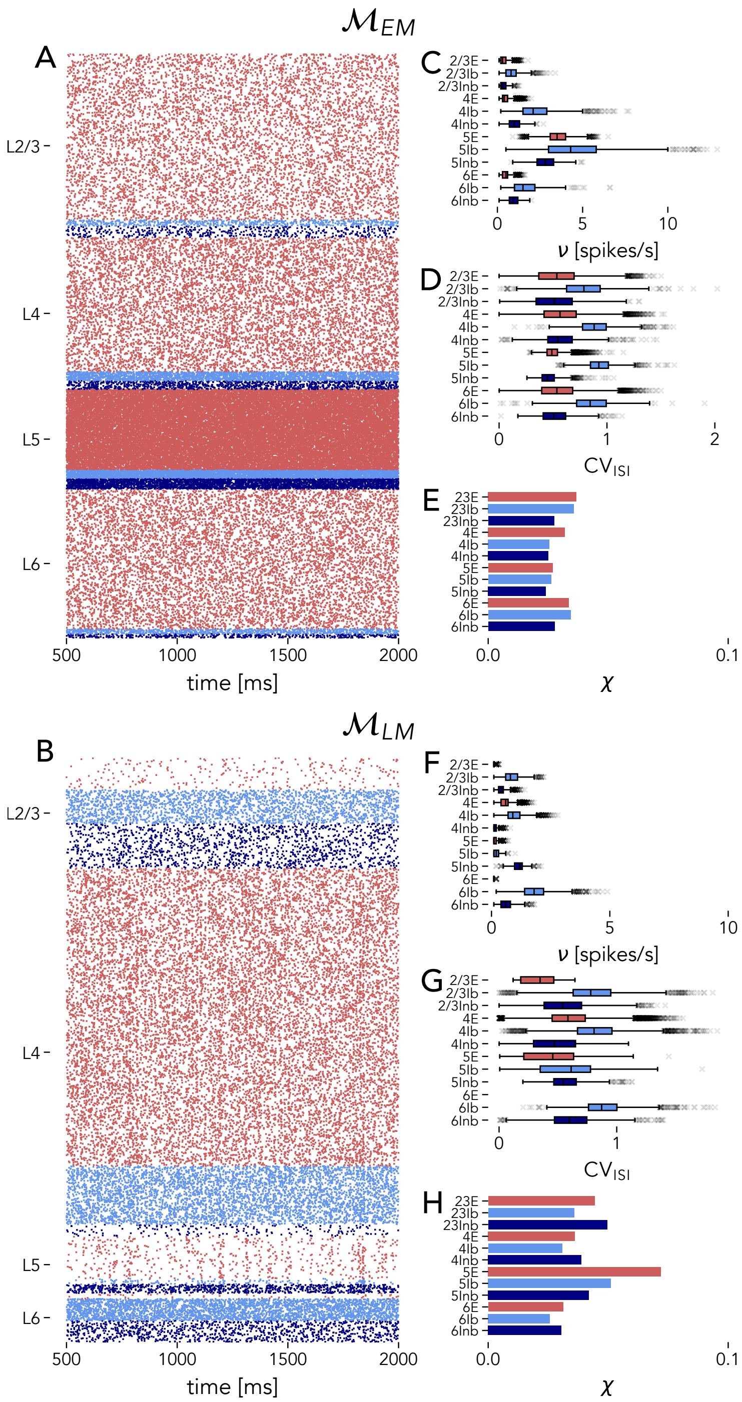

Figure 5.9: **Model activity with optimized external input.** Red indicates excitatory, light blue inhibitory basket, and dark blue inhibitory non-basket populations. The simulation gathers statistics over 10 s of biological time starting 0.5 s after simulation onset. **A, B** Raster plots (spikes marked as dots) of 50% of neurons that spiked at least once for the time stretch from 0.5 s to 2.0 s. **C–H** Population-resolved firing rates ν, coefficients of variation of the inter-spike interval $\mathrm{CV}_{\mathrm{ISI}}$ (Equation 5.8), and synchrony χ (Equation 5.9).

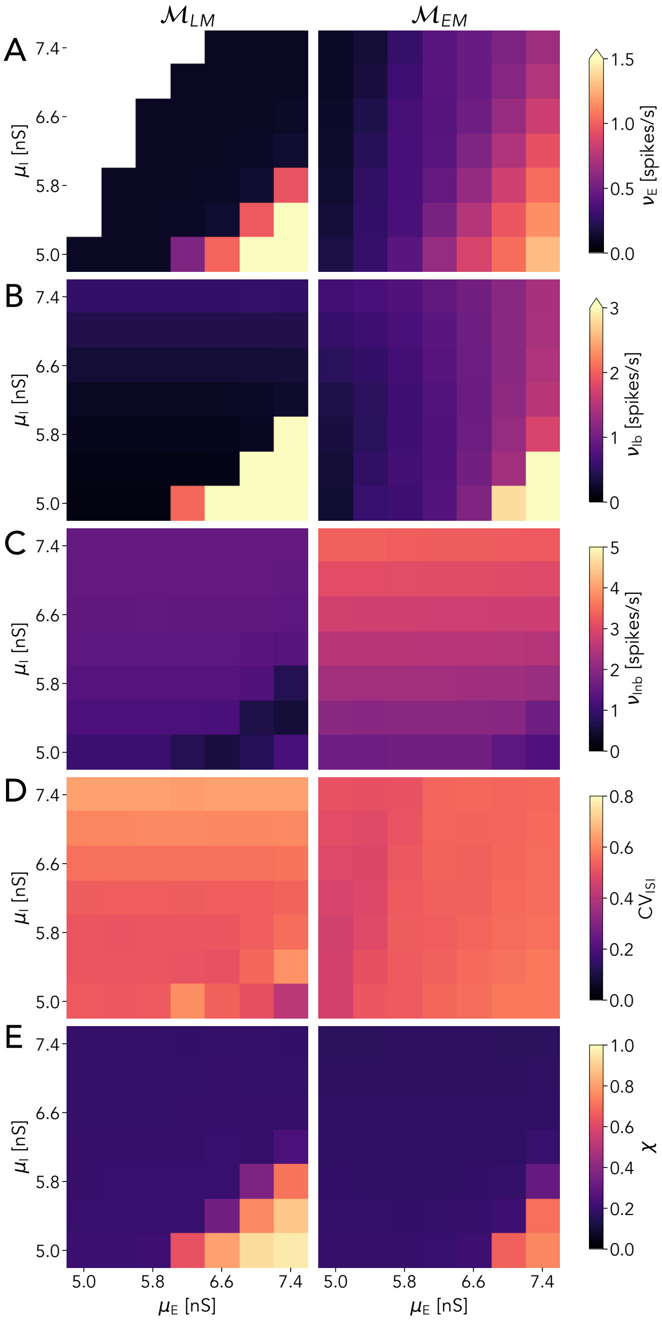

Figure 5.10: **Model activity under variation of extrinsic drive.** Simulation time of $T = 10\,\mathrm{s}$ with a simulation period of $T_{\mathrm{pre}} = 0.5\,\mathrm{s}$ before data acquisition to avoid transients due to initial conditions. The mean input to all excitatory (μ_E) and inhibitory (μ_I) neurons is varied. **A-C** Population-averaged firing rates of E, Ib, and Inb neurons. **D** Population-averaged coefficients of variation CV_{ISI} of the inter-spike interval (Equation 5.8). **E** Population-averaged synchrony χ (Equation 5.9).

We find that $\mathcal{M}_{LM}$ has only a narrow band of extrinsic drives for which the firing rates resemble experimentally observed activity: if μ_I is too large, the excitatory cells do not fire or have a vanishingly small firing rate, and if μ_E is too large, the network rapidly transitions to a highly synchronized state (Figure 5.10A-E left). In contrast, the activity of $\mathcal{M}_{EM}$ smoothly depends on the extrinsic drive with plausible firing rates, irregular spiking, and low synchrony across a large domain of the parameter space (Figure 5.10A-E right). Additionally, $\mathcal{M}_{EM}$ exhibits the paradoxical effect (Adesnik, 2017; Sanzeni et al., 2020; Tsodyks et al., 1997), where increasing μ_I decreases the firing rate of inhibitory basket cells. Note that also for $\mathcal{M}_{LM}$ the inhibitory basket cells reduce their firing rate when increasing μ_I for large μ_E and small μ_I. We do not consider this as a paradoxical effect, since for large μ_E and small μ_I, $\mathcal{M}_{LM}$ exhibits biologically implausible, highly synchronized activity, suggesting that in this case, the reduced inhibitory firing rate is due to a qualitative change in dynamical state.

5.2.4 *Linking structure and dynamics*

What is the structural reason for this discrepancy between the models? Consider the specificity of projections from a given source population to excitatory versus inhibitory neurons in a target layer, which we term *target specificity* (Equation 5.10, which is Equation 4.2 generalized for multiple inhibitory populations). The target specificity attains a positive (negative) value if a projection preferentially targets excitatory (inhibitory) neurons (Figure 5.11).

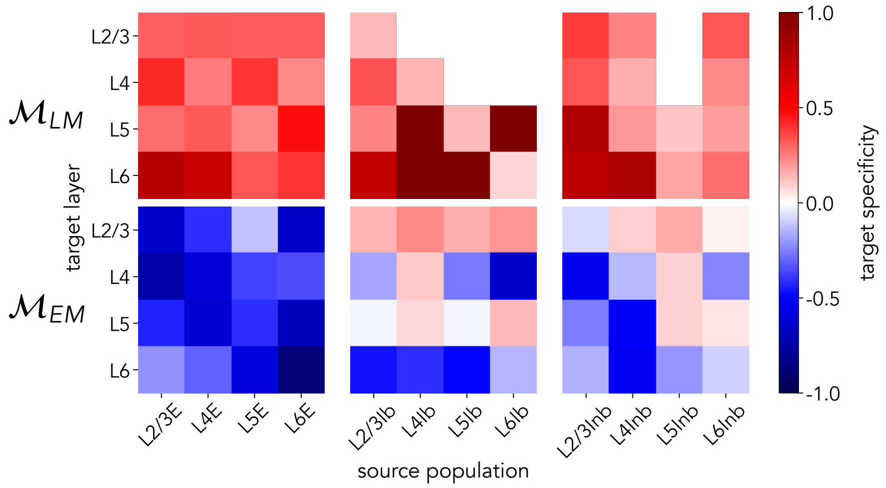

Figure 5.11: **Target specificity for both network models.** Positive values indicate a preferential targeting of excitatory neurons in a layer by a given source population; negative values indicate preference for inhibitory targets. Upper row: $\mathcal{M}_{LM}$; lower row: $\mathcal{M}_{EM}$.

For $\mathcal{M}_{LM}$, all presynaptic populations have a positive target specificity. For $\mathcal{M}_{EM}$, a more nuanced picture emerges where excitatory presynaptic populations preferentially target inhibitory neurons (left column), while there are inhibitory neurons of both cell types with positive as well as negative target specificities (middle and right column). This suggests that excitatory neurons that are more strongly innervated by recurrent connections, as is the case in $\mathcal{M}_{LM}$, are the origin of pathological dynamics. We test this in $\mathcal{M}_{EM}$ by redistributing synapses from excitatory neurons that previously targeted inhibitory neurons such that they target excitatory neurons instead, while conserving the total number of synapses (Figure 5.12).

When a moderate number of synapses are redistributed, the target specificity of excitatory neurons becomes balanced (Figure 5.12A, left). The resulting model retains the biologically plausible dynamics across a large part of the parameter space (Figure 5.12B-F, left column). Increasing the number of synapses that are redistributed until the target specificity of excitatory neurons resembles that of $\mathcal{M}_{LM}$ (Figure 5.12A, right), $\mathcal{M}_{EM}$ shows qualitatively similar dynamics to $\mathcal{M}_{LM}$ (Figure 5.12B-F, right column): for excitatory neurons, the transition between the firing regime at large μ_E and small μ_I and that at smaller μ_E and large μ_I becomes more abrupt. Further, the paradoxical effect in the activity of inhibitory neurons becomes less pronounced. Finally,the increased target specificity results in highly synchronous activity for a larger portion of the parameter space.

Thus, the biologically implausible activity of $\mathcal{M}_{LM}$ can in part be explained by the underlying positive target specificity, which in turn stems from the assumptions of the reconstruction of the local cortical circuit.

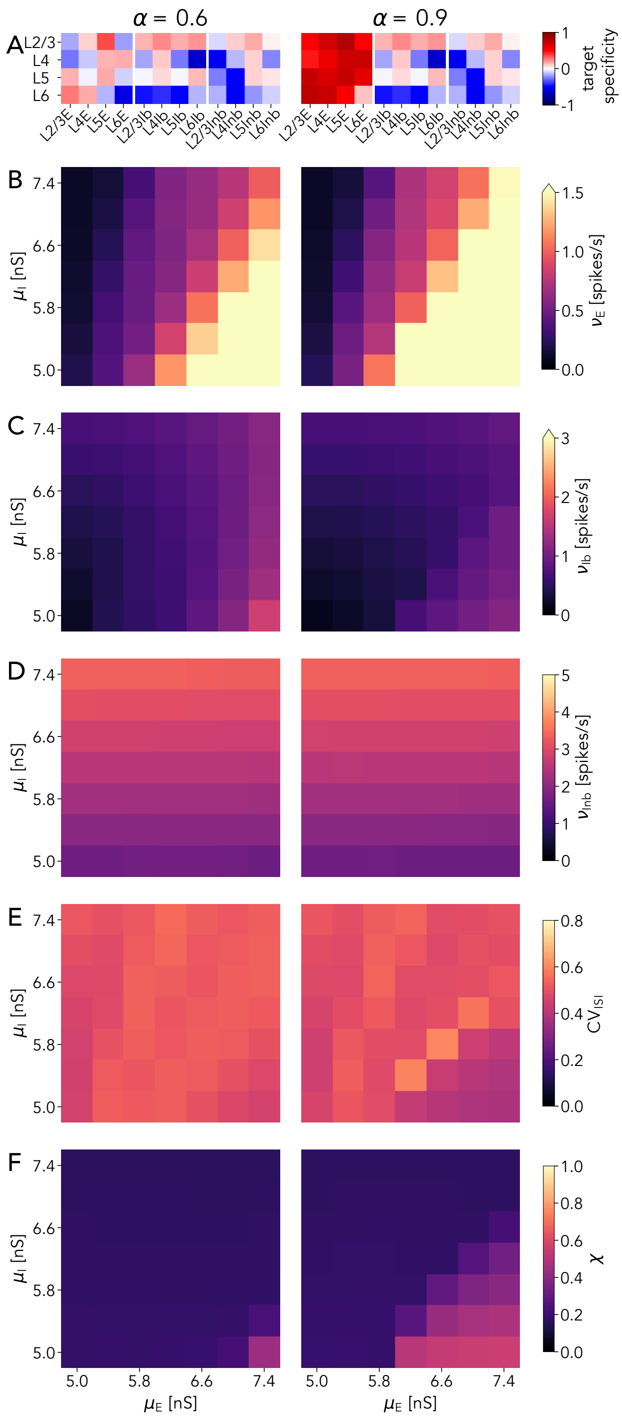

Figure 5.12: **Effect of increased target specificity in $\mathcal{M}_{EM}$.** Parameter $\alpha \in [0, 1]$ linearly controls redistribution of synapses onto excitatory neurons leading to an increase in target specificity (cf. Section 5.3.5). $\alpha = 0$ corresponds to the original model, $\alpha = 1$ implies that all synapses target excitatory neurons. Color code in top panels as in Figure 5.11, in bottom panels as in Figure 5.10.

5.2.5 *Amending the macaque V1 model*

In this section, we revisit the model of macaque V1 introduced in Chapter 4 which derives part of its connectivity from the same LM data set by Binzegger et al. (2004). In Section 5.2.4, we identify that the connectivity derived from Binzegger et al. (2004) leads to divergent activity in a microcircuit model due to its targeting patterns onto excitatory neurons, which is not the case using the connectivity data derived from the MICrONS Consortium et al. (2021) data set. First, we recall that in the V1 model, the Binzegger et al. (2004) data is used to derive the connection probabilities of given synapses in given layers to postsynaptic populations (paragraph 4.3.1.1). These probabilities can straight-forwardly be replaced by the newly obtained EM-based data, thereby introducing a more balanced recurrent targeting pattern into the model of V1. Figure 5.13 shows the characteristic activity of the V1 model amended by EM data for a range of inhibitory connection strength g and external input ν_{ext} without further fine-tuning.

First, we observe that the sharp transition between quiescent and pathological states observed in Section 4.2.3 are not present. Instead, the population-averaged firing rate is a smooth function of the two parameters (Figure 5.13A). Furthermore, the activity stays balanced even for small values of inhibitory connection strength which have been observed experimentally (Figure 4B of Billeh et al. (2020)). Figure 5.13B,C show the spiking activity and firing rate statistics for $g = 1$ and $\nu_{\text{ext}} = 11$ spikes/s, demonstrating that the first-order statistics are in agreement with biological expectations of asynchronous irregular activity and firing rates between 0.1 and 10 spikes/s. We leave optimizing this ground state to more accurately reflect population-specific firing rates to future studies, as this work conducted for $\mathcal{M}_{EM}$ and $\mathcal{M}_{EM}$ following the method by Isbister et al. (2023).

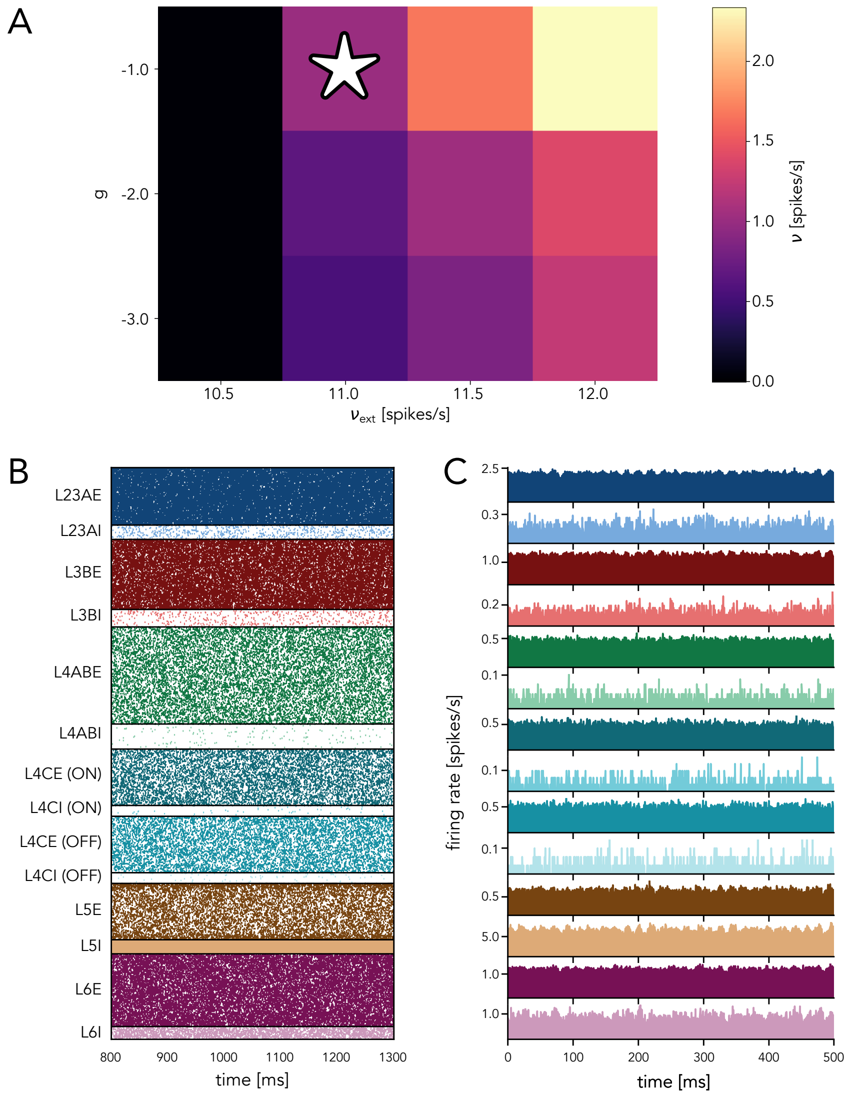

Figure 5.13: **Activity of spiking network of macaque V1 with EM connectivity data.** Homogeneous cortico-cortical input as in Figure 4.6. **A** Mean firing rate ν across all neurons depending g, the weight of inhibitory recurrent connections relative to the strength of excitatory ones, and ν_{ext}, the rate of external drive. Star indicates parameters for raster plot. **B** Raster plot for $g = -1$ and $\nu_{\text{ext}} = 11$ spikes/s subsampled to show 10 % of all spikes. **C** Average spontaneous activity of the model populations. Same color coding as in B.

5.3 METHODS

5.3.1 *Spatial connectivity*

To obtain estimates for the characteristic lengths of local cortical connectivity, we assume an exponential decay for the mean number of synapses with lateral distance between somata (Equation 5.1). The EM reconstruction allows for determining the density of the mean number of actual synapses $s_{AB}(d)$ between a single presynaptic proofread neuron and all postsynaptic neurons at inter-somatic distance d for pairs of the considered cell types. Using the distribution of uniformly sampled points in a rectangle with distance d (Sheng, 1985), we approximate the number of potential postsynaptic partners $\omega(d)$ given the dimensions of the reconstructed volume. We then fit

$$s_{AB}(d) = S^0_{AB} \cdot \omega(d) \cdot \exp\left(-d/\lambda^{\text{syn}}_{AB}\right) \tag{5.4}$$

to obtain λ_{AB} between presynaptic population B and postsynaptic population A.

Additionally, we estimate the distance-dependent connection probability, i.e., the probability of establishing at least one synapse. We define the connection probability $p_{AB}(d_1, d_2)$ between a presynaptic neuron of population B and a postsynaptic neuron of population A that have a horizontal inter-somatic distance d with $d_1 \leq d \leq d_2$ as the number of connected neuron pairs divided by the total number of neuron pairs in the same range of distances. Here, the total number of neuron pairs is given by the product of the number of neurons of populations A and B within the range of distances. We also assume an exponential decay for the continuous $p_{AB}(d)$, albeit with a different characteristic length scale (Equation 5.2). The quantity $c^\star = \langle \frac{\lambda^{\text{syn}}_{AB}}{\lambda^{\text{conn}}_{AB}} \rangle_{AB}$ measures the discrepancy between the distance dependence of the mean number of synapses and the connection probability based on the EM data set.

For the LM data set, Stepanyants et al. (2007) derive the connection probability using potential connectivity, defined as the probability that a source and target neuron of given cell types at inter-somatic distance d form at least one synapse based on morphological reconstructions of single neurons. This connection probability is given for a subset of pairs of populations. For these, we again fit Equation 5.2 which we convert to an estimate of the characteristic length of the mean number of synapses $\lambda^{\text{syn}}_{AB}$ using $c^\star$. For a different subset of pairs of populations, Stepanyants et al. (2007) derive the distance-dependent mean number of synapses $S_{AB}(d)$ under further assumptions, which we fit using Equation 5.1 to also obtain $\lambda^{\text{syn}}_{AB}$. These fits are displayed in Figure A.1.4. Finally, for pairs of populations in our model where

Stepanyants et al. (2007) provide neither of the two quantities, we generalize from the existing estimates as detailed in Section A.1.2.

Further, we assess the consistency of the LM potential connectivity with the potential and actual connectivity from the EM data. For this, we use the same morphology-based ansatz to derive a connection probability based on potential connectivity from the EM data. Compared to (Stepanyants et al., 2007), we use more single-cell reconstructions (266 compared to 41), and fewer cell positions within each box (100 compared to 1000). Then, we compare the resulting length scales of spatial connectivity with estimates from the actual connectivity derived above. For the EM data set, the two methodologies result in estimates of $\lambda_{AB}^{\mathrm{conn}}$ that are comparable across all pairs of populations.

In all cases, `scipy.optimize.curve_fit` (Jones et al., 2001) performs non-linear least-squares fits using the Levenberg-Marquardt algorithm (Moré, 2006).

5.3.2 *Connectivity maps*

To constrain the size of the models, we calculate

$$F(r) = \langle F_{AB}(r) \rangle_{AB} = \left\langle \frac{S_{AB}^{\mathrm{int},\leq r}}{S_{AB}^{\mathrm{total},\leq r}} \right\rangle , \tag{5.5}$$

the fraction of intra-model synapses for a circular patch of cortex with radius r. To determine the expected number of synapses from population B to population A of neurons within a circular patch with radius r, $S_{AB}^{\mathrm{int},\leq r}$, we uniformly distribute neurons with realistic densities in the patch and calculate the number of synapses according to Equation 5.1. Similarly, the total number of synapses between the populations, $S_{AB}^{\mathrm{total},\leq r}$, is determined by increasing the sampling radius of the presynaptic population to $3r$. The model radius $r^\star$ for both models is determined so that $F(r^\star) \approx 75\%$.

We calculate the number of synapses between populations A and B internal to the model as

$$S_{AB}^{\mathrm{int}} = \pi r^{\star 2} \cdot \rho_{AB}^{S} \cdot F_{AB}(r^\star) ,$$

where ρ_{AB}^{S} is the density of synapses from population B to population A extracted from the respective data set, and finally the distance-independent connection probability between a neuron in presynaptic population B and a neuron in postsynaptic population A as

$$p_{AB} = \frac{S_{AB}^{\mathrm{int}}}{N_A N_B}$$

where N_A, N_B are the number of neurons in populations A and B, respectively. Population pairs having insufficient data to constrain the characteristic length scales (cf. A.1.2) are neglected in determining the model sizes, but they are nevertheless connected according to the (scaled) total number of synapses in the respective data sets. Thus, also populations without a fitted length scale are connected, unless the corresponding number of reported synapses is zero.

5.3.3 *Network model and single-neuron dynamics*

Using the derived connectivity maps we instantiate network graphs for the two models using a pairwise Bernoulli connection rule with source- and target-population-specific connection probabilities p_{AB} (see the user-level documentation of NEST for an ontology based on (Senk et al., 2022)). The resulting networks are directed versions of Erdős-Rényi graphs (Erdős and Rényi, 1959).

Single neurons are modeled as leaky integrate-and-fire units with conductance-based synapses. The membrane potential for units representing inhibitory basket cells evolves as

$$\begin{aligned} C_m \dot{V}_m = &- g_L \left(V_m - E_L\right) - \left(g_{\text{ex}} + g_{\text{ext}}\right) \left(V_m - E_{\text{ex}}\right) \\ &- g_{\text{in}} \left(V_m - E_{\text{in}}\right) \end{aligned} \tag{5.6}$$

with the membrane capacitance C_m, leak conductance g_L and reversal potentials $E_L, E_{\text{ex}}, E_{\text{in}}$. The conductances obey

$$\begin{aligned} \tau_{\text{ex}} \dot{g_{\text{ex}}}(t) &= -g_{\text{ex}}(t) + \tau_{\text{ex}} \sum_j w_j \sum_{t_j} \delta(t - t_j - d_j) \\ \tau_{\text{in}} \dot{g_{\text{in}}}(t) &= -g_{\text{in}}(t) + \tau_{\text{in}} \sum_k w_k \sum_{t_k} \delta(t - t_k - d_k) \\ \tau_{\text{ext}} \dot{g_{\text{ext}}}(t) &= -(g_{\text{ext}}(t) - \mu) + \sigma \sqrt{2\tau_{\text{ext}}} \eta(t) \ , \end{aligned}$$

where j (k) is a presynaptic neuron, t_j (t_k) its spike times, w_j (w_k) the weight on a target neuron, d_j (d_k) is the connection delay, $\tau_{\text{ex}}, \tau_{\text{in}}$ are the synaptic time constants, μ is the mean and σ the standard deviation of the extrinsic drive modeled as an Ornstein-Uhlenbeck process (η denotes white noise), and τ_{ext} its timescale. The timescales of response conductances and the extrinsic conductance modulation can differ since the latter entails slower fluctuations from the not explicitly modeled neural tissue. For all simulations, we fix $\sigma/\mu = \xi = 0.2$ (Destexhe et al., 2001) while the mean varies between populations and experiments. If V_m crosses the threshold V_{th} at time t', the unit emits a spike with this time-stamp, is reset to V_{reset}, and is kept at this value for τ_{ref}. Units representing excitatory cells or inhibitory non-basket cells have additional currents that implement spike-frequency adaptation (a) and relative refractoriness (rr):

$$\begin{aligned} C_m \dot{V}_m = &-g_L \left(V_m - E_L\right) - \left(g_{\text{ex}} + g_{\text{ext}}\right) \left(V_m - E_{\text{ex}}\right) \\ &- g_{\text{in}} \left(V_m - E_{\text{in}}\right) - g_{\text{a}} \left(V_m - E_{\text{a}}\right) - g_{\text{rr}} \left(V_m - E_{\text{rr}}\right) . \end{aligned} \tag{5.7}$$

The corresponding conductances evolve according to

$$\tau_{\mathrm{a}} \dot{g_{\mathrm{a}}}(t) = -g_{\mathrm{a}}(t) + \tau_{\mathrm{a}} \sum_{t_i} \delta(t - t_i)$$
$$\tau_{\mathrm{rr}} \dot{g_{\mathrm{rr}}}(t) = -g_{\mathrm{rr}}(t) + \tau_{\mathrm{rr}} \sum_{t_i} \delta(t - t_i)$$

where t_i are the timings of the spikes emitted by the units.

