Correlations and Coding in Visual Cortex

vorgelegt von
Robert MEYER, M.Sc.
geb. in Olpe

von der Fakultät IV – Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften
— Dr. rer. nat. —

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr. Manfred OPPER
Gutachter: Prof. Dr. Klaus OBERMAYER
Gutachter: Prof. Dr. Lutz SCHIMANSKY-GEIER
Gutachter: Prof. Dr. Benjamin LINDNER

Tag der wissenschaftlichen Aussprache: 19. Juli 2016

Berlin 2016

"We're all mad here. I'm mad. You're mad."
"How do you know I'm mad?" said Alice.
"You must be," said the Cat, "or you wouldn't have come here."

Lewis Carroll, *Alice in Wonderland*

Abstract

Correlations and Coding in Visual Cortex

Understanding the neural code, that is deciphering how joint neural responses represent external stimuli, is one of the cardinal problems in Neuroscience. Since neurons are inherently noisy, the neural code is probabilistic. One key question regarding the probability distributions of neural responses is whether neural activity is correlated. Researchers observed so called noise correlations, shared variability among ensembles of neurons for repeated presentation of the same stimulus, in the visual cortex and other sensory areas. To this day the cause of these correlations remains unclear. Many hypotheses have been formulated about the origin of shared variability. In this thesis we investigate a particular hypothesis in depth. We study analytically and numerically the role of recurrent connectivity as a cause of noise correlations.

First, we introduce a novel Python library designed to support and manage numerical simulations such as spiking neuron networks. This library, called *pypet*, facilitates reproducible research by allowing the scientist to disentangle her core simulation from administrative tasks like scheduling or serialization of data. Besides being well tested and documented, the library provides a rich set of features including native multiprocessing and easy parameter exploration.

Next, we investigate analytically the relation between recurrent connectivity and correlations. We focus on a particular network topology called *Mexican hat* with shorter excitatory than inhibitory connections. Using a recent mean-field approach, we show that Mexican hat connectivity can amplify certain spatial frequencies. Moreover, we augment the model by introducing adaptation and demonstrate that the previous findings are still valid. However, because no analytical expression of the network state can be given, we reduce the model and compare it to an older approach with sinusoidal coupling. The resulting network state is a heterogeneous bump activity profile with an arbitrary phase. If noise or adaptation are incorporated, the bump starts moving along the spatial dimension of the mean-field model. Accordingly, the movement and phase changes across trials yield co-activation and co-inactivation of neighboring neurons which we hypothesize to cause noise correlations in spiking networks.

Thereafter, we test this hypothesis in a numerical experiment by simulating large networks of spiking neurons. For homogeneous input and Mexican hat connectivity, we observe the emergence of multiple moving bumps which, in turn, yield noise correlations. With an increasing distance between pairs of cells, the noise correlations are modulated sinusoidally and the amplitude decays exponentially. This holds for wide ranges of parameter settings as well as one- and two-dimensional network models. Moreover, noise correlations persist for heterogeneous stimuli, but the spatial modulation changes. Lastly, we test how the shared variability affects stimulus encoding. In general, the measured correlations decrease the stimulus encoding quality in terms of reduced Fisher information. However, if only a subset of neurons is taken into account to decode the stimulus from the spiking responses, Mexican hat networks achieve better performance than other topologies that do not produce noise correlations.

Zusammenfassung

Correlations and Coding in Visual Cortex

Ein fundamentale Problemstellung der Neurowissenschaft stellt das Entschlüsseln neuronaler Aktivität, also das Verständnis wie Neuronen gemeinsam externe Stimuli repräsentieren, dar. Neuronale Aktivität ist inhärent verrauscht, deshalb sind die Repräsentationen zwangsläufig probabilistisch. Ein wichtige Frage bezüglich der Wahrscheinlichkeitsverteilungen neuronaler Aktivität ist, ob und inwieweit diese Aktivität korreliert ist. Wissenschaftler konnten so genannte Noise-Korrelationen, korrelierte Aktivität bei wiederholter Präsentation des selben Stimulus, im visuellen Kortex und anderen sensorisches Arealen nachweisen. Bis heute ist der Ursprung dieser Korrelationen unklar. In dieser Arbeit untersuchen wir eine bestimmte Hypothese zu den Gründen der Noise-Korrelationen. Wir werden uns sowohl analytisch als auch numerisch mit der Rekurrenz als Ursache der Korrelationen auseinandersetzen.

Zunächst stellen wir eine neue Software Bibliothek vor. Das Python Programm, genannt *pypet*, unterstützt und verwaltet numerische Simulationen wie beispielsweise Netzwerke aus Spiking Neuronen. Die Software erleichtert reproduzierbare Ergebnisse indem sie Wissenschaftlern die Möglichkeit gibt den Kernteil ihrer Simulationen von administrativen Aufgaben zu trennen. *pypet* ist vollständig dokumentiert und umfassend getestet. Darüber hinaus bietet das Programm viele Funktionen wie beispielsweise automatische Parallelisierung und bequeme Parameter Explorationen.

Als nächstes befassen wir uns mit der Beziehung zwischen rekurrenten Netzwerken und Noise-Korrelationen. Wir konzentrieren uns auf eine bestimmte Topologie, genannt *Mexican Hat*, die aus länger reichweitigen inhibitorischen und kürzeren exzitatorischen Verbindungen besteht. Wir zeigen mit Hilfe eines Mean-Field Modells, dass Mexican Hat Verbindungen räumliche Frequenzen verstärken können. Außerdem erweitern wir das Model und berücksichtigen Adaptation. In diesem Fall gelten die Analysen weiterhin. Allerdings kann das Mean-Field Modell keine analytische Beschreibung der Netzwerkaktivität geben. Daher reduzieren wir das Modell und vergleichen es mit einer älteren Methodik, die auf sinusförmigen Verbindungen basiert. Der resultierende Netzwerkzustand ist eine inhomogene Antwort in Form einer Erhebung oder Beule, hiernach Bump genannt, mit einer beliebigen Phase. Unter Rauschen oder Adaptation bewegt sich der Bump in der räumlichen Ausdehnung des Netzwerks. Diese Bewegung und die Änderungen der Phase über verschiedene Trials führt zu gemeinsamer Aktivierung und Inaktivierung benachbarter Neuronen. Deshalb vermuten wir, dass die Bump Bewegung und Phasenänderungen in Netzwerken von Spiking Neuronen zu Noise-Korrelationen führen.

Anschließend testen wir diese Hypothese mit Hilfe von Simulationen großer Netzwerke mit Spiking Neuronen. Unter homogenen Netzwerkinput in Kombination mit Mexican Hat Verbindungen bilden sich multiple Bumps hervor, die sich bewegen. Die resultierenden Noise-Korrelationen weisen eine sinusförmige Modulation mit der Distanz zwischen Zellpaaren auf. Zusätzlich nimmt die Amplitude mit der Distanz ab. Noise-Korrelationen können auch für inhomogene Stimuli gemessen werden. Jedoch ändert sich die räumliche Modulation. Schlussendlich testen wir welchen Effekt die Korrelationen auf das Enkodieren von Stimuli haben. Im Allgemeinen wirken sich Korrelationen nachteilig auf die Kodierungsqualität aus und reduzieren die Fisher Information. Wird jedoch die Stichprobengröße der Zellen, die zum Dekodieren des Stimulus herangezogen werden, reduziert, eignen sich Mexican Hat Netzwerke besser als andere Topologien, die keine Noise-Korrelationen aufweisen.

Acknowledgements

First and foremost, I want to thank Prof. Klaus Obermayer, my supervisor, for his guidance and giving me the opportunity to work in his Neural Information Processing Group. He offered a great scientific environment, his feedback was always to the point, and he never imposed constraints on my work.

Moreover, many thanks to Prof. Lutz Schimansky-Geier and Prof. Benjamin Lindner for agreeing to be members of my PhD thesis committee.

I want to express my gratitude to Timm Lochmann for his help, advice, and the many discussions we had. Many thanks to Josef Ladenbauer, especially for his remarks on the analytical part of my work, and to Christian Donner for his comprehensive feedback. I also like to thank Veronika Koren and Maziar Hashemi-Nezhad for answering all my questions regarding real data and real biology. I am grateful to Robert Pröpper and Philipp Meier for their helping remarks on my Python project *pypet*. Likewise, I want to thank Mehmet Nevvaf Timur and Henri Bunting for their programming contributions to *pypet*. In addition, many thanks to Florian Aspart, Raphael Holca-Lamarre, and Caglar Cakan for their feedback on my Python toolkit. I highly appreciate the work done by Timm Lochmann, Moritz Augustin, Konstantin Mergenthaler, and Johannes Mohr regarding our server environment which I more than heavily relied on.

I am thankful to all members of the Neuro Information Processing Group and the members of the Bernstein Center for Computational Neuroscience in Berlin that made my project intellectual stimulating but also enjoyable through numerous coffee and lunch breaks, Christmas parties, Stammtische, and other social gatherings. And without Vanessa Casagrande, Margret Franke, Camilla Groiß, and Robert Martin, all the administrative tasks would have been tremendously difficult instead of a breeze.

I want to thank the numerous people who gave me feedback on my manuscript: My girlfriend Celia Sommer, her parents Petra Ebert-Sommer and Volker Sommer, my brother Thomas Meyer, Malte Lampart, Daniel König, and Daniel Wigger. Thanks to Nika and Fiona whose inputs unfortunately did not make it into the final version.

Finally, I am very grateful to my girlfriend, my friends, and especially my parents for their help, support, and encouragement when I needed it.

Contents

Introduction 1

UNDERSTANDING the neural code, that is deciphering how the brain encodes sensory stimuli, is one of the fundamental problems in Neurosciene. This thesis focuses on stimulus processing in a particular lower sensory area of the brain, the visual cortex. It is one of the most extensively studied parts of the nervous system. Therefore, for decades it has been subject to many experimental and computational studies investigating how neural activity represents the external visual world.

For instance, since the seminal work by Hubel and Wiesel (1959), we have known that most neurons in the primary visual cortex (V1) display a preference for particular stimulus features. The firing activity of simple cells is modulated by bar or grating stimuli of varying orientations. Such a stimulus, projected in a particular part of an animal's visual field, generates strong spiking activity of a neuron with a certain preferred orientation. With increasing disparity between the stimulus and this preferred orientation, the firing rate decreases.

During the past decades analysis of neural patterns has shifted from single cells to population codes, i.e. the joint responses of cell ensembles. New developments in extracellular recordings in vivo have facilitated this paradigm shift. For instance, multiple units in near vicinity can be recorded using a so called tetrode consisting of four close by micro electrodes (Harris et al., 2000). Moreover, multi-electrode arrays (MEA) can measure joint neural activity from neurons spanning areas over several square millimeters (Fejtl et al., 2006). As a matter of fact, the amount of simultaneously recorded cells doubled roughly every seven years within the last five decades (Stevenson and Kording, 2011).

It has been known for long that population codes are a reliable way to encode, transmit, and process sensory stimuli. Visual discrimination in the primary visual cortex is much more accurate than what would be predicted from single neuron responses (Paradiso, 1988). Similarly, Montijn et al. (2014) demonstrated that one can improve the performance of a variety of decoders estimating the stimulus from the joint neural responses by increasing the sample size of neurons.

This thesis is concerned with information encoded and transmitted between neurons via electrical pulses called action potentials or spikes. Hence, a response of an ensemble of neurons is fully characterized by the joint spiking activity of the cells. Moreover, we further simplify the notion of a response by ignoring exact timing of spikes. We assume that a single cell's response is specified by the number of action potentials observed within a fixed time window. Even though there is evidence that some information is encoded by precise spike timing (Rullen and Thorpe, 2001), our assumption allows estimation of statistics like correlation coefficients with considerably less data. Indeed, this coarse notion of neural responses is prominent among experimentalists (Bair et al., 2001; Rocha et al., 2007). As a consequence, comparisons of biological data to our simulation results are straightforward.

An important aspect regarding population coding is whether neural responses are correlated, especially when driven by the same stimulus. In this case one speaks of so called *noise correlations* or *shared variability* (Cohen and Kohn, 2011; Hansen et al., 2012). The variation around an expected mean response, the noise, is shared among cells. Such noise correlations have been widely observed in the visual cortex (Kohn and Smith, 2005; Martin and Schröder, 2013). Moreover, researchers reported that the magnitude of pairwise noise correlations decays with the distance between cell pairs (Smith and Kohn, 2008; Solomon et al., 2014). The magnitude and spatial structure of noise correlations are particularly interesting because these can have a considerable influence on stimulus processing. The correlations can potentially enhance or worsen the quality of a population code (Averbeck et al., 2006). Accordingly, shared variability can help, but also harm the brain's ability to discriminate between different or, rather, very similar but not equal stimuli.

Yet, it is still unclear what phenomena are causing the correlations. Hansen et al. (2012) hypothesized recurrent connectivity as one of the potential sources. They argue that axonal connections between cells residing in the same layer of the primary visual cortex lead to joint neural dynamics with noise correlations. This is a reasonable assumption considering that the vast majority of connections does not come from the sensory periphery or higher cortical areas, but originates within the visual cortex and projects locally (Markov et al., 2011).

We further explore this hypothesis on two counts. Firstly, we investigate mean-field descriptions and, secondly, we explore the dynamics of recurrent network models of spiking neurons. Indeed, much analytical as well as numerical work was done in the past (Brunel, 2000; Vogels et al., 2005; Augustin et al., 2013). In these studies researchers focused on activity patterns that can emerge in randomly connected networks without a real notion of spatial extent. Moreover, special attention was drawn to the emergence of asynchronous irregular (AI) activity in such networks (Brunel, 2000; Kumar et al., 2008). AI states are characterized by individual neurons firing in an irregular manner which leads to stationary, asynchronous global behavior. If patterns of neural activity are uncorrelated due to asynchronous firing, population coding properties are

relatively well known. For example, we have an understanding about which factors control the amount of information retained in uncorrelated population activity (Seung and Sompolinsky, 1993; Pouget et al., 2000), or how downstream networks can be constructed to optimally read out information from the population response (Deneve et al., 1999).

However, the assumption of random connectivity in spiking neuron networks limits the application of model results to cortical phenomena. For instance, the connection probability of cortex neurons decreases with the distance between pairs of excitatory pyramidal cells as well as inhibitory interneurons (Hellwig, 2000; Mariño et al., 2005; Wang, 2010). Therefore, in this thesis we are going to investigate the dynamics that arise in recurrent networks based on biologically more plausible connectivity profiles with local connections. Accordingly, we aim to provide a mechanistic explanation of how shared variability can arise in the visual cortex. In addition, the correlations' impact on stimulus processing is of interest. In particular, we are going to address the following questions:

- How can local recurrent connectivity cause noise correlations? What role do parameters like the coupling strengths and the spread of the local connections play in the emergence and spatial modulation of noise correlations?
- Do noise correlations in recurrent networks affect the encoding quality of stimuli in joint neural responses?

1.1 Outline

In order to answer the questions above we performed extensive parameter explorations with our network models. These explorations and the corresponding hundreds of gigabytes of numerical results imposed a challenge to our data management capacities. Accordingly, we developed a novel data management toolkit called *pypet* to adequately govern our network simulations. This novel framework will be introduced in chapter 2 besides brief presentations of different use cases.

Next, in chapter 3 we are going to review the literature on noise correlations. We will discuss the spatial and temporal scales that have been observed in animal experiments. Furthermore, hypotheses about potential causes of shared variability will be debated and we will relate noise correlations to the stimulus encoding quality of a population response.

Chapter 4 theoretically examines the consequences of local recurrent connectivity. Using mean-field models by Rosenbaum and Doiron (2014) and Hansel and Sompolinsky (1998) we will convey that a particular network topology with shorter excitatory than inhibitory connections, called Mexican hat, constitutes a potential source for noise correlations.

In chapter 5 we are going to present the results of our spiking neuron network simulations. We will verify the analytical predictions from the previous chapter and demonstrate how Mexican hat networks can cause noise correlations. Moreover, we will investigate the influence of shared variability on the stimulus information in joint neural responses.

Finally, chapter 6 summarizes the findings and we will put these into perspective for potential future lines of research.

Additionally, appendix A provides the details of our spiking neuron network models and appendix B presents some preliminary results.

pypet: A Python Toolkit for Parameter Exploration

2

THIS chapter introduces *pypet*, an open source software toolkit to manage numerical experiments in Python. At the beginning it started as a minor project to handle parameters and results of network simulations which we prepared and executed. Over the course of this dissertation *pypet* has grown into a mature software application with a rich set of features. The program is now used among scientists and engineers of different research groups and in the industry to guide and support their simulations and numerical experiments.

This chapter is mostly based on the freely available online documentation (`http://pypet.readthedocs.org`) also written by the thesis author. After an introduction and the motivation behind *pypet*, we are going to give an overview on other existing software packages and how these relate to *pypet*. Next, we will provide a brief explanation about *pypet*'s development, architecture, and program layout. Finally, some use case examples will be presented. However, a full description of all available features and all full demonstration of all of *pypet*'s capabilities is beyond the scope of this manuscript and the user is directed to the comprehensive online documentation.

2.1 Introduction and Motivation

Computational Neuroscience is a rapidly evolving scientific field. Researchers create a lot of highly complex computational models with great levels of biological details, for instance work by Reimann et al. (2013) or Potjans and Diesmann (2014). The increasing availability and decreasing costs of high-performance computing (HPC) have further facilitated this development. However, the Computational Neuroscience community lacks a standard in software tools and languages that are used to create models. Although many approaches and packages have been developed to tackle this problem, as discussed in the next section, many researchers rely on custom made software and their own data formats. Scripts and results of simulations are at most shared among researchers within the same group if ever shared with anyone. This makes numerical

experiments hard to reproduce. There is an ongoing debate about mandatory publication of source code in scientific research. Some researchers pledge to make this a necessity rather than optional (Ince et al., 2012).

Still, even the open availability of software does not guarantee reproducibility of results. For instance, if software is not well-documented and maintained, it is unlikely that other researchers are able to apply it. Even granted that they are willing to inspect the source code, just to get a basic understanding of existing but ill-documented code, researchers may have to invest an unacceptable amount of time to reimplement someone else's published computational model (Topalidou et al., 2015). Furthermore, simulations of neural phenomena are usually highly parametrized and encompass tens or even several hundreds of parameters. For example, the simulations in chapter 5 are relying on more than forty parameters. As a consequence, reproducing simulation results becomes difficult if the used parameter values are not tightly linked to the obtained results. Along these lines, Victoria Stodden (2011) even speaks about a "credibility crisis" of computational results. This is not only a problem for published numerical experiments, but difficulties can arise in the day-to-day work of scientists in the process of evolving simulations and research projects. If researchers try to replicate results of their own that were obtained just a few months ago, they already may fail to do so because they have no knowledge anymore about the full set of parameters of their previous simulations. Of course, the development of a scientific project, especially the early stages, is usually hidden from the public and only comes to light with a successful publication. Accordingly, there is no data available quantifying and measuring the struggle of researchers when it comes to data management of computational studies. Yet, the thesis author knows about a lot of such cases from personal communication with other scientists.

pypet is designed to fill this gap. The two main goals of the software package are, firstly, to allow easy but flexible exploration of large parameter spaces and, secondly, tightly link used parameters and obtained simulation results.

Accordingly, *pypet* stands for **Py**thon **P**arameter **E**xploration **T**oolkit. It is targeted to researchers and engineers executing numerical experiments of any kind, not only related to Neuroscience. It can support simulations written in the platform-independent and widely used programming language Python. Python is gaining more and more attention in Computational Neuroscience and it is likely to overtake MATLAB as the language of choice in the natural sciences (Koepke, 2010; Muenchen, 2012; Coelho, 2013).

With *pypet* the user can explore arbitrary parameter spaces simply by specifying Python lists of parameter points. In turn, these points define individual simulation runs and lead to particular numerical results. Tight linkage of parameters and results is achieved by storing all data together in the convenient HDF5 format (The HDF Group, 1996).

2.1.1 Main Features

Among the two main objectives of easy parameter exploration and tight linkage of parameters and results, *pypet* encompasses a variety of other features some of these are listed below:

- Novel tree container `Trajectory`, for handling and managing of parameters and results of numerical simulations
- Grouping of parameters and results into meaningful categories
- Accessing data via natural naming, e.g. `traj.parameters.neuron.gL`
- Support for many different data formats
  - Python native data types: `bool`, `int`, `long`, `float`, `str`, `complex`
  - Python containers: `list`, `tuple`, `dict`
  - NumPy arrays and matrices (Walt et al., 2011)
  - SciPy sparse matrices (Jones et al., 2001)
  - Pandas `Series`, `DataFrame`, and `Panel` (McKinney, 2011)
  - BRIAN and BRIAN2 quantities and monitors (Goodman and Brette, 2008; Stimberg et al., 2014)
- Easily extendable to other data formats
- Merging of trajectories residing in the same space
- Support for multiprocessing, *pypet* can run simulations in parallel
- Analyzing data on-the-fly during multiprocessing
- Adaptively exploring the parameter space combining *pypet* with optimization tools like the evolutionary algorithms framework DEAP (Fortin et al., 2012)
- Dynamic loading of parts of data one currently needs
- Resuming a crashed or halted simulation
- Annotation of parameters, results, and groups
- Git integration, *pypet* can make automatic commits of one's codebase
- Sumatra integration, *pypet* can automatically add one's simulations to the electronic lab notebook tool Sumatra (Davison, 2012)
- *pypet* can be used on computing clusters or multiple servers at once if it is combined with the SCOOP framework (Hold-Geoffroy et al., 2014)

Of note, *pypet*'s BRIAN2 sub-package is joint work with Henri Bunting as part of the Neural Networks Project course at the TU Berlin. In addition, some components of the SCOOP integration were programmed by Mehmet Nevvaf Timur.

2.2 Related Software

In recent years a couple of software projects dealing with data management in Computational Neuroscience have been developed. In this section we are going to discuss the most prominent approaches and emphasize their similarities and differences to *pypet*. Some of these programs can even be combined with *pypet* for advanced usage and enhanced management capabilities like grid or cluster computing.

NeuroManager (Stockton and Santamaria, 2015) facilitates automated scheduling of simulations with heterogeneous computational resources. Such computational resources can range from simply using the host computer — from which scheduling was started — to a network of other computers or even clusters and computer grids. NeuroManager, written in object-oriented MATLAB, allows the user to specify simulations in terms of pure MATLAB code or MATLAB code wrapping existing simulators like NEURON (Carnevale and Hines, 2006). The parameter space defined by the simulators can be explored using NeuroManager's scheduling routine by utilizing heterogenous computing resources; granted these resources support the needed software requirements like MATLAB licenses. Unfortunately, NeuroManager requires a license of the MATLAB Compiler Toolbox which is not necessarily part of a MATLAB standard environment. For instance, the Compiler Toolbox is not found in the student version of MATLAB. Hence, despite buying the regular MATLAB license, further costs may be associated with the NeuroManger requirements. In contrast, all of *pypet*'s requirements are open source and freely available.