The neuron models used in this study are specified using NESTML (Plotnikov et al., 2016). The state of the neurons (Equation 5.6, Equation 5.7) is integrated with an embedded Runge-Kutta-Fehlberg 4(5) method (Fehlberg, 1969). The Langevin equation describing the extrinsic drive is integrated using an exact scheme (Gillespie, 1996; Rotter and Diesmann, 1999).

5.3.4 *Analysis of dynamical data*

The firing rates are determined for each neuron by counting the spikes and dividing by the observation time. Coefficients of variation of the inter-spike intervals (ISIs) for each neuron are given by

$$\mathrm{CV}_{\mathrm{ISI}} = \frac{\text{standard deviation of ISIs}}{\text{mean of ISIs}} \,. \tag{5.8}$$

Both quantities are calculated using the Electrophysiology Analysis Toolkit `elephant` (Denker et al., 2022). To determine the synchrony χ of each population, we record the membrane potential of 50 neurons per population and calculate

$$\chi^2 = \frac{\left\langle \left(\langle V_m \rangle_i - \langle V_m \rangle_{i,T} \right)^2 \right\rangle_T}{\left\langle \left(V_m - \langle V_m \rangle_T \right)^2 \right\rangle_{i,T}} \tag{5.9}$$

where $\langle ... \rangle_i$ denotes the average over the neurons in one population, and $\langle ... \rangle_T$ denotes the time average.

5.3.5 *Target specificity*

For a presynaptic population B and a layer v, the target specificity is given by

$$\mathrm{TS}_{v,B} = \frac{p_{v\mathrm{E},B} - p_{v\mathrm{Ib},B} - p_{v\mathrm{Inb},B}}{p_{v\mathrm{E},B} + p_{v\mathrm{Ib},B} + p_{v\mathrm{Inb},B}} \in [-1, 1] \,. \tag{5.10}$$

A positive (negative) value indicates that B preferentially targets excitatory (inhibitory) neurons in layer v.

We redistribute synapses between populations to increase the value of the target specificity using a control parameter $\alpha \in [0, 1]$ while keeping the total number of synapses fixed. On the level of the mean number of synapses, this amounts to:

$$\begin{aligned}
S^{\mathrm{int},\alpha}_{u\mathrm{Ib},v\mathrm{E}} &= (1-\alpha)\cdot S^{\mathrm{int}}_{u\mathrm{Ib},v\mathrm{E}} \\
S^{\mathrm{int},\alpha}_{u\mathrm{Inb},v\mathrm{E}} &= (1-\alpha)\cdot S^{\mathrm{int}}_{u\mathrm{Inb},v\mathrm{E}} \\
S^{\mathrm{int},\alpha}_{u\mathrm{E},v\mathrm{E}} &= S_{v\mathrm{E},v\mathrm{E}} + S^{\mathrm{int}}_{u\mathrm{Ib},v\mathrm{E}} + S^{\mathrm{int}}_{v\mathrm{Inb},v\mathrm{E}}
\end{aligned}$$

This definition implies that $\alpha = 0$ leaves the connectivity maps (and thus the target specificity values) unchanged, while for $\alpha = 1$ all synapses exclusively target excitatory populations (TS $= 1$).

5.4 DISCUSSION

In this chapter, we revisit the local connectivity of the visual cortex in light of recent advances in using electron microscopy (EM) for reconstructing neuronal processes and synapses. We compare our findings to results from an established data set which is based on morphological reconstructions using light microscopy (LM). First, we investigate the lateral spatial spread of connections. Comparing the spatial decay of the mean number of synapses and the connection probability, we find that the former drops off faster than the latter with a robust factor of $c^{\star} = 0.7$ averaged over populations. Further, using the precise EM data set, we derive both the actual connection probability and the potential connection probability, the latter of which is based solely on the morphological reconstruction of axonal and dendritic arbors. In a subsequent comparison we demonstrate that potential connectivity delivers accurate estimates for the actual connection probability with distance.

Second, we derive connectivity maps between populations of different cell types in different layers that describe their recurrent connection probabilities. Between the two models, we observe some general similarities in connectivity patterns such as stronger sub-circuits between excitatory neurons in layer 2/3 and 5.

Third, we study the dynamical behavior of spiking microcircuit models built from the anatomical information. While the LM-based model can exhibit biologically plausible activity when fine-tuning the external input to the populations, systematically varying the input parameters reveals that this is only the case for a narrow band of external drive parameters. When slightly moving away from the optimized input, the activity in the LM-based microcircuit quickly vanishes or diverges. As such, it is not a good model of how the brain operates under varied input conditions. In contrast, the EM-based microcircuit exhibits biologically plausible activity for a wider range of input activity, depending smoothly on the provided input. Further, it exhibits the paradoxical effect of decreasing firing rates in inhibitory populations when their external drive is increased, which is also observed in experiments (Adesnik, 2017).

Fourth, we link the difference in dynamical behavior between the models to their underlying connectivity. Investigating the targeting patters, i.e., whether a neuron of a given population prefers to target excitatory or inhibitory neurons, reveals putative excitatory feedback loops in the LM-based connectivity. Conversely, excitatory neurons in the EM-based microcircuit prefer to target inhibitory neurons, while the targeting patterns of inhibitory neurons are more balanced. We show that shifting the targeting pattern of the EM-based circuit towards the one observed in the LM-based circuit, we reproduce the unstable dynamical behavior qualitatively. From this experiment, we conclude that targeting patterns play a key role in balanced activity of microcircuits. Due to the favorable dynamics observed in the EM-based model, we suggest that future simulation studies should use connectivity derived from EM data over LM-based connectivity data. We want to highlight that this work already provides the connectivity derived from EM data in a format that is directly usable for instantiating spiking neural networks. Thereby, LM-based connectivity structure can straight-forwardly be replaced in existing models.

Finally, we return to our spatial model of macaque V1 introduced in Chapter 4, where we use the same LM data to derive connection probabilities from existing synapses to postsynaptic neurons. In that chapter, we found that the subsequently instantiated spiking network model is unable to reach a stable ground state of biologically plausible activity. In Section 4.2.3 we hypothesize that the targeting patterns of the model are a main cause of this instability. Replacing the LM-based connectivity by the newly derived EM-based connectivity in our model of V1, we indeed confirm this hypothesis: the amended model of V1 exhibits asynchronous irregular activity with biologically plausible firing rates for a range of parameters without any further fine-tuning.

Exchanging the LM data for EM data in the model of macaque V1 also entails a change of species from which the connectivity data is derived: from cat, which has been under consideration for the LM data set, to mouse, which was investigated for the EM data set. On a large spatial scale, this might be problematic since the mouse lacks critical spatial connectivity principles that are of explicit interest for the model of macaque V1 developed here. In particular, neurons in mouse V1 are not organized according to an orientation tuning map, and are therefore also not connected via patchy connectivity which links iso-orientation domains across visual receptive fields. However, we here only use the mouse data for deriving the total number of synapses between populations, in the same way the cat data was used before (see Section 4.2.1 for an overview and Section 4.3.1 for a detailed derivation). Thus, the spatial connectivity employed in the model (Section 4.2.2) is not affected by this change of data source.

This concludes our endeavor of constructing a spatially organized model of macaque primary visual cortex. The V1 model in its final iteration, including the amendment by EM connectivity data, can act as a platform on which future studies can thoroughly investigate the role of spatial connectivity in the processing of visual information in cortical networks. Further, it can be extended by including similarly constructed models of areas higher in the cortical hierarchy to investigate the effect of converging and diverging spatial connectivity between areas, or refining the thalamic input model to provide even more realistic sensory drive to the cortical populations.

Part IV

TOWARDS PRECISE SPATIAL CONNECTIVITY FOR MODELING

PREFACE

In Part iii, we explore various spatial connectivity principles that govern the organization of local cortical circuits on different length scales. In particular, we investigate long-range patchy connectivity, short-range push-pull connectivity, and medium-range isotropic connectivity. In this part of the thesis, we challenge the assumptions of the latter, which neglects complex synaptic targeting patterns in favor of a simple description of connection probability that only depends on the inter-somatic distance. Leveraging the highly detailed connectivity data based on electron microscopy (EM) by MICrONS Consortium et al. (2021) already employed in Chapter 5, we take a closer look at the spatial distributions of synaptic connections—or *constellations*—that neurons of different cell types exhibit. Deriving population-averaged constellations for both postsynaptic sites on dendrites and presynaptic sites on axons, we derive a novel connectivity rule based on a measure of overlap of constellations.

This, however, raises the question of how to compare the accuracy of connectivity rules. Concretely, how do we assess which connectivity rule is more suited to capturing the inherent connectivity structure that can be observed by experiments? Opportunely, the EM data set provides a ground truth connectivity between the reconstructed neurons in the cortical tissue under consideration. Representing the network graph as an adjacency matrix, where each entry specifies the number of synapses between pairs of neurons, the comparison between connectivity rules condenses to assessing the similarity between the predicted and observed adjacency matrices. Thus, we start by developing *singular angle similarity* (SAS)—a similarity measure for matrices that can be used to assess the pairwise closeness of connectomes—in Chapter 6. Here, we extend existing approaches to non-symmetric matrices, which is necessary since local circuits in the brain are principally directed. While we thoroughly test the developed measure for this specific purpose, we note that the measure is generally applicable to any pair of real-valued matrices that have the same shape.

Then, in Chapter 7, we derive the novel connectivity rule based on the precise spatial constellations of synapses. Bringing the insights from both chapters together, we finally compare the predictive power of the constellation-based rule to a traditional, purely distance-dependent one by assessing the similarity to the experimental ground truth using SAS. We find that the constellation-based rule is able to robustly outperform the traditional rule, capturing the intricate details of spatial synaptic targeting patters that are characteristic for each neural population to a greater degree. This constellation-based predictive connectivity rule is specifically tailored for future computational studies of neural networks that aim at investigating the structure-dynamics relationship, akin to the work presented in Part iii.

6

QUANTIFYING THE SIMILARITY OF MATRICES WITH ARBITRARY SHAPE

This chapter is based on the following publication:
J. Albers*, A. C. Kurth*, R. Gutzen, A. Morales-Gregorio, M. Denker, S. Grün, S. J. van Albada, M. Diesmann: *Assessing the similarity of real matrices with arbitrary shape* (Albers et al., 2024). * indicates equal contributions.

Author contributions:
The author conceptualized this work together with Anno Christopher Kurth (ACK) and Robin Gutzen (RG). The author jointly constructed the methodology with ACK and RG. The software module was implemented by the author, RG, ACK, and Aitor Morales Gregorio (AMG). The author derived the formal analysis together with ACK, RG and AMG, with whom he also wrote the original draft. The review and editing was performed by the author, ACK, RG, AMG, Michael Denker (MDe), Sonja Grün (SG), Sacha J. van Albada (SvA) and Markus Diesmann (MDi). The visualization was done by the author, ACK, AMG, and RG. MDe, SvA, and MDi supervised the work. The funding was acquired by SG, SvA, and MDi.

6.1 INTRODUCTION

Many complex systems can be characterized by interactions between their constituent parts, for example social groups, transportation systems, chemical reactions, or brain networks. In the study of such systems, matrices are a useful tool to represent pairwise interactions, spatio-temporal dependencies, or the covariance structure. For example, the adjacency matrix of a neural network describes the recurrent connectivity between its neurons. Quantifying the similarity between such matrices provides valuable information for analyzing the underlying systems at hand.

Traditional measures of matrix similarity are typically derived from the Frobenius scalar product $\langle A, B\rangle_F = \mathrm{tr}(AB^T)$ between two matrices A and B, such as the Frobenius norm, $||A - B||_F^2 = \langle A - B, A - B\rangle_F$, or the cosine similarity, $\langle A, B\rangle_F$, where $||A||_F = ||B||_F = 1$ (Robert and Escoufier, 1976; Schönemann, 1966). However, these approaches neglect the inherent two-dimensional structure of matrices. Alternatively, similarity can be computed by calculating the canonical angles between the subspaces spanned by the matrix columns (Golub and

Zha, 1995). Because the embedding space needs to be of significantly higher dimension than the subspaces, this approach is limited to matrices where the number of rows is much larger than the number of columns (or vice-versa). To overcome these limitations, Gutzen et al. (2023) introduced a similarity measure based on the alignment of eigenvectors of the matrices under consideration. However, this method comes with its own set of limitations as it is only well-defined for square, symmetric, positive definite matrices.

Here, we present a measure that overcomes the limitations imposed by previous approaches. Using singular value decomposition (SVD), we derive the *singular angle similarity* (SAS) from the alignment of the respective left and right singular vectors. By construction, it takes the two-dimensional structure of matrices into account while being applicable to arbitrarily shaped real matrices.

We start by formally deriving SAS in Section 6.2.1. To provide an alternative viewpoint, Section 6.2.2 gives a geometric interpretation of the derived measure. Third, we evaluate how SAS compares to traditional measures of matrix similarity by testing its ability to recognize shared structure between classes of random matrices in Section 6.2.3. Fourth, we characterize how SAS behaves under change in matrix dimensionality, addition of noise, and introduction of degeneracy in Section 6.2.4. Then, we turn to concrete use cases: Section 6.2.5 demonstrates that SAS can provide a nuanced classification of random graphs based on their adjacency matrices, and Section 6.2.6 shows how SAS can be used to decode brain states from electrophysiological recordings.

Section 6.3 formally defines the random matrix classes (Section 6.3.1) and random graphs (Section 6.3.2) as well as the notion of introducing degeneracy to matrices (Section 6.3.4) used throughout this work. Further, we detail the preprocessing of neural data in Section 6.3.3.

Finally, Section 6.4 summarizes our findings and highlights how SAS can be used in the context of comparing predictive connectivity rules.

6.2 RESULTS

We present a measure for assessing the structural similarity between two arbitrary, real matrices named *singular angle similarity* (SAS). The measure is based on singular value decomposition (SVD), which introduces the left and right singular vectors with corresponding singular values (Trefethen and Bau, 2022). SAS exhibits the following properties:

- SAS attains values between 0 and 1 where higher values imply greater similarity.
- SAS is invariant under actions of identical orthogonal maps from the left or the right on the compared matrices; this includes the consistent permutation of rows and columns as a special case (Section A.2.2).
- SAS is invariant under transposition of both matrices (Section A.2.2).
- SAS is invariant under scaling with a positive factor; in particular, SAS $= 1$ for $M_b = c_1 M_a, c_1 \in \mathbb{R}^+$ (Section A.2.3).
- SAS is zero if the two compared matrices are equal up to a negative factor: SAS $= 0$ for $M_b = c_2 M_a, c_2 \in \mathbb{R}^-$ (Section A.2.3).

6.2.1 *Derivation of singular angle similarity*

Consider two arbitrary, real, $m \times n$ matrices M_a, M_b. Without loss of generality, we assume $m \leq n$. SVD guarantees the existence of orthogonal matrices $U_i \in \mathbb{R}^{m\times m}$ and $V_i \in \mathbb{R}^{n\times n}$, and diagonal matrices $\Sigma_i = \mathrm{diag}(\sigma_i^1, ..., \sigma_i^m) \in \mathbb{R}^{m\times n}$ where $\sigma_i^j \geq \sigma_i^l \geq 0$ for $l > j \geq 1$ such that

$$M_i = U_i \Sigma_i V_i^T \ . \tag{6.1}$$

Here, $i \in \{a, b\}$. SVD is schematically presented for a 2×2 case in Figure 6.1. The columns of V_i, denoted by $v_i^1, ..., v_i^n$, are the right singular vectors, and the columns of U_i, denoted by $u_i^1, ..., u_i^m$, are the left singular vectors. With this, the SVD can also be written in the form

$$M_i = \sum_{j=1}^{m} \sigma_i^j u_i^j \otimes v_i^j \ . \tag{6.2}$$

Here, $\otimes$ denotes the outer product of two vectors. Thus, under the action of M_i the vectors v_i^j are transformed into the vectors $\sigma_i^j u_i^j$. The singular values $\sigma_i^1, ..., \sigma_i^m$ are unique—they are the square root of the eigenvalues of $M_i M_i^T \in \mathbb{R}^{m\times m}$. We note that if both M_i are symmetric

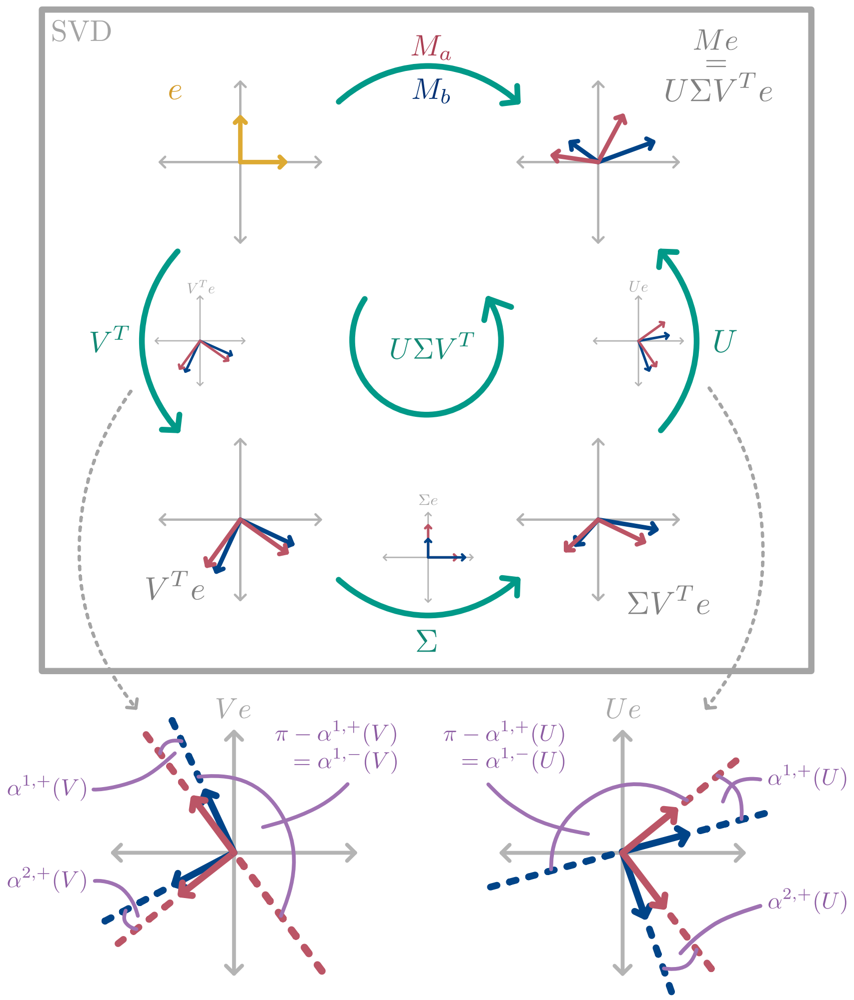

Figure 6.1: **Singular value decomposition and singular angles.** Schematic representation of the transformations applied on the basis vectors e (yellow) by the components of SVD for two 2×2 matrices M_a (red) and M_b (blue). Small graphs next to the green arrows illustrate the isolated action of the corresponding transformations V^T, Σ, and U on e. Below, the singular angles of the first ($\alpha^{j,+}(V)$ and $\alpha^{j,+}(U)$) and second ($\alpha^{j,-}(V)$ and $\alpha^{j,-}(U)$) kinds are shown as the angles between the column vectors of V_a and V_b, and U_a and U_b, respectively, as defined in Equation 6.3.

positive-definite matrices (e.g., covariance matrices), then $U_i = V_i$, and the singular values become the eigenvalues.

For simplicity of the derivations, we at first assume that there are no degenerate singular values except zero. The overwhelming majority of higher-dimensional matrices encountered in practice indeed satisfy this assumption.

Left and right singular vectors that correspond to non-degenerate singular values are uniquely determined up to a joint multiplication by -1. Thus, the vector pairs (u_i^j, v_i^j) and $(-u_i^j, -v_i^j)$ both are equally valid singular vectors to a non-degenerate singular value $\sigma_i^j \geq 0$.

If $\min\{\sigma_a^j, \sigma_b^j\} \neq 0$ we define the left singular angle $\alpha^{j,+}(U)$ and the right singular angle $\alpha^{j,+}(V)$ of the first kind as

$$\begin{aligned}
\alpha^{j,+}(U) &= \sphericalangle(u_a^j, u_b^j) = \arccos\left(\langle u_a^j, u_b^j\rangle\right), \\
\alpha^{j,+}(V) &= \sphericalangle(v_a^j, v_b^j) = \arccos\left(\langle v_a^j, v_b^j\rangle\right).
\end{aligned} \tag{6.3}$$

Due to the ambiguity in vector pairs of left and right singular vectors we additionally define left singular angles of the second kind as $\alpha^{j,-}(U) = \sphericalangle(-u_a^j, u_b^j) = \sphericalangle(u_a^j, -u_b^j)$ and *mutatis mutandis* for right singular angles of the second kind $\alpha^{j,-}(V)$. The singular angles of the first and second kinds are visualized in Figure 6.1. There, we have

$$\alpha^{j,+}(U) + \alpha^{j,-}(U) = \pi = \alpha^{j,+}(V) + \alpha^{j,-}(V)\,. \tag{6.4}$$

One has to consider either $\left(\alpha^{j,+}(U), \alpha^{j,+}(V)\right)$ or $\left(\alpha^{j,-}(U), \alpha^{j,-}(V)\right)$ together due to the ambiguity in the sign. We define the *singular angle* as the smaller average of the two choices

$$\begin{aligned}
\alpha^j &= \min\left\{\frac{\alpha^{j,+}(U) + \alpha^{j,+}(V)}{2}, \frac{\alpha^{j,-}(U) + \alpha^{j,-}(V)}{2}\right\} \\
&= \min\{\bar{\alpha}^j, \pi - \bar{\alpha}^j\}
\end{aligned} \tag{6.5}$$

where $\bar{\alpha}^j = (\alpha^{j,+}(U) + \alpha^{j,+}(V))/2$. Using the angular similarity

$$\Delta^j = 1 - \frac{\alpha^j}{\pi/2} \in [0,1] \tag{6.6}$$

and defining the *singular value score* as $w^j = w(\sigma_a^j, \sigma_b^j)$ where $w(x,y) \geq 0$ denotes a weight function, we calculate SAS as the weighted average of the angular similarities:

$$\mathrm{SAS} = \frac{\sum_j^k w^j \Delta^j}{\sum_j^k w^j} \in [0,1]\,. \tag{6.7}$$

Here, k is the largest natural number less than or equal to m such that $\min\{\sigma_a^k, \sigma_b^k\} \neq 0$. In the following, we choose

$$w(x,y) = (x+y)/2\,. \tag{6.8}$$

Another possible choice is $w(x,y) = \sqrt{(x^2 + y^2)/2}$ (cf. Gutzen et al. (2023)). One can substitute other vector-based similarity measures for the angular similarity defined in Equation 6.6. For example, substituting cosine similarity yields $\Delta^j = \cos(\alpha^j)$.

According to our definition of SAS in Equation 6.7, singular angles stemming from singular vectors of which at least one has a corresponding singular value of zero do not contribute to SAS.

DEGENERATE CASE If two or more nonzero singular values are identical, the described approach cannot be readily applied since there is no canonical choice for pairing singular vectors: left and right singular vectors $u_i^k, ..., u_i^{k+l}$ and $v_i^k, ..., v_i^{k+l}$ corresponding to the degenerate singular value $\sigma = \sigma_i^k = ... = \sigma_i^{k+l}$ are only unique up to an orthogonal transformation acting on the subspaces spanned by the singular vectors. This is the higher-dimensional generalization of the ambiguity regarding the factor -1 for the one-dimensional case described above. Writing the corresponding vectors as columns of matrices $U_i^{[k:k+l]} \in \mathbb{R}^{n \times l}$ and $V_i^{[k:k+l]} \in \mathbb{R}^{m \times l}$, this means that the columns of the matrices

$$U_i^{[k:k+l]} O \quad \text{and} \quad V_i^{[k:k+l]} O \tag{6.9}$$

are valid left and right singular vectors. Here, $O \in \mathbb{R}^{l \times l}$ is an arbitrary orthogonal matrix. Note that in order to maintain consistency between the singular vectors, the same orthogonal transformation O must be applied to both, $U_i^{[k:k+l]}$ and $V_i^{[k:k+l]}$. Thus, left and right singular vectors can only be given up to this ambiguity, and consequently singular angles according to Equation 6.3 are not well-defined.

This problem can be mitigated by using the *canonical angles* between the subspaces spanned by the left and right singular vectors of the matrices M_i (Jordan, 1875). If the two subspaces are given in terms of orthonormal bases B_i (written again as columns of matrices), the canonical angles are the angles between corresponding column vectors of $B_a O_a$ and $B_b O_b$, where O_a, O_b are suitable orthogonal transformations (Zhu and Knyazev, 2013) (see Section A.2.1). This can be interpreted as optimally aligning the two orthonormal coordinate systems while keeping the subspaces invariant.

We define singular angles for degenerate singular values building on that interpretation: assuming without loss of generality $\sigma = \sigma_a^k = ... = \sigma_a^{k+l}$, we define U-aligned left and right singular angles for $k \leq j \leq k + l$ as

$$\begin{aligned} \alpha^{j,U}(U) &= \sphericalangle \left(U_a^{[k:k+l]} O_a^U e_j, U_b^{[k:k+l]} O_b^U e_j \right) \\ \alpha^{j,U}(V) &= \sphericalangle \left(V_a^{[k:k+l]} O_a^U e_j, V_b^{[k:k+l]} O_b^U e_j \right) \ . \end{aligned} \tag{6.10}$$

Here, e_j is the j-th standard normal basis vector, and the orthogonal transformations O_a^U, O_b^U are chosen such that $\alpha^{j,U}(U)$ are the canonical angles. The U-aligned singular angles are then given by the mean of the U-aligned left and right singular angles:

$$\alpha^{j,U} = \frac{\alpha^{j,U}(U) + \alpha^{j,U}(V)}{2} \tag{6.11}$$

The corresponding angles in the V-aligned case are defined *mutatis mutandis*. The singular angles are then given by either the U- or V-

aligned singular angles, depending on which have the smaller sum:

$$\alpha^j = \alpha^{j,X} \text{ where } X = \begin{cases} U & \text{if } \sum_j \alpha^{j,U} < \sum_j \alpha^{j,V} \\ V & \text{if } \sum_j \alpha^{j,V} \leq \sum_j \alpha^{j,U} \end{cases} \tag{6.12}$$

In this way, the singular angles for singular vectors corresponding to degenerate singular values directly generalize the singular angles in the non-degenerate case. For their contribution to SAS, we define $w^j = w\left(\sigma_a^j, \sigma_b^j\right)$ for $k \leq j \leq k+l$, i.e. the singular values are combined in their matching order. A complication may arise when $\sigma_a^k = ... = \sigma_a^{k+l}$ and $\sigma_b^{k-m} = ... = \sigma_b^{k+n}$ for some m and n. Here, the degenerate subspaces of the two matrices are partially overlapping. We treat this case by applying the above described method to the subspaces given by $U_i^{[k-m:k+\max\{n,l\}]}$, $V_i^{[k-m:k+\max\{n,l\}]}$.

Additionally, if two singular values are close so that small perturbations can lead to a change in their order, the pairing of singular vectors will change, potentially leading to different SAS values. This can be avoided by rounding the singular values to a precision determined for the matrices at hand. Thereby, degeneracy is introduced which can be treated as described above.

6.2.2 *Geometric interpretation of SAS*

Singular angle similarity has a geometric interpretation. The left and right singular vectors of M_i ($i \in \{a, b\}$) are the respective eigenvectors of the square matrices $M_i M_i^T$ and $M_i^T M_i$. Further, $M_i M_i^T$ and $M_i^T M_i$ have the same eigenvalues (the squared singular values of M_i). Consider for each of these symmetric positive-definite matrices a hyperellipsoid spanned by the respective eigenvectors scaled by their eigenvalues (Figure 6.2).

The hyperellipsoid collapses in most dimensions as matrices M_i typically have only a small number of large singular values (cf. Marchenko and Pastur (1967)). Dimensions associated with the largest singular values dominate its shape, and the angle between the corresponding left and right singular vectors of the matrices M_a and M_b are of main relevance for SAS. Thus, a high SAS indicates that the hyperellipsoids are aligned, whereas a low SAS indicates misalignment or different shapes. If two matrices share two-dimensional structural features, their hyperellipsoids will be similarly shaped and point into similar directions, producing a high SAS. $M_i M_i^T$ and $M_i^T M_i$ are the correlation matrices up to a normalization by the number of rows and columns, respectively, and the subtraction of the mean. Therefore, SAS takes into account the correlation structure along both axes of the matrices. This distinguishes the measure from common methods such as cosine similarity and the Frobenius norm.

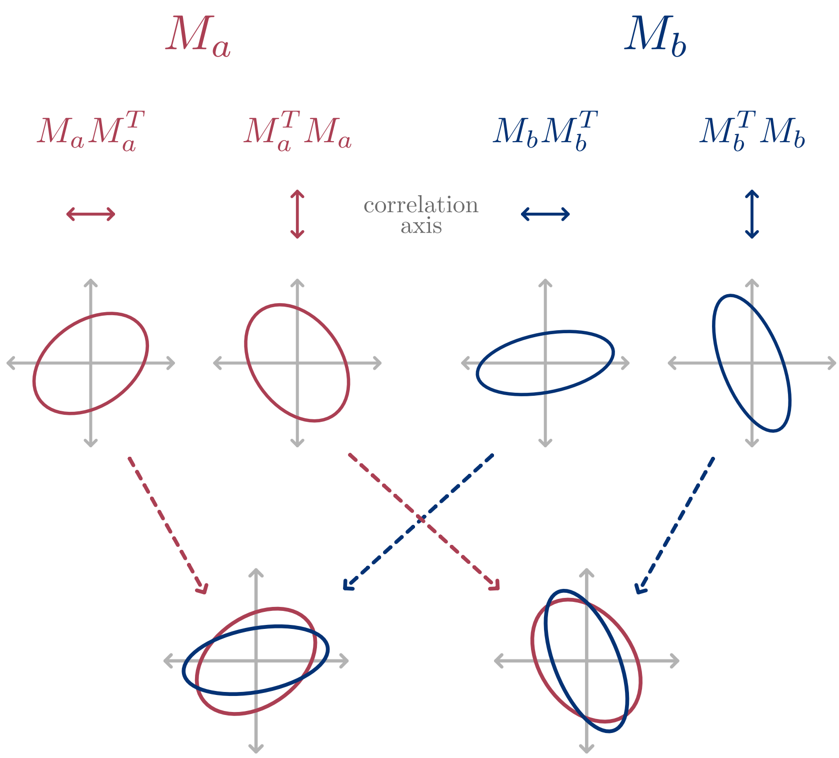

Figure 6.2: **Geometric interpretation of SAS.** Matrices M_a (red) and M_b (blue) as in Figure 6.1. The eigenvectors of $M_i M_i^T$ and $M_i^T M_i$ (same colors) scaled by the square root of their eigenvalues span the main axes of ellipsoids. These square matrices capture the correlation structure of M_i along the horizontal and vertical axis, respectively (double-headed colored arrows). SAS compares the angles between the corresponding ellipsoids (dashed colored arrows).

6.2.3 *Comparison with standard measures for random matrices*

By its very definition, SAS captures two-dimensional structures that are invisible to traditional measures of matrix similarity. Figure 6.3 shows the ability of different measures to discriminate between classes of random matrices with such structure.

Concretely, we compare SAS with cosine similarity,

$$\langle M_a, M_b \rangle_F = \mathrm{tr}(M_a M_b^T) \ , \tag{6.13}$$

and Frobenius distance,

$$||M_a - M_b||_F = \sqrt{\langle M_a - M_b, M_a - M_b \rangle_F}, \tag{6.14}$$

where we normalize the matrices such that

$$||M_a||_F = ||M_b||_F = 1 \ .$$

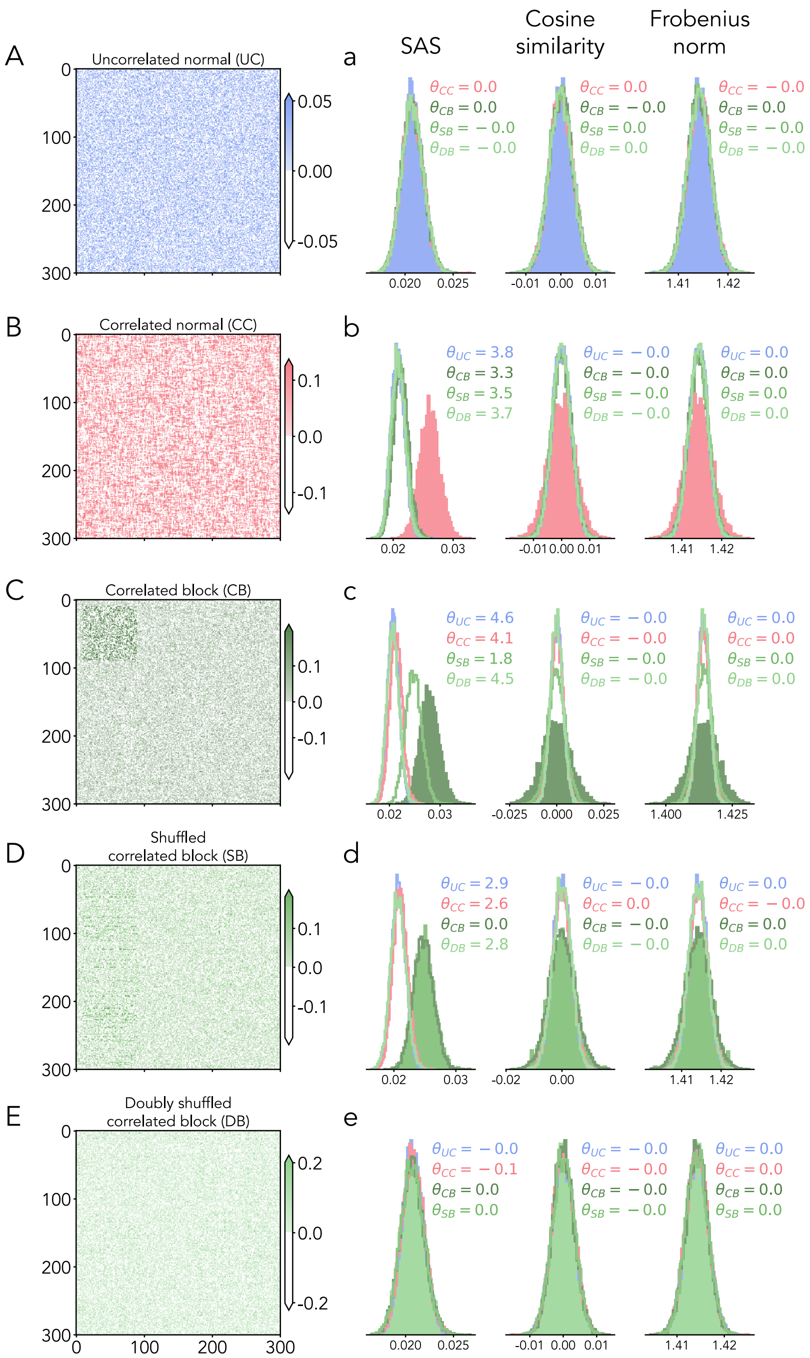

Figure 6.3: **Comparison of SAS with standard measures. A–E** Single instances of the different random matrix classes. For visibility, negative values are shown as white. **a–e** Histograms of SAS, cosine similarity, and the Frobenius norm between instances of the random matrix classes ($n = 100$, all pairs compared). Filled distributions indicate self-similarities (self-distances), non-filled ones indicate cross-similarities (cross-distances). Legends show effect sizes θ for the comparison between distributions.

Figure 6.3A–E show single instances of the five matrix classes defined in Section 6.3.1: uncorrelated normal matrices (UC) where matrix entries have no correlation; cross-correlated normal matrices (CC) where a 2d correlation between matrix entries is introduces; cross-correlated block matrices (CB) where the same correlation is introduced, but limited to a subset of matrix entries; shuffled, cross-correlated block matrices (SB) which are CB matrices with randomly shuffled rows; and doubly shuffled, cross-correlated block matrices which are CB matrices with randomly shuffled rows and columns.

We first calculate the *self-similarity*, the pairwise similarity between instances of the same random matrix class, and the *cross-similarity*, which refers to the similarity between instances of different classes (Figure 6.3a–e). We analogously define self- and cross-distance for the Frobenius distance. Subsequently, we investigate whether the different measures distinguish the random matrix classes from each other based on realizations of their particular structures. Fundamentally, this only works if the structure quantified by a measure is more similar between instances of the same class than across classes. Thus, for SAS and cosine similarity, the self-similarity must be meaningfully greater than the cross-similarities. Conversely, since the Frobenius norm measures a distance rather than a similarity, the self-distance must be smaller than the cross-distance. We call a difference meaningful if the effect size θ between pairs of distributions is greater than one. Assuming an underlying Gaussian model for the distributions, we employ the definition

$$\theta = \frac{\mu_{\text{self}} - \mu_{\text{cross}}}{\sqrt{\frac{\sigma^2_{\text{self}} + \sigma^2_{\text{cross}}}{2}}} \tag{6.15}$$

of the "Cohen's D" effect size Cohen (1988) underlying the common Student's and Welch's t-statistics Student (1908) and Welch (1947), where μ and σ are the mean and standard deviation of the self- and cross-similarity distributions. Thus, two distributions are meaningfully different if the distance of their means is greater than the quadratic mean of their standard deviations.

Figure 6.3a shows that UC matrices cannot be distinguished from the other matrix classes by any measure. This is expected: since the entries are independent, there is no detectable structure. In particular, this means that no structure is shared between different UC matrices or between UC matrices and matrices of other classes. Geometrically, this corresponds to ellipsoids that are oriented in random directions for each instance.

By definition, CC matrices exhibit such fluctuations. However, cosine similarity and the Frobenius norm fail to identify the common correlation structure (Figure 6.3b). Only SAS separates the self- and cross-similarity meaningfully and can thereby distinguish this matrix class from the others.

A similar conclusion holds true for CB matrices, where the correlated structure is embedded into an otherwise uncorrelated matrix (Figure 6.3c): again, only SAS separates the self- and cross-similarities meaningfully.

Next, we consider SB matrices. By construction, these are CB matrices with permuted rows. Between SB matrices, the CB correlation structure along the horizontal axis (quantified by MM^T) is destroyed while the correlation structure along the vertical axis (quantified by M^TM) stays the same. Since SAS takes into account both, it detects similarity between SB and CB matrices despite the permutation of the rows. This leads to a higher cross-similarity between CB and SB matrices than between CB and the other matrix classes (Figure 6.3c). Since the block structure exhibited by CB matrices can be viewed as one specific permutation of the rows, the self-similarity of SB and the cross-similarity between SB and CB follow the same distribution (Figure 6.3d) while SB matrices are separable from UC and CC matrices. The cosine similarity and the Frobenius norm fail to separate SB matrices from the other classes. The choice of the axis along which we permute is arbitrary; the results are identical if we permute columns instead of rows.

Finally, we turn to DB matrices (Figure 6.3e). Here, starting from CB matrices, both, the columns and rows are permuted. Consequently, between DB matrices a correlation structure is neither retained along the horizontal nor along the vertical axis. Neither SAS nor the other measures can detect similarity between DB matrices in comparison with DB matrices and matrices of the other classes.

Why are these examples relevant? They show that SAS captures certain two-dimensional correlation structures between instances. In contrast, the traditional measures cannot identify them. Additionally, SAS retains similarity even after permutation along one axis—including shifts as a special case. This is relevant in practical applications, for example in the analysis of highly parallel time series: even if the time series are not aligned, SAS exposes structural similarities. However, if both rows and columns are randomly permuted SAS fails to identify similarity. This is expected since in this case there is no two-dimensional correlation structure between instances left for SAS to identify.

6.2.4 *Characterization*

6.2.4.1 *Scale dependence and robustness*

To assess the dependence of SAS on matrix size, we calculate the self-similarity for increasing dimensionality N and observe a decreasing SAS for all random matrix classes (Figure 6.4A). Since the probability distribution for an angle between two random vectors increasingly cen-

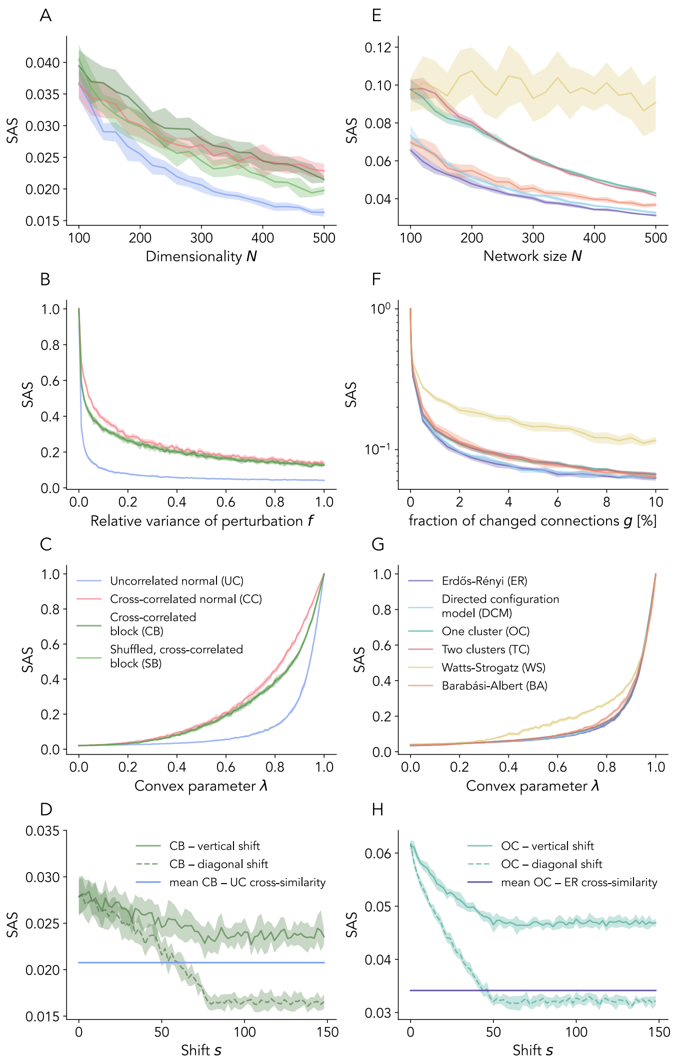

Figure 6.4: **Characterization of SAS.** In all panels, lines indicate the mean and shadings indicate the standard deviation over 10 realizations. In cases of constant matrix size, $N = 300$. A-C and E-G share the same legends. **A** Self-similarity for varying dimensionality N. **B** SAS between identical matrix instances for varying variance of an additive perturbation $\sigma^2_{\text{pert}} = f\sigma^2$. **C** SAS between matrices when increasing the degree of putative structural similarity quantified by λ. **D** SAS between CB matrices where the correlated block is shifted either vertically or diagonally for one matrix. **E** Self-similarity for varying network size N. **F** SAS between identical network model instances where the number of individual connections that are changed is gradually increased. **G** SAS between network model instances when increasing the degree of putative structural similarity quantified by λ. **H** SAS between OC matrices where the cluster is shifted either vertically or diagonally for one matrix.

ters around $\pi/2$ with increasing dimensionality (Cai et al., 2013), the resulting SAS between UC matrices decreases with increasing matrix dimensionality. This intuition generalizes to the other matrix classes. Hence, a quantitative comparison of SAS values is only reasonable for matrices of the same dimensionality.

Next, we investigate how SAS decreases between two identical copies of a matrix when gradually perturbing one of them. We analytically study SAS between a matrix M and a perturbed version of itself, $M + \sqrt{\epsilon}W$, using Rayleigh-Schrödinger perturbation theory (Landau and Lifshitz, 2013). Here, $\sqrt{\epsilon}$ is chosen as a perturbation parameter since this ensures a linear scaling of the variance of the perturbation matrix with ϵ. We find that, for a large class of perturbations, SAS follows $1 - \arccos\left(1 - O(\epsilon)\right) / \frac{\pi}{2}$ (see Section A.2.4). This implies that small differences are identified as dissimilarities arbitrarily fast ($\frac{d \arccos(1-x)}{dx} \to \infty$ for $x \to 0$). Thus, SAS is sensitive to small differences in the compared matrices. For an empirical analysis, we study the sensitivity of SAS under perturbations of the form $\tilde{M} = M + W$ where W is a matrix of the UC class with zero mean and variance $\sigma^2_{\text{pert}} = f\sigma^2$, $0 \leq f \leq 1$. As predicted analytically, Figure 6.4B shows a rapid fall-off for small perturbations that continues as a gradual decrease for each of the considered classes.

Next, we numerically study SAS while adding structure to a noise matrix. In particular, we calculate SAS between a matrix M and the convex combination of the same matrix with a UC matrix N:

$$(1-\lambda)N + \lambda M \qquad \text{for } \lambda \in [0,1] \, . \tag{6.16}$$

Figure 6.4C shows that SAS is well-behaved also for small values of λ: it increases smoothly when adding structure for all matrix classes.

Finally, starting from CB matrices, we investigate the change of cross-similarity when shifting the correlated block of one matrix either vertically or diagonally by s indices (Figure 6.4D). In both cases, SAS gradually decreases and saturates once the blocks are non-overlapping ($s = 80$). Importantly, SAS saturates to a value higher than the mean CB – UC cross-similarity for the vertical shift, whereas it saturates to a value lower than that for the diagonal shift. Thus, SAS identifies shared structure when it is shifted vertically, but not when it is shifted diagonally. While potentially counter-intuitive, this can be understood when considering the case of DB matrices (Figure 6.3e): the diagonal shift is a specific example of a permutation in both directions, and once the blocks are non-overlapping, there is no correlation structure between the two matrices that SAS can identify. We conclude that SAS cannot detect similarity between matrices if a sufficient amount of the relevant structure is moved across instances.

We perform an analogous analysis for network adjacency matrices of six different graph models: Erdős-Rényi (ER), directed configuration model (DCM), one cluster (OC), two clusters (TC), Watts-Strogatz

(WS), and Barabási-Albert (BA), as defined in Section 6.3.2. The results are qualitatively similar to those obtained for the four classes of random matrices: SAS decreases with increasing network size N (Figure 6.4E), SAS rapidly decreases for small perturbations (Figure 6.4F), SAS gradually increases when adding structure (Figure 6.4G), and SAS decreases when shifting either vertically or diagonally (Figure 6.4H). A notable exception is that for WS networks, SAS does not decrease when increasing N. In this network model the number of nearest neighbors each node is connected to scales with N. Therefore the correlation in the adjacency matrix also scales with N, rendering the similarity measured by SAS independent of N. For investigating the effect of a perturbation on identical network matrices (Figure 6.4F), we define the gradual change such that an increasing fraction g of matrix elements is altered. Specifically, for each of the gN^2 randomly selected matrix elements, existing connections are removed and missing connections are established with a multiplicity of one. For the binary adjacency matrices this corresponds to bit-flipping the corresponding entries. In the case of adding structure (Figure 6.4G), we choose the ER adjacency matrices as the noise component N. Note that while the sum over all entries stay the same on average under the convex combination, the entries are not confined to natural numbers anymore.

6.2.4.2 *Degenerate singular values*

We evaluate the discriminability of matrix classes with SAS in the presence of degenerate singular values (Figure 6.5). Specifically, we quantify how well CB matrices with non-overlapping blocks and OC matrices with non-overlapping clusters can be discriminated when making the singular value spectrum degenerate.

We introduce degeneracy gradually as described in Section 6.3.4. In the case of CB matrices, for which the singular value spectrum exhibits a smooth decay (not shown), SAS is able to distinguish matrices with blocks at different positions even for high degeneracy ($\theta > 1$ up to $D \approx 0.85$) as shown in Figure 6.5A–C.

In contrast, the spectrum of singular values of OC matrices is dominated by one singular value (not shown), which is of highest relevance for SAS. When introducing degeneracy from the center (Figure 6.5D) or starting from the smallest singular value (Figure 6.5E), θ remains roughly constant until $d = 299$, $d = 300$, respectively. At precisely these values, the largest singular value becomes degenerate and θ drops below 1. For the left-degenerate case (Figure 6.5F), some discriminability is retained for small values of d even though the largest singular value is made degenerate already at $d = 2$. These cases suggest that SAS is not principally limited by the presence of repeated singular values.

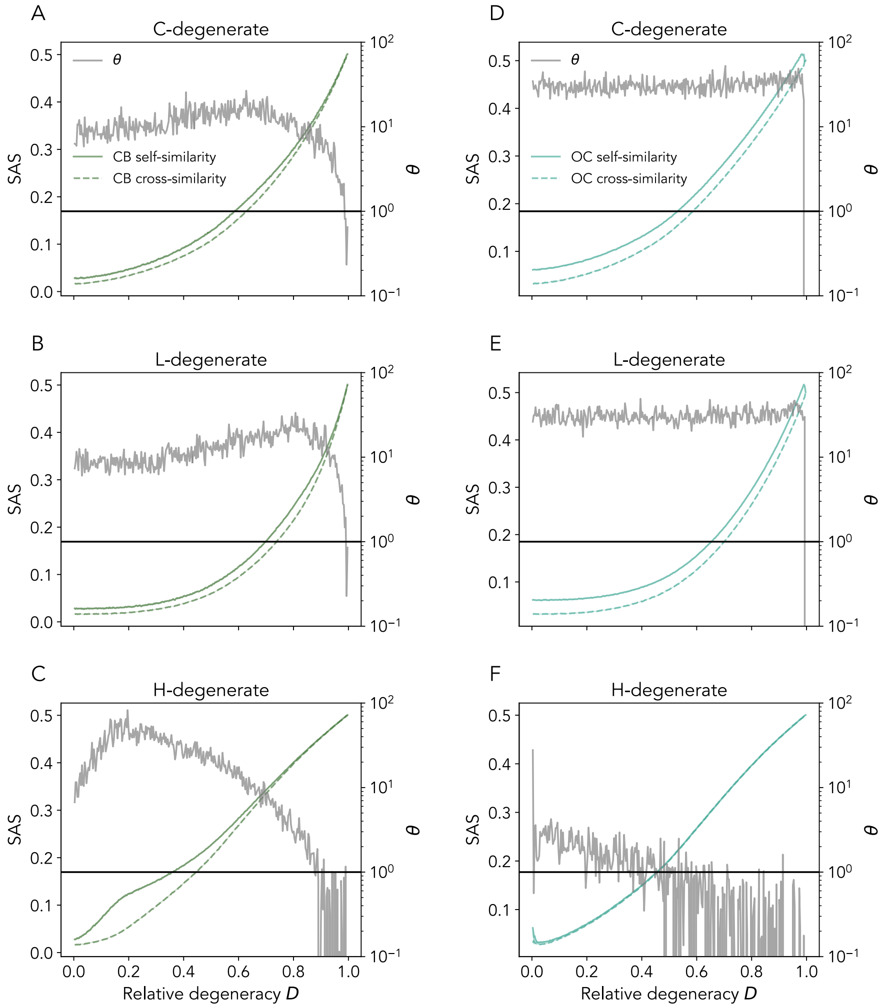

Figure 6.5: **SAS for degenerate matrices.** In all panels, lines indicate the mean, the shading the standard deviation over 10 realizations. Solid colored lines show self-similarity between CB (OC) matrices with identical parameters (same as in Table 6.1 and Table 6.2). Dashed colored lines show cross-similarity between two CB (OC) matrices where the second matrix exhibits a block (cluster) shifted to $b^{\text{lower}} = 210$ and $b^{\text{upper}} = 290$ ($b^{\text{lower}} = 200$ and $b^{\text{upper}} = 250$). Gray lines indicate effect size θ (right axes). In all panels, the fraction D of degenerate singular values is varied. $N = 300$. **A,D** C-degenerate matrices, where the singular values are made degenerate starting from the center singular value. **B,E** L-degenerate matrices, where the singular values are made degenerate starting from the lowest singular value. **C,F** H-degenerate matrices, where the singular values are made degenerate starting from the highest singular value.

6.2.5 *Categorization of random graphs*

We apply SAS to the adjacency matrices that describe the network architecture in various directed and undirected probabilistic graphs as defined in Section 6.3.2. Example adjacency matrices of network instances are shown in Figure 6.6A–F. Since these matrices M_i only contain non-negative entries, so do $M_i M_i^T$ and $M_i^T M_i$. The Perron-Frobenius theorem (Deuflhard and Hohmann, 2008) guarantees that the left and right singular vectors corresponding to the largest singular value have only non-negative or only non-positive entries (cf. Section A.2.5). As such, these singular vectors are confined to a single orthant of the N-dimensional vector space. Even if these vectors are random, they cannot be assumed to be orthogonal. Indeed, for the ER network, for which all other singular vectors are of random orientation, the first left and right singular vectors scatter around the vector $\left(1/\sqrt{N}, ..., 1/\sqrt{N}\right)^T$ across instances. Therefore, the first singular vectors necessarily enclose smaller angles across models compared to the other pairs of singular vectors. Consequently, information regarding the difference between models—which is encoded most strongly in the first singular vectors—is reduced. Thus, it is *a priori* not clear whether SAS reliably distinguishes between network models.

SELF-SIMILARITY OF NETWORK MODELS Firstly, we examine the self-similarity of the network models (Figure 6.6a–f). We find that ER networks exhibit the lowest self-similarity compared to all other network models. This is consistent with the ER network model maximizing the entropy under the constraint that the average number of connections is constant: ER networks have the least structure that is shared across instances. This can be also understood from their definition inasmuch as each connection is realized independently with the same probability. In this sense, ER networks are analogous to the UC random matrices. The other network models instead feature structural properties that are consistent across instances, stemming from shared variations in the connection probability. This is most obvious for the OC and TC network models (analogous to the CB random matrix class), where certain subgroups of nodes have a higher connection probability p among themselves compared to the rest of the network. Further, we expect WS networks to have reliably detectable structure, i.e., high self-similarity, as every node has dominant local connectivity. SAS confirms these expectations, as seen when comparing the respective self-similarity distributions in Figure 6.6a–f.

SELF-SIMILARITY VS. CROSS-SIMILARITY Secondly, we study whether SAS can differentiate between the particular structures present in the adjacency matrices of the network models. Using the effect size defined in Equation 6.15, Figure 6.6a–f confirm that the self-similarity

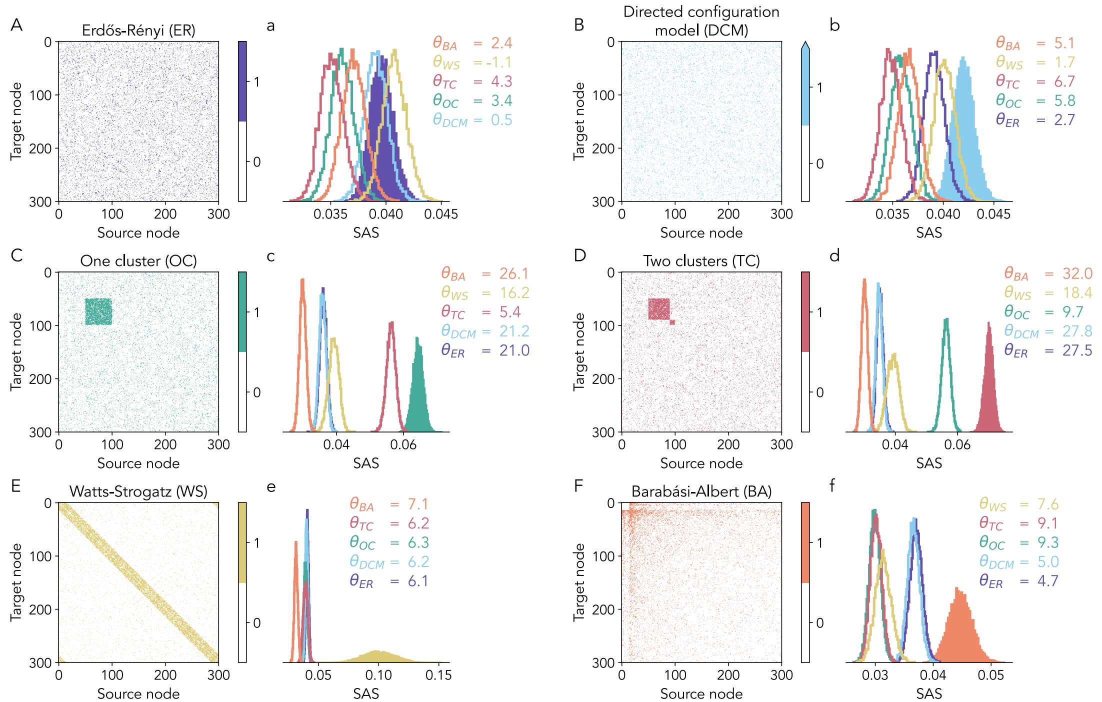

Figure 6.6: **Self- and cross-similarity of different network models. A–F** Single instances of the different network models. Colored matrix elements indicate a connection between nodes. For the DCM model, connections with a multiplicity higher than one are shown with the same color intensity as single connections. **a–f** Histograms of SAS between instances of the network models ($n = 100$, all pairs compared). Filled distributions indicates self-similarities, non-filled ones indicate cross-similarities. Legends show effect sizes θ for the comparison between self- and cross-similarities.

is meaningfully higher than the cross-similarity except for special cases.

First, ER networks do not exhibit $\theta > 1$ for all cases. As for the UC matrices, this is expected as all matrix elements are uncorrelated. As a matter of fact, the cross-similarity with WS networks yields even higher SAS values than the self-similarity of ER. Why is this the case? The first left and right singular vectors of both ER and WS networks scatter around $\left(1/\sqrt{N}, ..., 1/\sqrt{N}\right)^T$. The deviation between singular vectors of ER networks, however, is larger than that between those of WS networks across instances. This leads to a better alignment, i.e., a higher angular similarity, of the singular vectors, resulting in a higher SAS between ER and WS as compared to ER and ER networks.

Second, we note that SAS distinguishes between the OC and TC networks despite overlapping clusters. Figure 6.6c–d show that the respective self-similarities are closer to the cross-similarity of OC and TC than to the other cross-similarities. Thus, SAS identifies the clustered networks to be more similar among each other than compared to the remaining networks.

We conclude that SAS is sensitive to the structure present in matrices, enabling it to distinguish between model classes. The same conclusion also holds true for non-square matrices of network connectivity where a full graph is instantiated, but only subsamples are analyzed with SAS (see Section A.2.6).

6.2.6 *Separation of brain states*

We investigate brain activity originating from different experimental trials as a use case for SAS with non-square matrices of experimental data. The publicly available data set from Chen et al. (2022) is based on extracellular recordings from the visual cortex of a macaque monkey. In the experiment, bright bars move across a screen in one of four directions (rightward (R), leftward (L), upward (U) or downward (D)), evoking a strong neural response (Figure 6.7A). The data consists of the multi-unit activity envelope (see Section 6.3) yielding one 64×400 matrix per trial; sample matrices are shown in Figure 6.7B.