Mozaik (Antolík and Davison, 2013) is a Python data management toolkit especially designed for network simulations of two-dimensional neural sheets. It relies on the simulator environment PyNN (Davison, 2008). Its design goals are similar to *pypet*'s. Mozaik aims on integrating parameters and model descriptions with the simulator execution as well as the storage of results. However, the focus on two-dimensional networks makes it less flexible in comparison to *pypet*.

Lancet (Stevens et al., 2013) constitutes a more general approach to workflow management and integrates with IPython notebooks (Perez and Granger, 2007). Like *pypet*, Lancet is simulator agnostic. It even allows to interact with other programs not written in Python as long as these can be launched as a process and return output in form of files. Lancet is a well-designed alternative to *pypet*, especially for smaller projects that fit into the scope of a single IPython notebook. Nonetheless, Lancet's documentation is not as comprehensive as *pypet*'s and lacks detailed usage descriptions. The authors assume that users are willing to grasp and understand many ideas and use cases by directly inspecting Lancet's source code (Stevens et al., 2016).

Moreover, VisTrails (Bavoil et al., 2005) is a workflow and provenance management system written in Python that focuses on automation and standardization of visualizations. It is mainly operated through a graphical user interface (GUI) and targets on an audience less akin to programming. In comparison, *pypet* offers no GUI, but it is a

Python library that users can import and use in their own source code to write scripts and programs.

The primary goal of Sumatra (Davison, 2012) is to enhance reproducible research. Sumatra serves as an electronic lab notebook. The command line program does not only link all simulation parameters to obtained results, but also keeps track of the entire computing platform. It stores information like the used operating system and particular versions of software dependencies. Sumatra can be nicely integrated with *pypet* to automatically trigger a Sumatra record with every simulation start.

SAGA-Python (Merzky et al., 2015) is a software package that provides an application programming interface (API) for distributed computing. Users can submit and monitor jobs on computing cluster environments from a remote machine. They may even schedule automatic retrieval of output files. The *pypet* online documentation gives examples and explains in detail how to combine *pypet* with SAGA-Python.[1]

Lastly, SCOOP (Hold-Geoffroy et al., 2014) provides a Python library to seamlessly schedule parallel computing jobs on heterogeneous resources like computer grids or multiple servers. *pypet* is designed to integrate well with SCOOP to allow massive parameter explorations on cluster environments. We will briefly sketch the usage of *pypet* and SCOOP at the end of this chapter.

2.3 Software Architecture and Development

In this section we will discuss general design principles of *pypet* and briefly layout the architecture and structure of the Python package. First, we are going to start with *pypet*'s packaging and adhesion to the concept of test driven development. Next, we will present our conceptualization of parameter explorations. Furthermore, we are going to introduce the general layout followed by more detailed description of the individual components. Lastly, we will finish this chapter with some general use case examples.

2.3.1 Software Packaging and Testing

pypet is a pure Python *package*[2] and supports Python versions 2.6, 2.7, 3.3, 3.4, and 3.5. It is platform independent and runs under Linux, Windows, and OS X with 32-bit as well as 64-bit architectures. To allow further extensions of *pypet* effortlessly, the package is very modularized and *pypet* is designed following the concept of object oriented programming[3] (OOP).

[1] `http://pypet.readthedocs.org/en/latest/examplesdoc/example_22.html`

[2] In the Python universe the term *package* denotes a bundle of *modules*. Modules are specialized Python libraries that may contain related functions, objects, and general Python statements.

[3] In OOP user interactions involve so-called objects. These objects are special data structures that provide data in terms of object attributes. In addition, the object's methods grant functionality to modify the object and data therein.

Furthermore, the source code is openly available and hosted on the prominent *github*[4] code sharing platform. In addition, *pypet* is bundled on the Python Package index[5] (PyPI) to allow fast and easy installation using the package managing system *pip*.

Besides comprehensive documentation, it is very important for software packages — scientific ones in particular — that all functionality is well tested (Gewaltig and Cannon, 2014). Therefore, *pypet* is designed using test driven development. Accordingly, small features and single functions are already accompanied with corresponding test cases. In addition, we apply continuous integration testing. Every addition of new code triggers a full battery of package wide tests which are automatically started and deployed on independent build servers. *pypet* is tested using the services Travis-CI[6] with a Linux environment and AppVeyor[7] providing Windows servers. Every time a new code addition is pushed to the code repository on github, the unit and integration tests are automatically deployed by Travis-CI and AppVeyor. This guarantees that new features do not break existing functionality. In addition to continuous integration testing, we use the coveralls[8] web service to quantify how comprehensive the test suite is and how many relevant code lines are covered by the test battery. As of February 2016, *pypet*'s core modules encompass almost 10,000 lines of pure Python code of which more than 90% are hit by the test battery exceeding 900 tests. As a side remark, including the test suite, all documentation, and all example scripts the number of code lines increases more than five-fold.

2.3.2 Parameter Exploration and Conceptualization

pypet's goals are to allow for easy parameter exploration and storage of results as well as parameters side by side. Our definition of a parameter exploration is as follows: It is the process of sampling an n-dimensional parameter space of a simulation or model implementation with a pre-defined set of points within the space. Running the simulation or model independently with each point in the parameter space produces further data that is considered to be results. The dimensions of the parameter space can be heterogeneous, i.e. these may encompass integers, real values, or even less mathematical concepts like Python tuples, for instance. Therefore, we also refer to a dimension of the parameter space simply as a "parameter".

Moreover, we assume that from the n-dimensional space usually only a much smaller sub-space is sampled of size n' with $n' \ll n$. Accordingly, most parameters are fixed and only a minority are varied and explored. For instance, the network models in chapter 5 are based on several tens of parameters, but we only varied very few of these comprehensively (c.f. table A.1 in the appendix).

[4] `https://github.com/SmokinCaterpillar/pypet`
[5] `https://pypi.python.org/pypi/pypet`
[6] `https://travis-ci.org/SmokinCaterpillar/pypet`
[7] `https://ci.appveyor.com/project/SmokinCaterpillar/pypet`
[8] `https://coveralls.io/github/SmokinCaterpillar/pypet`

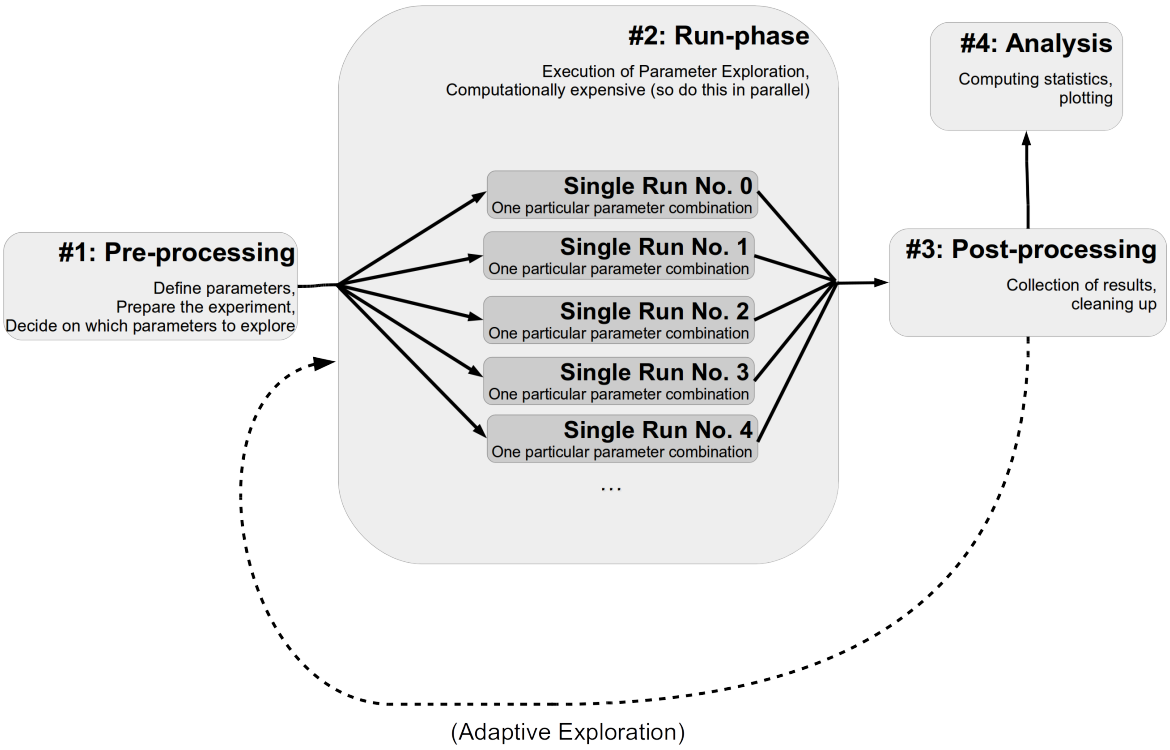

Figure 2.1: Conceptualization of a simulation or numerical experiment.

Furthermore, the set of points is sequentially ordered. The order may be arbitrary, but it is fixed such that the ith point in the parameter space corresponds to the ith run of a simulation or model. Due to the order, one may not just think of sampling the parameter space, but rather following a discrete trajectory through the space. Accordingly, the top-level container managing all parameters and results is called `Trajectory`.

Next, we will briefly discuss a particular layout of simulations that fits best with *pypet*. This conceptualization is also sketched in figure 2.1. We assume that numerical experiments or simulations usually comprise between two to four different stages:

1. **Pre-processing**: Parameter definition, preparation of the experiment
2. **Run phase**: Fan-out structure, usually parallel running of different parameter settings, gathering of individual results for each single run
3. **Post-processing** (optional): Cleaning up of the experiment, sorting results, etc.
4. **Analysis of results** (optional): Plotting, calculating statistics, etc.

The first pre-processing stage can be further divided into two sub-stages. In the beginning the definition of parameters is given and, secondly, one's experiment is initialized and configured. The definition of parameters also involves the decision how the parameter space is explored. Configuration and initialization might encompass creating particular Python objects or pre-computing some expensive functions that otherwise would be computed redundantly in every run in the next phase.

The second stage, the run phase, is the actual execution of one's numerical simulation. All different points in the parameter space that have been specified before for exploration are tested on the model. As a consequence, one obtains corresponding results for all parameter combinations. Since this stage is most likely the computationally expensive one, one probably wants to parallelize the simulations. We refer to an individual simulation execution with one particular parameter combination as a **single run**. Because such single runs are different individual simulation executions with different parameter settings, they are completely independent of each other. The results and outcomes of one single run should not influence another. This does not mean that non-independent runs cannot be handled by *pypet*, they can. However, keeping single runs independent greatly facilitates the parallelization of their execution.

Thirdly, after all individual single runs are completed one might perform post-processing. This could involve merging or collection of results of individual single runs or deleting some sensitive Python objects. In case one desires an adaptive or iterative exploration of the parameter space, one could restart the second phase. In this case the `Trajectory` can be extended. The user can iteratively add some more points of the parameter space and alternate the run phase and post-processing before terminating the experiment.

Finally, one may desire to do further analysis of the raw results obtained in the previous phases. This constitutes the final stage of an experiment and may include the generation of plots or calculation of statistics. For a strict separation of experimental raw data from its statistical analysis, one is advised to separate this final phase from the previous three. Thus, this could mean starting a completely different Python script than for the phases before.

2.3.3 General Package Structure

pypet encompasses five key modules. The `trajectory.py` module contains the `Trajectory` class that constitutes the main data container the user interacts with. User requests to a `Trajectory` are passed onto and processed by a service called `NaturalNamingInterface` residing in the `naturalnaming.py` module. Moreover, the `Trajectory` allows the arbitrary exploration of the parameter space and manages all data including parameters, results, as well as configuration specifications. All of these are further encapsulated by their own abstract containers which can be found in the `parameter.py` module. In case data is stored to disk, this is handled by the `HDF5StorageService` located in the `storageserivce.py` module. Currently, the data is saved in the HDF5 format (The HDF Group, 1996). Storage and loading of trajectories follow well-defined APIs. Hence, the implementation of other backends, like SQL for example, is possible without the need to change any other *pypet* core code. Finally, the `environment.py` module provides the so called `Environment` object for handling the running of simulations. This general structure of the *pypet* package is sketched in figure 2.2. Next, we will discuss the above introduced components in more detail.

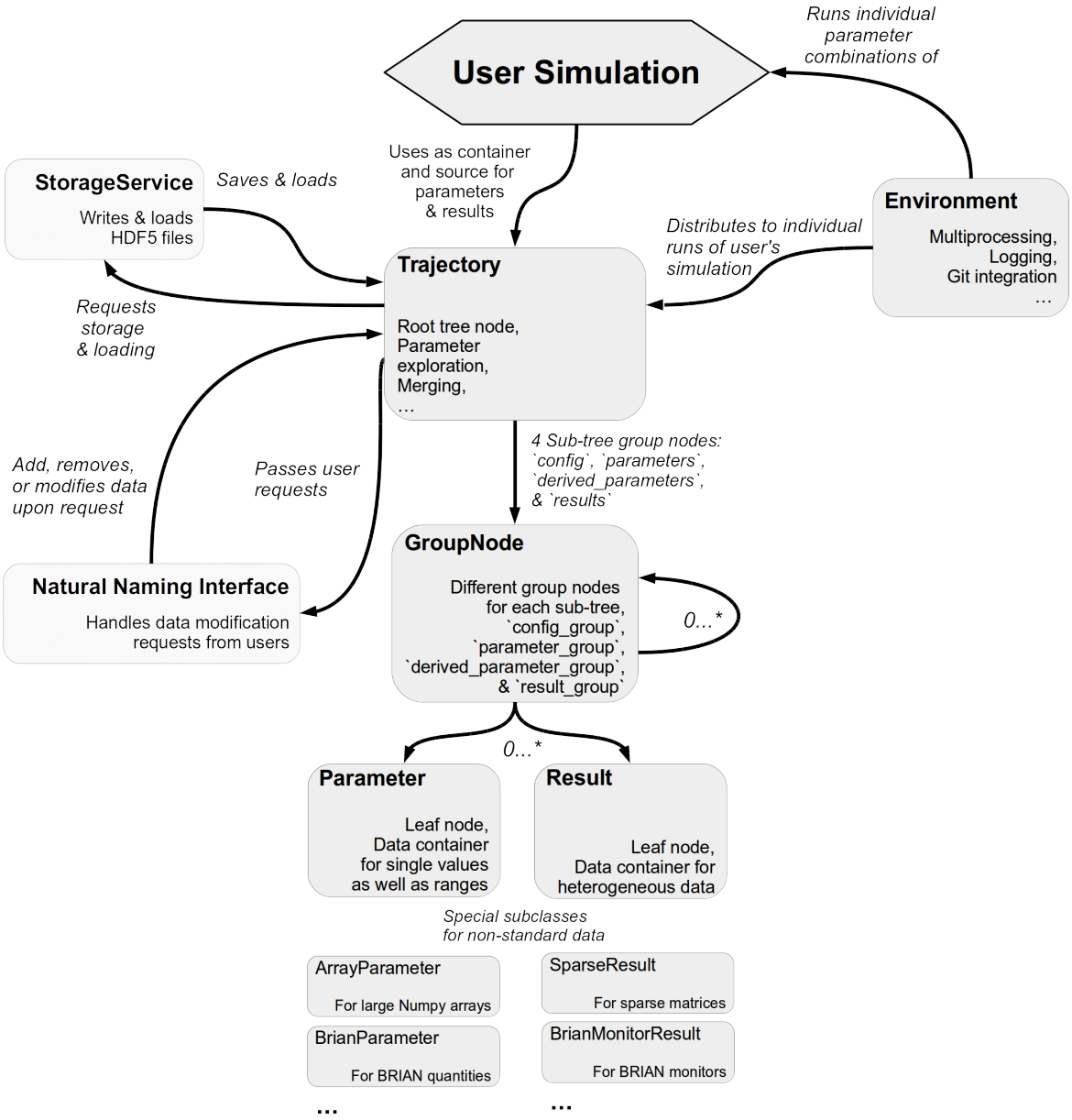

Figure 2.2: Main components of *pypet*. Elements in light grey mark objects that operate in the background with no direct user interaction.

2.3.4 Parameters and Results

The `parameter.py` module provides the so called `Parameter` class that follows a base API called `BaseParameter`. The `Parameter` contains data that is explicitly required as parameters for one's simulations. For the rest of the thesis we follow the convention that the stylized `Parameter` denotes the abstract container. The not stylized expression "parameter" refers to the user data that is managed by the `Parameter` container. In fact, `Parameter` is the name of the Python class constructor. Accordingly, instantiated `Parameter` objects are referred to by variable names in lower case letters, like `myparameter`, for example. This notation holds analogously for user results and variables like `myresult` referring to an instantiated container of the `Result` class which implements the base API `BaseResult`.

`Parameter`s follow two main principles. Firstly, a key concept in numerical experiments is the exploration of the parameter space. Therefore, the `Parameter` containers do not only manage a single value or data item, but they may also take a range of data items. Elements of such a range typically reside in the same dimension, i.e. only integers, only strings, only NumPy arrays, etc. The exploration is initiated via the `Trajectory`. This functionality will be introduced shortly. Secondly, a `Parameter` can be **locked**; meaning as soon as the `Parameter` container is assigned to hold a specific value or data item and the value or data item has already been used somewhere, it cannot be changed any longer (except after being explicitly unlocked). This prevents the cumbersome error of having a particular parameter value at the beginning of a simulation but changing it during runtime by accident. Such an error can be very difficult to track down. By definition parameters are fixed values that are defined before any simulation run. An exception to this rule is solely the exploration of the parameter space, but this requires to run a number of distinct simulations anyway. Hence, explored `Parameter`s are fixed and locked within each individual run.

`Parameter` containers accept a variety of different data items, these are

- Python natives (`int`, `str`, `bool`, `float`, `complex`),
- NumPy natives, arrays and matrices of type `np.int8` to `np.int64`, `np.uint8` to `np.uint64`, `np.float32`, `np.float64`, `np.complex`, and `np.str`
- Python homogeneous non-nested tuples and lists

For more complex data, there are specialized version of the `Parameter` container. For instance, the `SparseParameter` is a container for SciPy sparse matrices and the `BrianParameter` can manage quantities of the BRIAN simulator package (Goodman and Brette, 2008).

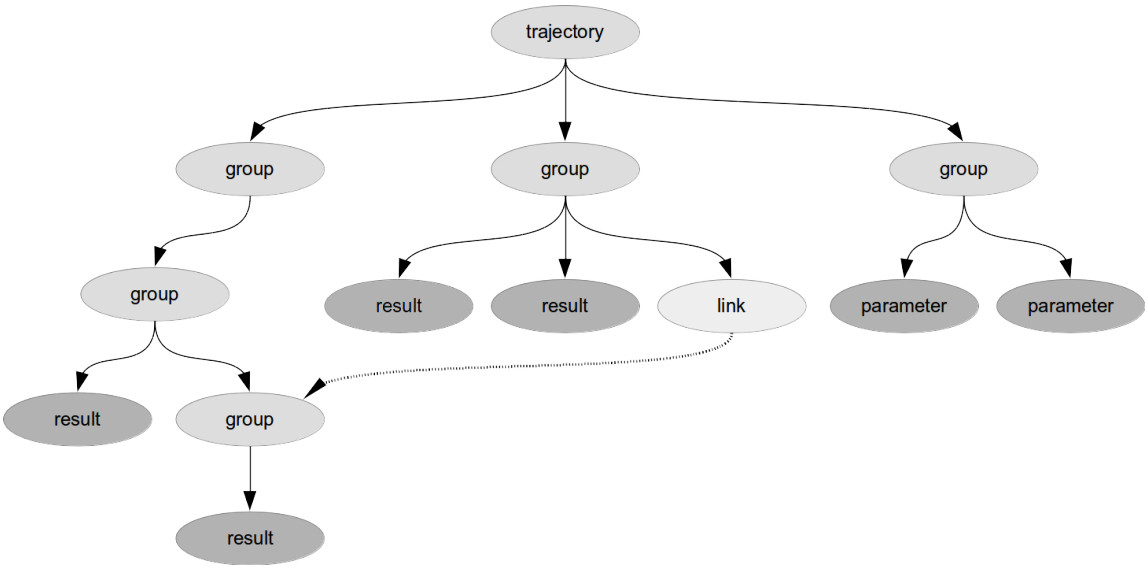

Figure 2.3: Depiction of the data tree.

Moreover, `Result` containers are less restrictive than `Parameter`s in terms of data they accept. They can also handle Python dictionaries and Pandas data frames (McKinney, 2011). In addition, they allow to add more than a single data item (or an exploration range). Individual data items can be assigned names and recalled. For instance, the following command

```
>>> myresult = Result('test', a=42, b='forty-two')
```

creates a `Result` named `'test'` that is referred to by `myresult`. It contains the integer and string data `42` and `'forty-two'`, respectively. Accordingly, these data items can be accessed via `myresult['a']` and `myresult['b']` or via `myresult.a` and `myresult.b`.

Similar to the `Parameter`, there exist specialized versions of a `Result`, like a `SparseResult`. In case the user relies on some custom data that is not supported by the `Result`, `Parameter`, or their specialized descendants containers, the user can implement a custom solution. Customized containers are straightforward and only need to follow the API specifications given by `BaseResult` and `BaseParameter`.

2.3.5 Trajectory and Natural Naming

The `Trajectory` is the container for all results and parameters of one's numerical experiments. The `Trajectory` instantiates a tree with **groups** and **leaf nodes**. The instantiated `Trajectory` object itself is the root node of the tree. The leaf nodes encapsulate the user data and are the `Parameter` and `Result` containers. Group nodes cannot contain user data directly, but may contain other groups and leaf nodes. This is sketched in figure 2.3. By using only groups and leaves there cannot be any cycles within the trajectory tree. However, one can introduce **links** that refer to other existing group or leaf nodes.

Furthermore, we implemented a general naming convention that applies to the `Trajectory`, all groups, and all containers that can encapsulate user data, i.e. the `Result` and `Parameter` introduced before. To avoid confusion with the natural naming scheme and the functionality provided by the `Trajectory`, we followed the idea by another software package *pypet* is based on, called PyTables (Alted et al., 2002), to use prefixes. That is `f_` for methods and functions and `v_` for Python variables, attributes, and properties.

For instance, given a particular instantiated `Result` denoted by the variable `myresult`, `myresult.v_comment` refers to the object's comment attribute and `myresult.f_set(mydata=42)` is the function for adding data to the `Result` container. Whereas `myresult.mydata` can be a data item named `mydata` provided by the user.