Neurons in the primary visual cortex (V1) respond according to their feature selectivity, primarily stimulus location (Tootell et al., 1982) and orientation (Hubel and Wiesel, 1959), but also movement direction De Valois et al. (1982). Beyond these well-known response properties of single neurons, the population activity—represented as a two-dimensional spatio-temporal matrix—may reveal additional information about brain dynamics. By applying SAS, we investigate shared variability across both time and neurons.

In the data set at hand, SAS reveals that neural activity of all trial types exhibits higher self- than cross-similarity with effect sizes $\theta > 1$ (Figure 6.7C). Trials with stimulus movement along the same axis (L-R

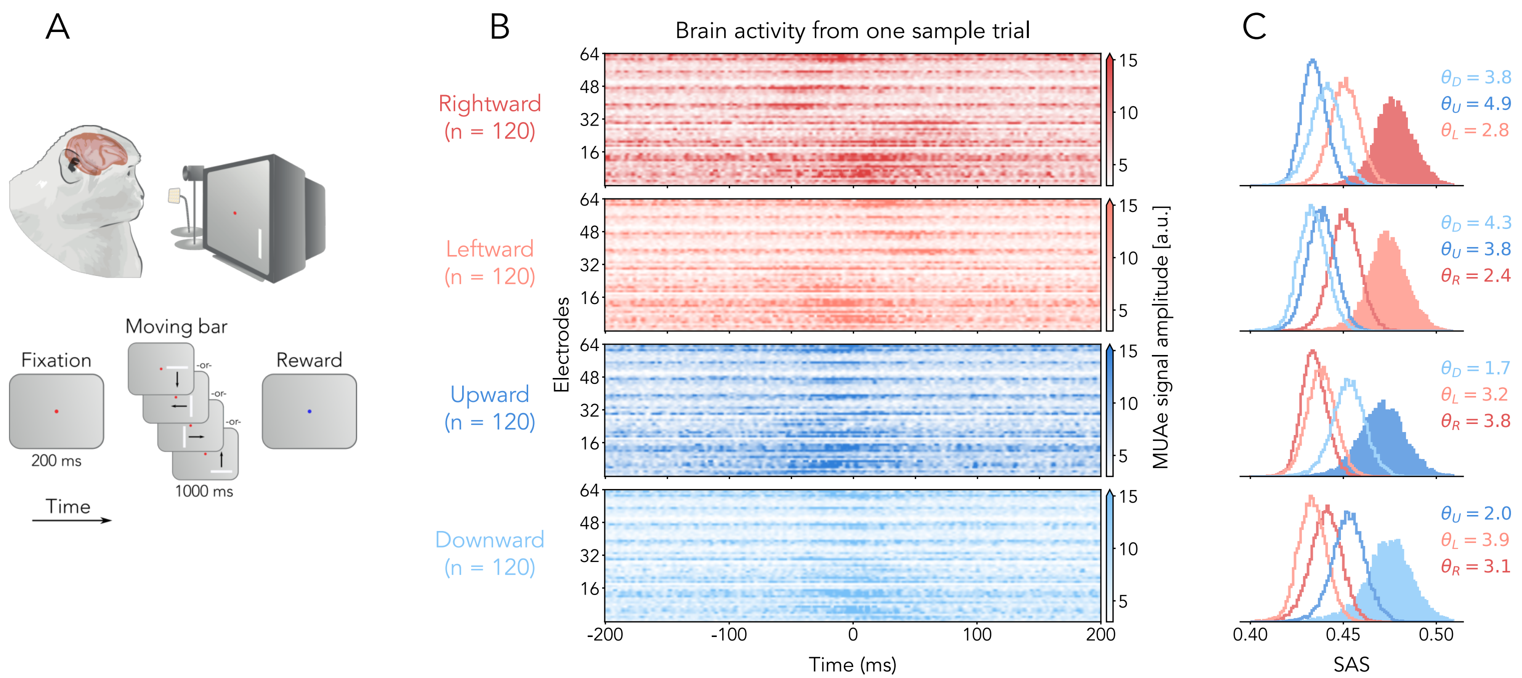

Figure 6.7: **Comparing non-square matrices of brain activity with SAS. A** Schematic diagram of the neuroscience experiment. **B** Single-trial brain activity at identical recording electrodes (64 electrodes, vertical) for each type as example (trial types distinguished by color). Larger values of matrix entries indicate higher MUAe activity (shading in color bar, arbitrary units). **C** Histograms of SAS between all ($n = 120$) individual trials. Filled distributions (color code as in B) indicate self-similarities, non-filled ones indicate cross-similarities. Legends show effect sizes θ for the comparison between self- and cross-similarities.

or U-D) are more similar to each other than to ones with orthogonal stimulus movement. This is a desirable outcome of SAS: in L-R (resp. U-D) trials, neurons that share an orientation tuning aligned with the stimulus are expected to have a higher probability of a strong response. The consideration implies that the shared orientation of the bar stimulus in L and R (resp. U and D) trials leads to more similar responses in these trial pairs.

In this use case, SAS outperforms common measures of matrix similarity. Both cosine similarity and the Frobenius norm can distinguish trial types—albeit with lower $|\theta|$ than SAS—but fail to identify the shared variability along the same axis (L-R or U-D) (Figure A.2.2). Therefore, the use case highlights the ability of SAS to identify the two-dimensional structure of matrices in experimental data.

6.3 METHODS

6.3.1 *Random matrices*

To compare SAS to standard measures of matrix similarity, we define the following classes of random matrices with shape $N \times N$ where each entry is drawn from a continuous probability distribution. Numerical values for the corresponding model parameters are summarized in Table 6.1.

UNCORRELATED NORMAL MATRIX (UC) For random matrices of this class, each entry is drawn independently from a normal distribution with the same mean μ and variance σ^2. Collections of these matrices are also referred to as real Ginibre Ensembles.

CROSS-CORRELATED NORMAL MATRIX (CC) We first independently sample N random vectors from an N-dimensional normal distribution with mean μ and covariance matrix C where

$$C_{ij} = \frac{a}{N} \cdot \exp\left(-b\frac{|i-j|}{N}\right) .$$

Thus, the entries of the correlation matrix decay exponentially with distance from the diagonal. The sampled vectors are the columns of an $N \times N$ matrix K^1. We repeat the process to obtain an independent matrix K^2 to finally define $K = (K^1 + K^{2^T})/2$. Since each entry of the K^i is normally distributed, so is each entry of their sum K, and the covariance between entries K_{kl} and K_{mn} is $(C_{kl} + C_{mn})/4$. Normalization by N in the argument of the exponential ensures that the strength of the correlation scales with the size of the matrix.

CROSS-CORRELATED BLOCK MATRIX (CB) Again, uncorrelated normal (UC) matrices B are sampled. Then, the entries B_{kl} where $b^{\text{lower}} \leq k \leq b^{\text{upper}}$ and $b^{\text{lower}} \leq l \leq b^{\text{upper}}$ are replaced by a correlated normal structure as defined above, forming a block on the diagonal.

SHUFFLED, CROSS-CORRELATED BLOCK MATRIX (SB) Matrices are sampled according to CB. Then, the rows are permuted randomly while the columns remain untouched.

DOUBLY SHUFFLED, CROSS-CORRELATED BLOCK MATRIX (DB) Matrices are sampled according to CB. Then, the rows and columns are permuted randomly.

Parameter	Model(s)	Meaning	Value
N	all	dimensionality	300
μ	all	mean of distribution	0
σ^2	UC	variance of distribution	$1/N$
a	CC	peak covariance	10
b	CC	inverse characteristic length	100
b^{lower}	CB	block index	10
b^{upper}	CB	block index	90

Table 6.1: **Parameters of random matrices.** Uncorrelated normal matrix (UC), cross-correlated normal matrix (CC), cross-correlated block matrix (CB), and shuffled, cross-correlated block matrix (SB).

6.3.2 *Random graphs*

We compare the adjacency matrices of six well-known network models with SAS. For all graphs, we derive the parameters such that the mean total number of connections N_c in the graph is conserved. Table 6.2 summarizes the numerical values chosen for the corresponding model parameters.

ERDŐS-RÉNYI (ER) In this network model Erdős and Rényi (1959), every connection has the same probability of being realized: $p = \frac{N_c}{N_e}$, where N_e is the total number of possible connections in the graph. Note that this network model maximizes the entropy under the constraint that the mean number of connections is fixed (Park and Newman, 2004).

DIRECTED CONFIGURATION MODEL (DCM) In a directed configuration model (Cooper and Frieze, 2004), a two-step probabilistic process is applied. First, random in-degrees and outdegrees are drawn for each node such that the total number of connections across nodes is preserved (we fix these numbers for all graph instances). Second, connections are established by randomly matching each outgoing connection with an incoming connection. Thus, two nodes can have more than one connection, and the resulting adjacency matrix is not strictly binary.

ONE-CLUSTER ERDŐS-RÉNYI (OC) Based on an ER graph, we introduce a single cluster by increasing p between a certain subset of nodes of the network, while uniformly decreasing p for all other connections such that N_c is conserved. The relative increase of p is denoted by r, and the location of the cluster on the diagonal is defined by the bounding indices b^{lower} and b^{upper}.

TWO-CLUSTER ERDŐS-RÉNYI (TC) For the two-cluster ER network, we create two non-overlapping clusters on the diagonal using the same method as in the OC model. The nodes that form the clusters are chosen such that there is maximal overlap with the single cluster of the OC model: the first cluster starts at the same index b^{lower} and extends up to index b^{mid}, and the second cluster starts at index $b^{\text{mid}} + 1$ and extends up to index b^{upper}.

WATTS-STROGATZ (WS) We create a small-world network following Watts and Strogatz (1998). Here, N_n nodes initially form a ring, where each node is connected to $k = \frac{N_c}{N_e}(N_n - 1)$ of its nearest neighbors. Afterwards, all connections are uniformly redistributed with probability p_{WS}. Note that this model is undirected.

BARABÁSI-ALBERT (BA) As an example of a scale-free network, we create Barabási-Albert networks as introduced in Albert and Barabási (2002). Here, from an initial star graph with $m = \frac{N_c}{N_e}(N_n - 1)/2$ nodes, new nodes are added subsequently until the desired number of nodes N_n is reached. Each added node is connected to m existing nodes, where the probability of each existing node being selected for a new connection is proportional to the number of connections it already has. Note that this model is undirected.

Param.	Model(s)	Meaning	Value
N_n	all	number of nodes	300
N_e	all	number of possible connections	90000
N_c	all	mean number of connections	9000
r	OC	relative increase of p in cluster	10
b^{lower}	OC	cluster index	50
b^{upper}	OC	cluster index	100
b^{mid}	TC	index between clusters	90
p_{WS}	WS	reconnection probability	0.3

Table 6.2: **Parameters of random graphs.** Erdős-Rényi (ER), directed configuration model (DCM), one cluster (OC), two clusters (TC), Watts-Strogatz (WS), and Barabási-Albert (BA).

6.3.3 *Brain data*

We apply SAS to compare non-square matrices of brain activity in response to visual stimuli. We use an openly available data set, which has an extensive description of the task and recording apparatus (Chen et al., 2022). In the experiments, the activity of neurons in the primary visual cortex (V1) of one macaque monkey (*Macaca mulatta*) was recorded using several extracellular electrode arrays (Utah arrays, 8×8 electrodes). The quality of the signals was assessed based on the signal-to-noise ratio and channel impedance. For details of the data recording and processing we refer to Chen et al. (2022). Here, we focus on a single array (ID $= 11$) during a receptive field mapping task. In this task, for each trial the macaque had to fixate its gaze to the center of the screen for 200 ms. Subsequently, one bright bar moved across the screen for 1000 ms in one of four directions: rightward (R), leftward (L), upward (U), or downward (D). The different task modalities are in the following referred to as trial types. For each trial type, there are $N = 120$ repetitions.

The activity time series recorded from the electrodes was processed to obtain the multi-unit activity envelope (MUAe) with a sampling rate of 1 kHz, a commonly used signal as a proxy for neuronal firing rates (Supèr and Roelfsema, 2005), see (Morales-Gregorio et al., 2023) for specific details of the processing. We align the trials to the peak response, defined as the maximum response from the average MUAe across electrodes, and cut data in a window $\pm$ 200 ms around the alignment trigger. This yields one 64×400 matrix per trial: 64 electrodes during 400 ms at 1 kHz; see examples in Figure 6.7B. We then group the matrices by trial type for comparison by SAS.

6.3.4 *Degenerate matrices*

To assess SAS in the presence of degenerate singular values we construct matrices with various levels of degeneracy from random matrices and random graphs. Given a matrix $M = U\Sigma V^T$, we introduce a degeneracy parameter d forcing

$$\sigma^i = ... = \sigma^{i+d} = \frac{1}{d}\sum_{j=i}^{i+d} \sigma^j \tag{6.17}$$

and define the degenerate matrix as

$$M^d = U\Sigma^d V^T \ , \tag{6.18}$$

where in Σ^d the corresponding singular values are replaced by their empirical mean. We distinguish three special cases of starting index i such that either the highest (first), lowest (last), or central singular value are first made degenerate, referred to as *H-*, *L-*, and *C-degenerate*, respectively. In the C-degenerate case, i is adjusted such that the central singular value lies in the center of the interval $[i, i + d]$. Finally, we denote the relative degeneracy of an $N \times N$ matrix as $D = d/N$.

6.4 DISCUSSION

In this chapter, we develop *singular angle similarity* (SAS), a novel measure for assessing the similarity of real matrices of arbitrary shape. SAS is based on the singular value decomposition of the matrices under consideration. First, the singular vectors are ordered by their singular value. Then, the corresponding vectors of the two matrices are matched pairwise. Finally, information about the alignment of the vectors is calculated and distilled into a single value between 0 and 1, indicating minimal and maximal similarity, respectively.

We show that SAS is able to reliably detect structural similarities in matrices where traditional measures fail or are not applicable by construction. Further, SAS is robust to noisy or degenerate data, making it readily applicable also to experimental data such as recordings of brain activity. SAS directly generalizes the eigenangle score for square, symmetric, positive definite matrices (Gutzen et al., 2023) and the angular similarity for vectors (Section A.2.7). Here, our choice of vector similarity measure (Equation 6.6) and weight function (Equation 6.8) can be straight-forwardly exchanged. Because SAS is sensitive to matrix size (Section 6.2.4), its results can only be compared in relative terms. Thus, a reference model is needed to which different matrices can be compared. Here, natural choices are an Erdős-Rényi graph for random networks, and an uncorrelated normal matrix for random matrices, as both contain no structural features.

In the context of this thesis, SAS's applicability to adjacency matrices is of particular interest. So far, we have only used matrices to display the average connection probabilities or mean number of synapses between populations of neurons. However, due to the unprecedented accuracy of electron microscopy imaging for neural anatomy, we are now in a position to evaluate the connectivity at neural resolution for large volumes of neural tissue. By construction, SAS is a more suitable measure than previously developed ones since such connectomes are generally non-square and non-symmetric, and contain possibly intricate connection motifs that cannot be captured by one-dimensional measures of similarity. In the next chapter, we set out to study whether the detailed information of preferred synaptic locations can be effectively used to predict local cortical connectivity. In the subsequent analysis, we use SAS to compare the predicted single-neuron connectomes to a traditional, distance based rule akin to the one used in our modeling efforts described in Part iii.

7 PREDICTING SPATIAL CONNECTIVITY FROM SYNAPTIC CONSTELLATIONS

Author contributions:
This project was supervised by Stefan Mihalas (SM). The author developed the constellation-based connectivity rule in close collaboration with Stefan Berteau (SB). The author devised and implemented the distance-based connectivity rule. Further, the author carried out the comparison between the two predictive rules using singular angle similarity. The work was continually refined in ongoing discussions with SB and SM.

7.1 INTRODUCTION

In the course of this thesis, we investigate spatial connectivity rules in the visual system that describe local cortical connectivity on three different spatial scales: long-range patchy connectivity (Section 4.2.2.2), short-range push-pull connectivity (Section 4.2.2.3), and medium-range isotropic connectivity (Section 4.2.2.1 and Section 5.2.1). In this chapter, we reconsider the latter connectivity principle. Previous studies have found that neurons vary in their preference of where they establish synapses on a given target process. For example, Beaulieu and Colonnier (1985) show that in area 17 of the cat visual cortex, inhibitory synapses are found more frequently on somata and proximal dendrites, while excitatory synapses are found more frequently on distal dendrites and spines. In the medial superior olive of Mongolian gerbils, which is part of the auditory brain stem, axons are found to follow dendrites and establish multiple synaptic sites along the way (Callan et al., 2021). Such findings bring into question the validity of assuming a connectivity rule that smoothly decays with inter-somatic distance. The recent data set by MICrONS Consortium et al. (2021), which derives the connectivity between neurons in the primary visual cortex of a mouse using electron microscopy (EM), allows to reexamine this assumption. In Section 5.2.1, we already make use of this very data set to update the isotropic connectivity rule that was formerly derived from reconstructions based on light microscopy. However, this derivation still assumes a probabilistic description of connectivity that only depends on the horizontal distance between somata.

In this chapter, we go beyond the commonly used isotropic description of local cortical connectivity and introduce a novel probabilistic connectivity rule that takes the detailed spatial distributions of synapses explicitly into account. We leverage the synapse-level resolution of the EM data set to derive *constellations*—the spatial distributions of synapses on both axons and dendrites. By averaging over single-neuron constellations, we obtain population-specific, characteristic constellations. Calculating spatial overlaps between presynaptic axonal and postsynaptic dendritic constellations, we predict the synaptic connectivity between single neurons based on their populations and relative three-dimensional positions. For validation, we compare the predictive power of the devised connectivity rule to the distance-base rule derived in Section 5.2.1: we assess the similarity to the experimental ground truth given by the EM data set using singular angle similarity (SAS), developed in Chapter 6. The comparison demonstrates that the constellation-based rule reliably outperforms the distance-dependent rule in the similarity of its predictions to the experimental ground truth connectivity. Thus, constellations can play a valuable role in informing the connectivity of future studies that aim to investigate the link between neural network dynamics and their underlying structure.

7.2 RESULTS

7.2.1 *Ground-truth connectivity*

From the data set by MICrONS Consortium et al. (2021), we first derive the ground truth connectivity for the recurrent circuitry between 266 fully reconstructed neurons of various cell types and cortical layers (see Section 7.3.1). This yields a matrix of size 266 $\times$ 266 where each integer-valued entry corresponds to the number of synapses between the respective pair of neurons. Figure 7.1 displays this matrix, where the neurons are sorted by population.

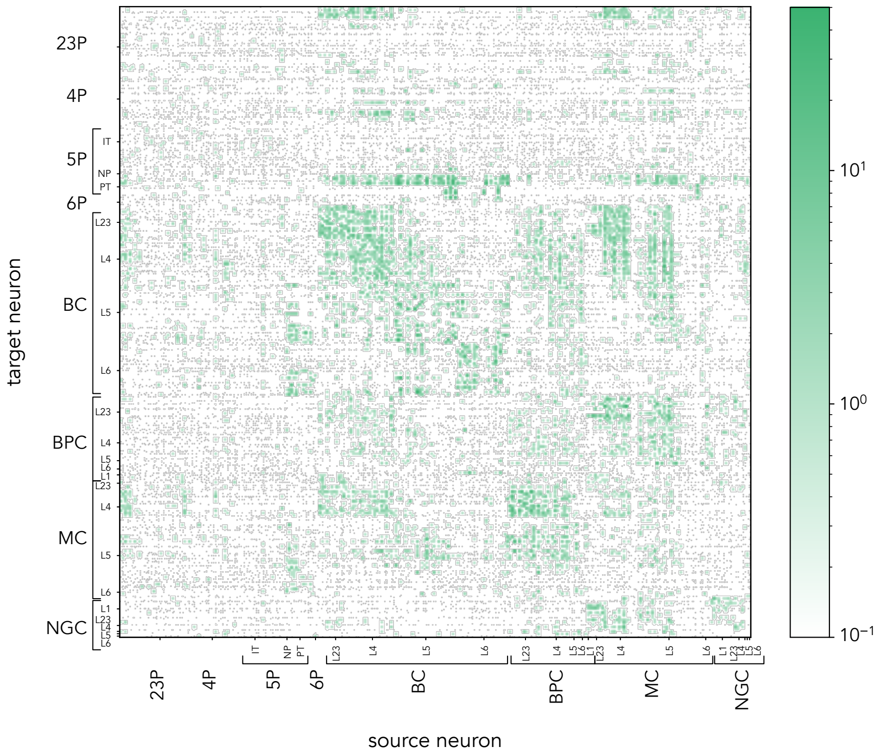

Figure 7.1: **Ground truth connectome.** Number of synapses between pre- and postsynaptic neurons. Logarithmic color scale. Nonexistent connections displayed as white. Large labels combine with small labels to create populations (see Table 7.1).

Here, we can observe a certain block-like structure, indicating that more connections exist within populations than across. This network graph acts as a baseline to which we can compare the connectomes calculated from the two employed statistical rules.

7.2.2 *Predictive connectomes*

Next, we use the positions and cell types of the subgraph of 266 neurons to calculate the predicted connectivity according to both the constellation-based and distance-dependent rule (for detailed derivations, see Section 7.3.2 and Section 7.3.3, respectively). These rules yield the expected number of synapses shown in Figure 7.2.

Both connectivity rules yield a real-valued estimation of the expected number of synapses between each pair of neurons. Therefore, a comparison to the ground truth connectivity (Figure 7.1) is not straight-forward: first, we have to sample concrete realizations, i.e., integer-valued numbers synapses, from these predictions. Here, we assume that the number of established synapses k is Poisson-distributed based on a maximum entropy argument (Section 7.3.4). Thus, for each entry of a predictive connectome specifying the mean number of expected synapses S, we draw the expected number of established synapses according to

$$p(k; S) = \frac{S^k}{k!} e^{-S} \ . \tag{7.1}$$

Examples for such instantiated, integer-valued connectomes are shown in Figure 7.3.

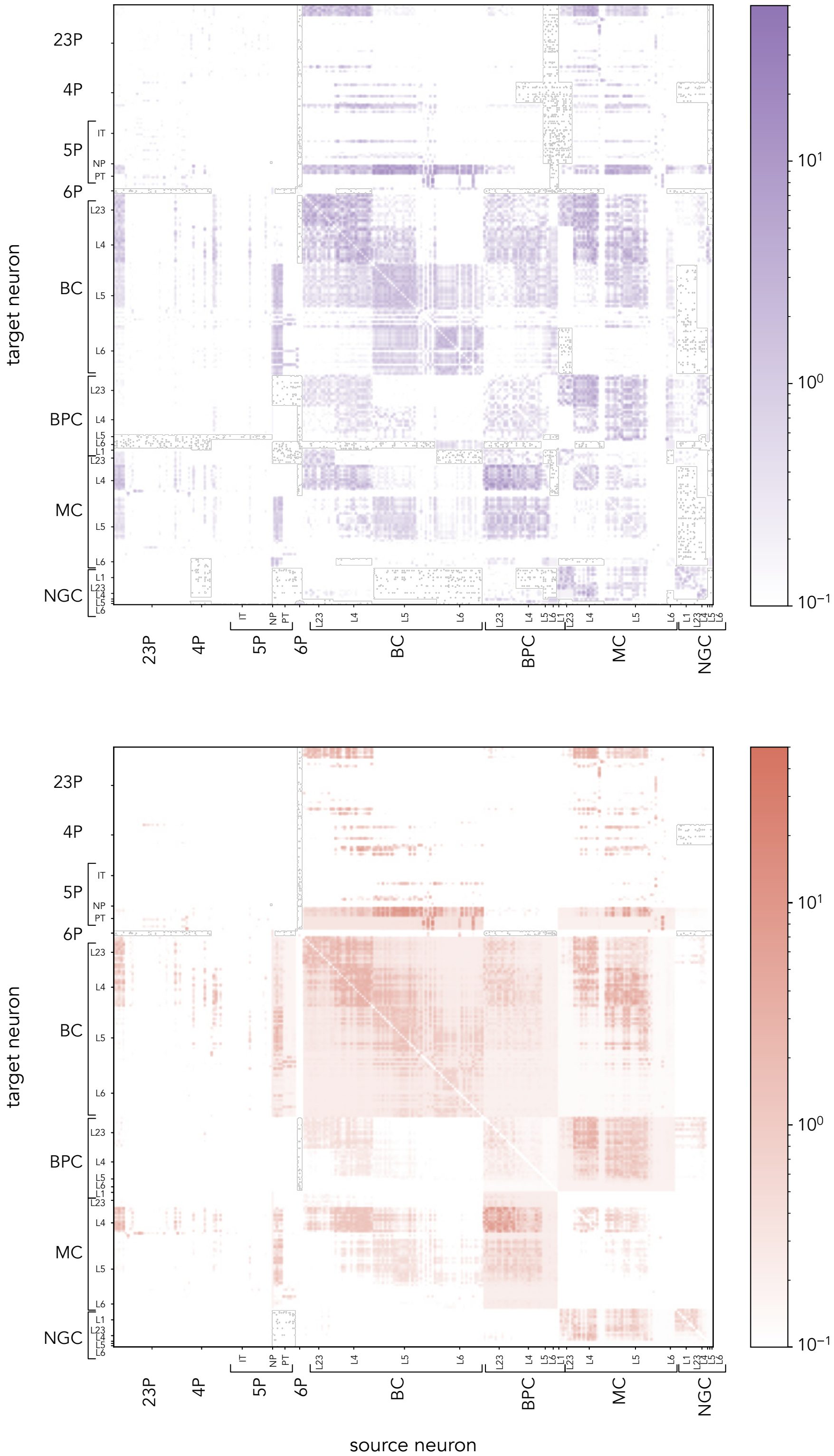

Figure 7.2: **Predictive connectomes.** Expected number of synapses between pre- and postsynaptic neurons. Logarithmic color scale. Large labels combine with small labels to create populations (see Table 7.1). **Top** Distance-dependent connectivity rule. **Bottom** Constellation-based connectivity rule.

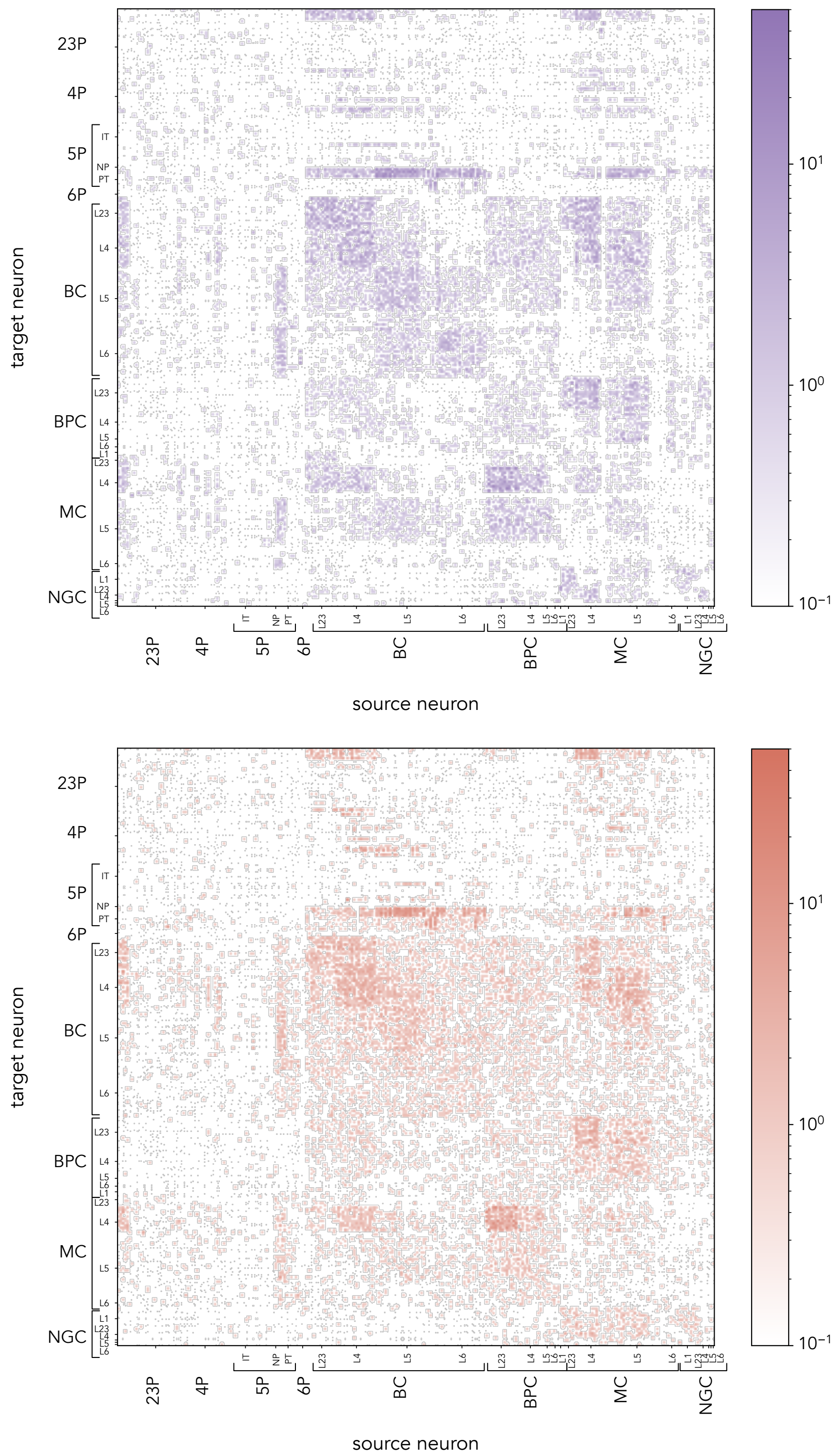

Figure 7.3: **Instantiated connectomes.** Instantiated number of synapses between pre- and postsynaptic neurons drawn according to a Poisson distribution with mean S given by the predictive rules. Logarithmic color scale. Large labels combine with small labels to create populations (see Table 7.1). **Top** Distance-dependent connectivity rule. **Bottom** Constellation-based connectivity rule.

To obtain a quantitative measure of which connectome more closely resembles the connectivity structure found in mouse visual cortex, we utilize the novel matrix similarity measure developed in Chapter 6, singular angle similarity (SAS). First, we draw 10 000 instantiations of possible connectomes from the probabilistic descriptions shown in Figure 7.2 according to the Poisson assumption (Equation 7.1). Then, we compare all instantiations to the ground truth given in Figure 7.1 using SAS as a measure of similarity. Figure 7.4 shows the resulting distributions of similarity.

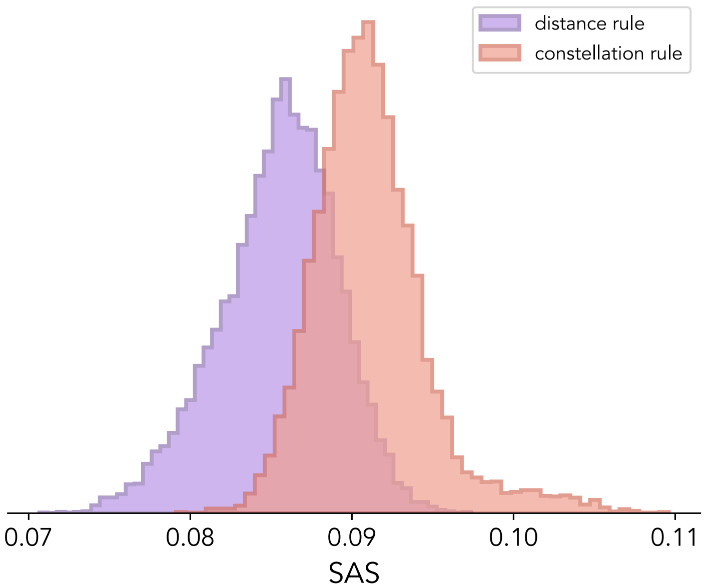

Figure 7.4: **Comparing the predicted connectomes to the ground truth using SAS.** Distribution of SAS between the ground truth and 10 000 instantiations of distance-based connectomes (purple) and constellation-based connectomes (peach), respectively.

We observe that the constellation-based connectivity rule produces predictions that achieve greater similarity to the ground truth connectome than predictions of the distance-dependent rule. The absolute values of SAS in Figure 7.4 may appear small with respect to the possible range SAS $\in [0, 1]$. However, recall that the absolute values of SAS are of no relevance as they are dependent on the characteristics of the matrix such as dimensionality and sparsity (Section 6.2.4). Therefore, only the comparison between the similarities of two models to a joint null model—here, the ground truth—is a meaningful measure of greater or lesser similarity. In Section 6.2.4, the distributions of SAS are separated by an effect size (Equation 6.15) of

$$\theta = \frac{\mu_{\text{constellation}} - \mu_{\text{distance}}}{\sqrt{\frac{\sigma^2_{\text{constellation}} + \sigma^2_{\text{distance}}}{2}}} = 1.6\,, \tag{7.2}$$

where $\theta > 1$ indicates robust separability. Thereby, we conclude that the constellation-based connectivity rule captures more of the inherent structure of connectivity observed in visual cortex than the distance-dependent one.

7.3 METHODS

The analyses in this chapter are based on the local cortical connectivity reported by MICrONS Consortium et al. (2021). In their study, the authors use electron microscopy (EM) to reconstruct the morphologies of around $\sim 75,000$ neurons in the primary visual areas of mouse cortex. The resolution of the obtained imaging data is high enough that single synapses can be resolved. Thereby, the resulting connectome is not just a probabilistic description, but represents actual ground truth connectivity between the observed neurons. Since the morphological reconstruction of axons is significantly more difficult than of dendrites, the authors found that automatized methods are not suitable for axon reconstruction. As a consequence, the axons need to be proofread by hand, which is a labor intensive process. Thus, only a limited subset of the $\sim 75,000$ neurons have fully proofread axons. We here consider the same subset of 266 fully manually proofread neurons as in Chapter 5, but only focus on the recurrent connectivity between neurons in this subset. Further, we disregard any autapses in the data set as their detection has been identified as unreliable by the authors.

7.3.1 *Preprocessing*

For the subgraph of 266 neurons, the data set provides all outgoing and incoming synapses, as well as all neuron positions. Compared to Chapter 5, we here choose a more detailed resolution on the cell typing side, differentiating between ten cell types in total (Table 7.1). For inhibitory subtypes, we further split the neurons into different populations depending on the cortical layer ℓ in which their soma resides, where $\ell \in \{\mathrm{L1}, \mathrm{L23}, \mathrm{L4}, \mathrm{L5}, \mathrm{L6}\}$.

Cell type	Description
23P	pyramidal cell, predominantly found in layer 2/3
4P	pyramidal cell, predominantly found in layer 4
5P-IT	pyramidal cell, predominantly found in layer 5, inter-telencephalic-projecting
5P-NP	pyramidal cell, predominantly found in layer 5, near-projecting
5P-PT	pyramidal cell, predominantly found in layer 5, extra-telencephalic-projecting (pyramidal tract)
6P	pyramidal cell, predominantly found in layer 6
BC	basket cell
BPC	bipolar cell
MC	Martinotti cell
NGC	neurogliaform cell

Table 7.1: **Cell type classification.** Fine-grained cell typing used in this work. Pyramidal (P) cells are excitatory; BPC, BC, MC, and NGC are inhibitory. Inhibitory cells are further subclassified depending on the cortical layer in which their soma resides.

7.3.2 *Constellation-based connectivity rule*

First, consider all neurons of a particular population A, i.e., of a cell type for excitatory neurons, or a cell type and layer combination for inhibitory neurons. For these neurons, we collect the spatial positions of their axonal and dendritic synapses relative to their soma body. We denote by X_A^{axon} the set of all three-dimensional vectors x_A^{axon} encoding the relative displacement of a single synapse, and similarly for dendritic synapses. Thereby, we obtain two collections of synaptic locations for each population, which we call *constellations*. Figure 7.5 shows examples of single-neuron constellations.

Next, we calculate the *constellation overlap* O_{ij} for all pairs of neurons in our data set. Assume a given a presynaptic neuron of population B with a soma location s_j and a postsynaptic neuron of population A with a soma at location s_i. For each axonal constellation element $x_B^{\text{axon}} \in X_B^{\text{axon}}$, we center a three-dimensional Gaussian function at its absolute position $s_j + x_B^{\text{axon}}$. Then, we iterate over all dendritic constellation elements $x_A^{\text{dend}} \in X_A^{\text{dend}}$ and evaluate the Gaussian function at their absolute positions $s_i + x_A^{\text{dend}}$.

Summing over the entire constellation, this procedure yields a measure of overlap O_{ij}, where O_{ij} is larger the more the two constellations spatially overlap:

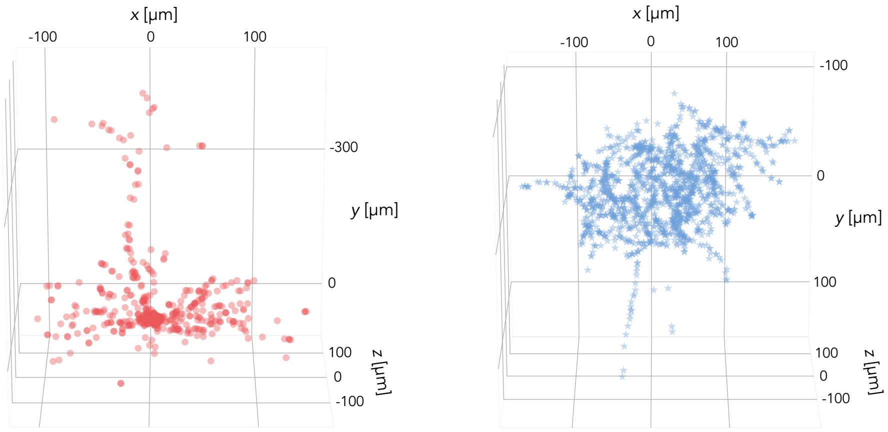

Figure 7.5: **Constellations of two example neurons.** Synaptic locations relative to the soma body. Vertical axis (y axis) corresponds to cortical depth, where negative values indicate locations closer to the cortical surface. Horizontal axes (x, z axes) correspond to directions parallel to the cortical surface. **Left** Dendritic constellation of an example pyramidal cell in layer 5 (cell type 5P-PT). **Right** Axonal constellation of an example basket cell in layer 2/3 (cell type BC, L23).

$$O_{ij} = \sum_{x_A^{\text{dend}} \in X_A^{\text{dend}}} \sum_{x_B^{\text{axon}} \in X_B^{\text{axon}}} \exp\left(- \frac{\left[\left(s_j + x_B^{\text{axon}}\right) - \left(s_i + x_A^{\text{dend}}\right) \right]^2}{2\sigma^2} \right) \tag{7.3}$$

Here, σ is a parameter of the model that controls how close an axonal and dendritic constellation element have to be to evoke the same overlap, which we set to $\sigma = 20\,\mu\text{m}$.

Finally, we normalize the overlap by introducing fitting parameters k_{AB} that fulfill

$$\sum_{j \in B} \sum_{i \in A} k_{AB} O_{ij} = S_{AB} \;, \tag{7.4}$$

where S_{AB} is the total number of synapses between all neurons of populations B and A. Note that k_{AB} only depend on the populations, not on the individual neurons. Figure 7.6 shows a schematic of the employed constellation scheme.

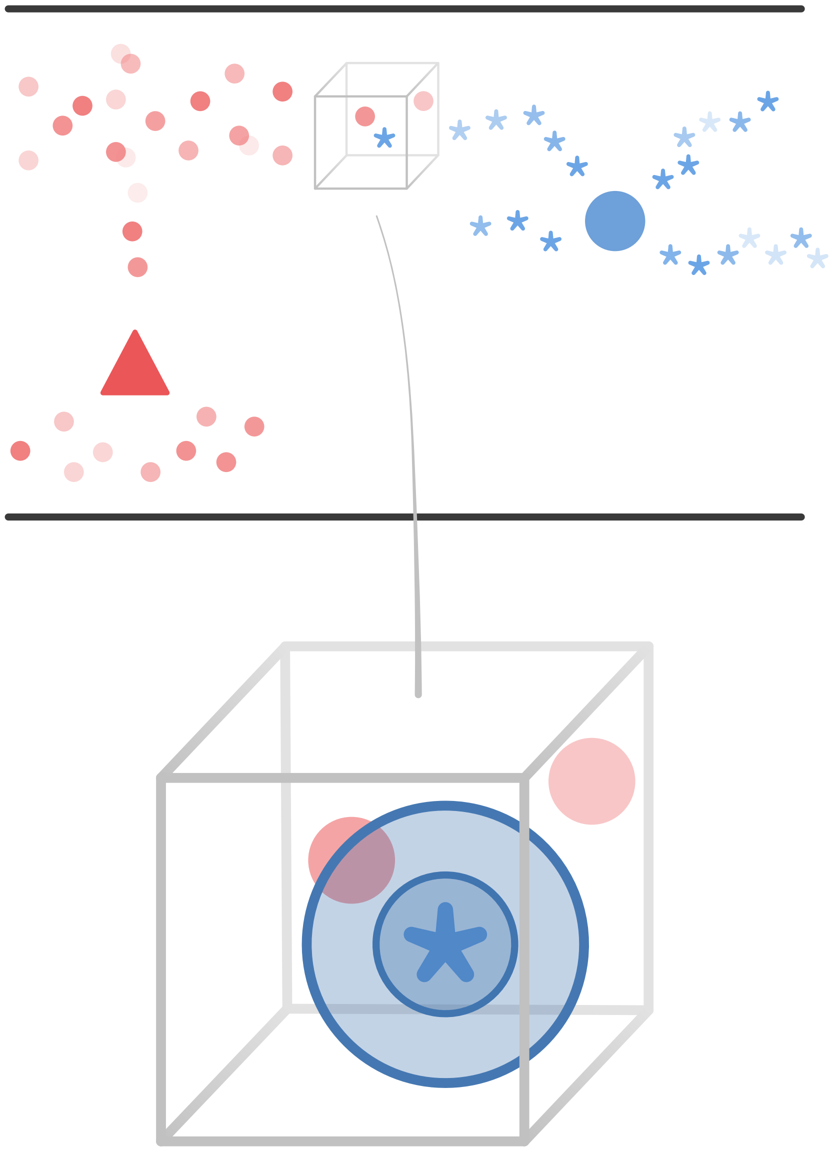

Figure 7.6: **Constellation connectivity scheme. Top** Large blue circle represents an inhibitory neuron of population B at its spatial position s_j; blue stars are the characteristic axonal constellation of population B, X_B^{axon}, with their origin shifted to coincide with s_j. Large red triangle represents an excitatory neuron of population A at its spatial position s_i; red circles are the characteristic dendritic constellation of population A, X_A^{dend}, with their origin shifted to coincide with s_i. **Bottom** Zoomed version of the gray cube of cortical space highlighted in top part of the figure. Dark blue spheres represent 1σ and 2σ distances of a Gaussian function (see Equation 7.3.2) centered on the axonal constellation element (blue star).

7.3.3 *Distance-dependent connectivity rule*

We employ the same approach as described in Section 5.2.1 to calculate the distance-dependence of the mean number of synapses $S_{AB}(d)$ between two neurons with a distance d between their somata:

$$S_{AB}(d) = S^0_{AB} \exp\left(-\frac{d}{\lambda_{AB}}\right) \tag{7.5}$$

In this formulation, S^0_{AB} marks the peak number of synapses, and λ_{AB} denotes the characteristic length of cortical connectivity between populations B and A. To estimate the observed distribution of distances between established connections, we have to account for the density of distances d between pre- and postsynaptic neurons, $\omega(d)$, which has been calculated analytically for uniformly sampled points in a rectangle by Sheng (1985). Thus, we fit

$$s_{AB}(d) = \omega(d) \cdot S_{AB}(d) = S^0_{AB} \cdot \omega(d) \cdot \exp\left(-\frac{d}{\lambda_{AB}}\right) \tag{7.6}$$

to the distance-distribution of synapses observed in the EM data set (cf. Equation 5.4, all fits in Figure A.3.1). This procedure yields the characteristic lengths λ_{AB} shown in Figure 7.7. For combinations of pre- and postsynaptic populations where less than five synapses exist, we omit the fit and instead assume a constant connection probability.

After obtaining the characteristic lengths, we normalize the connectivity rule by fitting parameters m_{AB} that fulfill

$$\sum_{j\in B}\sum_{i\in A} m^*_{AB} S_{AB}(d) = \sum_{j\in B}\sum_{i\in A} m_{AB} \exp\left(-\frac{d}{\lambda_{AB}}\right) = S_{AB}\,, \tag{7.7}$$

similar to the constellation-based connectivity rule (Section 7.3.2). We here absorb the normalization factor S^0_{AB} in m_{AB}. The predicted number of synapses between a presynaptic neuron of population B and a postsynaptic neuron of population A with a horizontal intersomatic distance d is then given by

$$m_{AB} \cdot \exp\left(-\frac{d}{\lambda_{AB}}\right)\,. \tag{7.8}$$

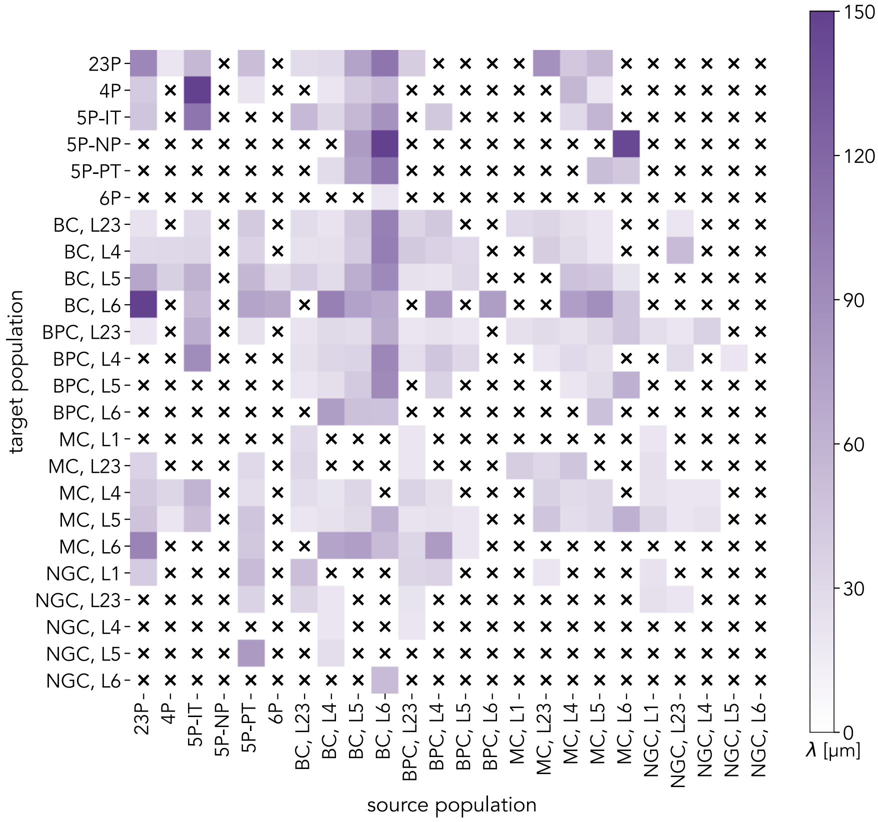

Figure 7.7: **Characteristic lengths of isotropic distance-dependent connectivity.** Characteristic lengths from fitting Equation 7.6 to the observed number of synapses (see Figure A.3.1 for fits). X indicates insufficient data (< 5 synapses).

7.3.4 *Poisson model for the distribution of the number of synapses between neurons*

Given the mean number of synapses S between two neurons, we assume the integer number of established synapses k between the two neurons to be Poisson distributed with mean S. This is motivated by the following model for the generation of synapses: each neuron pair has n sufficiently close sites on the neural processes where synapses can be formed. At each site, a synapse is formed independently with probability p. Since the mean number of synapses is S, it follows that $p = S/n$. The number of synapses thus is distributed according to the binomial distribution with n trials and success probability p, $\mathcal{B}(n; p)$. In contrast to the mean number of synapses, the number of sites on which synapses can be formed is difficult to estimate. Therefore, we consider the set of all binomial distributions that have fewer than n trials and mean S, $\mathfrak{B}(n; S)$. For all n, $\mathcal{B}(n; S/n)$ maximizes the entropy in $\mathfrak{B}(n; S)$ (provided $S/n \leq 1$). Letting n tend to infinity, $\mathcal{B}(n; S/n)$ converges to the Poisson distribution with mean S with respect to the Kullback-Leibler divergence (Harremoës, 2001). We thus identify the maximum entropy distribution in $\bigcup_{n\geq 1} \mathfrak{B}(n; S)$ with the Poisson distribution. The Poisson distribution is therefore the optimal model for the distribution of synapses provided the mean S is known while the number of possible sites at which synapses can be formed is unknown.

Lifting the assumption of homogeneity in synapse creation, we next consider the generalized binomial distribution $\mathcal{B}^{G}(n; p_1, ..., p_n)$. The probability for the natural number k is determined by the probability of k successes in n independent Bernoulli experiments where each experiment is successful with probability p_i. The mean of this distribution is given by $\sum_{i=1}^{n} p_i$. The generative process described above suggests a generalization of the model of the distribution of synapses where a different probability for synapse formation governs each site. We denote by $\mathfrak{B}^{G}(n; S)$ the set of all generalized binomial distributions with at most n trials and mean S. Note that $\mathfrak{B}(n; S) \subset \mathfrak{B}^{G}(n; S)$. The distribution maximizing the entropy in $\mathfrak{B}^{G}(n; S)$ is $\mathcal{B}(n; S/n)$ (provided $S/n \leq 1$). Thus, the Poisson distribution is also the maximum entropy distribution of the set of generalized binomial distributions with mean S, $\bigcup_{n\geq 1} \mathfrak{B}^{G}(n; S)$ and hence is the best model for the distribution of synapses given its mean, also under the more general model of synapse formation stemming from the generalized binomial distribution (Harremoës, 2001).

7.4 DISCUSSION

In this chapter, we present a novel rule for predicting local spatial connectivity in visual cortex based on the characteristic spatial distributions—or constellations—of axonal and dendritic synapses. First, we derive single-neuron constellations from a recent data set that reconstructs the morphology of neurons in mouse V1 using electron microscopy (EM), fully leveraging its synapse-level resolution. Next, we group neurons according to cell type and, for inhibitory neurons, also according to cortical layer, to calculate population-specific constellations. Based on the overlap of the axonal constellation centered on the soma of the presynaptic neuron and the dendritic constellation centered on the soma of the postsynaptic neuron, we predict the expected number of synapses for any two neurons. Notably, the connectivity rule can incorporate potential patterns of connectivity that are specific to spatial locations with respect to a cell's soma, thereby going beyond classical approaches that only depend on the distance between neurons.

We demonstrate that the constellation-based approach outperforms the commonly used distance-dependent connectivity rule: using singular angle similarity (Chapter 6) as a measure, the connectome predicted by the constellation-based rule is more similar to an experimental ground truth. This result holds robustly across sampled connectomes from the probabilistic descriptions.

The increased biological realism of the predicted connectivity comes at a computational cost, as the overlap calculations typically require more resources than the evaluations of a distance-dependent connectivity model. However, since this only has an influence on the construction of the network graph, subsequent dynamical simulations are not affected by this overhead as the same connectivity can be reused.

As Chapter 5 showed, updates to established notions of cortical connectivity driven by experimental anatomical data can strongly impact the resulting dynamical repertoire of spiking network models. The constellation-based connectivity presented in this chapter marks a further milestone in this endeavor by exploiting the unprecedented resolution of recent EM connectivity data. The hereby obtained precise spatial connectivity is of particular relevance for future computational modeling aimed at investigating the structure-dynamics interplay, akin to the modeling efforts described in Part iii.

Part V

CONCLUSIONS

8 CONCLUSIONS AND OUTLOOK

This thesis investigates the effect of spatial connectivity on the dynamics of spiking neural networks of visual cortex. It lays the technological foundation for simulations of large-scale neural networks (Chapter 3), leverages detailed anatomical data from a variety of experiments to constrain a comprehensive model of visual cortex of macaque V1 (Chapter 4), establishes links between the dynamics of instantiated cortical circuits and their structure (Chapter 5), derives a novel measure for assessing the similarity of adjacency matrices of cortical networks (Chapter 6), and devises a data-driven probabilistic connectivity rule for the next generation of *in silico* studies of spatial connectivity (Chapter 7). In the following, the technological and neuroscientific insights gained throughout this thesis are reflected, and an outlook is conceived on how future modeling studies can build on the presented results.

TECHNOLOGICAL FOUNDATIONS FOR LARGE-SCALE SIMULATIONS OF BIOLOGICALLY REALISTIC SPIKING NEURAL NETWORKS

The spatial connectivity observed in the primary visual cortex of macaque reaches distances of several millimeters, necessitating network models at the same scale for investigating the impact of spatial connectivity on cortical dynamics. At the same time, downscaling the number of neurons or synapses per unit volume alters the statistics of the emerging neural activity (van Albada et al., 2015a). Taken together, a computational approach requires models that consist of a large number of neurons and synapses. The routine execution of such network simulations is only viable if the models are efficiently implemented. In the development process, benchmarks are valuable tool for understanding bottlenecks in the performance of simulations. Benchmarking neural network simulators, however, proves to be complex when done in a collaborative fashion.

Part ii identifies the complexities that are particular to benchmarks of neural network simulators. Based on these observations, a platform is conceptualized and implemented, designed from the ground up for automating this task. In the process, it is demonstrated how different adaptations of a particular simulator code have positive performance implications for a variety of neural network architectures. Thereby, the necessary technological prerequisites for ensuring a continual development of neural network simulators are established: the implemented platform aids the development of dedicated simulation software that is the backbone of modern large-scale network simulations. Not only do current network simulations directly benefit from faster execution times, but the developed framework promises to facilitate future performance optimizations of neural network simulators, paving the way for the creation of network models that incorporate ever more biological detail.

While the benchmarking framework developed here automates the execution and analysis of single benchmarking runs, manual input is still needed in the selection of which simulator versions to test. Furthermore, the usage of the framework requires considerable expertise in handling high-performance computing resources, creating a substantial barrier of entry. These shortcomings can be addressed by integrating the benchmarking framework directly in the development workflow of the neural network simulators. This way, benchmarks could automatically be triggered as soon as performance-relevant changes are introduced in the code base. The full automation of benchmarking would eliminate the manual component, freeing up human resources to focus on the continual development of the performance of neural network simulators.

LINKING CORTICAL STRUCTURE TO DYNAMICS IN ANATOMICALLY CONSTRAINED SPIKING NEURAL NETWORK MODELS

With the technological foundations in place, Part iii leverages a broad collection of recent anatomical data to construct and simulate detailed neural network models of visual cortex at full neuronal and synaptic density.

Chapter 4 derives and implements three distinct spatial connectivity rules: (a) medium-range isotropic connectivity that decays in probability with the horizontal distance between neuron cell bodies; (b) long-range patchy connectivity between regions of visual cortex that share tuning properties to external stimuli, thereby establishing direct connections between neurons coding for similar properties across the visual scene; and (c) short-range push-pull connectivity between neurons that share input correlations which have been shown to promote contrast-invariant responses, matching experimental results. A dynamical analysis reveals that the model only operates in pathological states: when increasing the external drive onto the neurons in the network, the recurrent connectivity leads to runaway excitation, causing populations to fire at implausibly high rates. Calculating neural targeting patters in the underlying recurrent connectivity uncovers putative excitatory feedback loops which are hypothesized to be a root cause of this behavior.

Chapter 5 sets out to validate this hypothesis. The connectivity data used to constrain the model of V1 is based on probabilistic calculations of *potential connectivity*: in the data set in question (Binzegger et al., 2004), light microscopy (LM) is used to reconstruct the morphology of different neural cell types. Then, potential connectivity is calculated by calculating a spatial overlap of presynaptic axonal and postsynaptic dendritic arbors. Crucially, this method does not include information on where actual synapses are formed. Recent advances in the application of machine learning to imaging data from electron microscopy (EM) have made it possible to reconstruct substantial fractions of the actual connectivity, i.e., which neurons form synapses between each other, for volumes of cortex on the order of $1\,\mathrm{mm}^3$ (e.g., MICrONS Consortium et al. (2021)). To compare the two data sets, local cortical network models are derived which are constrained in their recurrent connectivity based on the respective anatomical data. Building on this, spiking neural network models are implemented and their dynamics are studied across regimes of external input that may stem from other brain areas. Thereby, it is revealed that feedback loops between excitatory neurons are indeed a key factor that contribute to pathological activity in models based on the LM data set. In contrast, the model based on the recent EM data avoids such dynamical regimes and exhibits biologically plausible activity that smoothly depends on the external drive.

These insights are utilized to derive an updated version of the model of macaque V1: replacing the LM connectivity data by EM connectivity data in the derivation of the number of synapses between populations of the model, simulations reveal biologically plausible first-order statistics of activity without fine-tuning of model parameters. Thus, the pathological dynamics are resolved in a data-driven way, highlighting the importance of high-resolution connectivity data constraining the structure of the model.

As such, the model of macaque V1 developed throughout Part iii provides a platform that future studies can use as a starting point to examine the effect of varying the spatial connectivity embedded within. Furthermore, the model is readily extendable. For example, future work could utilize the software pipelines developed for the model of V1 to create models of brain areas higher up the cortical hierarchy. In particular, adding areas V2 and V4—the main constituents of the ventral stream of visual processing—to form a joint multi-area model would unlock investigations of hierarchical processing. Due to its spatial nature, converging and diverging spatial connectivity schemes observed between V1, V2 and V4 could straight-forwardly be implemented. Considering the other direction of the cortical hierarchy, follow-up work could also implement a more sophisticated thalamic model which provides sensory input to the neurons in V1. Thereby, the propagation of activity could be investigated in a systematic fashion. Coupled with an assessment of performance metrics for tasks of processing visual input, the computational benefits of the spatial connectivity principles ingrained in the model could be studied, providing crucial information on the structure-function relationship of cortical processing.

DERIVING PRECISE SPATIAL CONNECTIVITY FOR FUTURE MODELING

Part iii demonstrates the critical nature of using high-resolution data for constraining the connectivity in spiking neural networks of visual cortex. Inspired by the unprecedented detail of connectivity data from EM-based experimental studies, Part iv investigates precise spatial connectivity that derives from the characteristic spatial distributions of synaptic locations. With a view towards computational modeling akin to the spiking neural networks employed in this thesis, Chapter 7 devises a predictive connectivity rule that estimates the number of synapses between neurons given their cell types and relative displacement. Naturally, the question arises whether the novel connectivity rule provides more accurate predictions than established ones. Since the EM data allows for an extraction of an experimental ground truth, this accuracy can, in principle, be tested. However, this raises the question of how to assess the similarity of networks as defined by their connectivity structure.