Results can be added to the `Trajectory` tree at any time. Parameters can only be introduced before the individual simulation runs are started. Assuming an instantiated `Trajectory` object denoted by the variable `traj`, a parameter can be added via

```
>>> traj.f_add_parameter('subgroup1.subgroup2.myparam', 42,
        comment='I am an example')
```

where `'subgroup1.subgroup2.myparam'` is the full name of the parameter. This will automatically add the groups `subgroup1` and `subgroup2` on the fly if they do not exist, yet. Here `42` is the user's standard or default value of the `Parameter`. Additionally, one can always provide a descriptive comment to inform potential other users or researchers about the parameter's scope and meaning. Results are added analogously. Still, because a `Result` can handle heterogeneous data, one may provide additional keyword arguments that denominate each data item:

```
>>> traj.f_add_result('anothergroup1.anothergroup2.myresult',
        a=42, b='forty_two', c=44.4,
        comment='I am an example result')
```

There exist some more ways to add data to the tree. For a detailed explanation the reader is directed to the online documentation.

Data can be accessed using *natural naming*, i.e. the user can rely on "." notation to retrieve data. For instance, given the result from above, one may recall the data via

```
>>> traj.anothergroup1.anothergroup2.myresult.c
```

which will return `44.4`. Internally these request are not handled by the `Trajectory` class, but are forwarded to the `NaturalNamingInterface`.

Furthermore, links allow alternative paths through the tree. For instance, given the following link addition

```
>>> traj.anothergroup1.f_add_link('resultlink',
        traj.anothergroup1.anothergroup2.myresult)
```

this allows access to the result data also with

```
>>> traj.anothergroup1.resultlink.c
44.4
```

The user is encouraged to group and structure results as fine-grained as possible. To spare the user an excessive amount of typing, the `Trajectory` supports so called *shortcuts*. If the user leaves out intermediate groups in the natural naming request, a breadth first search is applied by the `NaturalNamingInterface` to find the corresponding group or leaf:

```
>>> traj.myresult.b
'forty_two'
```

Search is established by a very fast look up and usually needs much less time than the complexity of $O(N)$, where N is the total number nodes in the tree. Due to internal look up tables the corresponding node is most often found in $O(1)$ or sometimes in $O(d)$, with d the depth of the tree. Given that shortcuts are not unique, that is one could potentially find several solutions for a natural naming search in the tree, *pypet* will return the first item it finds via breadth first search. If there are several items with the same name but in different depths, the one with the lowest depth is returned. For performance reasons, *pypet* actually stops the search if an item was found and there are no other items within the tree with the same name and same depth. If there are two or more items with the same name within the same depth, the user request is considered ambiguous and *pypet* will raise a `NotUniqueNodeError`.

Exploration can be prepared with the `Trjectory`'s `f_explore()` method. This function takes a dictionary with parameter names as keys and lists specifying the parameter ranges as values. Note that all lists need to be of the same length. For example:

```
>>> traj.f_add_parameter('x', 0)
>>> traj.f_add_parameter('y', 0)
>>> traj.f_explore({'x':[42, 44, 45, 46], 'y' :[1, 4, 6, 6]})
```

These statements produce a `Trajectory` of length 4 and explore the four parameter space points $(42, 1)$, $(44, 4)$, $(45, 6)$, and $(46, 6)$. Additionally, the Cartesian product of two ranges can be automatically created using a helper function called `cartesian_product()`:

```
>>> traj.f_explore(cartesian_product(
    {'x':[42, 44, 45, 46],
     'y' :[1, 4, 6, 7,8, 9, 10]}))
```

In this case the list lengths may differ because the `cartesian_product()` function will return lists with appropriately matching lengths.

2.3.6 Data Storage and Loading

Storage and loading of the `Trajectory` container and all its content are not carried out by the `Trajectory` itself but by a service in the background. Currently, all data is stored into a single HDF5 file (The HDF Group, 1996) via the `HDF5StorageService`. To interface HDF5, the storage services uses the excellent PyTables library (Alted et al., 2002).

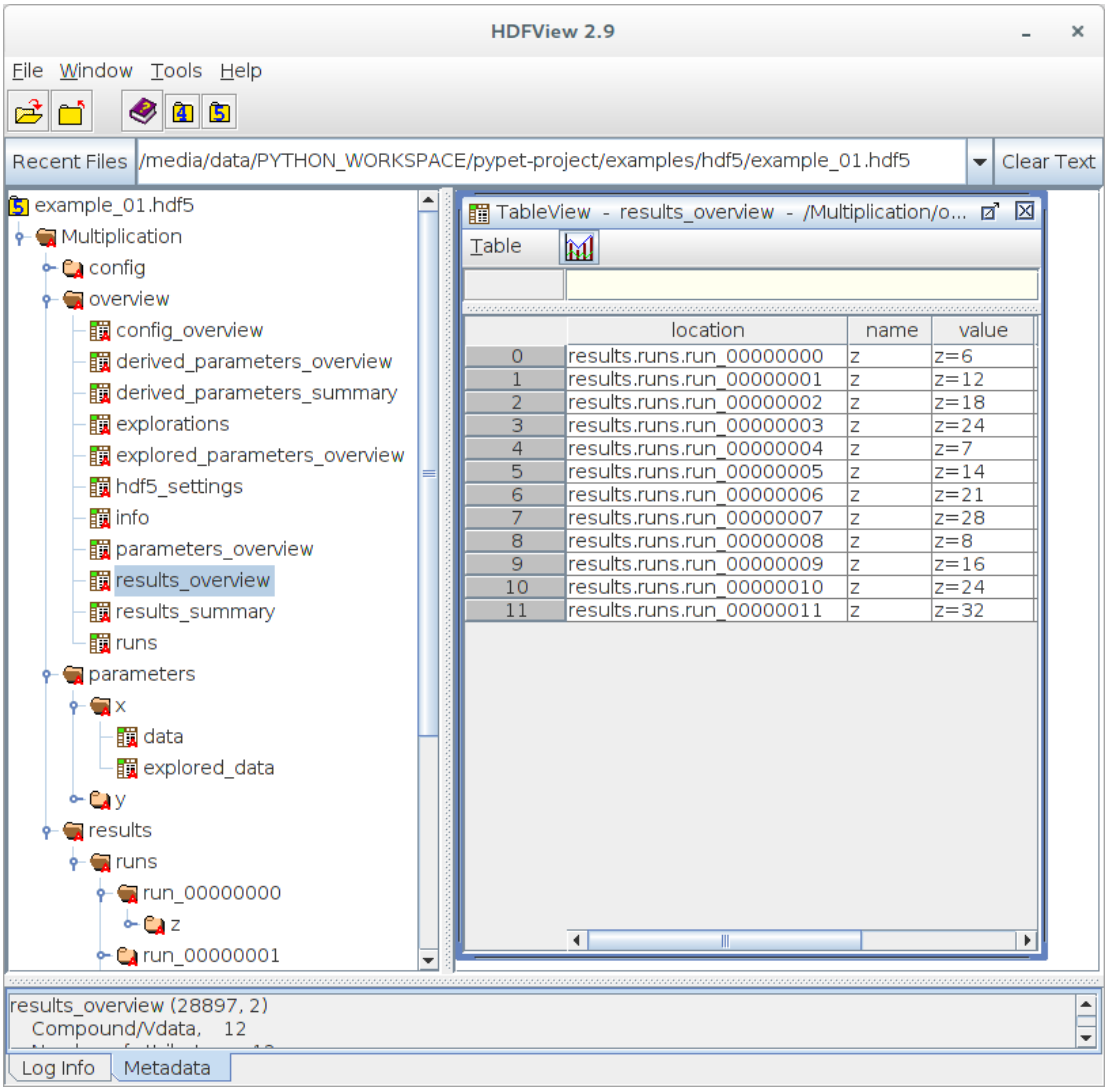

Figure 2.4: Inspection of a strored `Trajectory` in an HDF5 file using the HDFView tool (The HDF Group, 1996).

The acronym HDF5 stands for the fifth version of the *Hierarchical Data Format*. It is a very convenient format because it allows compressed reading and writing of data to the hard disk with high performance. More important, as its name suggests, data is ordered in hierarchies that are very similar to the file and folder structure of most operating systems. The data hierarchies and the numerical data therein can directly be inspected with tools like *HDFView*. Not surprisingly, the tree structure of the `Trajectory` is mapped one to one to the hierarchical structure in the HDF5 file. For example, figure 2.4 depicts the inspection of a stored *pypet* `Trajectory`.

Usually, the storage of a `Trajectory` is automatically triggered by *pypet* in regular intervals. Additionally, the user can manually initiate storing using the `Trajectroy`'s `f_store()` method. Loading is analogous via `f_load()`. One can specify to load parts of the tree or only specific data items. Detailed advice is provided in the online documentation. In addition, *pypet* supports automatic loading of data as soon as the user

needs it. No explicit loading is necessary and data is recovered from the HDF5 file on-the-fly. Automatic loading will be demonstrated in one of the usage examples at the end of this chapter.

2.3.7 Environment

The `Environment` defines a scheduler for the numerical experiments. It constitutes a general framework in which the user can embed her simulations. In principle, the `Environemnt` allows the user to disentangle the core simulation from administrative tasks like distribution and repeated execution of runs and data serialization. The `Environment` takes care about setting up new `Trajectory`s, keeps log files, or can be used for parallelization of one's simulations onto several CPUs.

Usually, the user starts simulations by creating an `Environment` object:

```
>>> env = Environment(trajectory='mytrajectory',
                      comment='A useful comment',
                      multiproc=True, ncores=4,
                      git_repository='~/myproject')
```

There are plenty of customizations possible that can be passed as keyword arguments to the `Environment`. For instance, *pypet* natively supports parallelization using the Python multiprocessing library. As shown above, to run all simulation runs on 4 cores at the same time, one can use the `multiproc` and `ncores` keywords. Moreover, *pypet* integrates nicely with code bases under git version control. If the path to the main folder of the user's git repository is passed to the `Environment` via the `git_repository` keyword, *pypet* automatically triggers a git commit if it finds changes in the code base.[9] The toolkit remembers the commit's SHA identifier. Consequently, the user can always recall the exact version of a simulation with which particular results were obtained.

Furthermore, the `Environment` provides a Trajectory container for the user:

```
>>> traj = env.traj
```

Next, the user can fill the `traj` container with parameters and specify the exploration of the parameter space. Afterwards, she can start the individual simulation runs via

```
>>> env.run(mysimulation)
```

where `mysimulation` is the top-level simulation function that takes a `Trajectory` container as the first argument. Accordingly, the function definition requires the following form:

```
def mysimulation(traj, ...):
    # Do some sophisticated simulation
```

where `traj` is the `Trajectory` container. The function is allowed to take other positional as well as keyword arguments. The user is advised that the top-level simulation

[9] Note that this feature requires the GitPython package (Trier et al., 2016).

function `mysimulation` should not return any results, but put them directly into the `Trajectory` container using the `traj.f_add_result()` method.

Upon calling `env.run(mysimulation)`, the `Environment` invokes `mysimulation` for every point in the parameter space independently. In case of long running simulations or many runs, the `Environment` notifies the user about the progress and gives an estimate of the remaining time in regular intervals. Furthermore, the `Environment` will automatically trigger the storage of obtained results after every individual simulation run. In addition, it monitors the simulation execution in terms of keeping log-files. By default, general log-files are accompanied by specialized error logs. The latter display only error messages to allow easier identification and debugging in case there are errors in the user's simulation.

2.4 Usage

So far we have introduced *pypet*'s main components and sketched their functionality. In this section we will give three brief usage examples. But first, we are going to demonstrate how easily *pypet* can be installed. Secondly, we will provide a very basic example to allow the reader getting used to the general structure of a simulation managed by *pypet*. Thirdly, a more comprehensive use case will be examined. We are going to simulate the theoretical model by Hansel and Sompolinsky (1998). Note that this model will be discussed in great detail in the chapter 4. It will be shown how *pypet* can be wrapped around an existing simulation code. Additionally, we are going to demonstrate how stored simulation data can be loaded for further analyses and plotting. Finally, it will be explained how simulations can be scheduled and executed on a computing cluster environment using *pypet* in combination with SCOOP.

2.4.1 Installation

Because *pypet* is a pure Python package, its installation is straightforward and does not require more involved steps like compilation of source code. If the Python package manager *pip* is available,[10] one can simply install *pypet* from the command line:

```
$ pip install pypet
```

Alternatively, one can download *pypet* from the PyPI[11] web page, unpack it, and run

```
$ python setup.py install
```

in a terminal.

Note that *pypet*'s four core prerequisites are NumPy, SciPy, PyTables, and Pandas. These are standard libraries in scientific Python and have most likely been installed

[10] From Python version 3.4 on, *pip* is part of the Python distribution and is no longer needed to be installed manually.

[11] `https://pypi.python.org/pypi/pypet`

already on most computer systems. For a fresh Python environment, however, one needs to install these before setting up *pypet*.

2.4.2 Basic Example

Here we are going to describe a very basic usage example. We will simulate the multiplication of two values, i.e. $z = x \cdot y$.
Before discussing the details of the simulation, we provide the full script below for an overview:

```
from pypet import Environment, cartesian_product

def multiply(traj):
    """Simulation that involves multiplying two values."""
    z = traj.x * traj.y
    traj.f_add_result('z', z)

# Create an environment that handles running our simulation
env = Environment()
# Get the trajectory from the environment
traj = env.traj
# Add both parameters
traj.f_add_parameter('x', 1.0, comment='First dimension')
traj.f_add_parameter('y', 1.0, comment='Second dimension')
# Explore the parameters with a Cartesian product
traj.f_explore(cartesian_product({'x': [1.0,2.0,3.0,4.0],
                                  'y': [6.0,7.0,8.0]}))
# Run the simulation with all parameter combinations
env.run(multiply)
```

The top-level simulation function `multiply` is defined as

```
def multiply(traj):
    """Simulation that involves multiplying two values."""
    z = traj.x * traj.y
    traj.f_add_result('z', z)
```

The function makes use of a `Trajectory` container `traj` which manages our parameters. Because `multiply` is invoked with all points in the parameter space, here `traj` holds a particular choice of x and y. We can access the parameters simply by natural naming, as seen above via `traj.x` and `traj.y`. Next, the value of `z` is added as a result to the `traj` container.

After the definition of the job that we want to simulate, we create an `Environment` denoted by `env` that runs the simulation. Moreover, the `Environment` will take care that the function `multiply` is called with each choice of parameters once. We do not pass any arguments and simply use *pypet*'s default settings for an `Environment`:

```
env = Environment()
```

The `Environment` will automatically generate a `Trajectory` for us which we can access via the `env.traj` property. Next, we can populate the container with the parameters. We add them using default values $x = y = 1.0$:

```
# Add both parameters
traj.f_add_parameter('x', 1.0, comment='First dimension')
traj.f_add_parameter('y', 1.0, comment='Second dimension')
```

Afterwards, we decide upon how to explore the parameter space. More precisely, we are interested in the Cartesian product set $\{1.0, 2.0, 3.0, 4.0\} \times \{6.0, 7.0, 8.0\}$. Therefore, we use `f_explore()` in combination with the builder function `cartesian_product()`:

```
# Explore the parameters with a Cartesian product
traj.f_explore(cartesian_product({'x': [1.0,2.0,3.0,4.0],
                                  'y': [6.0,7.0,8.0]}))
```

Finally, we need to tell the `Environment` to run our job `multiply` with all parameter combinations:

```
env.run(multiply)
```

This will invoke our simulation twelve times with the parameter points $(1.0, 6.0), (2.0, 6.0), ..., (4.0, 8.0)$. The `Trajectory` and all results are automatically stored into an HDF5 file; similar to what is shown in figure 2.4.

2.4.3 Rate Model Simulation

We are going to demonstrate how to use *pypet* in the context of a more sophisticated simulation. We will numerically integrate the following differential equation modeling an ensemble of coupled neurons (Hansel and Sompolinsky, 1998):

$$\frac{d\nu_i(t)}{dt} = -\nu_i(t) + \sqrt{N}\left[\sum_{k=1}^{N} \nu_k(t)\left(j_0 + j_1\, cos(2\pi\,(k-i)/N)\right) + j_A\right]_+, \qquad (2.1)$$

where ν_i is the activity of the ith neuron, N the number of neurons, j_A denotes afferent input and j_0 as well as j_1 are the recurrent coupling strengths. Moreover, $[\cdot]_+$ means rectification, i.e. $[x]_+ = x$ if $x > 0$ else 0. The model will be discussed extensively in chapter 4. Here the model simply serves as an example of a numerical simulation and its details are not important. We will use Euler integration to simulate the development of ν_i over time.

The corresponding Python implementation is given below:

```
def rate_model(j0, j1, jA, N, dt, steps):
    """Rate model by Hansel and Sompolinsky, 1998"""
    # Random initial conditions
    v = np.zeros((N, steps))
    v[:, 0] = np.random.rand(N)
    # Coupling Kernel
    x = np.linspace(0, 1, N, endpoint=False)
    kernel = j0 + j1 * np.cos( 2 * np.pi * x)
    # Iterative update of DiffEq
    for irun in range(1, steps):
        # Current network state
        v_curr = v[:, irun-1]
        # Mean current using circular convolution
        mu = jA + np.convolve(np.hstack((v_curr, v_curr[:-1])),
                              kernel, mode='valid')
        # Rectification
        mu[mu < 0] = 0
        # Euler Update
        dv = -v_curr + np.sqrt(N) * mu
        v_new = v_curr + dv*dt
        v[:, irun] = v_new
    return v[:,-1]
```

Besides the parameters of the differential equation, the model relies on `dt`, the Euler stepsize with a fixed amount of `steps`. The details of the implementation do not matter. The important part here is that the function `rate_model()` returns the full state vector ν of the last time step.

Given such existing simulator functionality that takes parameters and returns a result, *pypet* can be added to operate on top of the code base. One simply needs a wrapper function that passes parameters from and results back to *pypet*:

```
def pypet_model_wrapper(traj):
    v = rate_model(traj.j0, traj.j1, traj.jA,
                   traj.N, traj.dt, traj.steps)
    traj.f_add_result('v', v)
```

Still, some boiler-plate code is missing to add parameters, decide what to explore, and start the simulation:

```
import numpy as np
from pypet import Environment
# Create the environment
env = Environment(trajectory='rate_model',
                  filename='./HDF5/rate_model.hdf5')
traj = env.traj
# Add parameters
traj.f_add_parameter('j0', -0.02, comment='Global weight')
traj.f_add_parameter('j1', -0.01, comment='Cosine weight')
traj.f_add_parameter('jA', 10, comment='Afferent input')
traj.f_add_parameter('N', 500, comment='Number of neurons')
traj.f_add_parameter('dt', 0.01, comment='Euler stepsize')
traj.f_add_parameter('steps', 5000, comment='Euler steps')
# Explore the cosine strength
traj.f_explore({'j1': [-0.01, 0.005, 0.01, 0.02]})
# Run all (4) simulations
env.run(pypet_model_wrapper)
```

In contrast to the previous example, we passed `trajectory='rate_model'` and `filename='./HDF5/rate_model.hdf5'` to the `Environment` constructor to explicitly specify the `Trajectory`'s name and the resulting HDF5 file.

Next, we want to plot the obtained results. According to the conceptualization discussed previously, we assume that this is done in a different script and it is independently executed from the previous simulation. The full script reads:

```
import matplotlib.pyplot as plt
from pypet import load_trajectory
# Load trajecrory wihtout data
traj = load_trajectory(filename='./HDF5/rate_model.hdf5',
                       name='rate_model',
                       load_all=0)
traj.v_auto_load = True
# Plot all runs
for run_name in traj.f_iter_runs():
    # Load data on-the-fly
    j1 = traj.parameters.j1
    v = traj.results.runs.crun.v
    plt.plot(v)
```

We use the `load_trajectroy()` function to recover the container from the HDF5 file. Note the keyword `load_all=0` which enforces *pypet* to only load the root node of the tree and skip the rest of the data. This is particularly useful if our data is large, potentially hundreds of gigabytes. Thus, we do not load all data on start-up, but only when we need it; hence the statement `traj.v_auto_load = True` in line 7. This allows loading of data on-the-fly without explicit user request.

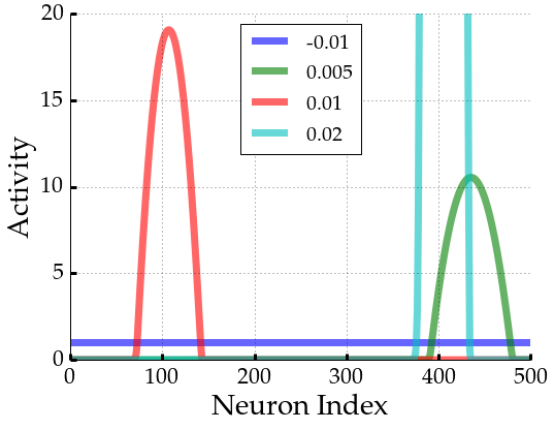

Figure 2.5: Activity ν_i for different coupling strength j_1. For $j_1 = -0.01$ one observes a flat profile. Setting $j_1 = 0.005$ or 0.01 yields a bump profile. Positions of the bumps are arbitrary and depend on the initial conditions. Too strong coupling $j_1 = 0.02$ yields runaway excitation and for some neurons firing diverges. Accordingly, the activity of the cyan bump grows unboundedly over time.

Moreover, the method `traj.f_iter_runs()` (line 9) iterates all runs (here 4) sequentially and modifies all explored parameters accordingly (here only `j1`). The explored parameter `j1` is iteratively set to its explored value of the corresponding run. This is helpful for natural naming requests which will return the value of the current run iteration. Consequently, `traj.parameters.j1` (line 11) will return -0.01 in the first loop iteration, followed by 0.005, 0.01, and 0.02.

This applies analogously to the statement `traj.results.runs.crun.v` (line 12) to return the state vector ν of each run. Due to `traj.v_auto_load = True` (line 7), there is no explicit loading with `f_load()` necessary, but *pypet* loads the network states ν in the background as soon as the natural naming request `traj.results.runs.crun.v` is processed. In addition, one may notice the identifier `crun`, short for *current run*. All results added via `f_add_result()` are automatically sorted into the `Trajectory` tree in the branch *results.runs.run_X*, where X is the index of the corresponding run. In combination with `f_iter_runs()`, `crun` maps always to the run currently processed within the for-loop. For instance, in the first iteration this is the 0th run, *run_0* (*pypet* starts counting at 0), followed by *run_1* and so on.

As a side remark, instead of using `f_iter_runs()`, one can manually set a `Trajectory` to a particular run via `traj.v_idx = 2`, for example. As a consequence, all explored parameters are set to the values of the second run and `crun` maps to *run_2*. For undoing this and to recover the default settings, one writes `traj.v_idx = -1`.

Finally, if we add labels and polish the resulting plot (code not shown), our rate model simulation yields the interesting curves in figure 2.5.

2.4.4 *pypet* and SCOOP

In combination with SCOOP package, *pypet* can be used on computing clusters as well as multiple servers sharing a home directory.
One simply needs to create an environment as follows:

```
env = Environment(multiproc=True,
                  use_scoop=True
                  wrap_mode='LOCAL')
```

and start the main Python script with the `-m scoop` option:

```
$ python -m scoop mysimulation.py
```

No other changes of one's simulation code are required to use a simulation with SCOOP. Moreover, the `wrap_mode` defines how data storage is synchronized among the computational resources. This is necessary because PyTables does not allow concurrent access of HDF5 files. Available modes are `'LOCAL'`, `'NETLOCK'`, and `'NETQUEUE'`. In the former case all data is collected from the SCOOP workers and stored by the main Python process running the `mysimulation.py` script. `'NETLOCK'` means locks are shared across the computer network to allow only one process at a time to write data to disk. Lastly, in the `'NETQUEUE'` mode an additional server process is started that accepts data via the TCP protocol. This process internally manages a storage queue from which data is popped and subsequently stored to disk. This mode is a contribution by our former research group intern Mehmet Nevvaf Timur.

One can easily use *pypet* in a multi-server or cluster framework. If one has access to multiple servers sharing the same home directory, one can distribute the simulation runs by starting the Python simulation script using SCOOP's `--hostfile` flag. For instance,

```
$ python -m scoop --hostfile hosts.txt -vv -n 16 mysimulation.py
```

starts 16 workers on particular resources as listed in `hosts.txt`. This is a simple text file specifying the available servers. It has the following format:

```
some_host 10
130.148.250.11
another_host 4
```

The file contains names of hosts or, alternatively, IP addresses followed by an optional number of SCOOP workers one wants to launch on the corresponding server.

To use *pypet* and SCOOP on a computing cluster, one additionally needs a bash start-up script. For instance, for a sun grid engine (SGE), the bash script might look like the following:

```
#!/bin/bash
#$ -l h_rt=3600
#$ -N mysimulation
#$ -pe mp 4
#$ -cwd

# Launch the simulation with SCOOP
python -m scoop -vv mysimulation.py
```

Most important is the `-pe` parallel environment flag to let the computer grid and SCOOP know how many workers to spawn (here 4). Other options may encompass parameters like `-l h_rt` defining the maximum runtime, `-N` assigning a name, or `-cwd` using the current directory as the working directory. The particular options depend on the cluster environment and requirements of the grid provider. This job script, for example named `mybash.sh`, can be submitted via

```
$ qsub mybash.sh
```

Accordingly, the simulation `mysimulation.py` gets queued and eventually executed in parallel on the computer grid as soon as resources are available.

For many more usage examples and detailed explanations the reader is directed to the *pypet* and SCOOP online documentations.

2.5 Summary

This chapter described *pypet*, a flexible Python tool to manage numerical experiments and simulations. *pypet* has a rich set of features and its main objectives are easy exploration of high dimensional parameter spaces and fostering ties between the parameters and simulation results. *pypet* provides a novel container called `Trajectroy` that governs all parameters and results. The data handled by a `Trajectory` is automatically stored to disk in the convenient HDF5 format. The tree structure of the `Trajectroy` maps one-to-one to the data hierarchy in a HDF5 file. In addition, *pypet*'s `Environment` forms a general framework for simulations. It schedules individual runs of the user's experiments and can be used to parallelize simulations using multiple CPUs. In combination with SCOOP *pypet* can also distribute simulation runs among multiple servers or a computer grid engine. In conclusion, by supporting data management via various features and by tightly linking numerical results and the underlying parameters, *pypet* enhances reproducible research in Computational Neuroscience and other disciplines exercising computer simulations in Python.

Correlations and Coding in Visual Cortex

3

In this chapter different concepts of correlations in the context of Neuroscience will be reviewed. It will be discussed how these correlations can be quantified and what scales and patterns of correlations are reported in experiments. Furthermore, it will be shown how correlations affect information processing. More precisely, we are going to discuss how stimulus information among spiking activity of neuron populations is modulated by correlations.

Deciphering the neural code, i.e. understanding how the brain encodes sensory stimuli, is one of the fundamental problems in Neuroscience. No doubt, decoding of neural activity poses a difficult challenge, especially because neuron responses are inherently noisy. Repeating the same experiment twice, one would most certainly never encounter the same activity pattern. As a consequence, the neural code is probabilistic.

Thus, understanding the neural code requires an understanding of the joint probability distributions of neural activity $p(\boldsymbol{r})$, where $\boldsymbol{r} = (r_1, r_2, ..., r_N)^T$ is the neural response vector. Unfortunately, the exact definition of a neural response r_i is unclear because we do not have a complete understanding about all mechanisms that govern information transmission in the brain. Are action potentials a sufficient source of information? Does exact spike timing matter? Can we coarsen our measure without losing information, like binning spikes, i.e. sorting spike events into discrete time intervals? How small do the bins need to be? Hence, the definition of a neural response r_i already includes a lot of assumptions about the underlying neural code (Latham and Roudi, 2011). The two most widely used ones are binning spikes on millisecond scales or assuming that spike counts within a fixed time window matter. The particular choice depends heavily on the research question at hand and time scales of interest. As we will see in the next sections, many experimenters use the latter, especially when they are interested in effects happening on longer time scales such as within a full trial of stimulus presentation. Additionally, coarser measures most often also require less data for estimation of their statistical properties. Consequently, usually r_i denotes the spike count within a fixed time window of a particular neuron indexed by i. As already mentioned in the introduction, this definition will also be used throughout this thesis.

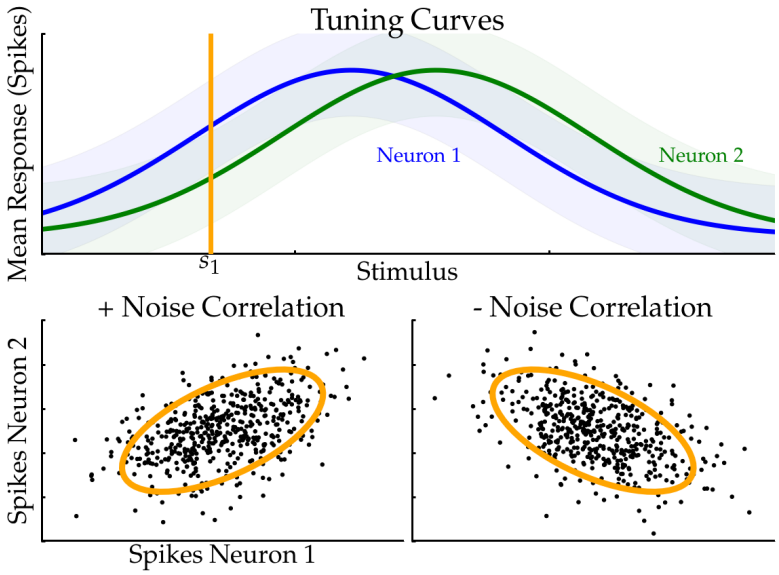

Figure 3.1: Tuning curve plus noise model. Top graph shows average responses of two neurons as a function of a one-dimensional stimulus. Envelopes depict the individual neuron's variability around the mean response. Bottom graphs show potential positive and negative noise correlations for a fixed stimulus s_1. Black dots represent spike responses of both neurons in different trials. Orange ellipses depict 95% confidence intervals. Image based on Averbeck et al. (2006).

Certainly, one key question regarding the neural code is whether neural responses are correlated. One can distinguish between two types of correlations, signal and noise correlations (Nirenberg and Latham, 2003).

Neural responses $\boldsymbol{r} = (r_1, r_2, ..., r_N)^T$ of N different neurons are signal correlated if and only if

$$p(\boldsymbol{r}) \neq \prod_{i=1}^{N} p(r_i), \tag{3.1}$$

i.e. the joint distribution of responses $p(\boldsymbol{r})$ does not simply factorize into the marginal response distributions. Moreover, neurons are said to be noise correlated given a fixed stimulus s if and only if

$$p(\boldsymbol{r}|s) \neq \prod_{i=1}^{N} p(r_i|s), \tag{3.2}$$

that is the joint distribution conditioned on s does not factorize into the marginal distributions.

The former distribution $p(\boldsymbol{r})$ implicitly incorporates all potential stimuli that could be used to drive the neuron ensemble. Thus, as soon as neurons show enhanced or reduced responses following a particular stimulus or stimulus feature, they are signal correlated. For instance, since the seminal work by Hubel and Wiesel (1959) it has been

known that neurons in the visual cortex are tuned to features like the orientation of a grating stimulus.

Noise correlations can be basically understood as lying on top of the signal correlations. A prominent depiction is the so called *tuning curve plus noise* model (Averbeck et al., 2006). Average neuron firing activity can be described by tuning curve functions; for example, the orientation tuned neurons in primary visual cortex as mentioned before. Due to variability, observed responses across different trials will vary around this average, i.e. on top of the average response there exists noise. If the neurons are noise correlated, they share a certain amount of variation around the mean. Thus, noise correlations are also often termed *shared* or *correlated variability* (Hansen et al., 2012). Figure 3.1 sketches a pair of linearly noise correlated neurons and their tuning functions. Note that we use the terms correlations, noise correlations, and shared or correlated variability interchangeably. In case we refer to signal correlations, we explicitly say so.

Noise correlations can affect the stimulus encoding quality of a population code. The effects of correlation on stimulus processing will be discussed at the end of this chapter. But first, we will focus on how noise correlations can be measured and what is usually reported in biology.

3.1 Measuring Noise Correlations

In this section we are going to introduce and discuss several statistical measures of correlations in the context of Neuroscience.

As previously mentioned, neurons are said to be noise correlated in case the joint response distribution $p(\boldsymbol{r}|s)$ cannot be factorized into the marginal response distributions $p(r_i|s)$. However, estimating full joint response distributions $p(\boldsymbol{r}|s)$ is notoriously hard endeavor. Doing this for already a few neurons requires a lot of data, i.e. many repeated trials of the same experiment. With growing numbers of neurons one faces a combinatorial explosion of potential response patterns which makes accurate estimates of $p(\boldsymbol{r}|s)$ intractable. To overcome this problem researchers may parametrize the distribution for the sake of estimating fewer parameters. Alternatively, they may look at simpler forms of correlation, like only up to second order (linear). Linear correlation can be measured without full knowledge about $p(\boldsymbol{r}|s)$.

Nonetheless, there is evidence that higher order correlations beyond linear pairwise ones play a role in cortical activity (Ohiorhenuan et al., 2010; Yu et al., 2011). One approach to quantify and model such higher order neural interactions is using copulas (Berkes et al., 2008; Onken et al., 2012). Copulas are multivariate distributions based on a parametrized linkage of the marginal distributions. Hence, $p(\boldsymbol{r}|s)$ may be expressed in terms of $p(r_i|s)$ and a few copula parameters. As a consequence, less amount of data is required because only the copula parameters and the individual marginals $p(r_i|s)$ need to be estimated; or these may even be parametrized, too. The number of parameters depends on the particular copula family.

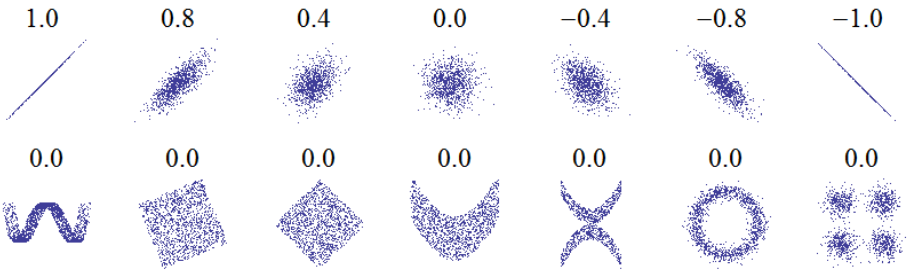

Figure 3.2: Correlation coefficients for different distributions of the data. Top row shows well suited data and the corresponding correlation coefficient. Bottom row shows complex relationships which cannot be captured by the measure. Image taken from Wikipedia (`en.wikipedia.org/wiki/File:Correlation_examples2.svg`, accessed 26.10.2015)

Another prominent approach is based on maximum entropy models (Schneidman et al., 2006; Shlens et al., 2006; Tang et al., 2008; Ganmor et al., 2011). Researchers can fix the maximum order of interactions K they want to use in their model and estimate all interaction parameters up to this limit based on the data. One can fit an exponential distribution by maximizing entropy subject to the observed data. The maximum entropy distribution is of the following form:

$$p_K(x_1, x_2, ..., x_N) = \frac{1}{Z} \exp\left(\sum_i h_i\, x_i + \frac{1}{2}\sum_{i\neq j} h_{ij}\, x_i\, x_j + ...\right), \tag{3.3}$$

where Z is a normalization factor and the kth order moments $h_{ij...z}$ are estimated from the data. Note that here x_i are binned spike responses in very short time windows, usually on the order of a millisecond. x_i is either 1 if the neuron spiked at that particular time or -1 otherwise. Usually, the moments $h_{ij...z}$ are cut off after the second or third order due to the combinatorial growth of parameters. This approach also offers the advantage that one can directly assess if second or third order terms are relevant.

Yet, most of the experimental publications report linear correlations, for example see Gawne et al. (1996), Bair et al. (2001), and Gutnisky and Dragoi (2008). By looking only at linear correlations one can easily quantify the degree of shared variability among pairs of neurons. A popular measure is the spike count correlation coefficient r_{SC} for a fixed stimulus s:

$$\begin{aligned} r_{SC} &= \frac{\sum_{k=1}^{n}(r_i^k - \langle r_i\rangle)(r_j^k - \langle r_j\rangle)}{\sqrt{\sum_{k=1}^{n}(r_i^k - \langle r_i\rangle)^2}\sqrt{\sum_{k=1}^{n}(r_j^k - \langle r_j\rangle)^2}} \\ &= \frac{\mathrm{Cov}(r_i, r_j)}{\mathrm{Std}(r_i)\ \mathrm{Std}(r_j)}, \end{aligned} \tag{3.4}$$

with r_i^k the number of spikes of neuron i in trial k over a fixed time window while presenting s, and $\langle r_i\rangle$ denotes the average spike count of neuron i across trials. The correlation coefficient is expressed as the ratio between the covariance ($\mathrm{Cov}(r_i, r_j)$) and

the product of the spike count standard deviations ($\mathrm{Std}(r_i)\ \mathrm{Std}(r_j)$). Typically, r_{SC} is measured for a full trial length of one stimulus presentation to capture correlations on long time scales. Since r_{SC} is a stochastic variable, either distributions are reported or values are averaged across many cell pairs. Moreover, r_{SC} ranges from -1, perfect negative correlation, to 1, perfect positive correlation. Of course, this measure can only capture linear dependencies and may fail to uncover more complex relationships as sketched in figure 3.2.

r_{SC} is not necessarily required to be computed over trials. If data is stationary — which we can assure for simulations — the number of spikes r_i^k may be computed using a sliding window over a single or a few long trials (Renart et al., 2010) via

$$r_i^k = \Lambda_T * x_i(k) = \sum_{k'} \Lambda_T(k' - k)\, x_i(k), \tag{3.5}$$

where $*$ denotes discrete temporal convolution, Λ_T is a normalized square kernel of window size T, and $x_i(k)$ is the binned spike train of neuron i at position k. $x_i(k)$ is either 1 or 0 if the neurons spiked within the time bin or not, respectively. Nonetheless, if simulation time is too short this may yield a biased estimate of r_{SC}. If the trajectory of an attractor state is potentially large, it might have been observed only partially and not randomly sampled as compared to measuring r_{SC} over trials.

Furthermore, noise correlations are often estimated from the cross-correlograms (CCG) of experimental recordings (Kohn and Smith, 2005; Smith and Kohn, 2008). This measure is denoted by r_{CCG} and was originally introduced by Bair et al. (2001). It allows the investigation of different time scales of linear correlations. Cross-correlograms are computed as

$$CCG(\tau) = \frac{\frac{1}{M}\sum_{i=1}^{M}\sum_{t=1}^{H} x_1^i(t)x_2^i(t+\tau)}{\Theta(\tau)\sqrt{\nu_1\nu_2}}, \tag{3.6}$$

with M the number of trials with H bins each, x_j^i is the binned spike train of neuron j in trial i, τ is the time lag, and ν_j is the mean firing rate of neuron j. $\Theta(\tau)$ is a triangular function that corrects for overlap:

$$\Theta(\tau) = T - |\tau|, \tag{3.7}$$

with T the trial length. The CCGs are usually corrected for signal correlations induced by the stimulus by subtracting a shift predictor. The shift predictor is computed as the CCG in equation 3.6, but correlating x_1^i with x_2^{i+1} from the following trial. Moreover, auto-correlograms (ACG) are also computed as in equation 3.6, but correlating x_1^i with

x_1^i from the same trial and same neuron. The shared variability measure r_{CCG} for a fixed stimulus s is then defined as

$$r_{CCG}(t) = \frac{\sum_{\tau=-t}^{t} CCG_{\text{corrected}}(\tau)}{\sqrt{\left(\sum_{\tau=-t}^{t} ACG1(\tau)\right)\left(\sum_{\tau=-t}^{t} ACG2(\tau)\right)}}, \tag{3.8}$$

where t is the size of the integration time window. If one considers the whole trial length T, it holds $r_{CCG}(T) = r_{SC}$ (Bair et al., 2001). Thus, for stationary data $r_{CCG}(t)$ is equal to the r_{SC} computed with a sliding window of size $t \leq T$.

3.2 Spatial and Temporal Scales of Noise Correlations

In the previous section several correlation measures were introduced. Here we will review what values have been observed in experimental data. Typically, measures are based on in vivo animal recordings. An awake or anesthetized animal is shown repeated trials of the same stimulus while researchers record with devices like tetrodes or multi-electrode arrays from ensembles of cells in the animal's cortex (Smith and Kohn, 2008; Hansen et al., 2012). The raw recordings are processed and spike responses of individual units are identified via a process called spike sorting (Lewicki, 1998). Note the term *unit* is used instead of cell or neuron to indicate that mistakes in sorting are possible. Problems that may arise due to suboptimal spike sorting will be discussed in the following section. Finally, the sorted unit activities are used to compute statistical measures like r_{SC} and r_{CCG} (Kohn and Smith, 2005).

In terms of r_{SC} and r_{CCG} throughout the literature weak positive noise correlations on the order between 0.1 and 0.4 are reported for nearby cells with similar response properties (Zohary et al., 1994; Lee et al., 1998; Huang and Lisberger, 2009). This holds across species, for instance in cat (Martin and Schröder, 2013), mouse (Montijn et al., 2014), or monkey (Constantinidis and Goldman-Rakic, 2002); as well as across different lower and higher visual areas, like in V1 (Gutnisky and Dragoi, 2008), V2 (Smith et al., 2013), or middle temporal (MT) visual area (Bair et al., 2001). Although many studies reported noise correlations in the visual cortex, their existence is still under debate. In particular, Ecker et al. (2010) found almost no noise correlations in the visual cortex of awake monkeys. They measured average r_{SC} values of only 0.02 for nearby cell pairs. Accordingly, the authors argued that the commonly observed positive correlations may be attributed to the experimental setup and artifacts rather than to neural phenomena. Potential pitfalls of measuring correlations and prominent hypothesis about neural causes of noise correlations will be discussed in the subsequent section.

A decay of correlations with distance between cell pairs is often reported in the literature. One finds different magnitudes of decay. For example, Kohn and Smith (2005) and Smith and Kohn (2008) measured a decay lasting over several millimeters in primary visual cortex of monkeys, see figure 3.3A. Whereas Solomon et al. (2014)

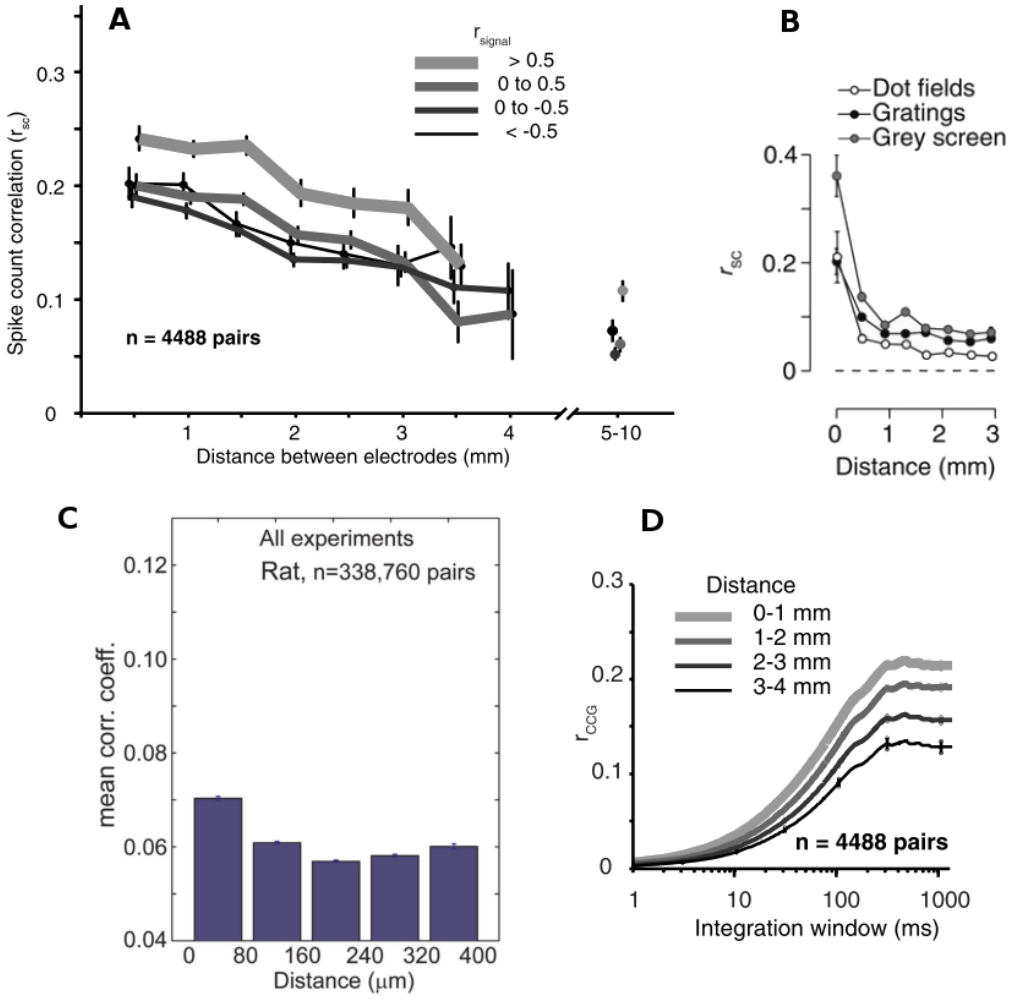

Figure 3.3: Top left (A): Spatial scales of r_{SC} measured in primary visual cortex (V1) of an anesthetized macaque monkey. One can observe a slow decay of noise correlations with distance. Different lines mark correlation as a function of distance according to the signal correlation value of cell pairs. Top right (B): Distance decay of correlations in middle temporal visual area (MT) of anesthetized marmoset monkey. In comparison to (A) the decay happens on shorter spatial scales. Strength of correlations depends on the stimulus type with strongest values for a blank grey screen. Bottom left (C): Significant sinusoidal modulation of average r_{SC} with distance in spontaneous activity in rat visual cortex. Bottom right (D): r_{CCG} as a function of integration window size. Images A and D taken with permission from Smith and Kohn (2008), image B taken with permission from Solomon et al. (2014), and image C taken with permission from Ch'ng and Reid (2010)

Contrast level in %	Average r_{SC}
1.25 (worst contrast)	0.38
12.5	0.44
50	0.39
100 (best contrast)	0.25

Table 3.1: r_{SC} modulated by stimulus contrast as reported by Kohn and Smith (2005). Average noise correlations were computed using a grating stimulus that evoked the best response in a cell population with 57 units. Differences between the highest and the other contrast levels are significant (ANOVA, p<0.001).

observed a decay on shorter spatial scales. According to their data, the correlation coefficient almost fully decreased within the first millimeter in the area MT, as shown in figure 3.3B.