Chapter 6 addresses this in a formal way: since the information about recurrent connectivity in a network can be distilled into an adjacency matrix, which contains the number of connections between all pairs of neurons, the problem reduces to the quantification of similarity between matrices. In this chapter, a novel similarity measure is developed that captures the two-dimensional structure of matrices by design, thereby outperforming existing approaches.

Using this measure, Chapter 7 demonstrates that a predictive connectivity rule based on the precise spatial arrangements of synaptic contact locations—or *constellations*—produces network connectivities that are robustly more similar to the experimental ground truth than predictions from a purely distance-dependent rule.

Thereby, future computational studies can use a constellation-based rule to capture more of the intricate detail of spatial connectivity in spiking neural network simulations. Once additional EM data sets of comparable quality are available, future work could test the generalization capabilities of the constellation-based rule by predicting the connectivity across individual brains of the same species. Finally, incorporating constellation-based connectivity in spatially resolved spiking neural networks, akin to the model of macaque V1 developed in this thesis, could uncover the impact of precise spatial connectivity on the dynamics of cortical circuit models.

CONCLUDING REMARKS

Overall, this thesis investigates links between spatial connectivity and resulting cortical dynamics in a computational framework. In the course of this work, three key areas of development are addressed. First, the technological foundations for simulations of large-scale spiking neural networks are established. Second, anatomically detailed models of visual cortical areas are created in a data-driven way. Simulating those networks reveals fundamental links between the structure of cortical networks and their emerging activity, highlighting the impact of accurate estimates of spatial connectivity. Finally, a precise spatial connectivity rule with a view towards computational modeling is derived that can inform the structure of spiking neural network models of the next generation. Thereby, this thesis provides a thorough account on spatial connectivity in cortical neural networks, establishing frameworks for future studies to build upon. Linking the spatial connectivity structure to the dynamics of the studied large-scale spiking neural networks, it contributes to the ultimate goal of unraveling the structure-dynamics-function relationship of the brain.

Part VI

APPENDIX

A SUPPLEMENTARY MATERIALS

A.1 SUPPLEMENT TO "REVISITING ARCHITECTURES OF LOCAL CIRCUITS ..."

A.1.1 *Population-specific characteristic lengths*

We fit an exponential decay of the mean number of synapses using Equation 5.1 and the connection probability using Equation 5.2 to the actual and potential connectivity derived from EM and LM data to obtain the population-specific characteristic lengths. Figure A.1.1 shows the fits for the mean number of synapses from EM data; Figure A.1.2 shows the fits for the connection probability from EM data (actual connectivity); Figure A.1.3 shows the fits for the connection probability from EM data (potential connectivity); and Figure A.1.4 shows the fits for the connection probability and mean number of synapses from LM data.

Figure A.1.5 compares the length scale of the connection probability obtained from actual and potential connectivity using EM data. We find that the two methods produce similar estimates:

$$\gamma_X = \frac{\lambda^{\text{conn}}_{A,vX}}{\lambda^{\text{conn,pot}}_{A,vX}} \approx 1 \quad \forall v, X \tag{A.1.1}$$

where v is the layer and X is the cell type of the presynaptic population.

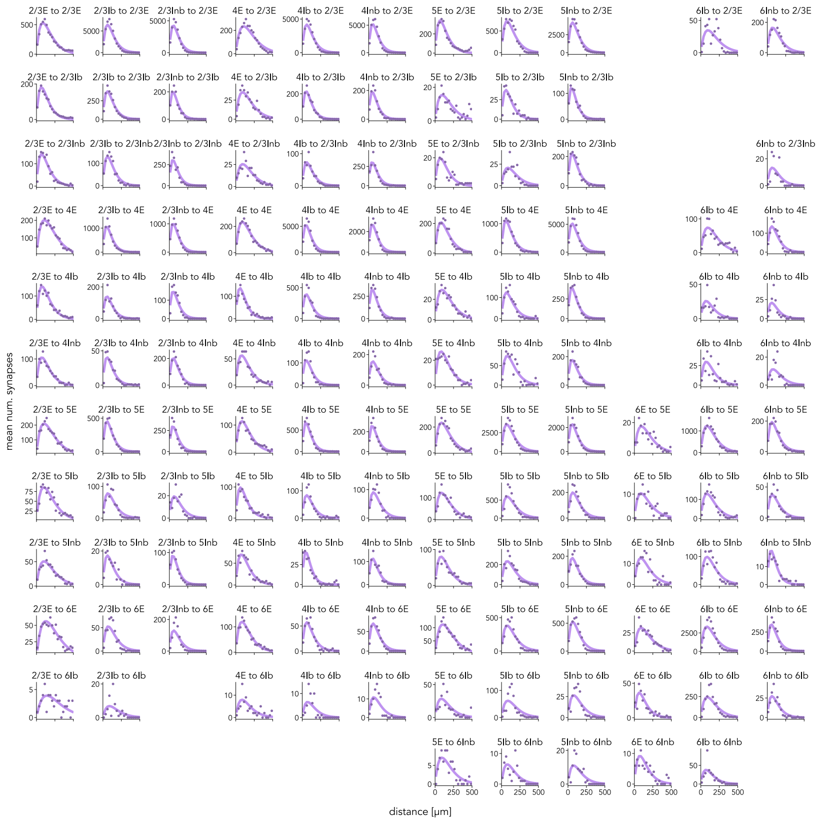

Figure A.1.1: **Density of mean number of synapses between pre- and postsynaptic populations from EM data, actual connectivity.** Purple dots indicate experimentally observed number of connections at distance d, purple curves are fits to the expected number of connections given the density of the mean number of synapses of one presynaptic neuron with all possible partners from the postsynaptic population at distance d (Equation 5.4). No fits for combinations of populations where the observed number of synapses is smaller than 50.

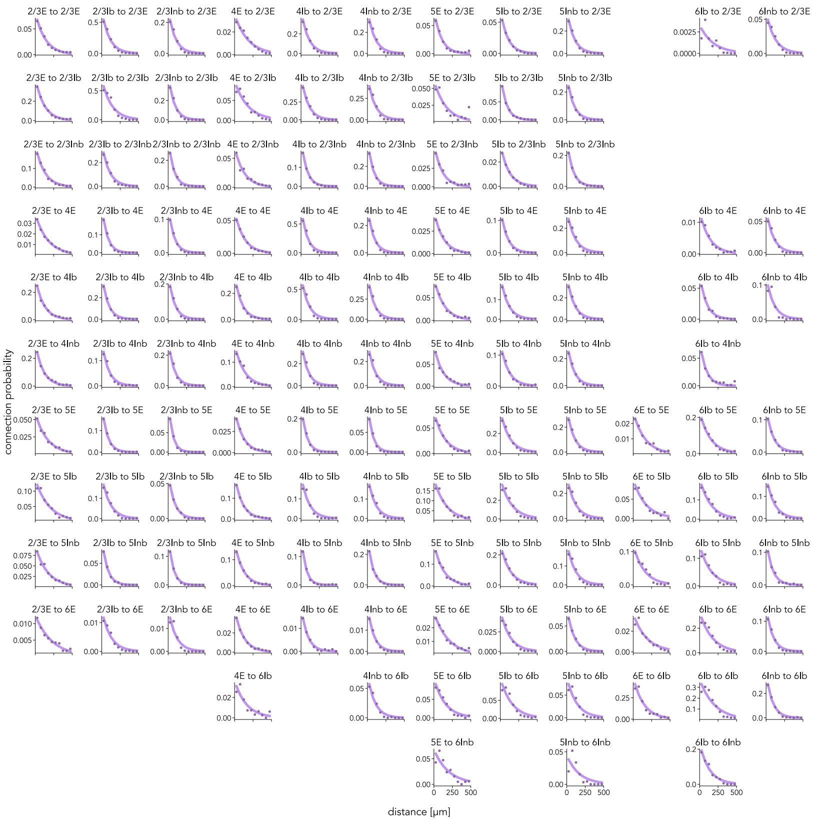

Figure A.1.2: **Connection probability between pre- and postsynaptic populations from EM data, actual connectivity.** Purple dots indicate experimentally observed connection probability at distance d, purple curves are fits to the expected connection probability (Equation 5.2). No fits for combinations of populations where the observed number of connections is smaller than 50.

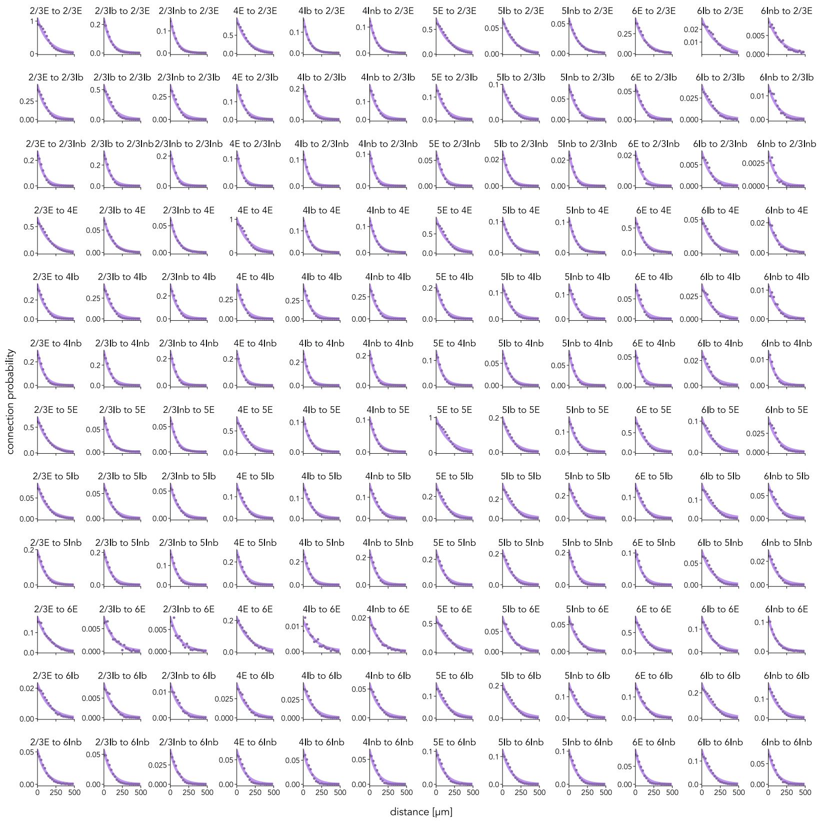

Figure A.1.3: **Connection probability between pre- and postsynaptic populations from EM data, potential connectivity.** Purple dots indicate connection probability at distance *d*, purple curves are fits to the expected connection probability (Equation 5.2).

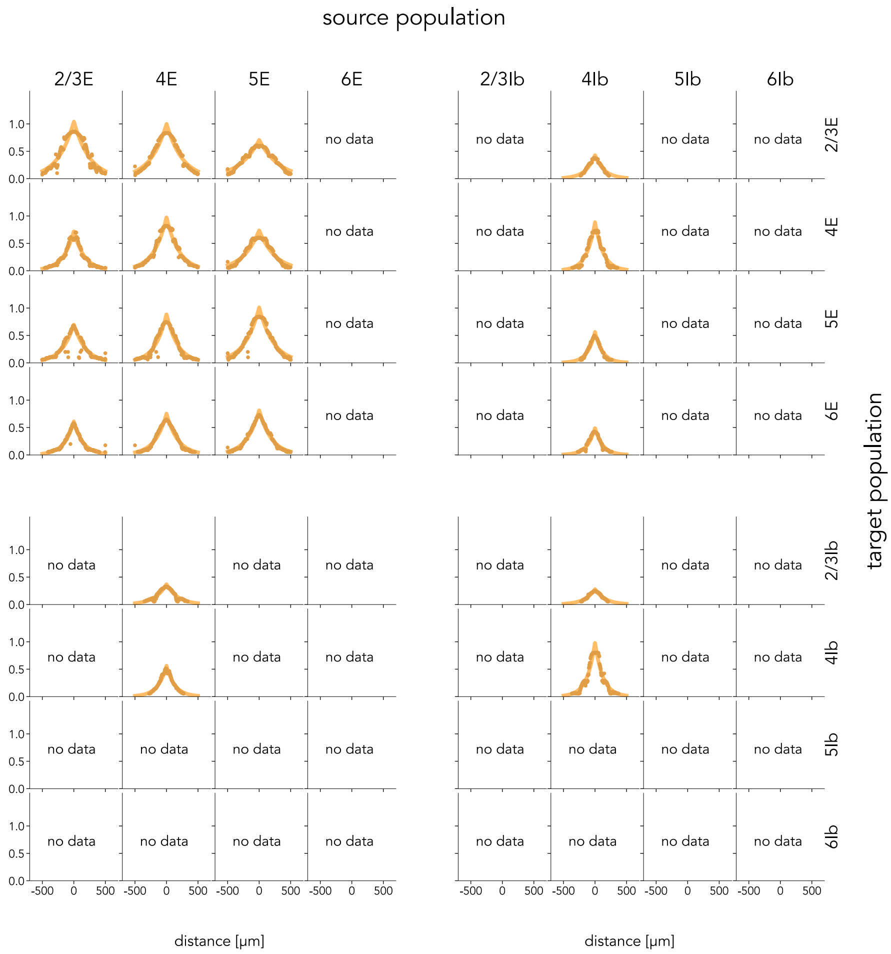

Figure A.1.4: **Potential connectivity between pre- and postsynaptic populations from LM data.** Source population horizontal, target population vertical. Panels show potential connectivity (vertical) at a lateral displacement d as orange dots. Orange curves are fits to the connection probability according to Equation 5.2 for E-to-E connections, and mean number of synapses according to Equation 5.1 for all other types of connections.

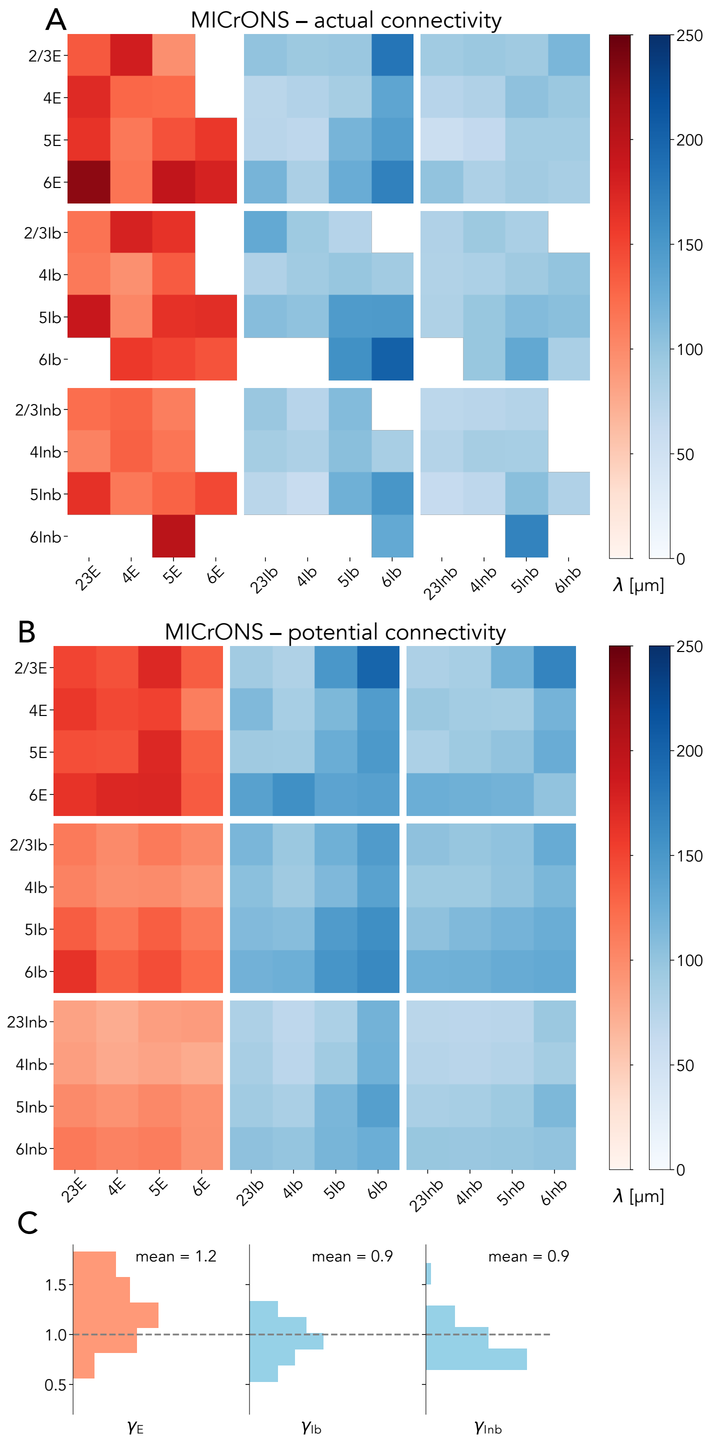

Figure A.1.5: **Comparison of the length scales of connection probability derived from actual and potential connectivity of EM data. A** Characteristic length scale $\lambda_{AB}^{\text{conn}}$ derived from actual connectivity. **B** Characteristic length scale $\lambda_{AB}^{\text{conn,pot}}$ derived from potential connectivity. **C** Fraction of length scales $\gamma_X = \lambda_{A,vX}^{\text{conn}} / \lambda_{A,vX}^{\text{conn,pot}}$ where $X \in \{\text{E}, \text{Ib}, \text{Inb}\}$.

A.1.2 *Characteristic lengths for LM-based model*

Since the data of Stepanyants et al. (2007) do not have the required resolution for the derivation in Section 5.2.2, we generalize the missing values from the estimated $\lambda^{\text{syn}}_{AB}$. For inhibitory source populations, we choose

$$\lambda^{\text{syn}}_{5\text{Ib},4\text{Ib}} = \lambda^{\text{syn}}_{5\text{E},4\text{Ib}} \tag{A.1.2}$$
$$\lambda^{\text{syn}}_{6\text{Ib},4\text{Ib}} = \lambda^{\text{syn}}_{6\text{E},4\text{Ib}} \tag{A.1.3}$$
$$\lambda^{\text{syn}}_{A,v\text{Ib}} = \lambda^{\text{syn}}_{A,4\text{Ib}} \tag{A.1.4}$$

where A represents any target population, and v represents any source layer. Since Stepanyants et al. (2007) only provide data for inhibitory basket cells, we use the same characteristic length scales also for inhibitory non-basket cells.

For excitatory source populations, we choose

$$\lambda^{\text{syn}}_{u\text{Ib},v\text{E}} = \lambda^{\text{syn}}_{u\text{E},v\text{E}} \tag{A.1.5}$$

for all source layers except $v = 6$ since data for this layer is lacking. There, we choose

$$\lambda^{\text{syn}}_{A,6\text{E}} = \lambda^{\text{syn}}_{A,5\text{E}} \,, \tag{A.1.6}$$

see Table A.1.1.

Characteristic lengths $\lambda_{AB}^{\text{syn}}$ of the mean number of synapses

Target population	**Source population**							
	2/3E	**2/3Ib**	**4E**	**4Ib**	**5E**	**5Ib**	**6E**	**6Ib**
2/3E	171 µm	146 µm	171 µm	146 µm	216 µm	146 µm	216 µm	146 µm
2/3Ib	171 µm	160 µm	179 µm	160 µm	216 µm	160 µm	216 µm	160 µm
4E	115 µm	131 µm	136 µm	131 µm	173 µm	131 µm	173 µm	131 µm
4Ib	115 µm	124 µm	134 µm	124 µm	173 µm	124 µm	173 µm	124 µm
5E	129 µm	117 µm	123 µm	117 µm	154 µm	117 µm	154 µm	117 µm
5Ib	129 µm	117 µm	123 µm	117 µm	154 µm	117 µm	154 µm	117 µm
6E	103 µm	120 µm	123 µm	120 µm	132 µm	120 µm	132 µm	120 µm
6Ib	102 µm	120 µm	123 µm	120 µm	132 µm	120 µm	132 µm	120 µm

Table A.1.1: **Characteristic lengths for $\mathcal{M}_{LM}$.** Characteristic lengths $\lambda_{AB}^{\text{syn}}$ of the mean number of synapses for model $\mathcal{M}_{LM}$. Green values are estimated directly from Stepanyants et al. (2007), black values are generalized from these estimates.

A.1.3 *Connection strengths*

We follow Kraynyukova et al. (2022) in defining the connection strength J^v_{XY} in a layer v from neurons of population vY to vX as

J^v_{XY} = average in-degree to neuron in vX from $vY \times$ corresponding PSP.

To compare our results with theirs, we only consider inhibitory basket cell populations and discard populations of inhibitory non-basket cells. We find that the weights chosen in the present work (cf. Section 5.2.3) satisfy the hierarchy of connection strengths found in Kraynyukova et al. (2022) (called connectivity weights in their study) in most cases. Additionally, we display the synaptic weights (assessed by unitary PSPs) used in Kraynyukova et al. (2022) originating from previous measurements (their reference Allen Institute for Brain Science, 2019). For these synaptic weights, the hierarchy is satisfied for the EM model in all cases except L6 EI-EE (Figure A.1.7, lower left panel). This is consistent with their findings using connection probabilities in mouse V1 obtained by electrophysiological means (their Figure 4).

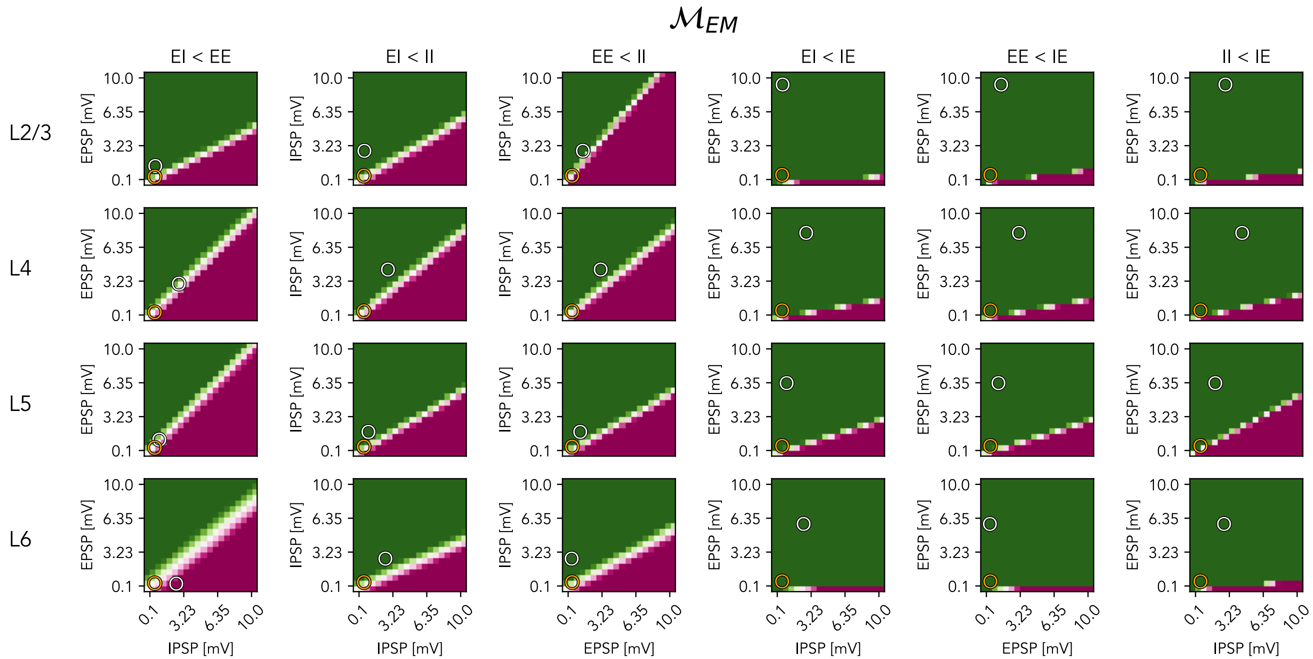

Figure A.1.6: **Connection strengths for $\mathcal{M}_{EM}$.** Layer- and population-pair-resolved connection strengths when systematically varying the PSP of presynaptic populations. On the horizontal (vertical) axis, the PSP of presynaptic populations with putatively smaller (larger) connection strengths is varied. Weights chosen in this work marked in orange, weights by the Allen Institute marked in white. Green (purple) indicates that the inequality given in the column title is true (false) and therefore the hierarchy of connection strengths is fulfilled (not fulfilled).

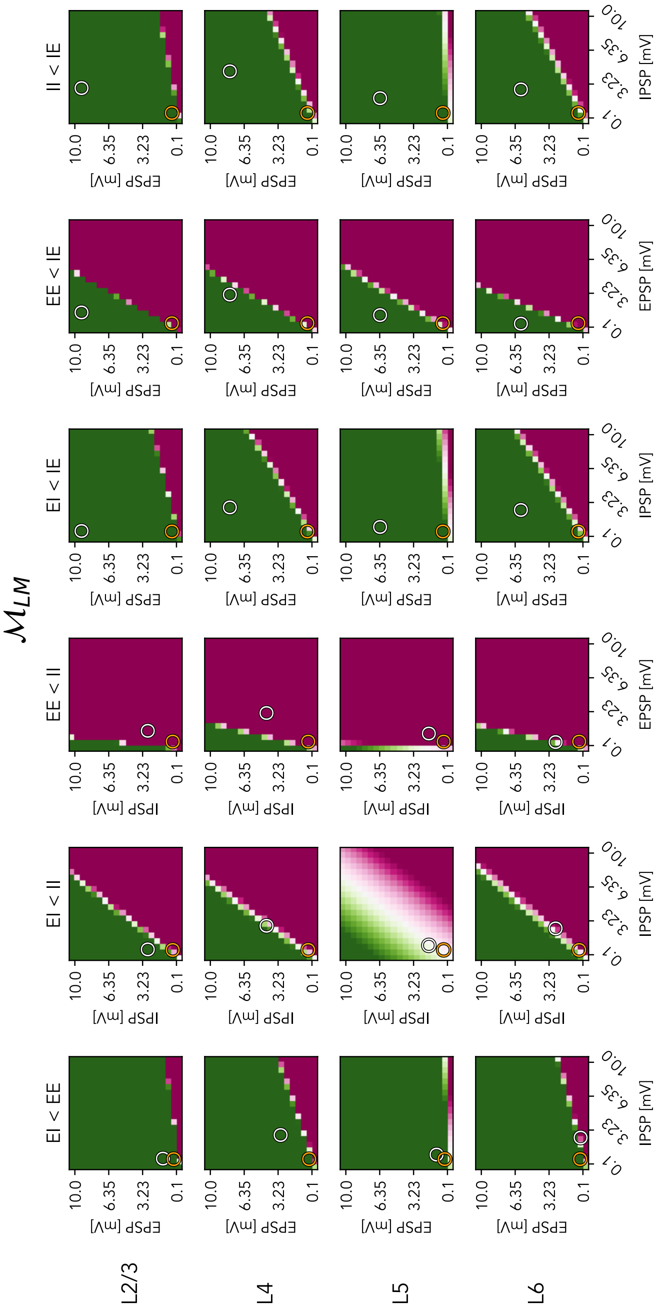

Figure A.1.7: **Connection strengths for $\mathcal{M}_{LM}$.** As in Figure A.1.6.

A.1.4 *Tables summarizing model definitions*

The network architecture and the list of parameters are summarized in the style of Nordlie et al. (2009). Table A.1.2 summarizes the models, Table A.1.3 specifies neuron and synapses models, Table A.1.4, Table A.1.5, and Table A.1.6 give the numeric values used in the simulations.

Structure	Multi-layer network of excitatory, inhibitory basket and non-basket neurons
Populations	12 cortical populations in 4 layers (L2/3, L4, L5, L6), populations vY for all v in layers, $Y \in \{\mathrm{E}, \mathrm{Ib}, \mathrm{Inb}\}$
Measurements	Spikes and membrane potential of neurons in all populations
Neuron Model	Leaky integrate-and-fire neurons, E and Inb neurons with adaptation
Synapse Model	Conductance-based synapses with exponentially shaped kernel
Network Model	Pairwise Bernoulli connectivity rule
External Drive	Modulations of excitatory conductances with an Ornstein-Uhlenbeck process

Table A.1.2: **Model summaries for $\mathcal{M}_{LM}$ and $\mathcal{M}_{EM}$.**

Neuron and synapse model for E and Inb cells	
Dynamics	$C_m \dot{V}_m = -g_L (V_m - E_L) - (g_{\mathrm{ex}} + g_{\mathrm{ext}})(V_m - E_{\mathrm{ex}})$ $\quad -g_{\mathrm{in}}(V_m - E_{\mathrm{in}}) - g_{\mathrm{a}}(V_m - E_{\mathrm{a}})$ $\quad -g_{\mathrm{rr}}(V_m - E_{\mathrm{rr}})$ $\tau_{\mathrm{ex}} \dot{g_{\mathrm{ex}}}(t) = -g_{\mathrm{ex}}(t) + \tau_{\mathrm{ex}} \sum_j w_j \sum_{t_j} \delta(t - t_j - d_j)$ $\tau_{\mathrm{in}} \dot{g_{\mathrm{in}}}(t) = -g_{\mathrm{in}}(t) + \tau_{\mathrm{in}} \sum_k w_k \sum_{t_k} \delta(t - t_k - d_k)$ $\tau_{\mathrm{a}} \dot{g_{\mathrm{a}}}(t) = -g_{\mathrm{a}}(t) + \tau_{\mathrm{a}} \sum_{t_i} \delta(t - t_i)$ $\tau_{\mathrm{rr}} \dot{g_{\mathrm{a}}}(t) = -g_{\mathrm{rr}}(t) + \tau_{\mathrm{rr}} \sum_{t_i} \delta(t - t_i)$ where t_i is the timing of the spike emitted by the neuron.
Spiking	If $V(t-) < V_{\mathrm{th}}$ and $V(t+) \geq V_{\mathrm{th}}$: 1. Set $t^* = t$ and $V(t) = V_{\mathrm{reset}}$ in $(t^*, t^* + \tau_r]$. 2. Emit spike with time stamp t^*.
Neuron and synapse model for inhibitory basket cells	
Dynamics	$C_m \dot{V}_m = -g_L (V_m - E_L) - (g_{\mathrm{ex}} + g_{\mathrm{ext}})(V_m - E_{\mathrm{ex}})$ $\quad -g_{\mathrm{in}}(V_m - E_{\mathrm{in}})$

Spiking	$\tau_{\text{ex}}\dot{g_{\text{ex}}}(t) = -g_{\text{ex}}(t) + \tau_{\text{ex}} \sum_j w_j \sum_{t_j} \delta(t - t_j - d_j)$ $\tau_{\text{in}}\dot{g_{\text{in}}}(t) = -g_{\text{in}}(t) + \tau_{\text{in}} \sum_k w_k \sum_{t_k} \delta(t - t_k - d_k)$ 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^*.
Computation time step	
Resolution	$\mathrm{d}t = 0.1\,\mathrm{ms}$
Synaptic delays	
Delay	Synaptic delay is exponentially distributed with a mean depending on the pre- and postsynaptic cell type. The minimal synaptic delay equals dt.
Synaptic weights	
Weights	Synaptic weights are lognormally distributed with a fixed relative (to the mean) standard deviation and the mean depending on the pre- and postsynaptic cell type.
Stimulation	
Ext. drive	$I_{\text{ext}} = X\,(V_m - E_{\text{ex}})$, $\frac{\mathrm{d}X}{\mathrm{d}t} = \frac{-(X-\mu)}{\tau_X} + \sigma\sqrt{\frac{2}{\tau_X}}\eta$ where η denotes white noise, μ and σ are population-specific.

Table A.1.3: **Neuron and synapse models.** Models are used for spiking neural network simulations of both $\mathcal{M}_{LM}$ and $\mathcal{M}_{EM}$.

Parameter	**Value**	**Description**
	Neuron parameters: excitatory neurons	
C_m	250 pF	Membrane capacitance
t_{ref}	2 ms	Refractory period
τ_{ex}	2 ms	Excitatory synaptic time constant
τ_{in}	5 ms	Inhibitory synaptic time constant
τ_X	10 ms	Time constant of Ornstein-Uhlenbeck noise
g_L	16.7 nS	Leak conductance
E_L	−70 mV	Resting potential
E_{in}	−75 mV	Inhibitory reversal potential
E_{ex}	0 mV	Excitatory reversal potential
V_{th}	−50 mV	Threshold
V_{reset}	−60 mV	Reset membrane potential
V_m	−60 mV	Initial membrane potential
	Neuron parameters: inhibitory basket neurons	
C_m	250 pF	Membrane capacitance
t_{ref}	4 ms	Refractory period
τ_{ex}	2 ms	Excitatory synaptic time constant
τ_{in}	5 ms	Inhibitory synaptic time constant
τ_X	10 ms	Time constant of Ornstein-Uhlenbeck noise
g_L	16.7 nS	Leak conductance
E_L	−70 mV	Resting potential
E_{in}	−75 mV	Inhibitory reversal potential
E_{ex}	0 mV	Excitatory reversal potential
V_{th}	−50 mV	Threshold
V_{reset}	−60 mV	Reset membrane potential
V_m	−60 mV	Initial membrane potential
	Neuron parameters: inhibitory non-basket neurons	
C_m	250 pF	Membrane capacitance
t_{ref}	4 ms	Refractory period
τ_{ex}	2 ms	Excitatory synaptic time constant

τ_{in}	5 ms	Inhibitory synaptic time constant
τ_X	10 ms	Time constant of Ornstein-Uhlenbeck noise
g_L	16.7 nS	Leak conductance
E_L	−65 mV	Resting potential
E_{in}	−75 mV	Inhibitory reversal potential
E_{ex}	0 mV	Excitatory reversal potential
V_{th}	−50 mV	Threshold
V_{reset}	−60 mV	Reset membrane potential
V_m	−60 mV	Initial membrane potential
Adaptation parameters		
q_{sfa}	14.48 nS	Quantal spike-frequency adaptation conductance increase
q_{rr}	3214 nS	Quantal relative refractory conductance increase
τ_{sfa}	110 ms	Time constant of spike-frequency adaptation
τ_{rr}	1.97 ms	Time constant of relative refractoriness
E_{sfa}	−70 mV	Spike-frequency reversal potential
E_{rr}	−70 mV	Relative refractory reversal potential

Table A.1.4: **Neuron parameters for $\mathcal{M}_{LM}$ and $\mathcal{M}_{EM}$.**

Mean delay

Target cell type	**Source cell type**		
	E	**Ib**	**Inb**
E	2 ms	0.8 ms	1.5 ms
Ib	1.2 ms	1.5 ms	1.5 ms
Inb	1.5 ms	1.2 ms	1.5 ms

Table A.1.5: **Mean delays between populations of given cell types for $\mathcal{M}_{LM}$ and $\mathcal{M}_{EM}$.** Numerical values inspired by measurements from Avermann et al. (2012).

Mean connection weights			
Target cell type	**Source cell type**		
	E	**Ib**	**Inb**
E	0.4 nS	4.7 nS	4.0 nS
Ib	0.8 nS	4.7 nS	3.2 nS
Inb	0.4 nS	6.5 nS	4.0 nS

Table A.1.6: **Mean connection weights between populations of given cell type for $\mathcal{M}_{LM}$ and $\mathcal{M}_{EM}$.** Numerical values inspired by measurements from Avermann et al. (2012). Relative standard deviation of weights equals 0.5.

μ_X for $\mathcal{M}_{LM}$			
Layer	**Cell type**		
	E	**Ib**	**Inb**
L2/3	6.34 nS	6.79 nS	5.78 nS
L4	8.07 nS	6.67 nS	5.57 nS
L5	6.79 nS	5.61 nS	5.66 nS
L6	8.00 nS	9.12 nS	8.22 nS

Table A.1.7: **Mean of conductance fluctuations.** Change in conductances modeled as OU processes providing external drive for the tuned $\mathcal{M}_{LM}$ network.

μ_X for $\mathcal{M}_{EM}$			
Layer	**Cell type**		
	E	**Ib**	**Inb**
L2/3	7.29 nS	6.55 nS	4.50 nS
L4	7.85 nS	8.45 nS	5.72 nS
L5	11.98 nS	11.54 nS	7.58 nS
L6	5.60 nS	5.45 nS	3.94 nS

Table A.1.8: **Mean of conductance fluctuations.** Change in conductances modeled as OU processes providing external drive for the tuned $\mathcal{M}_{EM}$ network.

A.2 SUPPLEMENT TO "QUANTIFYING THE SIMILARITY OF MATRICES ..."

A.2.1 *Canonical angles between subspaces*

We follow (Zhu and Knyazev, 2013). Given two subspaces H_a and H_b of $\mathbb{R}^n$ of dimension p and q, the canonical angles θ_i can be iteratively defined as

$$\cos(\theta_i) = \max_{v_i \in H_a, w_i \in H_b} \{ \langle v_i, w_i \rangle \text{ s.t. } ||v_i|| = ||w_i|| = 1 \qquad \text{(A.2.1)}$$
$$\text{and } v_i \perp v_1, ..., v_{i-1}, w_i \perp w_1, ..., w_{i-1} \},$$

or equivalently

$$\theta_i = \min_{v_i \in H_a, w_i \in H_b} \{ \arccos \left(\langle v_i, w_i \rangle \right) \text{ s.t. } ||v_i|| = ||w_i|| = 1 \qquad \text{(A.2.2)}$$
$$\text{and } v_i \perp v_1, ..., v_{i-1}, w_i \perp w_1, ..., w_{i-1} \} .$$

for $i = 1, ..., \min\{p, q\}$. The vectors v_i and w_i are called the *canonical vectors* and determine the alignment of the subspaces. If B_a, B_b are orthonormal bases of H_a and H_b respectively (written as matrices of $n \times p$ and $n \times q$, respectively), one can determine the canonical angles by means of the SVD of $B_a B_b^T$: writing

$$B_a B_b^T = U \Sigma V^T \qquad \text{(A.2.3)}$$

where $U \in \mathbb{R}^{p \times p}$ and $V \in \mathbb{R}^{q \times q}$ are orthogonal matrices and $\Sigma \in \mathbb{R}^{p \times q} = \text{diag}(\sigma_1, ..., \sigma_{\min\{p,q\}})$ is a diagonal matrix with entries greater than or equal to zero. As in the main text we assume that the singular values are in decreasing order. The canonical angles are then given by $\theta_i = \arccos(\sigma_i)$, and $v_i = B_a U e_i$ as well as $w_i = B_b V e_i$.

A.2.2 *Invariance properties of SAS*

Let $M_a, M_b \in \mathbb{R}^{m \times n}$ with SVDs

$$M_i = U_i \Sigma_i V_i^T \qquad \text{(A.2.4)}$$

for $i \in \{a, b\}$. Noting that the transposition of an orthogonal transformation is again orthogonal, the SVDs of M_i^T are

$$M_i^T = V_i \Sigma_i^T U_i^T \ . \qquad \text{(A.2.5)}$$

Thus, by its definition, the SAS between M_a and M_b is invariant under transposition of both matrices. If working with the eigendecomposition instead of the SVD, one obtains, in the general case,

$$M_i = P_i \Lambda_i P_i^{-1} \qquad \text{(A.2.6)}$$

where Λ_i is a diagonal matrix containing the eigenvalues, and the columns of P_i are the eigenvectors. Here, the eigenvectors are not guaranteed to be orthogonal. Thus,

$$(P_i^{-1})^T \neq P_i \tag{A.2.7}$$

in general. Consequently, any measure comparing the eigenvectors is not invariant under transposition of the compared matrices. Specifically, this includes the eigenangle score Gutzen et al. (2023).

Additionally, SAS is invariant under actions of identical orthogonal matrices from the left or right onto M_i. We denote the orthogonal matrices as O_1 and O_2^T. The action of the matrices from the left and from the rights yields

$$\tilde{M}_i = O_1 M_i O_2^T \ , \tag{A.2.8}$$

and the corresponding SVD is

$$\tilde{M}_i = \tilde{U}_i \Sigma_i \tilde{V}_i^T = \Big(O_1 U_i\Big)\Sigma_i\Big(O_2 V_i\Big)^T \ . \tag{A.2.9}$$

Note that the singular values remain identical. Since the left and right singular vectors of the matrices $\tilde{M}_i$ are $\tilde{u}_i^j = \tilde{U}_i e^j$ and $\tilde{v}_i^j = \tilde{V}_i e^j$, we compute the left and right singular angles as

$$\begin{aligned}\alpha^j(\tilde{U}) &= \arccos\Big(\langle \tilde{U}_a e^j, \tilde{U}_b e^j\rangle\Big) = \arccos\Big(\langle O_1 U_a e^j, O_1 U_b e^j\rangle\Big)\\ &= \arccos\Big(\langle U_a e^j, U_b e^j\rangle\Big) = \alpha^j(U)\end{aligned} \tag{A.2.10}$$

and *mutatis mutandis* for $\alpha^j(\tilde{V})$, resp. $\alpha^j(\tilde{U}^-), \alpha^j(\tilde{V}^-)$. We here used the orthogonality of the matrices O_i. Thus, the singular angles, the singular values, and consequently the similarity score, are invariant under the action of orthogonal matrices from the left and the right.

A.2.3 *Change in SAS under scaling*

Consider a matrix M with SVD

$$M = U\Sigma V^T \tag{A.2.11}$$

and scalars $c_1 > 0$ as well as $c_2 < 0$. In the following, our aim is to determine the SAS between M and $c_i M$ for $j = 1, 2$.

POSITIVE SCALING We note that the SVD of $c_1 M$ is

$$c_1 M = U(c_1\Sigma)V^T \ . \tag{A.2.12}$$

Hence, only the singular values are scaled while the left and right singular vectors remain constant. This implies $\alpha^j = 0\ \forall j$, and thus an angular similarity of $\Delta^j = 1\ \forall j$. Therefore, SAS between M and $c_1 M$ equals 1.

NEGATIVE SCALING We write $c_2 = -1 \cdot |c_2|$ and determine the SVD of $c_2 M$ as

$$c_2 M = (-1 \cdot U)|c_2|\Sigma V^T = U|c_2|\Sigma(-1 \cdot V^T) \,. \qquad \text{(A.2.13)}$$

In the following, we focus on the first of the two representations. The second one can be treated analogously. We first calculate

$$\alpha^j(U) = \arccos\left(\langle u^j, -u^j\rangle\right) = \arccos(-1) = \pi, \quad \alpha^j(V) = 0 \,. \qquad \text{(A.2.14)}$$

Thus, we obtain $\bar{\alpha}^j = \pi/2$ and consequently $\alpha^j = \pi/2 \ \forall j$. Therefore, the angular similarity $\Delta^j = 1 - \alpha^j / \frac{\pi}{2} = 0$, and the SAS between M and $c_2 M$ equals 0.

A.2.4 *Change in SAS under small perturbations*

Consider a matrix M with SVD

$$M = U\Sigma V^T \,. \qquad \text{(A.2.15)}$$

For simplicity, we assume that $M \in \mathbb{R}^{n\times n}$ is a random matrix with non-degenerate singular values. First, we relate the singular values as well as the left and right singular vectors to the eigenvalues and eigenvectors of certain symmetric matrices. We calculate

$$MM^T = U\Sigma^2 U^T, \quad M^T M = V\Sigma^2 V^T \qquad \text{(A.2.16)}$$

and observe that MM^T and $M^T M$ are symmetric, the squared singular values are the eigenvalues of the matrices, and since U and V are orthogonal matrices, the left and right singular vectors are the eigenvectors. We perturb M by a random $\sqrt{\epsilon}W$, thus considering $M + \sqrt{\epsilon}W$. This induces a perturbation up to first order (in $\sqrt{\epsilon}$) of $\sqrt{\epsilon}\left(MW^T + WM^T\right)$ in MM^T and $\sqrt{\epsilon}\left(M^T W + W^T M\right)$ in $M^T M$. Following Rayleigh-Schrödinger perturbation theory (Landau and Lifshitz, 2013) , the perturbed left singular vectors $\tilde{u}^j$ read

$$\tilde{u}^j = u^j + \sqrt{\epsilon} \sum_{k\neq j} \frac{\langle u^k, \left(MW^T + WM^T\right) u^j\rangle}{\sigma^{j^2} - \sigma^{k^2}} u^k + \epsilon \cdot \text{ terms} \perp \text{to } u^j \,. \qquad \text{(A.2.17)}$$

while the perturbed eigenvalues are

$$\tilde{\sigma}^{j^2} = \sigma^{j^2} + \sqrt{\epsilon}\langle u^j, \left(MW^T + WM^T\right) u^j\rangle + \epsilon \sum_{k\neq j} \frac{\langle u^k, \left(MW^T + WM^T\right) u^j\rangle^2}{\sigma^{j^2} - \sigma^{k^2}} \,. \qquad \text{(A.2.18)}$$

To investigate how the eigenvalues change under perturbation, we calculate:

$$\begin{aligned}
&\langle u^j, \left(MW^T + WM^T\right) u^j \rangle \\
=&\langle u^j, MW^T u^j \rangle + \langle u^j, WM^T u^j \rangle \\
=&\langle WM^T u^j, u^j \rangle + \langle u^j, WM^T u^j \rangle \\
=&2\langle WM^T u^j, u^j \rangle \\
=&2\sigma^j \langle Wv^j, u^j \rangle \ .
\end{aligned} \tag{A.2.19}$$

Similarly,

$$\langle u^k, \left(MW^T + WM^T\right) u^j \rangle = \sigma^k \langle Wv^k, u^j \rangle + \sigma^j \langle u^k, Wv^j \rangle \ . \tag{A.2.20}$$

Writing the perturbation matrix W as an expansion of the left and right singular vectors, we get

$$W = \sum_{j,k} \bar{w}_{jk} u^j (v^k)^T \tag{A.2.21}$$

where $\bar{w}_{ij} \in \mathbb{R}$ are the corresponding coefficients. Thus

$$\tilde{\sigma}^{j^2} = \sigma^{j^2} + \sqrt{\epsilon} 2\sigma^j \bar{w}_{jj} + \epsilon \sum_{k \neq j} \frac{\left(\sigma^k \bar{w}_{jk} + \sigma^j \bar{w}_{kj}\right)^2}{\sigma^{j^2} - \sigma^{k^2}} \ . \tag{A.2.22}$$

Consequently, it is easy to construct perturbations that leave certain singular values unchanged. For a random perturbation, however, this is unlikely: even when the perturbation changes a single entry of M, since the left and right singular vectors generally have no resemblance to the standard normal basis of $\mathbb{R}^n$, most of the $\bar{w}_{ij}$ differ from zero. Neglecting the special case in which the first and second order exactly cancel, we can assume that all eigenvalues change under perturbation.

Writing

$$c_{kj} = \frac{\langle u^k, \left(MW^T + WM^T\right) u^j \rangle}{\sigma^{j^2} - \sigma^{k^2}} \ , \tag{A.2.23}$$

normalizing the eigenvectors yields

$$\tilde{u}^j = \left(1 - \frac{\epsilon}{2} \sum_{k \neq j} c_{kj}^2\right) u^j + \sqrt{\epsilon} \cdot \text{ terms} \perp \text{to } u^j \tag{A.2.24}$$

and *mutatis mutandis* for the perturbed right singular vectors $\tilde{v}^j$. This implies for the angles between the left singular vectors of the unperturbed and perturbed matrix

$$\begin{aligned}\alpha^j(U) &= \arccos\left(\langle u^j, \tilde{u}^j\rangle\right)\\ &= \arccos\left(\langle u^j, \left(1-\frac{\epsilon}{2}\sum_{k\neq j} c_{kj}^2\right)u^j + \sqrt{\epsilon}\cdot \text{ terms} \perp \text{to } u^j\rangle\right)\\ &= \arccos\left(1-\frac{\epsilon}{2}\sum_{k\neq j} c_{kj}^2\right) \qquad \text{(A.2.25)}\end{aligned}$$

where we used the orthonormality of the u^i.

By a similar argument as for the eigenvalues, we can also assume that for a generic perturbation, the eigenvectors change. Thus, $\sum_{k\neq j} c_{kj}^2 > 0$ since $\langle u^j, \tilde{u}^j\rangle < 1$, and we denote by $C(M, W, U)$ the smallest of those terms. Hence, we obtain

$$\alpha^j(U) \leq \arccos(1 - \epsilon C(M, W, U)) \qquad \text{(A.2.26)}$$

where the constant also absorbs the factor $\frac{1}{2}$. Since the same relations hold for the right singular vectors, we have $\alpha^j \leq \arccos(1 - \epsilon^2 C(M, W))$ where $C(M, W) = \min\{C(M, W, U), C(M, W, V)\}$. And as this holds true for all angles α^j, it implies that SAS between the matrices M and $M + \sqrt{\epsilon}W$ is $1 - \arccos(1 - \epsilon C(M, W))/\frac{\pi}{2}$ up to first order in ϵ.

A.2.5 *The Perron-Frobenius Theorem and the Perron vector*

The Perron-Frobenius theorem (Deuflhard and Hohmann, 2008) for non-negative square matrices asserts that if a matrix N has only non-negative entries, then

- there is a real eigenvalue $\lambda_P \geq 0$ of N such that, for all other eigenvalues λ of N, $|\lambda| \leq \lambda_P$, and
- the eigenvector corresponding to the eigenvalue λ_P has only non-negative entries. This eigenvector is called the Perron vector.

For the random graphs in this work, SAS considers $N = M_i M_i^T$ and $N = M_i^T M_i$ where M_i is an adjacency matrix with non-negative entries. In both cases, N is a symmetric positive-definite matrix. Both N share all non-zero eigenvalues (the squares of the singular values of M_i), and the eigenvectors of N are the right and left singular vectors (up to a potential multiplication with -1.) Since by assumption non-trivial singular values are unique, and under the assumption that at least one non-trivial singular value exists, the above inequalities become strict, and the eigenvector to λ_P becomes unique.

A.2.6 *Non-square matrices from subsampled networks*

We define subsampled graphs as graphs where the connectivity is only known for a random subset of source nodes. Thus, the adjacency matrices are non-square. We keep the number of possible connections in the network the same by instantiating a full graph of size $N_> = \frac{1}{f} N_N$ where $f < 1$, and then subsample $N_< = f N_N$ source nodes. For all networks we here choose $f = 2/3$ and for the comparison between subsampled graphs in this work, the list of source nodes for which information is available is shared across network instance and models. We use SAS to test how similar the structures of subsampled networks are.

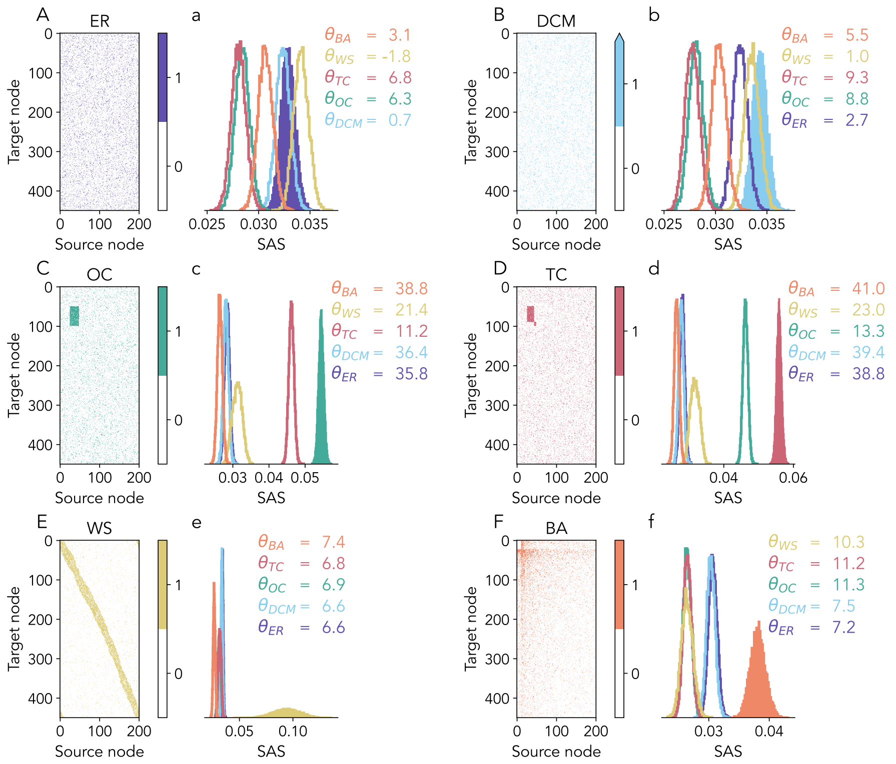

Figure A.2.1: **Self- and cross-similarity of subsampled network models. A–F** Single instances of the different network models. Colored matrix elements indicate a connection between nodes. **a–f** Histograms of SAS between instances of the network models ($n = 100$, all pairs compared). Filled distributions indicate self-similarities, non-filled ones indicate cross-similarities. Effect sizes θ between self- and cross-similarities also shown.

Instances of these subsampled graphs, their respective self- and cross-similarities, and the corresponding effect sizes are shown in Figure A.2.1. In subsampled graphs, some of the original structure is inevitably invisible, and we therefore expect SAS to find less distinguishable structure. This is especially apparent for the DCM network, which does not have enough structure to be reliably distinguished from the other network models. All other subsampled network graphs retain enough structure such that SAS can distinguish between them.

A.2.7 *SAS for* $1 \times n$ *matrices*

Let $v_i \in \mathbb{R}^{1\times n}$ where $i \in \{a, b\}$. In this case, the SVD reads

$$v_i = 1 \cdot \mathrm{diag}(\|v_i\|, 0, ..., 0) \cdot \left[\begin{array}{c|c|c|c} \frac{v_i}{\|v_i\|_2} & * & \dots & * \end{array} \right]. \quad \text{(A.2.27)}$$

Thus, there is only one non-zero singular value, resulting in the weighted sum used in the definitions of SAS to consist of one summand, with weight one. Thus, SAS is the angular similarity between v_a and v_b in this case.

A.2.8 *Comparison of SAS with common measures on brain data*

We here compare how similarity is assessed by SAS, cosine similarity, and the Frobenius norm between the different trial types of brain activity data (Section 6.3.3). Figure A.2.2 shows that only SAS identifies that the recorded activity is more similar between trials where the stimulus has the same orientation (U-D and L-R).

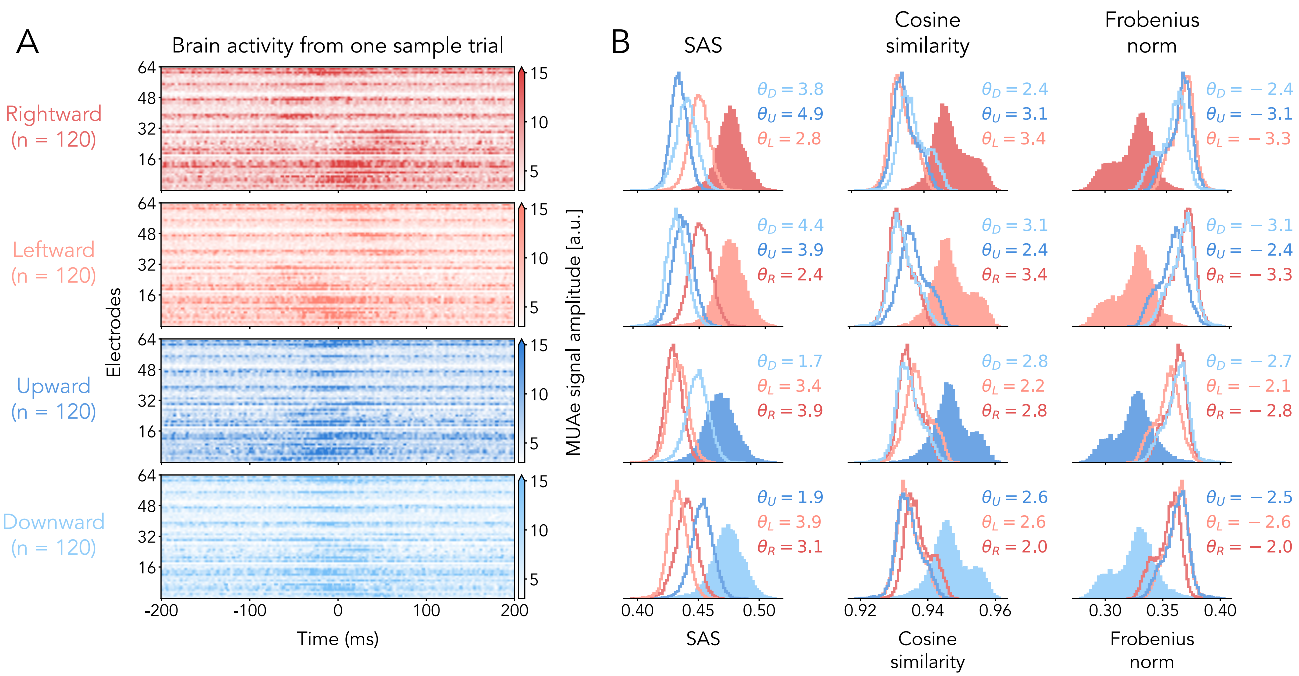

Figure A.2.2: **Comparing non-square matrices of brain activity with SAS, cosine similarity, and the Frobenius norm. A** Brain activity of all recorded electrodes for one example trial of each type (different trials as shown in Figure 6.7). **B** Histograms of SAS, cosine similarity and the Frobenius norm between all individual trials. Filled distributions indicate self-similarities, non-filled ones indicate cross-similarities. Effect sizes θ for the comparison between self- and cross-similarities also shown.

A.3 SUPPLEMENT TO "PREDICTING SPATIAL CONNECTIVITY ..."

A.3.1 *Population-specific characteristic lengths*

We fit an exponential decay (Equation 7.6) to the distance-resolved number of synapses to obtain the population-specific characteristic lengths λ_{AB}.