Furthermore, Ohiorhenuan et al. (2010) used maximum entropy models to asses correlations on different scales in visual cortex of anesthetized macaque monkey. They found that second order correlations are predominant on short spatial scales up to 1 mm. Even more so for higher order correlations, these almost exclusively play a role in cell pairs with at most 300 µm distance between them.

Moreover, cells with similar response properties, i.e. signal correlations close to 1, also show higher correlated variability (Kohn and Smith, 2005, Smith and Kohn, 2008, and c.f. figure 3.3A). For instance, cells in V1 with similar preferred orientations exhibit stronger shared variability than pairs that differ in their preferred orientations (Hansen et al., 2012).

With respect to temporal scales, noise correlations increase with larger integration time windows (Bair et al., 2001). For example, Smith and Kohn (2008) demonstrated that their measured r_{CCG} saturated for sizes between 500 milliseconds to 1 second. This is shown in figure 3.3. Others reported that saturation happens on slightly longer time scales. For instance, Reich (2001) showed that the average r_{SC} saturated for integration windows beyond 1 second.

The type of stimulus matters as well. Blank screen stimuli yield larger correlations than orientation gratings (Solomon et al., 2014). Moreover, in figure 3.3B one can see that in contrast to random dots or gratings, the decrease with distance for a grey blank screen stimulus is non-monotonic and the magnitude of noise correlations slightly fluctuates with distance. Nonetheless, these fluctuations are not statistically significant. Still, Ch'ng and Reid (2010) reported significant sinusoidal like fluctuations of noise correlations with distance in spontaneous activity in primary visual cortex of rats, see figure 3.3C.

Finally, noise correlations are modulated by the contrast of stimuli, with a tendency to higher correlation values observed for lower and intermediate contrasts. For example, table 3.1 lists values reported by Kohn and Smith (2005). Similar effects were also observed by Oliveira et al. (1997).

3.3 Potential Causes and Confounds of Noise Correlations

Previously we discussed the magnitude and spatial as well as temporal scales of experimentally measured noise correlations. It is still not fully understood which mechanisms govern the emergence of correlated variability. In the following we are going to review a few explanations and hypotheses regarding the causes of noise correlations.

Mathematically, the correlations between discrete variables like spike counts are not dependent on the absolute magnitude of the variables. However, strong relations between the value of r_{SC} and the firing rates of pairs of neurons were observed (Rocha et al., 2007; Schulz et al., 2015). This dependency is not a mathematical one, but grounded in biology. If the mean firing rates of pairs of cells are weak, noise correlations are hidden by the spiking threshold (Cohen and Kohn, 2011). Co-fluctuations in the synaptic input in neuron pairs yield co-fluctuations in the membrane potential. If the firing rate of the cells is large, these co-fluctuations can be measured as noise correlations. In contrast, if firing rates are weak, co-fluctuations are happening on a sub-threshold scale and cannot be observed by only measuring spike responses (Dorn and Ringach, 2003; Kazama and Wilson, 2009). Similarly, this explains why the magnitude of correlations increases and saturates with increasing integration windows. For short windows only a few spikes can be observed per cell and correlations are hidden by the threshold. For sufficiently large windows enough spikes are counted and the full spectrum of correlations is uncovered. Further lengthening of the integration time window is without avail and one observes a saturation of correlated variability (Cohen and Kohn, 2011).

Many experimental factors can influence and cause correlated variability. Erroneous spike sorting can yield biased and false estimates of noise correlations. Electrodes implanted in the animal's brain record a compound signal of ongoing activity from all cells in their vicinity. In order to infer single unit activity, a multitude of methods exist based on individual spike waveforms filtered from the recorded data, for example one may ave a look at the work of Franke et al. (2015). In fact, since the ground truth of such a sorting cannot be known and spike sorting is a statistical process, the isolated sources of spikes are termed units and not neurons. Schulz et al. (2015) reported that among units which could be well isolated correlated variability was reduced. The quality of an isolation was assessed as the distance of the unit's activity cluster to the "multiunit" cluster, containing many unsorted, low-amplitude spikes. Poorly isolated units showed stronger noise correlations. The authors proposed that correlations are increased because these units contain many falsely attributed spikes based on overlapping activity from other synchronously firing neurons. Similarly, Cohen and Kohn (2011) showed that r_{SC} exhibits a steady increase if more and more units are lumped together and regarded as a single unit. Moreover, an abrupt or slow movement of the recording electrode may disturb spike sorting and thereby modulate the r_{SC} measure as well (Ecker et al., 2010).

Other important factors influencing noise correlations are internal states of the animal. For example, shifts in the animal's attention strongly affect correlated variability (Cohen and Maunsell, 2009). Furthermore, the role of anesthesia regarding correlated variability is debated. Ecker et al. (2014) found a significant difference between noise correlation in primary visual cortex of anesthetized and awake macaque monkeys. For the former, averaged r_{SC} values up to 0.2 were measured. Whereas for awake monkeys the authors reported basically an absence of noise correlations. They argued that already subtle variations in the level of anesthesia can cause common changes in the firing rate of many cells over widespread distances.

A prominent hypothesis is that shared incoming connections may cause correlations. As mentioned above, co-fluctuations in the input currents yield co-fluctuations in the membrane potential which, in turn, cause correlated variability in the spike responses (Aertsen et al., 1989; Shadlen and Newsome, 1998; Kazama and Wilson, 2009). In fact, anatomical studies found that about 10% of inputs to excitatory neurons in the visual cortex are shared among cells in close vicinity (Braitenberg and Schüz, 1991; Hellwig et al., 1994).

However, in a seminal study Renart et al. (2010) theoretically proved that even if substantial amount of input is shared among cells, recurrent connectivity still can decorrelate spiking activity for a wide range of parameter settings. In this case recurrent inhibitory currents cancel excitatory ones such that the net amount is always close to zero. Basically, these balanced conditions in form of tight tracking of excitatory activity by recurrent inhibitory activity can generate negative correlations that compensate for the shared input. According to the authors the current correlation c among cell pairs can be decomposed into three parts c_{EE}, c_{II}, and c_{EI}. c_{EE} denotes correlation of excitatory currents among cell pairs, c_{II} refers to inhibitory currents, and c_{EI} (which is equal to c_{IE}) to correlations between excitatory and inhibitory currents. Both c_{II} and c_{EE} are large and positive correlations due to common inputs and very weak recurrent correlations. Yet, c_{EI} is large and negative because of correlations among excitatory and inhibitory cells due to tracking; thus, giving the following cancellation:

$$c = c_{EE} + c_{II} + 2c_{EI} \sim O(1/\sqrt{N}), \tag{3.9}$$

where N refers to the size of the network. For larger networks tracking gets better and c decreases. So if the input currents to cell pairs are not correlated ($c = 0$), neither are the spike responses ($r_{SC} = 0$).

Notwithstanding the decorrelating effects mentioned above, Hansen et al. (2012) hypothesized that noise correlations may be caused by recurrent connectivity in the first place. Indeed, this connectivity is a good candidate as a source of correlated activity because the majority of connections are recurrent. As much as 80% of a cell's inputs originate from within the same cortex area (Markov et al., 2011). The interesting study

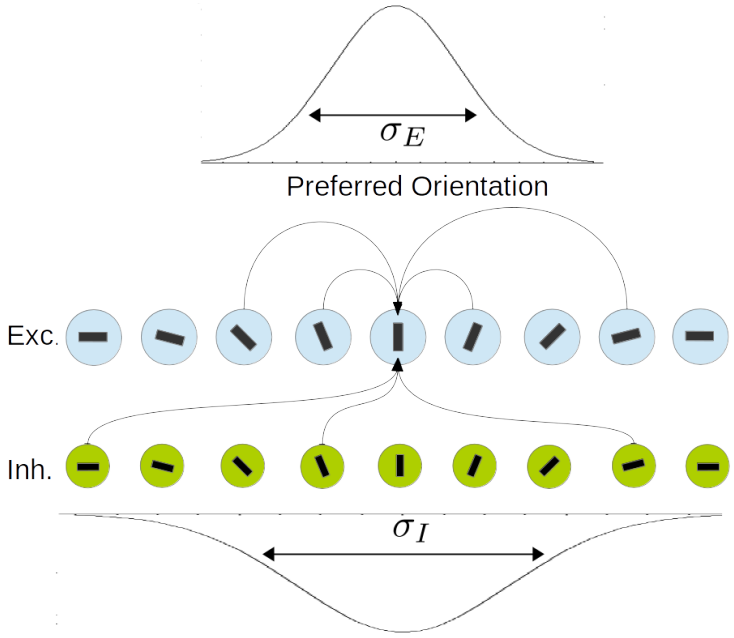

Figure 3.4: Spiking neural network layout as used by Hansen et al. (2012). Excitatory neurons are depicted in blue and inhibitory neurons in green, preferred orientations are sketched by black bars. Presynaptic connections are drawn randomly from Gaussian distributions defined over the preferred orientations with widths σ_E and σ_I.

by Hansen et al. (2012) and the hypothesis formulated therein laid the basis for the further analysis in this thesis. Accordingly, we are going to take a more detailed look in the following section.

3.3.1 Noise Correlations due to Recurrent Connectivity

Hansen et al. (2012) recorded multi-neuron activity in primary visual cortex of two awake primates (Macaca mulatta) with multi-contact laminar electrodes. Moreover, the monkeys were trained to fixate the center of a screen. A single orientation stimulus was flashed for 300 ms while the monkeys were looking. Eight different orientations in steps of 22.5° were shown randomly across trials with 50 trials per orientation. Noise correlation coefficients r_{SC} were calculated averaging over different orientations for all recorded cell pairs.

The authors reported different correlation coefficients for different layers in V1. In the fourth layer, also referred to as the granular layer, correlations were almost non-existent with an average r_{SC} of 0.04 (± 0.03 SEM). They further distinguished between layers above and below the fourth layer, that is the supra (SG) and infragranular (IG) layers, respectively. There correlations were significantly larger with average r_{SC} equal to 0.24 (± 0.01 SEM) in the SG layer and 0.23 (± 0.01 SEM) in the IG layer. A similar dependency of r_{SC} on laminar depth with low correlation in the fourth layer of V1 has been reported by Smith et al. (2013) as well.

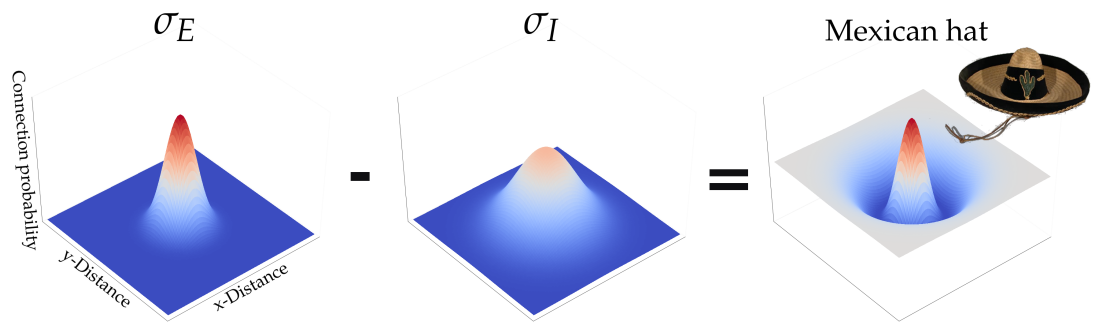

Figure 3.5: Depiction of Mexican hat connectivity scheme on a 2D plane. If both connectivity kernels with widths σ_E and σ_I are subtracted from each other, the resulting difference remotely resembles a sombrero. If the kernels are first multiplied by the connection strength before subtracting, one may interpret the result as the expected connectivity strength as a function of distance in each dimension between cell pairs. Small sombrero image taken from Wikipedia (`https://upload.wikimedia.org/wikipedia/commons/a/ac/Harry_S_Truman_sombrero.jpg`, accessed 26.10.2015)

Hansen et al. (2012) hypothesized that these differences in noise correlations arise due to disparity in the spatial spread and specificity in horizontal recurrent connections. More precisely, recurrent excitatory inputs to neurons in the infra- and supragranular layers are more sharply tuned to orientation than the inhibitory recurrent input. The tuned input might be realized via long range connections across cortical hypercolumns with similar orientation preference.

To fortify this hypothesis Hansen et al. (2012) implemented a ring network model of conductance based integrate and fire neurons positioned according to their preferred orientations. Recurrent connections were randomly drawn from Gaussian distributions over differences in preferred orientations between the pre- and the post-synaptic neurons. The width of these Gaussians for the inhibitory σ_I and excitatory neurons σ_E were varied and their influence on noise correlations were investigated. Simulations were performed by showing repeated trials of the same orientation to the network to estimate r_{SC}. Results were averaged across several orientation stimuli. A sketch of the network is provided in figure 3.4.

Given a configuration with narrow excitatory spread compared to a large inhibitory connection width ($\sigma_E < \sigma_I$), networks featured correlated variability among pairs of neurons with similar orientation tuning. The networks exhibited average correlation coefficients of more than 0.3 for neurons with similar orientation preference. Additionally, Hansen et al. (2012) observed a smooth decline of correlations with an increase in orientation preference difference between pairs of cells.

From throughout the thesis to simplify naming, network topologies with the property of $\sigma_E < \sigma_I$ are termed **Mexican hat** networks. One can image that subtracting two Gaussian connectivity kernels (negative sign for inhibition) from each other yields a curve whose shape remotely resembles the silhouette of a sombrero. The resemblance to a sombrero becomes even more apparent if the Gaussian kernels are defined over a

two-dimensional space, as depicted in figure 3.5. Likewise, topologies with $\sigma_E > \sigma_I$ are termed **inverse Mexican hat**. Lastly, the setting $\sigma_E = \sigma_I$ is called a **balanced hat**. This naming scheme for topologies featuring two Gaussian kernels can be found throughout the literature, for example in work by Kang et al. (2003), Blumenfeld et al. (2006), and Bressloff (2012). Note because σ_E and σ_I denote the widths of Gaussian functions, but in biological terms refer to the axonal spread and the resulting spread of connectivity, we use the expressions width and spread interchangeably.

3.4 Influence of Noise Correlations on Population Coding

Why do correlations matter in the first place? Correlations can affect — improve or worsen — the brain's ability to make inferences about the state of the outside world (Averbeck et al., 2006).

Visual stimuli are translated into spike trains by the retina and transmitted via the lateral geniculate nucleus (LGN) to the brain (Kandel et al., 2000). As mentioned before, this process is probabilistic and very different spike trains can be caused by the very same stimulus; hence the distribution $p(\boldsymbol{r}|s)$ (c.f. equation 3.2). The quantity the brain is interested in is arguably not $\boldsymbol{r}$, but the stimulus s that caused the neural response. Basically, the brain has to invert the relation between stimulus and response and wants to know $p(s|\boldsymbol{r})$ (Latham and Roudi, 2011). As a consequence, it can determine an appropriate reaction to s. For instance, a goal keeper observes the trajectory of a soccer ball and initiates an appropriate movement to catch it and prevent a goal.

The relation between $p(\boldsymbol{r}|s)$ and $p(s|\boldsymbol{r})$ is given via *Bayes rule*:

$$p(s|\boldsymbol{r}) = \frac{p(\boldsymbol{r}|s)\, p(s)}{p(\boldsymbol{r})}, \tag{3.10}$$

with $p(s)$ the prior distribution over all stimuli and $p(\boldsymbol{r})$ the distribution over all responses. This notion of a *Bayesian brain* is prominent among scholars (Knill and Pouget, 2004; Doya et al., 2007). Still, we do not know for sure if the brain really bothers with this Bayesian inference problem at all. Alternatively, the brain could just associate each stimulus with an appropriate response (Latham and Roudi, 2011). But it is highly unlikely that the goal keeper remembers a response for all potential ball trajectories. She rather evaluates the ball's trajectory for every observation anew. To prevent the scoring of a goal it is necessary to estimate when the soccer ball will be close and have a notion about the certainty and the potential errors regarding the ball's position. As a matter of fact, there exist many studies suggesting that the brain takes such uncertainties into account when making decisions (Körding and Wolpert, 2004; Chater et al., 2006; Ma et al., 2011).

Already by the definition of noise correlations in equation 3.2,

$$p(\boldsymbol{r}|s) \neq \prod_{i=1}^{N} p(r_i|s), \qquad \text{(3.2 revisited)}$$

we identify that correlations have a direct effect on the inference problem. The shared variability can influence the shape of the $p(\boldsymbol{r}|s)$ which is part of the numerator in equation 3.10. Nonetheless, how the inference is carried out in particular by the brain is a non-trivial problem. This poses already some difficult questions. What is the distribution of all stimuli $p(s)$ and how should the brain know about it? No doubt, these problems are beyond the scope of this thesis. Still, at least if we are willing to assume that "if there is information in spike trains, the brain uses it" (Latham and Roudi, 2011), we may just look at how much information about the stimulus is encoded by a population response. Thus, we can ask if correlations increase or decrease information about the stimulus. Of course, in the former case correlations would benefit the inference problem, and in the latter harm it. Admittedly, as we will later see, in order to estimate this information we will use a decoder to infer s from $\boldsymbol{r}$. Indeed, the form of the decoder itself is a strong assumption, but our claim is not that the brain works this way.

Next, we are going to introduce two widely used information measures. Thereafter, it will be discussed how these can be applied to investigate the effect of correlations on stimulus encoding.

3.4.1 Measuring Information

Two commonly applied information measures in Neuroscience are *mutual information*, often also termed *Shannon information*, and *Fisher information* (Seriès et al., 2004; Averbeck et al., 2006; Yarrow et al., 2012).

Mutual information measures the dependence of two random variables. For instance, in a Neuroscience experiment these are the sets of applied stimuli and observed neural responses. Mutual information is based on *Shannon entropy*. In 1948 the mathematician Claude E. Shannon introduced this concept in his seminal paper on information theory (Shannon, 1948). Entropy H is a measure of the *uncertainty* or *randomness* of a random variable X with possible outcomes x_1 to x_n and probability mass function p:

$$H(X) - \sum_{i=1}^{n} p(x_i) \log_b p(x_i). \qquad (3.11)$$

If the logarithm basis is $b = 2$, entropy is calculated in *bits*. Moreover, if the distribution of X is uniform, X is most unpredictable. Hence, the entropy is maximal (c.f. Rieke et al., 1997).

Furthermore, given two random variables X and Y with potentials values x_1 to x_n and y_1 to y_m the *joint entropy* is defined as follows:

$$H(X,Y) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p(x_i, x_j), \log_b p(x_i, y_j), \tag{3.12}$$

where $p(x_i, x_j)$ is the joint distribution of X and Y. If X and Y are independent, it holds $H(X,Y) = H(X) + H(Y)$. Yet, if they are dependent variables, one may look at the conditional entropy $H(X|Y)$. It is the difference between the joint and individual entropy:

$$H(X|Y) = H(X,Y) - H(Y). \tag{3.13}$$

Mutual information $I_{\text{mutual}}(X,Y)$ describes the overlap between the entropy of two random variables:

$$\begin{aligned} I_{\text{mutual}}(X,Y) &= H(X) + H(Y) - H(X,Y) \\ &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X). \end{aligned} \tag{3.14}$$

Using a bit of algebra $I_{\text{mutual}}(X,Y)$ can also be directly expressed in terms of the distributions (Dayan and Abbott, 2005), as given below:

$$I_{\text{mutual}}(X,Y) = \sum_{i=1}^{n}\sum_{j=1}^{m} p(x_i, y_j) \log_b \frac{p(x_i, y_j)}{p(x_i)\, p(x_j)}. \tag{3.15}$$

Mutual information is large if there exist considerable amount of entropy overlap between both random variables. Hence, the outcome of one random variable is very informative about the outcome of the other. Given high mutual information between stimuli and responses, observing a neural response $\boldsymbol{r}$ can tell a lot about the stimulus s that caused the response and vice versa. Thus, an increase in mutual information benefits decoding s from $\boldsymbol{r}$.

Mutual information measures information of an entire response set about an entire stimulus set. However, mutual information does not make any statements with which precision specific stimuli are encoded (Yarrow et al., 2012). Additionally, mutual information is only defined over discrete events. For continuous domains, like orientation of a bar stimulus, it is not applicable.

For such problem settings Fisher information is the better choice. It is defined as (Dayan and Abbott, 2005)

$$I_{\text{Fisher}}(s) = \left\langle \left(\frac{\partial}{\partial s} \ln p(r|s) \right)^2 \middle| s \right\rangle, \tag{3.16}$$

where $\langle\cdot|s\rangle$ is the expectation operation over responses for a fixed stimulus s. Units are in reciprocal of the variance of the quantity at hand. So for orientation stimuli this is deg^{-2}. Fisher information has a direct link to decoding. The inverse of Fisher information is the lower bound on the variance of an unbiased estimator $\hat{s}$ of stimulus s, also termed the Cramèr-Rao bound (Gabbiani and Cox, 2010):

$$\mathrm{Var}(\hat{s}) \geq \frac{1}{I_{\mathrm{Fisher}}(s)} \tag{3.17}$$

Hence, large Fisher information means a low variance of an optimal decoder of s and, consequently, a good decoding precision. Fisher information is a local measure of coding quality. For instance, it tells how well small changes in orientation of a grating stimulus can be detected by a neural ensemble.

Given Gaussian variability, neurons' tuning curve functions $\boldsymbol{f}(s)$, and covariance matrix $\boldsymbol{Q}(s)$, Fisher information can be directly computed (Abbott and Dayan, 1999):

$$I_{\mathrm{Fisher}}(s) = \boldsymbol{f}'(s)^T \boldsymbol{Q}(s)^{-1} \boldsymbol{f}' + \frac{1}{2}\,\mathrm{Tr}[\boldsymbol{Q}'(s)\,\boldsymbol{Q}(s)^{-1}\,\boldsymbol{Q}'(s)\,\boldsymbol{Q}(s)^{-1}], \tag{3.18}$$

with $\mathrm{Tr}[\cdot]$ giving the trace of a matrix, and $\boldsymbol{f}'(s)$ and $\boldsymbol{Q}'(s)$ being the derivatives of the tuning curve $\boldsymbol{f}$ and covariance matrix $\boldsymbol{Q}$ with respect to s. However, calculating Fisher information with equation 3.18 may be intractable. Reliable estimates of $\boldsymbol{Q}(s)$ and $\boldsymbol{Q}'(s)$ require a very large amount of data.

Fortunately, there is an approximation to Fisher information based on a linear estimator called I_{LOLE} that requires much less data (Latham et al., 2003; Seriès et al., 2004). LOLE stands for a *locally optimal linear estimator*. The estimated value of a stimulus, that is $\hat{s}$, is computed as follows:

$$\hat{s} = \boldsymbol{w}\,\boldsymbol{r} + w_0, \tag{3.19}$$

where $\boldsymbol{w}$ is a columnar weight vector, $\boldsymbol{r}$ the spike response of the neural ensemble over a fixed time window, and w_0 a bias weight. $\boldsymbol{w}$ and w_0 are optimized to reduce decoding error based on given training data. The data are several trials of two rather similar stimuli, for example orientation gratings with rotations of 0° and 1°. Next, Fisher information can be approximated via

$$I_{\mathrm{LOLE}} = \frac{\left(\frac{\langle\hat{s}_2\rangle - \langle\hat{s}_1\rangle}{s_2 - s_1}\right)^2}{\frac{1}{2}(\mathrm{Var}(\hat{s}_2) + \mathrm{Var}(\hat{s}_1))}, \tag{3.20}$$

where $\langle\cdot\rangle$ denotes expectation and $\mathrm{Var}(\hat{s}_i)$ is the variance of the estimates for one particular stimulus across trials. I_{LOLE} gives a lower bound of the real Fisher information. In case the LOLE performance is close to the theoretically optimal Bayesian decoder, i.e. there's no information that can only be recovered by non-linear methods, this bound is tight. As a consequence, I_{LOLE} provides a good approximation to the real amount of

Neuron	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	**Total**
Neuron I	1	2	3	4	5	**15**
Neuron II	1	1	0	3	3	**8**
Neuron III	1	2	0	5	4	**12**

Neuron	Shuff 1	Shuff 2	Shuff 3	Shuff 4	Shuff 5	**Total**
Neuron I	3	4	1	5	2	**15**
Neuron II	3	0	1	1	3	**8**
Neuron III	4	1	5	2	0	**12**

Table 3.2: These tables give an example of shuffled trial data that can be used to compute I_{shuff}. The top table lists the original spike counts observed for 3 hypothetical neurons over 5 trials. The bottom table depicts the shuffling of trials. The observed spike counts are shuffled for each neuron. Note that counts are shuffled across trials but not across neurons. Hence, the totally observed number of spikes for each neuron (last column) remains equal between the non-shuffled and shuffled condition.

Fisher information. In fact, linear decoders can be well suited for decoding of experimental recordings. For instance, Marre et al. (2015) demonstrated that a linear decoder can read out stimuli from retinal activity with high precision.

3.4.2 Effect of Correlations on Information

Given the information measures from the previous section we can investigate if correlations modulate information about stimuli in neural population responses. First, we are going to deal with the question whether correlations increase or decrease information. Secondly, do correlations themselves carry information or may downstream neurons safely ignore them?

If we want know how correlations affect population coding we can look at the quantity

$$\Delta I_{\text{shuff}} = I - I_{\text{shuff}}. \tag{3.21}$$

It is the difference between the amount of information I about stimuli in the original population responses and the amount of information I_{shuff} given correlations were removed artificially (Seriès et al., 2004; Averbeck and Lee, 2006). The generic term I can be replaced by the corresponding type of information one is interested in, like Fisher or mutual information discussed in the previous section. Moreover, the index *shuff(led)* comes from the fact that correlations in the data are removed by shuffling trials. An example of shuffling trial data is provided in table 3.2.

In case $\Delta I_{\text{shuff}} < 0$, correlations are detrimental to the population code and decrease the available information. Considering a population of two neurons and two stimuli, this relationship can be nicely sketched, see figure 3.6A. In this case the response distributions of the cell pair for the two different stimuli may overlap considerably. The

95% confidence intervals of the distributions of both stimuli are depicted by the orange and red ellipses. Accordingly, for a response that falls within the overlapping region it is hard to tell if the response was caused by one or the other stimulus. Even given the optimally placed decision boundary (black dotted line), a decoder trying to infer s from $\boldsymbol{r}$ would inevitably produce a couple of wrong inference results. In contrast, if trials are shuffled, the ellipses become circles and, as a result, less overlap means better decoding of the stimulus.

In the opposite case $\Delta I_{\text{shuff}} > 0$ (figure 3.6B), correlations benefit the population code. The parallel ellipses of the original distributions exhibit less overlap than the distribution of the shuffled trials. As one can see from the black decision boundary, this increases inference performance.

Consequently, one has to be careful when calculating information based only on single cell recordings. If information is solely estimated by combining trials from several individually recorded neurons, one can only compute I_{shuff}. Hence, one may under or overestimate the available information. This emphasizes the importance of modern approaches of multi-cell recordings.

Finally, there can be the case that even though correlations exists, they have no effect on information: $\Delta I_{\text{shuff}} = 0$ (figure 3.6C). Response distributions are aligned such that the overlap remains the same regardless of shuffling or no shuffling.

In summary, there is no straightforward answer how correlations modulate the quality of a population code. Indeed, this depends on the structure of the correlations and the particular combination of signal and noise correlations.

As mentioned before, another question one can ask is more from the perspective of decoding: Does a decoder actually have to care about correlations? Or from the brain's point of view: Would downstream neurons lose information if they ignored the correlations from their inputs? To answer these questions one can look at the measure of

$$\Delta I_{\text{diag}} = I - I_{\text{diag}}. \tag{3.22}$$

This is the difference between the information in the original data I and the information I_{diag} that can be retrieved from an optimal decoder that is trained on the shuffled but applied to the original data. The subscript *diag(onal)* refers to the decoder's assumption of neural responses having a diagonal covariance matrix; as is the case for shuffled data. The value of I_{diag} can only be equal to zero, correlations can be ignored, or larger than zero, correlations carry information relevant for the decoder (Seriès et al., 2004; Averbeck and Lee, 2006).

In fact, there can be situations in which $\Delta I_{\text{shuff}} \neq 0$ but $\Delta I_{\text{diag}} = 0$ as well as the opposite where ΔI_{shuff} is equal to 0, but a decoder cannot safely ignore correlations without losing information (Nirenberg and Latham, 2003; Averbeck and Lee, 2006). These two cases are sketched in figure 3.7. The decision boundaries (black and brown

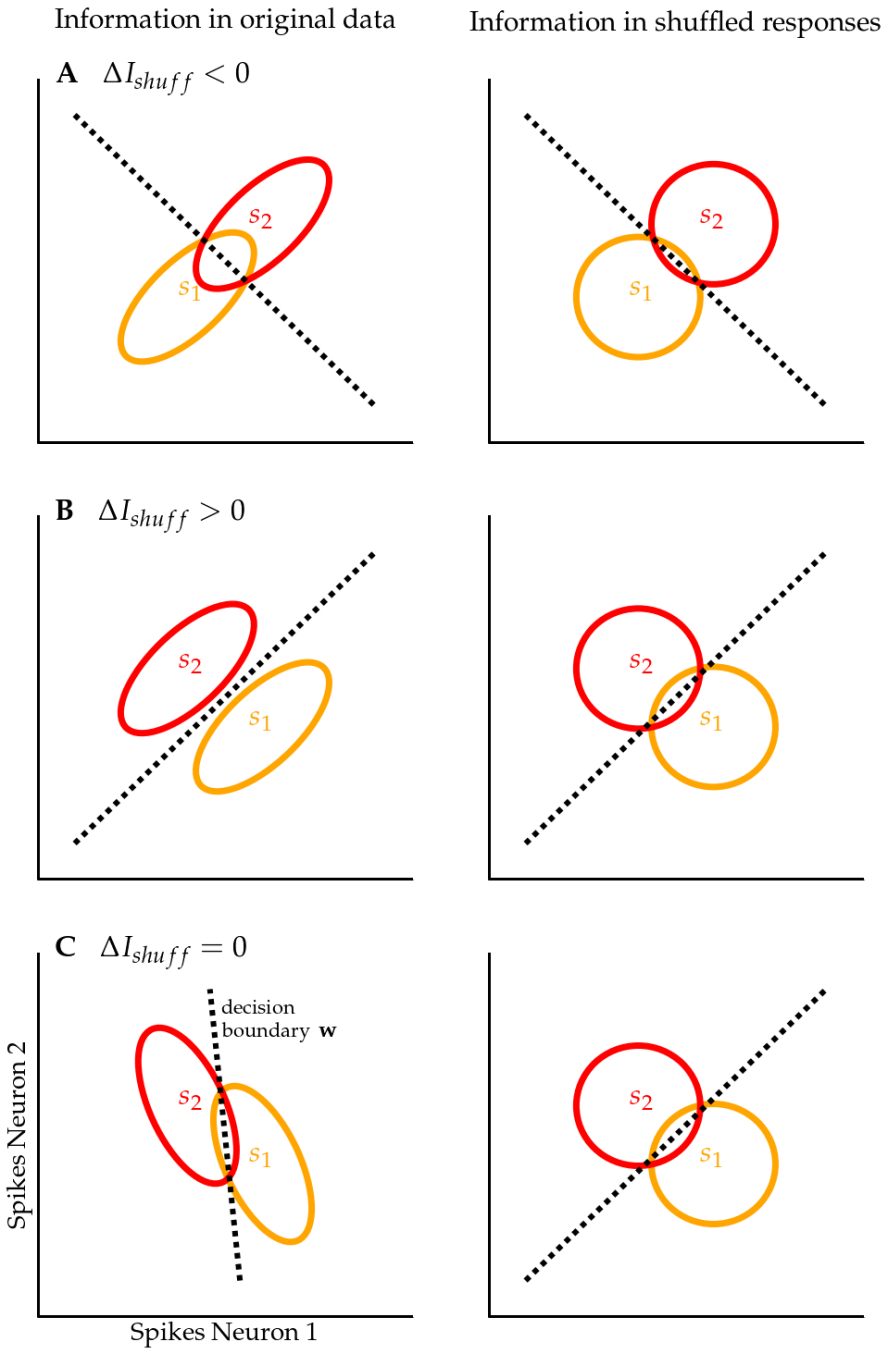

Figure 3.6: Effects of correlations on information. Top row (A): ΔI_{shuff} is negative. Correlations are reducing information. The two ellipses represent the 95% confidence intervals of joint response distributions for two different stimuli. There exists considerable amount of overlap between the correlated responses. Hence, an optimal decoder, depicted by the black dotted decision boundary, would produce some erroneous stimulus classifications. Whereas for the shuffled data, represented by the circles on the right, overlap is reduced. Middle row (B): ΔI_{shuff} is positive. Correlations are increasing information. Response distribution are aligned in such a way that overlap is reduced in comparison to the shuffled responses. Bottom row (C): ΔI_{shuff} is zero. Although responses on the left are correlated, the amount of overlap remains the same if trials are shuffled. Image based on Averbeck et al. (2006).

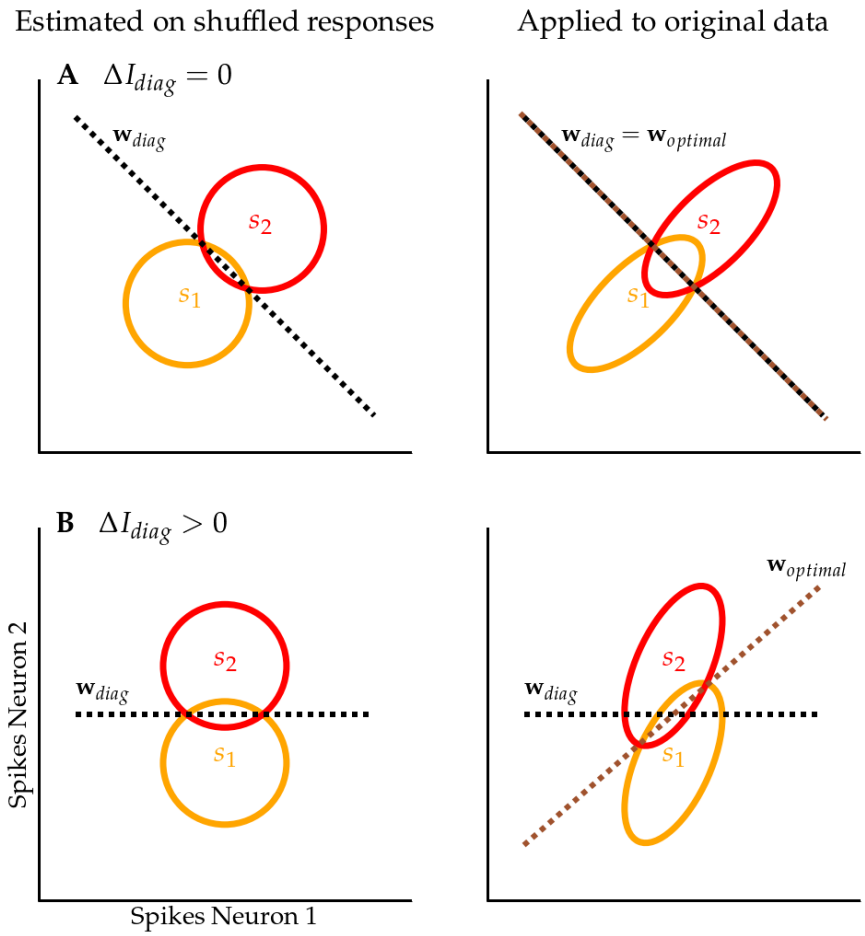

Figure 3.7: Effects of correlations on decoding. Top row (A): ΔI_{diag} is zero. Correlations can be safely ignored. The two ellipses represent the 95% confidence intervals of joint response distributions for two different stimuli. It makes no difference if the decoder, depicted by the black dotted line, is trained on the original or shuffled data. As the right hand side shows, the optimal decision boundary, depicted as the brown dotted line, is equal to the decoder trained on the shuffled data. Bottom row (B): ΔI_{diag} is positive. Correlations carry additional information relevant to a decoder. The decoding decision boundary trained on the shuffled data on the left is different from the optimal boundary of the correlated data on the right. Image based on Averbeck et al. (2006).

dotted lines on top of each other) are equal in figure 3.7A despite considerable amount of correlation between the neuron responses. Whereas in figure 3.7B ignoring correlations would yield a suboptimal decision boundary (black dotted line vs. brown dotted one).

Correlations increasing ($\Delta I_{\text{shuff}} > 0$) as well as harming information ($\Delta I_{\text{shuff}} < 0$) were reported for stimulus encoding experiments in the visual cortex (Golledge et al., 2003; Hansen et al., 2012). In addition, there is evidence that correlations matter for decoding of visual information ($\Delta I_{\text{diag}} > 0$) (Golledge et al., 2003). Yet, often only small influence of correlations on coding were measured (Averbeck and Lee, 2004). Still, this does not mean that correlations can be neglected. Even if the effects of correlations are

small regarding only a couple of neurons, their influence might still matter a lot for larger populations on the order of biologically relevant magnitudes, like columns in the visual cortex (Zohary et al., 1994; Shamir and Sompolinsky, 2004). For example, according to simulations by Averbeck et al. (2006), noise correlations on the order of $r_{SC} = 0.1$ in combination with positive signal correlations, i.e. neurons having similar tuning curves, yielded a negligible information difference between I and I_{shuff} of less than 1% if the authors only looked at cell pairs. However, for thousands of neurons the available information I in the original data was about 25 times smaller than the shuffled information I_{shuff}.

Along these lines are the following theoretical considerations from Sompolinsky et al. (2001). Given infinitively precise input, positive signal correlations, and positive noise correlations, the amount of information of a population code is bounded and saturates for infinite network sizes. This means even for infinitely many neurons and the input of infinite information, only a finite amount of information can be recovered from the network. In contrast, the shuffled information of the network is not bounded and grows infinitely with the network size. Even more so for negative noise correlations, in this case information increases even faster with network size than the shuffled information.

Furthermore, theoretical studies by Abbott and Dayan (1999) and Wilke and Eurich (2002) showed that the often reported positive noise correlations that decrease with distance, as discussed in the previous sections, are in general harmful to the population code. They yield saturation effects with increasing network size. Nonetheless, whether information is bounded is arguably of lesser importance to the brain. On one hand, the brain is made of an admittedly large, but still finite number of neurons. On the other hand, the amount of information about a stimulus that can be extracted by sensors like the retina is finite as well. Of course, following processing steps cannot add new information. Hence, "the question is not whether information saturates in the nervous system — it does — it's how quickly it saturates as the number of neurons increases" (Averbeck et al., 2006).

3.5 Summary

In this chapter we reviewed the importance of noise correlations in spiking activity of ensembles of neurons and the effects these can have on stimulus encoding.

A prominent measure to quantify correlations among pairs of neurons is the spike count correlation coefficient r_{SC}. It is widely used among experimentalists. Often it is reported that correlations are positive for nearby neurons with similar response properties in the visual cortex of different species. These correlations are profound on longer time scales and one often observes a decay with distance between cell pairs. Additionally, for blank stimuli the distance dependency may exhibit a sinusoidal modulation.

We presented common confounds and hypothesized causes on the origin of noise correlations. A very interesting explanation was offered by Hansen et al. (2012) who argued that a particular recurrent network topology with narrower excitatory than inhibitory connection spread — termed Mexican hat — yields correlated variability.

Furthermore, we discussed two particular measures of information, namely mutual and Fisher information. The former is suitable for discrete sets of stimuli, the latter is applied in continuous domains and can be approximated using a linear decoder. We further reviewed that correlations can increase, decrease, or have no effect on the information about a stimulus encoded by spike responses. In addition, correlations can either carry valuable information or decoders may safely ignore correlations without losing information about the stimulus.

Theoretical Motivation: Mexican Hat Connectivity

4

In the previous chapter we discussed possible hypotheses about the origin of correlated variability in visual cortex. The proposal by Hansen et al. (2012) emphasized recurrent connectivity as a source of correlations. Yet, seminal work by Renart et al. (2010) showed that balanced recurrent connectivity can lead to the opposite effect and one may observe a decorrelation of activity with an increase in network size. This holds even if correlated afferent input is present and in case of dense recurrent connectivity. To shed light on this discrepancy, we are going to review recent theoretical findings by Rosenbaum and Doiron (2014) in this chapter. Their findings encompass conditions under which the balance between excitation and inhibition can no longer be maintained. Accordingly, such cases are potential candidates for noise correlations.

In the following sections we are going to provide a detailed review of the study by Rosenbaum and Doiron (2014). In addition, we will augment their work by introducing spike frequency adaptation (SFA). We are going to demonstrate that the findings by Rosenbaum and Doiron (2014) still hold in this extended setting. Furthermore, we will make an extensive analysis under which conditions balanced recurrent activity cannot be maintained any longer. However, in this case no explicit solution of the resulting network state can be given due to the combination of the model's non-linearity and coupling topology. Therefore, we are going to compare the model by Rosenbaum and Doiron (2014) to older work by Hansel and Sompolinsky (1998). This simpler but analytically tractable model gives an explicit solution if the balance between recurrent excitatory and inhibitory activity does not hold any longer. Under certain conditions one observes spatially inhomogeneous activity that moves within the spatial extent of the network. We will illustrate the resemblance between the two models and discuss which conclusions can be drawn for the more complex model and the numerical simulations presented in the following chapter. Namely, the movement and phase changes of spatial inhomogeneous activity cause joint modulations of firing rates. In turn, these modulations yield noise correlations among spike counts.

4.1 Vanilla Rosenbaum and Doiron Model

A detailed explanation of the model by Rosenbaum and Doiron (2014) will be given in this section. Rosenbaum and Doiron (2014) discuss a general mean-field approach to spiking neuron networks. They investigate under which recurrent coupling schemes the balance between excitation and inhibition destabilizes.

Accordingly, they use the following definition of balanced conditions: A network is said to be balanced if and only if in the continuum limit it exhibits finite or unsaturated, and non-zero firing rates which for constant input are constant in time. Hence, inhibition must counterbalance afferent and recurrent excitatory inputs to prevent diverging activity.

To anticipate the main finding, the mean-field system becomes unstable in case of Mexican hat coupling schemes where inhibitory recurrent connections are wider than excitatory ones. Rosenbaum and Doiron (2014) obtain this result by taking the spatial Fourier transform of the network dynamics. From this transformed state one can infer whether some spatial frequencies are amplified. If so, the balance between excitation and inhibition cannot be maintained. Homogeneous asynchronous activity is no longer stable and spatially heterogeneous activity can arise. Movement and phase changes of the spatially inhomogeneous activity, in turn, make such networks potential candidates for the emergence of noise correlations.

Of note, in the following analysis an attempt is made to derive the work in the original paper in a clear and concise manner. Accordingly, we are going to provide some derivations that were spared by Rosenbaum and Doiron (2014).