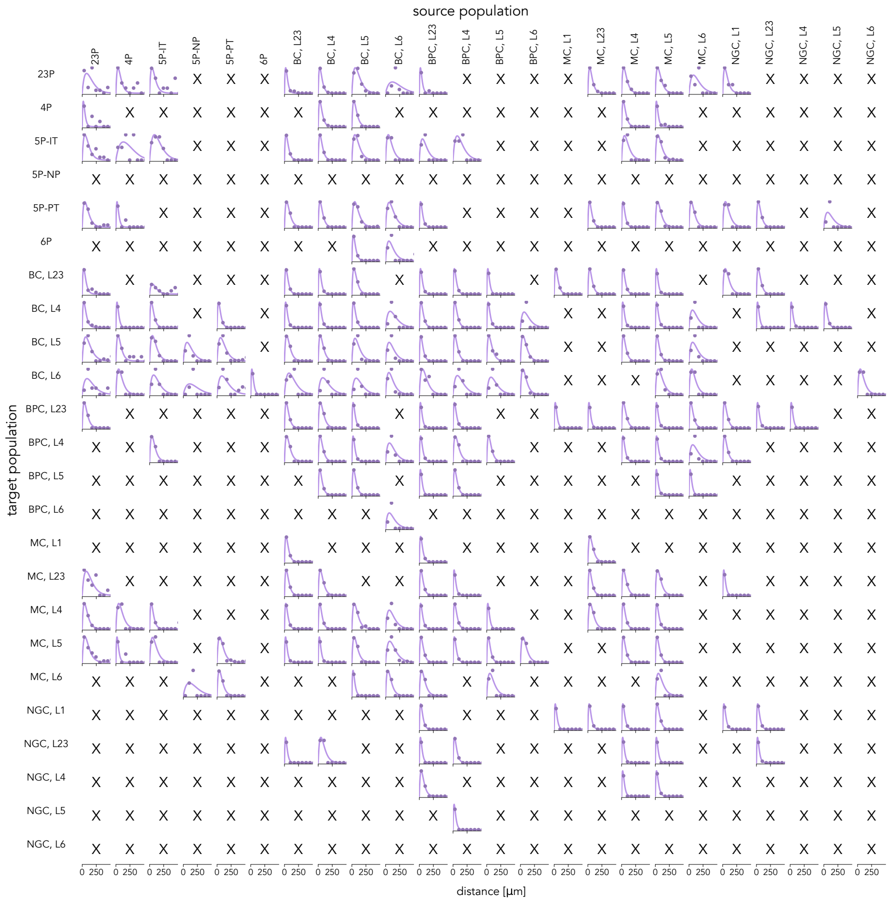

Figure A.3.1: **Distance-dependent connectivity fits.** Points indicate experimentally observed number of synapses at a distance d, lines are fits to the expected number of synapses given an exponential rule (Equation 7.6). No fits for combinations of populations where the observed number of connections is smaller than 5.

BIBLIOGRAPHY

Adesnik, Hillel (2017). “Synaptic mechanisms of feature coding in the visual cortex of awake mice.” In: *Neuron* 95.5, pp. 1147–1159. DOI: 10.1016/j.neuron.2017.08.014 (cit. on pp. 110, 120).

Akar, Nora Abi et al. (2019). “Arbor —A Morphologically-Detailed Neural Network Simulation Library for Contemporary High-Performance Computing Architectures.” In: *2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*. IEEE, pp. 274–282. DOI: 10.1109/EMPDP.2019.8671560 (cit. on p. 15).

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 pp. xi, 17).

Albers, Jasper et al. (2024). *Assessing the similarity of real matrices with arbitrary shape*. arXiv: 2403.17687 [q-bio.NC] (cit. on pp. xi, 127).

Albert, Réka and Albert-László Barabási (2002). “Statistical mechanics of complex networks.” In: *Rev. Mod. Phys.* 74, pp. 47–97 (cit. on p. 149).

Amir, Y., M. Harel, and R. Malach (1993). “Cortical Hierarchy Reflected in the Organization of Intrinsic Connections in Macaque Monkey Visual Cortex.” In: *J. Comp. Neurol.* 334, pp. 19–46 (cit. on p. 84).

Amit, Daniel J. and Nicolas Brunel (Apr. 1997). “Model of Global Spontaneous Activity and Local Structured Activity During Delay periods in the Cerebral Cortex.” In: *Cereb. Cortex* 7, pp. 237–252. DOI: 10.1093/cercor/7.3.237 (cit. on p. 49).

Angelucci, Alessandra, Jonathan B. Levitt, and Jennifer S. Lund (2002). “Anatomical origins of the classical receptive field and modulatory surround field of single neurons in macaque visual cortical area V1.” In: *Prog. Brain Res.* 136, pp. 373–388 (cit. on p. 59).

Antolík, Ján et al. (Aug. 2024). “A comprehensive data-driven model of cat primary visual cortex.” In: *PLOS Computational Biology* 20.8, pp. 1–42. DOI: 10.1371/journal.pcbi.1012342 (cit. on pp. 50, 53).

Atallah, Bassam V et al. (2012). “Parvalbumin-expressing interneurons linearly transform cortical responses to visual stimuli.” In: *Neuron* 73.1, pp. 159–170. DOI: 10.1016/j.neuron.2011.12.013 (cit. on p. 107).

Avermann, M et al. (2012). “Microcircuits of excitatory and inhibitory neurons in layer 2/3 of mouse barrel cortex.” In: *J. Neurophysiol.* 107.11, pp. 3116–3134 (cit. on pp. 107, 193, 194).

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. 97, 153).

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 p. 97).

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. 54, 113).

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. 56, 65–68, 70, 75, 76, 93, 95, 97, 113, 172).

Blasdel, Gary G (1992). "Orientation selectivity, preference, and continuity in monkey striate cortex." In: *Journal of Neuroscience* 12.8, pp. 3139–3161 (cit. on p. 85).

Blasdel, Gary G and DAVID Fitzpatrick (1984). "Physiological organization of layer 4 in macaque striate cortex." In: *Journal of Neuroscience* 4.3, pp. 880–895 (cit. on p. 57).

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. 81, 82).

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 pp. 59, 86).

Box, George EP (1976). "Science and statistics." In: *Journal of the American Statistical Association* 71.356, pp. 791–799 (cit. on p. 7).

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 p. 97).

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 p. 49).

Cai, T Tony, Jianqing Fan, and Tiefeng Jiang (2013). "Distributions of angles in random packing on spheres." In: *Journal of Machine Learning Research* 14, pp. 1837–1864. DOI: 10.1098/rspb.2001.1767 (cit. on p. 139).

Callan, Alexander R. et al. (2021). "Arrangement of Excitatory Synaptic Inputs on Dendrites of the Medial Superior Olive." In: *Journal of Neuroscience* 41.2, pp. 269–283. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.1055-20.2020. eprint: https://www.jneurosci.org/content/41/2/269.full.pdf (cit. on p. 153).

Carnevale, Nicholas T. and Michael L. Hines (2006). *The NEURON Book*. Cambridge: Cambridge University Press (cit. on p. 15).

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. 144, 149).

Cohen, Jacob (1988). *Statistical Power Analysis for the The Behavioral Sciences*. L. Erlbaum Associates, p. 567. ISBN: 0805802835 (cit. on p. 136).

Cohen, Marlene R. and Adam Kohn (July 2011). "Measuring and interpreting neuronal correlations." In: *Nat. Rev. Neurosci.* 14.7, pp. 811–819. DOI: 10.1038/nn.2842 (cit. on p. 107).

Cooper, Colin and Alan Frieze (2004). "The Size of the Largest Strongly Connected Component of a Random Digraph with a Given Degree Sequence." In: *Combinatorics, Probability and Computing* 13.3, 319–337. DOI: 10.1017/S096354830400611X (cit. on p. 148).

Crook, Sharon M., Andrew P. Davison, and Hans E. Plesser (2013). "Learning from the Past: Approaches for Reproducibility in Computational Neuroscience." In: *Bower J. (eds) 20 Years of Computational Neuroscience. Springer Series in Computational Neuroscience*. Springer New York, pp. 73–102. DOI: 10.1007/978-1-4614-1424-7_4 (cit. on p. 17).

Dayan, P. and L. F. Abbott (2001). *Theoretical Neuroscience*. Cambridge: MIT Press. DOI: 9780262041997 (cit. on p. 107).

De Valois, Russell L, Duane G Albrecht, and Lisa G Thorell (1982). "Spatial frequency selectivity of cells in macaque visual cortex." In: *Vision Res.* 22.5, pp. 545–559 (cit. on p. 144).

Denker, Michael et al. (Apr. 2022). *Elephant 0.11.1*. DOI: 10.5281/ZENODO.6470226 (cit. on p. 118).

Destexhe, A et al. (Nov. 2001). "Fluctuating synaptic conductances recreate in vivo-like activity in neocortical neurons." In: *Neuroscience* 107.1, pp. 13–24. DOI: 10.1016/s0306-4522(01)00344-x (cit. on pp. 107, 117).

Deuflhard, Peter and Andreas Hohmann (2008). *Numerische Mathematik 1: Eine algorithmisch orientierte Einführung*. Walter de Gruyter. DOI: 10.1515/9783110203554 (cit. on pp. 142, 199).

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. 16).

Douglas, R. J. and K. A. C. Martin (2004). "Neuronal Circuits of the Neocortex." In: *Annu. Rev. Neurosci.* 27, pp. 419–451 (cit. on p. 56).

Ecker, Alexander S. et al. (Jan. 2010). "Decorrelated Neuronal Firing in Cortical Microcircuits." In: *Science* 327.5965, pp. 584–587. DOI: 10.1126/science.1179867 (cit. on p. 107).

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. 7, 15).

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. 24, 26, 40).

Erdős, P. and A. Rényi (1959). "On random graphs." In: *Publ. Math.* 6, pp. 290–297 (cit. on pp. 117, 148).

Fardet, Tanguy et al. (Aug. 2021). *NEST 2.20.2*. DOI: 10.5281/zenodo.5242954 (cit. on pp. 32, 33).

Fehlberg, Erwin (1969). *Low-order classical Runge-Kutta formulas with stepsize control and their application to some heat transfer problems*. Vol. 315. National aeronautics and space administration (cit. on p. 118).

Feldmeyer, Dirk et al. (2018). "Inhibitory interneurons and their circuit motifs in the many layers of the barrel cortex." In: *Neuroscience* 368, pp. 132–151 (cit. on p. 107).

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. 81, 82).

Galvani, Luigi (1791). "De viribus electricitatis in motu musculari. Commentarius." In: *De Bonoiensi Scientiarum et Artium Intituo atque Academie Commentarii* 7, pp. 363–418 (cit. on p. 5).

Gamblin, Todd et al. (Nov. 2015). "The Spack package manager." In: ACM, pp. 1–12. DOI: 10.1145/2807591.2807623 (cit. on p. 23).

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 p. 57).

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. 23).

Gentet, Luc J et al. (2012). "Unique functional properties of somatostatin-expressing GABAergic neurons in mouse barrel cortex." In: *Nat. Neurosci.* 15.4, pp. 607–612 (cit. on p. 107).

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. 15, 23, 24, 55, 107).

Gillespie, Daniel T. (Aug. 1996). "Exact numerical simulation of the Ornstein-Uhlenbeck process and its integral." In: *Phys. Rev. E* 54.2, pp. 2084–2091 (cit. on p. 118).

Gleeson, Padraig et al. (Aug. 2019). "Open Source Brain: A Collaborative Resource for Visualizing, Analyzing, Simulating, and Developing Standardized Models of Neurons and Circuits." In: *Neuron* 103.3, 395–411.e5. DOI: 10.1016/j.neuron.2019.05.019 (cit. on p. 17).

Golgi, Camillo (1873). "Sulla struttura della sostanza grigia del cervello." In: *Gaz. Med. Ital. Lomb.* 33, pp. 244–46 (cit. on p. 3).

Golomb, David and John Rinzel (1994). "Clustering in globally coupled inhibitory neurons." In: *Physica D* 72.3, pp. 259–282. DOI: 10.1016/0167-2789(94)90214-3 (cit. on p. 107).

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 p. 15).

Golub, Gene H and Hongyuan Zha (1995). "The canonical correlations of matrix pairs and their numerical computation." In: *Linear Algebra for Signal Processing*. Vol. 69. The IMA Volumes in Mathematics and its Applications. Springer. DOI: https://doi.org/10.1007/978-1-4612-4228-4_3 (cit. on p. 127).

Goodman, Dan and Romain Brette (2008). "Brian: a simulator for spiking neural networks in Python." In: *Front. Neuroinform.* 2, p. 5. DOI: 10.3389/neuro.11.005.2008 (cit. on p. 15).

Gutzen, Robin, Sonja Grün, and Michael Denker (2023). "Evaluating the statistical similarity of neural network activity and connectivity via eigenvector angles." In: *BioSystems* 223, p. 104813. DOI: https://doi.org/10.1016/j.biosystems.2022.104813 (cit. on pp. 128, 131, 151, 196).

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 p. 17).

Hahne, Jan et al. (June 2021). *NEST 3.0*. Version 3.0. DOI: 10.5281/zenodo.4739103 (cit. on pp. 33, 40).

Harremoës, Peter (2001). "Binomial and Poisson distributions as maximum entropy distributions." In: *IEEE Trans. Inf. Theory* 47.5, pp. 2039–2041 (cit. on p. 167).

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. 15, 40).

Hellwig, Bernhard (Feb. 2000). "A quantitative analysis of the local connectivity between pyramidal neurons in layers 2/3 of the rat visual cortex." In: *Biol. Cybern.* 2.82, pp. 111–121. DOI: 10.1007/pl00007964 (cit. on p. 101).

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. 49, 59, 144).

– (1962). "Receptive Fields, Binocular Interaction, and Functional Architecture in the Cat's Visual Cortex." In: *J. Physiol.* 160, pp. 106–154 (cit. on p. 49).

Hubel, D. H. and T. N. Wiesel (1968). "Receptive Fields and Functional Architecture of Monkey Striate Cortex." In: *J. Neurophysiol.* 195.1, pp. 215–243 (cit. on pp. 49, 88).

– (1977). "Ferrier lecture. Functional architecture of macaque monkey visual cortex." In: *Proc. Biol. Sci.* 198.1130, pp. 1–59 (cit. on p. 59).

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. 30).

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. 15).

Isbister, James B et al. (2023). "Modeling and Simulation of Neocortical Micro-and Mesocircuitry. Part II: Physiology and Experimentation." In: *bioRxiv*, pp. 2023–05 (cit. on pp. 107, 113).

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 p. 97).

Jones, Eric, Travis Oliphant, Pearu Peterson, et al. (2001). *SciPy: Open source scientific tools for Python*. http://www.scipy.org/ (cit. on p. 116).

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. 87).

Jordan, Camille (1875). "Essai sur la géométrie à n dimensions." In: *Bull. Soc. Math. Fr.* 3, pp. 103–174 (cit. on p. 132).

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. 15, 16, 25, 33, 34, 36, 40–42).

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. 84).

Jülich Supercomputing Centre (2015). *JUQUEEN: IBM Blue Gene/Q® Supercomputer System at the Jülich Supercomputing Centre*. DOI: 10.17815/jlsrf-1-18 (cit. on p. 33).

Kandel, Eric R., James H. Schwartz, and Thomas M. Jessel (2000). *Principles of Neural Science*. 4th ed. ISBN 978-0838577011. New York: McGraw-Hill (cit. on p. 3).

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 p. 15).

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 p. 15).

Knight, James C and Thomas Nowotny (2021). "Larger GPU-accelerated brain simulations with procedural connectivity." In: *Nat. Comput. Sci.* 1.2, pp. 136–142. DOI: 10.1038/s43588-020-00022-7 (cit. on p. 15).

Knoll, Max and Ernst Ruska (1932). "Das elektronenmikroskop." In: *Zeitschrift für physik* 78, pp. 318–339 (cit. on p. 95).

Korcsak-Gorzo, Agnes et al. (2024). "Phenomenological Modeling of Diverse and Heterogeneous Synaptic Dynamics at Natural Density." In: *New Aspects in Analyzing the Synaptic Organization of the Brain*. Ed. by Joachim H.R. Lübke and Astrid Rollenhagen. New York, NY: Springer US, pp. 277–321. ISBN: 978-1-0716-4019-7. DOI: 10.1007/978-1-0716-4019-7_15 (cit. on p. xi).

Kraynyukova, Nataliya et al. (2022). "In vivo extracellular recordings of thalamic and cortical visual responses reveal V1 connectivity rules." In: *Proc. Natl. Acad. Sci. USA* 119.41, e2207032119. DOI: 10.1073/pnas.2207032119 (cit. on pp. 107, 187).

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, 80).

Kunkel, Susanne and Wolfram Schenck (2017). "The NEST dry-run mode: Efficient dynamic analysis of neuronal network simulation code." In: *Front. Neuroinform.* 11, p. 40. ISSN: 1662-5196. DOI: 10.3389/fninf.2017.00040 (cit. on p. 15).

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. 15, 40).

Kunkel, Susanne et al. (Oct. 2014). "Spiking network simulation code for petascale computers." In: *Front. Neuroinform.* 8, p. 78. DOI: 10.3389/fninf.2014.00078 (cit. on pp. 15, 16, 40).

Kunkel, Susanne et al. (Mar. 2017). *NEST 2.12.0*. DOI: 10.5281/zenodo.259534 (cit. on p. 40).

Kurth, Anno C. et al. (2021). "Sub-realtime simulation of a neuronal network of natural density." In: *ArXiv*, 2111.04398 [cs.DC] (cit. on p. 38).

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. 16).

Kurth, Anno Christopher et al. (2024). "Cell-type specific projection patterns promote balanced activity in cortical microcircuits." In: *bioRxiv*. DOI: 10.1101/2024.10.03.616539. eprint: https://www.

biorxiv.org/content/early/2024/10/03/2024.10.03.616539.full.pdf (cit. on pp. xi, 95).

Landau, Lev Davidovich and Evgenii Mikhailovich Lifshitz (2013). *Quantum mechanics: non-relativistic theory*. Vol. 3. Elsevier (cit. on pp. 139, 197).

Levy, Robert B. and Alex D. Reyes (Apr. 2012). "Spatial profile of excitatory and inhibitory synaptic connectivity in mouse primary auditory cortex." In: *J. Neurosci.* 32.16, pp. 5609–5619. DOI: 10.1523/jneurosci.5158-11.2012 (cit. on p. 101).

Li, Xue et al. (2022). "Global, regional, and national burden of Alzheimer's disease and other dementias, 1990–2019." In: *Frontiers in Aging Neuroscience* 14, p. 937486 (cit. on p. 6).

Linssen, Charl et al. (Aug. 2018). *NEST 2.16.0*. DOI: 10.5281/zenodo.1400175 (cit. on p. 40).

Livingstone, M. S. and D. H. Hubel (Nov. 1984). "Specificity of intrinsic connections in primate primary visual cortex." In: *J. Neurosci.* 4.11, pp. 2830–2835. DOI: 10.1523/jneurosci.04-11-02830.1984 (cit. on p. 53).

Lytton, William W. et al. (Oct. 2016). "Simulation neurotechnologies for advancing brain research: parallelizing large networks in NEURON." In: *Neural Comput.* 28.10, pp. 2063–2090. DOI: 10.1162/neco_a_00876 (cit. on p. 15).

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. 6, 95, 97, 113, 125, 153, 155, 161, 173).

Malach, R. et al. (Nov. 1993). "Relationship between intrinsic connections and functional architecture revealed by optical imaging and in vivo targeted biocytin injections in primate striate cortex." In: *Proc. Natl. Acad. Sci. USA* 90, pp. 10469–10473 (cit. on p. 85).

Marchenko, Vladimir Alexandrovich and Leonid Andreevich Pastur (1967). "Distribution of eigenvalues for some sets of random matrices." In: *Matematicheskii Sbornik* 114.4, pp. 507–536 (cit. on p. 133).

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. 70, 77).

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. 55).

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. 61).

Mattson, Peter et al. (2019). *MLPerf Training Benchmark*. arXiv: 1910.01500 [cs.LG] (cit. on p. 16).

McDougal, Robert A, Anna S Bulanova, and William W Lytton (2016). "Reproducibility in computational neuroscience models and simulations." In: *IEEE Trans. Biomed. Eng.* 63.10, pp. 2021–2035. DOI: 10.1109/TBME.2016.2539602 (cit. on p. 17).

Merger, Claudia et al. (2024). "Spurious self-feedback of mean-field predictions inflates infection curves." In: *Physical Review E* 110.2, p. 024308 (cit. on p. xi).

Message Passing Interface Forum (2009). *MPI: A Message-Passing Interface Standard, Version 2.2*. Tech. rep. Knoxville, Tennessee (cit. on p. 24).

Migliore, M. et al. (2006). "Parallel network simulations with NEURON." In: *J. Comput. Neurosci.* 21.2, pp. 119–223. DOI: 10.1007/s10827-006-7949-5 (cit. on p. 15).

Miyazaki, Hiroyuki et al. (2012). "Overview of the K computer System." In: *Fujitsu Sci. Tech. J.* 48.3, pp. 255–265 (cit. on p. 34).

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. 150).

Moré, Jorge J (2006). "The Levenberg-Marquardt algorithm: implementation and theory." In: *Numerical analysis: proceedings of the biennial Conference held at Dundee, June 28–July 1, 1977*. Springer, pp. 105–116 (cit. on p. 116).

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. 15).

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. 42).

Nordlie, Eilen, Marc-Oliver Gewaltig, and Hans Ekkehard Plesser (Aug. 2009). "Towards Reproducible Descriptions of Neuronal Network Models." In: *PLOS Comput. Biol.* 5.8, e1000456. DOI: 10.1371/journal.pcbi.1000456 (cit. on p. 190).

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. 62).

OpenMP Architecture Review Board (2008). *OpenMP Application Program Interface*. http://www.openmp.org/mp-documents/spec30.pdf. Specification. Accessed: 2016-09-27 (cit. on p. 24).

Ostrau, Christoph et al. (Mar. 2020). "Benchmarking of Neuromorphic Hardware Systems." In: *Proceedings of the Neuro-Inspired Computational Elements Workshop*. Vol. 16. NICE '20. New York, NY, USA:

Association for Computing Machinery, p. 4. ISBN: 9781450377188. DOI: 10.1145/3381755.3381772 (cit. on p. 16).

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. 78–80, 101).

Park, Juyong and Mark EJ Newman (2004). "Statistical mechanics of networks." In: *Phys. Rev. E* 70.6, p. 066117 (cit. on p. 148).

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. 17).

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 p. 101).

Peters, Alan and Martin L. Feldman (1976). "The projection of the lateral geniculate nucleus to area 17 of the rat cerebral cortex. I. General description." In: *J. Neurocytol.* 5.1, pp. 63–84 (cit. on p. 97).

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. 16).

Plesser, H. E. et al. (2007). "Efficient Parallel Simulation of Large-Scale Neuronal Networks on Clusters of Multiprocessor Computers." In: *Euro-Par 2007: Parallel Processing*. Ed. by A.-M. Kermarrec, L. Bougé, and T. Priol. Vol. 4641. Lecture Notes in Computer Science. Berlin: Springer-Verlag, pp. 672–681. DOI: 10.1007/978-3-540-74466-5_71 (cit. on p. 15).

Plotnikov, Dimitri et al. (Mar. 2016). "NESTML: a modeling language for spiking neurons." In: *Modellierung 2016*. Ed. by Andreas Oberweis and Ralf Reussner. Vol. P-254. Lecture Notes in Informatics (LNI). Modellierung 2016, Karlsruhe (Germany), 17 Mar 2016 - 19 Mar 2016. Gesellschaft für Informatik e.V. (GI), pp. 93–108 (cit. on p. 118).

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. 26, 37, 50, 53, 65, 93).

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. 15, 16, 39).

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).

Ramon y Cajal, Santiago (1899). *Comparative study of the sensory areas of the human cortex*. Clark University (cit. on p. 4).

Reimann, Michael W. et al. (June 2017). "Morphological diversity strongly constrains synaptic connectivity and plasticity." In: *Cereb. Cortex* 27.9, pp. 4570–4585. DOI: 10.1093/cercor/bhx150 (cit. on p. 101).

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. 26, 36).

Ringach, Dario L, Robert M Shapley, and Michael J Hawken (2002). "Orientation selectivity in macaque V1: diversity and laminar dependence." In: *J. Neurosci.* 22.13, pp. 5639–5651. DOI: 10.1523/JNEUROSCI.22-13-05639.2002 (cit. on p. 107).

Robert, Paul and Yves Escoufier (1976). "A unifying tool for linear multivariate statistical methods: the RV-coefficient." In: *Journal of the Royal Statistical Society Series C: Applied Statistics* 25.3, pp. 257–265 (cit. on p. 127).

Rotter, Stefan and Markus Diesmann (1999). "Exact digital simulation of time-invariant linear systems with applications to neuronal modeling." In: *Biol. Cybern.* 81.5-6, pp. 381–402. DOI: 10.1007/s004220050570 (cit. on p. 118).

Rougier, Nicolas P. et al. (Dec. 2017). "Sustainable computational science: the ReScience initiative." In: *PeerJ Comput. Sci.* 3, e142. DOI: 10.7717/peerj-cs.142 (cit. on p. 17).

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 pp. 59, 82).

Sanzeni, Alessandro et al. (2020). "Inhibition stabilization is a widespread property of cortical networks." In: *eLife* 9, e54875 (cit. on p. 110).

Savica, Rodolfo et al. (Aug. 2016). "Time Trends in the Incidence of Parkinson Disease." In: *JAMA Neurology* 73.8, pp. 981–989. ISSN: 2168-6149. DOI: 10.1001/jamaneurol.2016.0947. eprint: https://jamanetwork.com/journals/jamaneurology/articlepdf/2529538/noi160029.pdf (cit. on p. 6).

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. 16).

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 p. 26).

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. 26, 35, 36, 50).

Schönemann, Peter H (1966). "A generalized solution of the orthogonal procrustes problem." In: *Psychometrika* 31.1, pp. 1–10 (cit. on p. 127).

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. 117).

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 p. 53).

Sheng, T. K. (1985). "The distance between two random points in plane regions." In: *Adv. Appl. Probab.* 17.4, pp. 748–773. DOI: 10.2307/1427086 (cit. on pp. 115, 165).

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. 58, 81, 82, 86).

Snyder, Adam C, Michael J Morais, and Matthew A Smith (2016). "Dynamics of excitatory and inhibitory networks are differentially altered by selective attention." In: *J. Neurophysiol.* 116.4, pp. 1807–1820. DOI: 10.1152/jn.00343.2016 (cit. on p. 107).

Softky, William R. and Christof Koch (1993). "The Highly Irregular Firing of Cortical Cells Is Inconsistent with Temporal Integration of Random EPSPs." In: *J. Neurosci.* 13.1, pp. 334–350. DOI: 10.1523/jneurosci.13-01-00334.1993 (cit. on p. 107).

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. 101–103, 115, 116, 185, 186).

Stettler, D.D. et al. (2002). "Lateral Connectivity and Contextual Interactions in Macaque Primary Visual Cortex." In: *Neuron* 36, pp. 739–750 (cit. on p. 59).

Stimberg, Marcel, Romain Brette, and Dan FM Goodman (Aug. 2019). "Brian 2, an intuitive and efficient neural simulator." In: *eLife* 8. DOI: 10.7554/elife.47314 (cit. on p. 15).

Stimberg, Marcel, Dan F. M. Goodman, and Thomas Nowotny (Jan. 2020). "Brian2GeNN: accelerating spiking neural network simulations with graphics hardware." In: *Sci. Rep.* 10.1. DOI: 10.1038/s41598-019-54957-7 (cit. on p. 15).

Student (1908). "The probable error of a mean." In: *Biometrika*, pp. 1–25 (cit. on p. 136).

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 p. 150).

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 p. 34).
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. xi).
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 pp. 88, 144).
Trefethen, Lloyd N and David Bau (2022). *Numerical linear algebra*. Vol. 181. Siam (cit. on p. 129).
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. 53, 61, 87, 89).
Tsodyks, M. et al. (1997). "Paradoxical effects of external modulation of inhibitory interneurons." In: *J. Neurosci.* 17.11, pp. 4382–8 (cit. on p. 110).
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. 49, 54, 171).
Van Albada, Sacha et al. (2015b). *NEST 2.2.0*. DOI: 10.5281/zenodo.5772624 (cit. on p. 40).
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. 49).
Vanni, Simo et al. (Jan. 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. 55, 66, 70, 75).
Watts, D. J. and S. H. Strogatz (1998). "Collective dynamics of small-world networks." In: *Nature* 393, pp. 440–442. DOI: 10.1038/30918 (cit. on p. 149).
Welch, Bernard L (1947). "The generalization of Student's problem when several different population varlances are involved." In: *Biometrika* 34.1-2, pp. 28–35. DOI: https://doi.org/10.1093/biomet/34.1-2.28 (cit. on p. 136).
Wimo, Anders et al. (2023). "The worldwide costs of dementia in 2019." In: *Alzheimer's & Dementia* 19.7, pp. 2865–2873 (cit. on p. 6).
Wohrer, Adrien, Mark D Humphries, and Christian K Machens (2013). "Population-wide distributions of neural activity during perceptual

decision-making." In: *Prog. Neurobiol.* 103, pp. 156–193. DOI: 10.1016/j.pneurobio.2012.09.004 (cit. on p. 107).

Yabuta, N Harumi and Edward M Callaway (1998). "Functional streams and local connections of layer 4C neurons in primary visual cortex of the macaque monkey." In: *Journal of Neuroscience* 18.22, pp. 9489–9499 (cit. on p. 57).

Yacoub, Essa, Noam Harel, and Kâmil Uğurbil (2008). "High-field fMRI unveils orientation columns in humans." In: *Proceedings of the National Academy of Sciences* 105.30, pp. 10607–10612 (cit. on p. 85).

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. 15).

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. 28).

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. 24).

Zhu, Peizhen and Andrew V Knyazev (2013). "Angles between subspaces and their tangents." In: *Journal of Numerical Mathematics* 21.4, pp. 325–340 (cit. on pp. 132, 195).