4.1.1 Model Description

We will start with a detailed description of the unitless system used by Rosenbaum and Doiron (2014). They assume a network of N spiking neurons, with $N_E = q\,N$ excitatory neurons and $N_I = (1-q)\,N$ inhibitory neurons. Therefore, q denotes the fraction of excitatory neurons. The neurons are evenly spaced on a ring $\Gamma = (0, 1]$ with periodic boundary conditions. Accordingly, the kth excitatory neuron is placed at $x = k/N_E$. Analogously, for inhibitory neurons $x = k/N_I$ marks the location of kth inhibitory cell. The input current composed of the recurrent as well as afferent inputs to each neuron is defined as:

$$I_X(x,t) = \sum_{i=1}^{N_E} J_{XE}^{\kappa^\Gamma}(x - i/N_E)\, S_{E,i}(t) - \sum_{i=1}^{N_I} J_{XI}^{\kappa^\Gamma}(x - i/N_I)\, S_{I,i}(t) + J_{XA}(x), \tag{4.1}$$

where $S_{X,i} = \sum_j \delta(t - t_{X,i}^j)$ is the spike train of the ith pre-synaptic neuron in population $X \in \{E, I\}$ modeled as a sum of Dirac delta functions. $J_{XA}(x)$ denotes the static afferent input at position x. Moreover, recurrent connectivity is probabilistic. The synaptic weight $J_{XY}^{\kappa^\Gamma}$ with $X, Y \in \{E, I\}$ is either equal to 0 or a positive constant

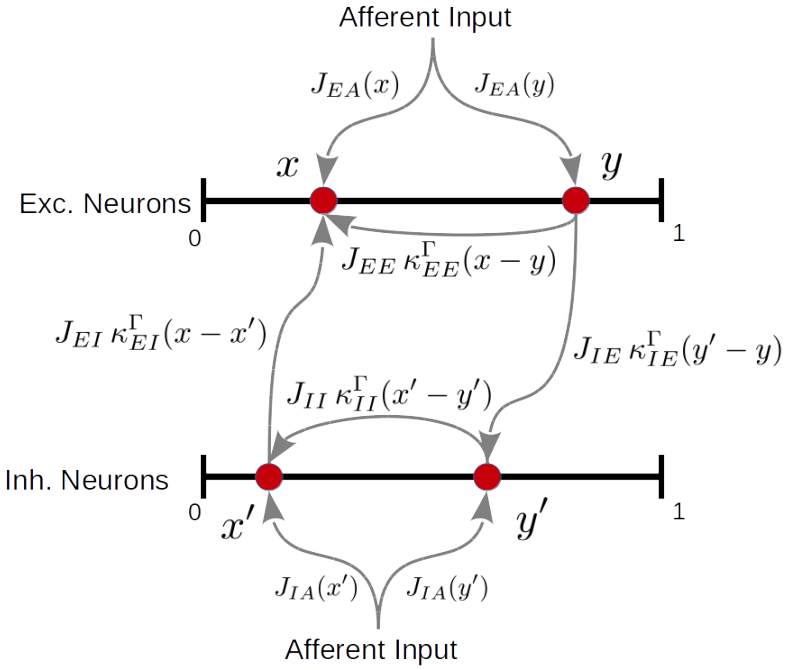

Figure 4.1: Depiction of the model by Rosenbaum and Doiron (2014). Neurons are arranged on the ring $\Gamma = (0, 1]$ with periodic boundary conditions. For better visibility excitatory and inhibitory neurons are depicted side by side. An excitatory neuron at positions x receives inputs of strength J_{EE} from another one at position y with probability $\kappa^{\Gamma}_{EE}(x-y)$, other connections κ^{Γ}_{EI}, κ^{Γ}_{IE}, and κ^{Γ}_{II} are analogously applied. In addition, neurons are driven by static afferent inputs $J_{EA}(x)$ and $J_{IA}(x)$.

value J_{XY} with probability $\kappa^{\Gamma}_{XY}(x-y)$. Accordingly, inhibition is not captured by negative coupling strengths but by subtraction, as in equation 4.1, for example. Moreover, to account for periodic boundary conditions, $\kappa^{\Gamma}_{XY}(x-y)$ is the periodic summation of probabilities:

$$\kappa^{\Gamma}_{XY}(x-y) = \sum_{l=-\infty}^{\infty} \kappa_{XY}(x-y+l), \tag{4.2}$$

where κ_{XY} is a probability distribution defined over discrete points on the real axis, for example a discrete Gaussian kernel. The periodic summation of κ_{XY} wraps the distribution infinitely often around the ring Γ such that the kernel domain changes from discrete points on the real axis to discrete points on the interval $(0, 1]$. Moreover, $x = k/N_X$ and $y = i/N_Y$ are the positions of the post- and pre-synaptic neurons on the ring. Due to the periodic boundary conditions of κ^{Γ}_{XY}, the difference $x-y$ corresponds to the shortest distance between the post- and pre-synaptic neuron on Γ. Hence, κ^{Γ}_{XY} is the spatial connectivity profile originating in neural population Y projecting onto population X. The model's layout is sketched in figure 4.1.

To study neural dynamics, the current as defined in equation 4.1 can be used in a particular neuron model, like the a leaky integrate and fire (LIF) one given below (c.f. Izhikevich, 2007):

$$\frac{\partial V_X(x,t)}{\partial t} = -\frac{1}{\tau_V} V_X(x,t) + I_X(x,t), \tag{4.3}$$

where V_X is the membrane potential of a neuron from population $X \in \{E, I\}$ at position $x \in \Gamma$ at time t. In addition, τ_V denotes cell's membrane time constant. A spike is emitted whenever the potential reaches a threshold $V_X(x,t) = 1$ and it is subsequently reset to 0, the resting state. However, we will not investigate the dynamics of this model directly. Instead, we are going to use a simple firing rate approximation that will be introduced in the following sections.

Moreover, increasing network size N leaves one with the question of how to scale the synaptic coupling strengths J_{XY} for a growing number of neurons. Many studies assume a scaling of order $O(1/\sqrt{N})$ (Vreeswijk and Sompolinsky, 1996; Renart et al., 2010) and so do Rosenbaum and Doiron (2014). Under these assumptions and given a neuron model with a spiking mechanism like the LIF above, a neuron receives $O(N)$ connections, but requires only excitatory spiking input of order $O(\sqrt{N})$ to emit a spike itself. Hence, in order to maintain finite or unsaturated firing rates in the limit of infinite network size, one requires a stable balance between excitation and inhibition (Vreeswijk and Sompolinsky, 1998; Renart et al., 2010).

As a side remark, in our simulations in the following chapter a biologically plausible range of connection strengths was used. Under the given scaling of $O(1/\sqrt{N})$ increasing network size would render the strengths too small. Thus, in our simulations the weights were fixed. In fact, as we will discuss later, even if weights scale with $O(1)$, the findings of Rosenbaum and Doiron (2014) are still valid.

For simplicity and if not stated otherwise, in this chapter we will use the scaling of the original paper. Namely, Rosenbaum and Doiron (2014) assume a distance between spike threshold and resting potential for each neuron of $O(1)$ and use the following scaling properties with an increase in network size N: $J_{XY} \sim O(1/\sqrt{N})$, $J_{XA} \sim O(\sqrt{N})$, $\kappa_{XY} \sim O(1)$, and $q \sim O(1)$ for $N \to \infty$. In order to make the connection weights independent of the network size, they introduce rescaled weights denoted by lower case $j_{XY} = \sqrt{N}\, J_{XY}$ and $j_{XA} = J_{XA}/\sqrt{N}$.

4.1.2 Balanced State in the Continuum Limit

We are now going to turn from the spike based model to a mean-field approximation with firing rates. The instantaneous mean firing rate ν_X of the kth neuron of population X at position $x = k/N$ at time t can be defined as

$$\nu_X(x,t) \coloneqq \langle \mathrm{E}[S_{X,k}(t)] \rangle, \tag{4.4}$$

where $\mathrm{E}[\cdot]$ is the expectation operation in the mean-field limit over $M \to \infty$ different network realizations, i.e. sampling over κ_{XY}. In addition, $\langle\cdot\rangle$ denotes expectation over a vanishing time interval Δt (c.f. Augustin et al., 2013):

$$\langle \mathrm{E}[S_{X,k}(t)]\rangle = \lim_{\substack{M\to\infty,\\ \Delta t\to 0}} \frac{1}{\Delta t\, M} \int_t^{t+\Delta t} \sum_{m=1}^{M} \sum_j \delta(s - t_{X,k}^j)\, ds. \tag{4.5}$$

Similarly, we can compute the expected coupling strength between the ith pre-synaptic neuron and the post-synaptic neuron at position x for the different neuron populations:

$$\begin{aligned} \langle \mathrm{E}[J_{XY}^{\kappa^\Gamma}(x - i/N_Y)]\rangle =& \mathrm{E}[J_{XY}^{\kappa^\Gamma}(x - i/N_Y)] \\ =& \lim_{M\to\infty} \frac{1}{M} \sum_{m=1}^{M} J_{XY}^{\kappa^\Gamma}(x - i/N_Y) \\ =& J_{XY}\, \kappa_Y^\Gamma(x - i/N_Y). \end{aligned} \tag{4.6}$$

The expectation over the vanishing time interval can be ignored for the expected coupling because it is static over time.

Furthermore, Rosenbaum and Doiron (2014) assume a sparse connectivity, that is $\kappa_{XY}(x) \ll 1$. Consequently, correlations among the fluctuating part of the synaptic inputs of different neurons can be neglected. Nevertheless, this does not mean that spike trains are always uncorrelated. Of course, these can be correlated due to correlations among instantaneous rates, but for given rates the spikes are assumed to be independent (c.f. Brunel, 2000). Accordingly, the expectations over the coupling $J_{XY}^{\kappa^\Gamma}$ and spike trains $S_{X,i}(t)$ can be taken independently, i.e. $\langle \mathrm{E}[J_{XY}^{\kappa^\Gamma}(x - i/N_Y)\, S_{X,i}(t)]\rangle = \langle \mathrm{E}[J_{XY}^{\kappa^\Gamma}(x - i/N_Y)]\rangle\, \langle \mathrm{E}[S_{X,i}(t)]\rangle$. Under these assumptions the mean current for a neuron at position x in population X is

$$\begin{aligned} \langle \mathrm{E}[I_X(x,t)]\rangle =& \left\langle \mathrm{E}\left[\sum_{i=1}^{N_E} J_{XE}^{\kappa^\Gamma}(x - i/N_E)\, S_{E,i}(t) - \sum_{i=1}^{N_I} J_{XI}^{\kappa^\Gamma}(x - i/N_I)\, S_{I,i}(t) + J_{XA}(x) \right] \right\rangle \\ =& \sum_{i=1}^{N_E} J_{XE}\, \kappa_E^\Gamma(x - i/N_E)\, \nu_E(i/N_E, t) \\ & - \sum_{i=1}^{N_I} J_{XI}\, \kappa_I^\Gamma(x - i/N_I)\, \nu_I(i/N_I, t) + J_{XA}(x). \end{aligned} \tag{4.7}$$

Using $N_E = q\, N$, $N_I = (1-q)\, N$, $J_{XY} = j_{XY}/\sqrt{N}$, and $J_{XA} = \sqrt{N}\, j_{XA}$ we obtain

$$\begin{aligned} \langle \mathrm{E}[I_X(x,t)]\rangle =& \sqrt{N} \sum_{i=1}^{N_E} j_{XE}\, q\, \kappa_E^\Gamma(x - i/N_E)\, \nu_E(i/N_E, t)\, \frac{1}{N_E} \\ & - \sqrt{N} \sum_{i=1}^{N_I} j_{XI}\, (1-q)\, \kappa_I^\Gamma(x - i/N_I)\, \nu_I(i/N_I, t)\, \frac{1}{N_I} + \sqrt{N}\, j_{XA}(x). \end{aligned} \tag{4.8}$$

In the continuum limit, that is turning from a network with a discrete number of neurons to a continuous neural field on Γ, the summation turns into an integration, i.e. $\sum_{i=1}^{N_X} f(i/N_X)/N_X$ becomes $\int_0^1 f(y)\,dy$. Of note, one does not take the limit $N \to \infty$ yet, but Rosenbaum and Doiron (2014) simply assume a large enough N such that a continuous instead of discrete network is a feasible simplification. This simplification has the advantage that we get rid of involved and inelegant operations in a discrete network space, but can apply continuous functions instead. For example, we are able to apply a continuous Fourier transform and we can use continuous kernels κ, like Gaussian functions. Additionally, not taking the limit at this point is important because we want to investigate the effects of finite in comparison to infinite network sizes in detail in subsequent sections. Under these assumptions, we obtain

$$\begin{aligned}\langle \mathrm{E}[I_X(x,t)]\rangle =& \sqrt{N}\int_0^1 j_{XE}\,q\,\kappa_I^\Gamma(x-y)\,\nu_E(y,t)\,dy \\ &- \sqrt{N}\int_0^1 j_{XI}\,(1-q)\,\kappa_E^\Gamma(x-y)\,\nu_I(y,t)\,dy + \sqrt{N}\,j_{XA}(x) \\ =& \sqrt{N}[(j_{XE}\,q\,\kappa_E^\Gamma) * \nu_E(x,t) \\ &- (j_{XI}\,(1-q)\,\kappa_I^\Gamma) * \nu_I(x,t) + j_{XA}(x)],\end{aligned} \tag{4.9}$$

where $*$ denotes circular convolution on the ring $\Gamma = (0,1]$. Thus, the mean input current to population X, from now on denoted by μ_X, can be defined as

$$\begin{aligned}\mu_X(x,t) &\coloneqq \langle \mathrm{E}[I_X(x,t)]\rangle \\ &= \sqrt{N}\,[\omega_{XE} * \nu_E(x,t) - \omega_{XI} * \nu_I(x,t) + j_{XA}(x)],\end{aligned} \tag{4.10}$$

with $\omega_{XY}(x)$ being the effective coupling from population Y to X, i.e. the product of the coupling strength, population fraction, and connection probability:

$$\omega_{XE}(x) = j_{XE}\,q\,\kappa_{XE}^\Gamma(x), \tag{4.11}$$

$$\omega_{XI}(x) = j_{XI}\,(1-q)\,\kappa_{XI}^\Gamma(x). \tag{4.12}$$

For a balanced solution with positive firing rates ($\forall x, \forall t : \nu_X(x,t) > 0$) to be stable for $N \to \infty$, it is required that ν_X and μ_X converge to finite values. Therefore, the following must hold:

$$\omega_{XE} * \nu_E(x,t) - \omega_{XI} * \nu_I(x,t) + j_{XA}(x) = O(1/\sqrt{N}). \tag{4.13}$$

For $N \to \infty$ we obtain a Fredholm equation of the first kind:

$$\omega_{XE} * \nu_E(x,t) - \omega_{XI} * \nu_I(x,t) + j_{XA}(x) = 0. \tag{4.14}$$

Next, one can take the spatial Fourier transform $\tilde{f}(n) = \int_0^1 \exp(-2\pi\,\mathrm{i}\,n\,x)\,f^\Gamma(x)\,dx$. This translates the system defined over the finite interval $\Gamma = (0,1]$ with periodic

boundary conditions to the Fourier domain with discrete Fourier modes $n \in \mathbb{Z}$. Accordingly, the inverse transform is $f^{\Gamma}(x) = \sum_{n=-\infty}^{\infty} \tilde{f}(n) \exp(2\pi \,\mathrm{i}\, n\, x)$.

Moreover, taking the Fourier transform has the advantage that the spatial convolution $*$ turns into multiplication in the Fourier domain:

$$\tilde{\omega}_{EE}(n)\, \tilde{\nu}_E(n,t) - \tilde{\omega}_{EI}(n)\, \tilde{\nu}_I(n,t) + \tilde{j}_{EA}(n) = 0, \tag{4.15}$$

$$\tilde{\omega}_{IE}(n)\, \tilde{\nu}_E(n,t) - \tilde{\omega}_{II}(n)\, \tilde{\nu}_I(n,t) + \tilde{j}_{IA}(n) = 0. \tag{4.16}$$

This set of linear equations can be solved and gives a steady-state firing rate fixpoint:

$$\nu_E^0(n) = \frac{\tilde{j}_{EA}\, \tilde{\omega}_{II} - \tilde{j}_{IA}\, \tilde{\omega}_{EI}}{\tilde{\omega}_{EI}\, \tilde{\omega}_{IE} - \tilde{\omega}_{EE}\, \tilde{\omega}_{II}}, \tag{4.17}$$

$$\nu_I^0(n) = \frac{\tilde{j}_{EA}\, \tilde{\omega}_{IE} - \tilde{j}_{IA}\, \tilde{\omega}_{EE}}{\tilde{\omega}_{EI}\, \tilde{\omega}_{IE} - \tilde{\omega}_{EE}\, \tilde{\omega}_{II}}. \tag{4.18}$$

Note that we omitted the variable dependency — here the Fourier mode n — on the right hand sides of the equations. If readability benefits and to avoid too long equations, we are going to apply this shortening in some of the following equations in this chapter too. The two equations 4.17 and 4.18 must hold at every Fourier mode $n \in \mathbb{Z}$ for which the denominator is non-zero, i.e. $\tilde{\omega}_{EI}(n)\, \tilde{\omega}_{IE}(n) - \tilde{\omega}_{EE}(n)\, \tilde{\omega}_{II}(n) \neq 0$. If $\tilde{\omega}_{EI}(n)\, \tilde{\omega}_{IE}(n) - \tilde{\omega}_{EE}(n)\, \tilde{\omega}_{II}(n) = 0$ at some Fourier mode n, then for a solution to exist, it must also hold

$$\tilde{j}_{EA}(n)\, \tilde{\omega}_{II}(n) - \tilde{j}_{IA}(n)\, \tilde{\omega}_{EI}(n) = \tilde{j}_{EA}(n)\, \tilde{\omega}_{IE}(n) - \tilde{j}_{IA}(n)\, \tilde{\omega}_{EE}(n) = 0. \tag{4.19}$$

Finally, the solutions of equations 4.15 and 4.16 are only valid if $\tilde{\nu}_X$ has a well-defined inverse Fourier transform; therefore, one requires at least that

$$\lim_{n \to \pm\infty} \frac{\tilde{j}_{EA}(n)\, \tilde{\omega}_{IY}(n) - \tilde{j}_{IA}(n)\, \tilde{\omega}_{EY}(n)}{\tilde{\omega}_{EI}(n)\, \tilde{\omega}_{IE}(n) - \tilde{\omega}_{EE}(n)\, \tilde{\omega}_{II}(n)} = 0, \tag{4.20}$$

for $Y \in \{E, I\}$. In other words, if external input is spatially tuned and not homogeneous, its tuning must be broader than the recurrent coupling. To be precise, the Fourier coefficients $\tilde{j}_{EA}$ and $\tilde{j}_{IA}$ in the numerator need to decay quicker than the ones of the recurrent connectivity profile in the denominator.

Trivially, for the case of spatially homogeneous input, i.e. the input is constant regardless of the post-synaptic neuron's position, $j_{XA}(x) = j_{XA}$, we see that there exists only a homogeneous solution for ν_E and ν_I. There, all neurons exhibit the same firing rate because the Fourier transform $\tilde{j}_{XA}(n)$ of a constant input j_{XA} is the constant value j_{XA} itself for the 0th Fourier mode and 0 for $n \neq 0$; hence, $\tilde{\nu}_X(n) = 0$ for $n \neq 0$. Because we are interested in stability due to recurrent coupling and heterogeneous activity despite homogeneous input, for the rest of the analysis we will assume input that is not only constant in time but also in space if not explicitly stated otherwise. Hence, we will use constant j_{XA} and neglect the space dependency.

4.1.3 Stability of the Homogeneous Fixpoint

So far we discussed the existence of a fixpoint of homogeneous activity in the limit of infinite size of a continuous neural field. Next, we are going to consider the stability of the fixpoint for a particular rate model and the influence of finite size effects. Given large mean input and a simple spiking neuron model like the leaky integrate and fire neuron (c.f. equation 4.3), the firing rate of a spiking neuron can be approximated as a threshold linear function (Rosenbaum and Doiron, 2014). Accordingly, one can define the rate dynamics of the excitatory and inhibitory populations by the following set of differential equations (c.f. Wilson and Cowan, 1972):

$$\tau \frac{\partial \nu_X(x,t)}{\partial t} = -\nu_X(x,t) + \gamma \left[\mu_X(x,t)\right]_+, \tag{4.21}$$

where $X \in \{E, I\}$, τ is the characteristic time scale of the network, and γ denotes the gain of the neurons. Moreover, $[\cdot]_+$ means rectification that accounts for the fact that firing rates can only be zero or positive. It is defined as

$$[x]_+ = \begin{cases} x & \text{for } x > 0 \\ 0 & \text{else.} \end{cases} \tag{4.22}$$

Because Rosenbaum and Doiron (2014) always use $\gamma = 1$ and different values of γ could also simply be factored into the afferent input and recurrent coupling strengths, the gain γ is dropped in our analysis. Further replacing μ_X by its definition from equation 4.10, we obtain the following system of coupled equations:

$$\tau \frac{\partial \nu_E(x,t)}{\partial t} = -\nu_E(x,t) + \sqrt{N}\left[\omega_{EE} * \nu_E(x,t) - \omega_{EI} * \nu_I(x,t) + j_{EA}\right]_+, \tag{4.23}$$

$$\tau \frac{\partial \nu_I(x,t)}{\partial t} = -\nu_I(x,t) + \sqrt{N}\left[\omega_{IE} * \nu_E(x,t) - \omega_{II} * \nu_I(x,t) + j_{IA}\right]_+. \tag{4.24}$$

Assuming a solution with strictly positive firing rates, that is $\forall t, \forall x : \nu_X(x,t) > 0$, it must also hold $\forall t, \forall x : \mu_X(x,t) > 0$, and one can safely ignore the non-linear rectification in equations 4.21, 4.23, and 4.24, i.e. $[\mu_X]_+ = \mu_X$. As a consequence, one can straightforwardly translate these equations into the spatial Fourier domain:

$$\tau \frac{\partial \tilde{\nu}_E(n,t)}{\partial t} = -\tilde{\nu}_E(n,t) + \sqrt{N}\left[\tilde{\omega}_{EE}(n)\,\tilde{\nu}_E(n,t) - \tilde{\omega}_{EI}(n)\,\tilde{\nu}_I(n,t) + \tilde{j}_{EA}(n)\right], \tag{4.25}$$

$$\tau \frac{\partial \tilde{\nu}_I(n,t)}{\partial t} = -\tilde{\nu}_I(n,t) + \sqrt{N}\left[\tilde{\omega}_{IE}(n)\,\tilde{\nu}_E(n,t) - \tilde{\omega}_{II}(n)\,\tilde{\nu}_I(n,t) + \tilde{j}_{IA}(n)\right]. \tag{4.26}$$

As a result, one can compute the corresponding fixpoint, given in the Fourier domain by

$$\tilde{\nu}_E^0(n) = \frac{\epsilon\,\tilde{j}_{EA} + \tilde{j}_{EA}\,\tilde{\omega}_{II} - \tilde{j}_{IA}\,\tilde{\omega}_{EE}}{\epsilon^2 - \epsilon\,\tilde{\omega}_{EE} + \epsilon\,\tilde{\omega}_{ii} + \tilde{\omega}_{EI}\,\tilde{\omega}_{IE} - \tilde{\omega}_{EE}\,\tilde{\omega}_{II}}, \tag{4.27}$$

$$\tilde{\nu}_I^0(n) = \frac{\epsilon\,\tilde{j}_{IA} + \tilde{j}_{EA}\,\tilde{\omega}_{IE} - \tilde{j}_{IA}\,\tilde{\omega}_{EE}}{\epsilon^2 - \epsilon\,\tilde{\omega}_{EE} + \epsilon\,\tilde{\omega}_{ii} + \tilde{\omega}_{EI}\,\tilde{\omega}_{IE} - \tilde{\omega}_{EE}\,\tilde{\omega}_{II}}, \tag{4.28}$$

with $\epsilon = 1/\sqrt{N}$. ϵ can be interpreted as a factor that accounts for finite size effects and vanishes with $N \to \infty$. Clearly, the finite size coefficient $\epsilon = 1/\sqrt{N}$ stems from the factor $\sqrt{N}$ in equations 4.25 and 4.26 which, in turn, is a direct consequence of the $1/\sqrt{N}$ scaling of the recurrent weights. If we take the limit of an infinite network size, i.e. $N \to \infty$ yielding $\epsilon \to 0$, the fixpoint is the same as in equations 4.15 and 4.16.

We can analyze the stability of the system by linearizing around the fixpoint:

$$\begin{pmatrix} \frac{\partial \tilde{\nu}_E(n,t)}{\partial t} \\ \frac{\partial \tilde{\nu}_I(n,t)}{\partial t} \end{pmatrix} = \boldsymbol{A}(n) \begin{pmatrix} \tilde{\nu}_E(n,t) - \tilde{\nu}_E^0(n) \\ \tilde{\nu}_I(n,t) - \tilde{\nu}_I^0(n) \end{pmatrix}, \tag{4.29}$$

with

$$\boldsymbol{A}(n) = \begin{pmatrix} \frac{1}{\tau}\left(-1 + \sqrt{N}\,\tilde{\omega}_{EE}(n)\right) & -\frac{1}{\tau}\sqrt{N}\,\tilde{\omega}_{EI}(n) \\ \frac{1}{\tau}\sqrt{N}\,\tilde{\omega}_{IE}(n) & \frac{1}{\tau}\left(-1 - \sqrt{N}\,\tilde{\omega}_{II}(n)\right) \end{pmatrix}. \tag{4.30}$$

The stability of the fixpoint can be determined by looking at the eigenvalues of the Jacobian $\boldsymbol{A}$ at every Fourier mode $n \in \mathbb{Z}$. If there exists at least one eigenvalue with positive real part for any Fourier mode n, the homogeneous fixpoint ν_X^0 is unstable. Especially situations in which $\nu_X^0(0)$ is stable, but at least one $\nu_X^0(n)$ for $n \neq 0$ is unstable are interesting. The 0th Fourier mode corresponds to the system's mean activity and all other modes $n \neq 0$ to spatial frequencies with $|n|$ revolutions per network extent Γ. On the one hand, the average activity in the network may remain stable. One does not necessarily encounter a runaway of excitation to theoretically infinite rates as one certainly would for $\boldsymbol{A}(0)$ having eigenvalues with positive real parts. On the other hand, since at least some non-zero spatial frequencies are amplified, spatially inhomogeneous patterns of activity can be observed even for homogeneous input. The potential structure and dynamics of such heterogeneous activity will be discussed in subsequent sections of this chapter.

Of note, the stability of the fixpoint is independent of the network input j_{XA} and is only determined by the recurrent weights and the network size. The time scale τ is not important because it is a common positive coefficient and cannot change the sign of any eigenvalue of $\boldsymbol{A}$. Because only the relative magnitudes — especially the signs — of $\boldsymbol{A}$'s eigenvalues are of interest and the absolute magnitudes are not important for the

stability analysis, Rosenbaum and Doiron (2014) rescale the Jacobian with $\tau\,\epsilon$; where ϵ is the finite size factor $\epsilon = 1/\sqrt{N}$. For this rescaling we obtain

$$\boldsymbol{A}_\epsilon(n) = \begin{pmatrix} -\epsilon + \tilde{\omega}_{EE}(n) & -\tilde{\omega}_{EI}(n) \\ \tilde{\omega}_{IE}(n) & -\epsilon - \tilde{\omega}_{II}(n) \end{pmatrix}. \tag{4.31}$$

As before, the fixpoint is stable if for all Fourier modes n the matrix's eigenvalues have only negative real parts. Given all $\tilde{\omega}_{XY}$ are real valued and $\boldsymbol{A}_\epsilon$ is a real matrix, this is equivalent to the conditions

$$\tilde{\omega}_{EI}(n)\,\tilde{\omega}_{IE}(n) - \tilde{\omega}_{EE}(n)\,\tilde{\omega}_{II}(n) - \epsilon\,(\tilde{\omega}_{EE}(n) - \tilde{\omega}_{II}(n)) + \epsilon^2 > 0 \tag{4.32}$$

and

$$\tilde{\omega}_{EE}(n) - \tilde{\omega}_{II}(n) - 2\epsilon < 0 \tag{4.33}$$

for all n.

In the limit of infinite network sizes $N \to \infty$ and $\epsilon \to 0$, one recognizes that the stability of the fixpoint is only determined by the topology and strength of the recurrent coupling, that is all ω_{XY}. For networks of a finite size with $\epsilon > 0$, the *rate leak terms* of equations 4.23 and 4.24 ($-\nu_E$ and $-\nu_I$) have an influence on the stability of the homogeneous fixpoint. As we will see later on, this rate leakage can stabilize the homogeneous activity.

4.1.4 Weight Scaling

Do the previous findings hold for other weight scalings, too? In the following we are going to discuss the effect of arbitrary weight scaling. This analysis is a contribution of the thesis author and not a review of the original paper.

Let $\eta(N)$ be an arbitrary scaling function for the recurrent weights J_{XY}, i.e. $J_{XY} \sim \eta(N)$. The afferent inputs scale with $N\,\eta(N)$ to guarantee that they are always of the same order as the recurrent inputs regardless of network size. For this scaling $\eta(N)$ we define rescaled weights $j_{XY} = J_{XY}/\eta(N)$ and $j_{XY} = J_{XA}/(N\,\eta(N))$. Accordingly, the mean current (c.f. equation 4.7) becomes

$$\begin{aligned} \langle \mathrm{E}[I_X(x,t)] \rangle =& N\,\eta(N) \sum_{i=1}^{N_E} j_{XE}\,q\,\kappa_E^\Gamma(x - i/N_E)\,\nu_E(i/N_E)\,\frac{1}{N_E} \\ & - N\,\eta(N) \sum_{i=1}^{N_I} j_{XI}\,(1-q)\,\kappa_I^\Gamma(x - i/N_I)\,\nu_I(i/N_I)\,\frac{1}{N_I} + N\,\eta(N)\,j_{XA}(x). \end{aligned} \tag{4.34}$$

This yields in the continuum limit:

$$\langle \mathrm{E}[I_X(x,t)] \rangle = N\,\eta(N)\,[\omega_{XE} * \nu_E(x,t) - \omega_{XI} * \nu_I(x,t) + j_{XA}], \tag{4.35}$$

with ω_{XE} and ω_{XI} defined as in equations 4.11 and 4.12. Note the factor $N\,\eta(N)$ in equation 4.35. For instance, for the original scaling with $\eta(N) = 1/\sqrt{N}$, we obtain $\sqrt{N}$ as in equation 4.10.

Moreover, the finite size coefficient ϵ is simply the inverse of the factor:

$$\epsilon = \frac{1}{N\,\eta(N)}. \tag{4.36}$$

Thus, if recurrent weights are not scaled with network size, viz. $\eta(N) = 1$, we obtain $\epsilon = 1/N$. Hence, it still holds that $\epsilon \to 0$ for $N \to \infty$. In fact, for any recurrent weight scaling $> O(1/N)$, we have $\epsilon \to 0$ in the limit of infinite networks. Consequently, for scalings $> O(1/N)$, the findings from the previous and subsequent sections are still valid. Of course, this is based on the assumption that the mean-field model is valid approximation to a spiking neuron network. One always requires that the system is well-defined by the average rates of neurons and fluctuations can be ignored.

Furthermore, for a scaling of $\eta(N) = 1/N$, the finite size factor basically becomes irrelevant with $\epsilon = 1$ for any network size N. In this case the system may still destabilize, but the homogeneous fixpoint can be stable for wider parameter ranges. The influence of nonzero ϵ values on stability will be discussed in the following section.

For simplicity, in the rest of the analysis in this chapter we will return to the original scaling by Rosenbaum and Doiron (2014) of $O(1/\sqrt{N})$ for recurrent weights.

4.1.5 Gaussian Connectivity Profile

We discussed under which conditions the fixpoint of homogeneous activity with strictly positive firing rates becomes unstable according to Rosenbaum and Doiron (2014). So far the setting was very general, comprising a simple threshold linear rate model for arbitrary effective couplings ω_{XY}. Next, to be more specific we are going to look at a particular realization, that is Gaussian shaped connectivity kernels.

Let $\kappa^{\Gamma}_{XY}(x) = g(x;\sigma_Y)$ with

$$g(x;\sigma_Y) = \frac{1}{\sqrt{2\pi}\sigma_Y} \sum_{l=-\infty}^{\infty} \exp\left(-\frac{(x+l)^2}{2\sigma_Y^2}\right) \tag{4.37}$$

is a wrapped Gaussian defined over the space $\Gamma = (0, 1]$. Due to the wrapping of the Gaussian, the network exhibits periodic boundary conditions. The corresponding Fourier transform is (Rosenbaum and Doiron, 2014)

$$\tilde{g}(n;\sigma_Y) = \exp(-2\pi^2\, n^2\, \sigma_Y^2). \tag{4.38}$$

We recall from equation 4.32 that in the limit of large network sizes ($\epsilon \to 0$) in order to have stable homogeneous activity one requires

$$\frac{\tilde{\omega}_{EE}(n)}{\tilde{\omega}_{II}(n)} < 1 \tag{4.39}$$

for all Fourier modes n. However, if we assume a Mexican hat profile, i.e. $\sigma_E < \sigma_I$, this condition is always violated regardless of the connection strengths $j_{EE} > 0$ and $j_{II} > 0$ for some $n \neq 0$, see below. Substituting equations 4.11, 4.12, and 4.38 into expression 4.39 yields

$$\begin{aligned} \frac{\omega_{EE}(n)}{\omega_{II}(n)} &= \frac{j_{EE}\, q\, \exp(-2\pi^2\, n^2\, \sigma_E^2)}{j_{II}\, (1-q)\, \exp(-2\pi^2\, n^2\, \sigma_I^2)} \\ &= \frac{j_{EE}\, q}{j_{II}\, (1-q)} \exp(-2\pi^2\, n^2\, (\sigma_E^2 - \sigma_I^2)). \end{aligned} \tag{4.40}$$

Due to the assumption of $\sigma_E < \sigma_I$, it holds that $\exp(-2\pi^2\, n^2\, (\sigma_E^2 - \sigma_I^2))$ grows without bounds for $n \to \pm\infty$. Additionally, for any choice of $j_{EE} > 0$ and $j_{II} > 0$ and $q \in [0, 1)$, the factor $(j_{EE}\, q)/(j_{II}\, (1-q))$ is positive and constant for all n. Hence, a large enough n can always be found such that

$$\frac{j_{EE}\, q}{j_{II}\, (1-q)} \exp(-2\pi^2\, n^2\, (\sigma_E^2 - \sigma_I^2)) = \frac{\tilde{\omega}_{EE}(n)}{\tilde{\omega}_{II}(n)} > 1 \tag{4.41}$$

Nevertheless, if the network size is finite,[1] the fixpoint may still be stable. Then ϵ is large enough such that

$$\frac{\tilde{\omega}_{EE}(n)}{2\epsilon + \tilde{\omega}_{II}(n)} < 1 \tag{4.42}$$

holds for all Fourier modes n even in case $\sigma_E < \sigma_I$.

Moreover, we can immediately see that for inverse Mexican hat and balanced profiles, i.e. $\sigma_E \geq \sigma_I$, even in an infinite system ($\epsilon = 0$), the fixpoint is stable for all Fourier modes n as soon as $\boldsymbol{A}_\epsilon(0)$ is stable. This is trivial because $0 < \exp(-2\pi^2\, n^2\, (\sigma_E^2 - \sigma_I^2)) \leq 1$ with $\sigma_E \geq \sigma_I$ for all modes n, and $\tilde{\omega}_{EE}(0)/\tilde{\omega}_{II}(0) < 1$, so we have

$$\frac{\tilde{\omega}_{EE}(n)}{\tilde{\omega}_{II}(n)} = \frac{\tilde{\omega}_{EE}(0)}{\tilde{\omega}_{II}(0)} \exp(-2\pi^2\, n^2\, (\sigma_E^2 - \sigma_I^2)) \leq \frac{\tilde{\omega}_{EE}(0)}{\tilde{\omega}_{II}(0)} < 1. \tag{4.43}$$

[1] Analogously, one can also think of infinite networks with weights that scale with $O(1/N)$ and use $\epsilon = 1$.

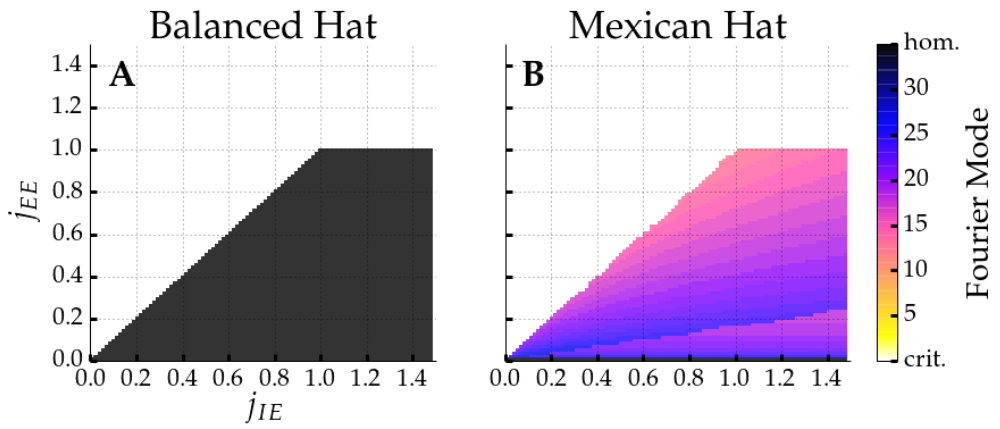

Figure 4.2: The Fourier mode $n \geq 0$ with the largest real part of all eigenvalues is color coded. Excitatory to excitatory j_{EE} and excitatory to inhibitory values j_{IE} are explored. Inhibitory connection strengths are fixed with $j_{II} = j_{EI} = 4.0$ and we set $q = 0.8$. For the balanced case it holds $\sigma_E = \sigma_I = 0.0125$, whereas for the Mexican hat we use $\sigma_E = 0.0125 < \sigma_I = 0.025$. Homogeneous activity where no eigenvalue has a positive real part are depicted in black. Critical runaway excitation where already Fourier mode $n = 0$ is unstable is depicted in white. Inhomogeneous non-runaway activity can only be observed for the Mexican hat network, see color gradient on the right (B).

Additionally, we have $\exp(-2\pi^2\, n^2\, (\sigma_E^2 + \sigma_I^2)) > 0$ for all modes n. Since $\boldsymbol{A}_\epsilon(0)$ is stable and with respect to equation 4.32, we get $\tilde{\omega}_{EI}(0)\,\tilde{\omega}_{IE}(0) - \tilde{\omega}_{EE}(0)\,\tilde{\omega}_{II}(0) > 0$. Hence,

$$\begin{aligned}\tilde{\omega}_{EI}(n)\,\tilde{\omega}_{IE}(n) - \tilde{\omega}_{EE}(n)\,\tilde{\omega}_{II}(n) =& [\tilde{\omega}_{EI}(0)\,\tilde{\omega}_{IE}(0) - \tilde{\omega}_{EE}(0)\,\tilde{\omega}_{II}(0)] \\ & \exp(-2\pi^2\, n^2\, (\sigma_E^2 + \sigma_I^2)) \\ >& 0.\end{aligned} \tag{4.44}$$

So for $\sigma_E \geq \sigma_I$ and $\boldsymbol{A}_\epsilon(0)$ having only negative real eigenvalues both conditions for stability are fulfilled for all Fourier modes n.

The previous findings are illustrated in figure 4.2. The Fourier modes $n \geq 0$ for which $\boldsymbol{A}_\epsilon(n)$ exhibits eigenvalues with the largest positive real part are depicted for a range of coupling strengths and two connection schemes with $\epsilon = 0$.[2] As one can see for a balanced network ($\sigma_E = \sigma_I$, figure 4.2A) we observe either homogeneous activity (black), or runaway excitation (white). In the former case the homogeneous fixpoint is stable and in the latter $\tilde{\nu}_X^0(0)$ is already unstable. The mean activity is amplified and firing rates grow without bounds. For Mexican hat networks (B) we have runaway excitation (white), too, but the homogeneous fixpoint is only stable for $j_{EE} = 0$. For the rest of the values there exists a third regime where $\boldsymbol{A}_\epsilon(0)$ has only eigenvalues with negative real parts, but for some $n \neq 0$ there exists an $\boldsymbol{A}_\epsilon(n)$ with eigenvalues having positive real parts. The corresponding modes n that are strongest amplified are coded

[2] It is sufficient to look at $n \geq 0$ because the Fourier transforms for n and $-n$ are only complex conjugates of each other, i.e. $\tilde{f}(n) = \tilde{f}^*(-n)$; hence $\mathrm{Re}(\tilde{f}(n)) = \mathrm{Re}(\tilde{f}(-n))$.

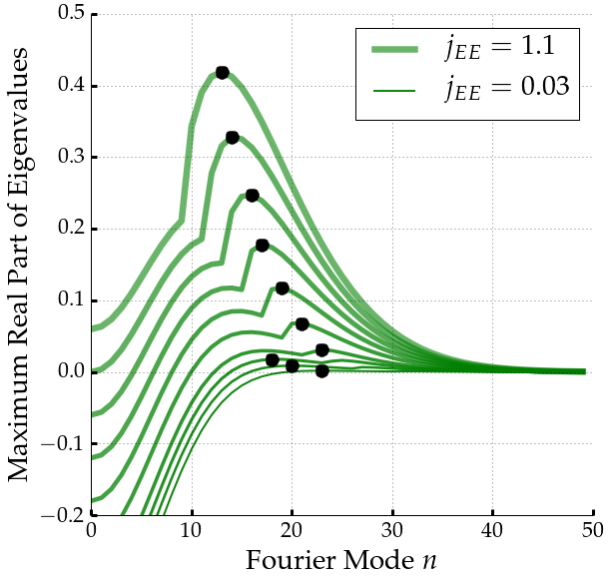

Figure 4.3: Maximum real part of all eigenvalues for all Fourier modes $n \geq 0$ up to mode 50 in a Mexican hat network. Parameters as for the Mexican hat network in figure 4.2 with $j_{IE} = 1.4$. The global maximum of each curve is indicated by the black dot.

by the color gradient in figure 4.2B. Similarly, figure 4.3 depicts the largest real parts of the eigenvalues for the first 50 Fourier modes for various connection strengths. One can see that for some coupling strengths, the 0th Fourier mode is not amplified, but some modes n with $n > 0$ are. There the homogeneous fixpoint is unstable and spatially inhomogeneous patterns may emerge.

Furthermore, we visualized the influence of finite size effects in figure 4.4. For $\epsilon > 0$ one can observe homogeneous activity even for Mexican hat networks with $j_{EE} > 0$. As expected, increasing ϵ also widens the parameter ranges where homogeneous activity is preserved.

Moreover, figure 4.5 shows the maximally amplified Fourier mode n for fixed connection strengths but varied widths σ_E and σ_I. Only for Mexican hat topologies $\sigma_E < \sigma_I$ the homogeneous fixpoint is unstable. Moreover, introducing finite size effects $\epsilon > 0$ (figure 4.5B), we can also observe homogeneous activity for some Mexican hat topologies.

Next, before going into detail what network activity is expected in case the homogeneous fixpoint is unstable, but rates could remain finite, we will discuss the effects of an extension of the model to a two-dimensional space as well as introducing adaptation.

4.1.6 Two-Dimensional Circular Gaussian

In the previous sections we discussed a one-dimensional mean-field approach. Here we are going to use the model by Rosenbaum and Doiron (2014) in a setting with a second spatial dimension.

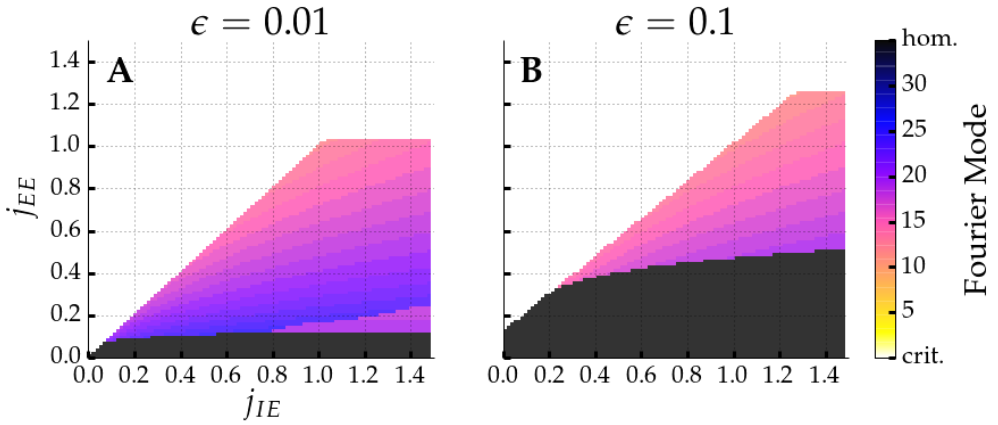

Figure 4.4: The Fourier mode $n \geq 0$ with the largest real part of all eigenvalues is color coded as a function of excitatory connection strengths in a Mexican hat network. Parameters as for the Mexican hat network in figure 4.2 except that $\epsilon = 0.01$ on the left (A) and $\epsilon = 0.1$ on the right (B). Homogeneous activity where no eigenvalue has a positive real part is depicted in black. Critical runaway excitation where already Fourier mode $n = 0$ is unstable is depicted in white. Increasing ϵ increases the parameter range where homogeneous activity is observed.

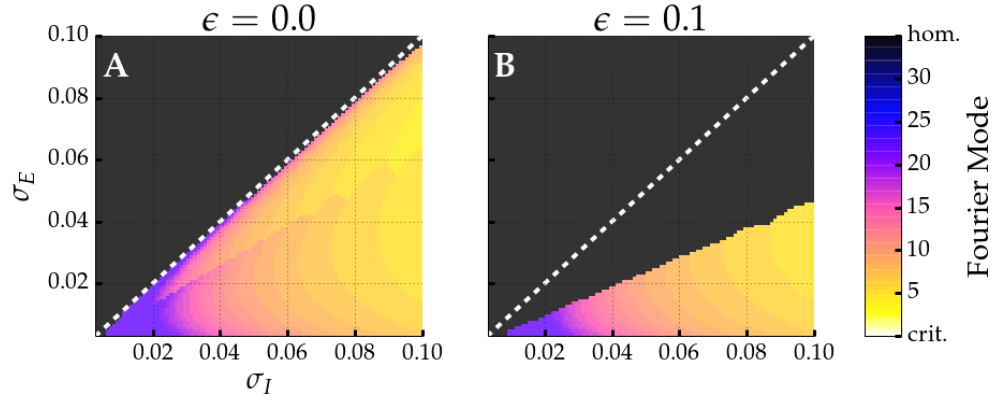

Figure 4.5: The Fourier mode $n \geq 0$ with the largest real part of all eigenvalues is color coded as a function of connectivity spread. Parameters as for networks in figure 4.2 except that $j_{EE} = 0.4$, $j_{IE} = 0.8$, and σ_E and σ_I are varied. Additionally, on the right (B) $\epsilon = 0.1$. White dotted line marks equality $\sigma_E = \sigma_I$. Inhomogenous activity can only be observed for Mexican hat topolgies. Given finite size networks $\epsilon > 0$ (B), for some Mexican hat topologies homogenous activity is stable.

The expansion of the one-dimensional to a two-dimensional model with $\Gamma_2 = (0,1] \times (0,1]$ and circular Gaussian connectivity profiles is straightforward. The circular Gaussian profile with width σ_Y wrapped around Γ_2 is defined as

$$\begin{aligned} g_2(x,y;\sigma_Y) &= \frac{1}{2\pi\sigma_Y^2} \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} \exp\left(-\frac{(x+k)^2+(y+l)^2}{2\sigma_Y^2}\right) \\ &= \frac{1}{2\pi\sigma_Y^2} \sum_{k=-\infty}^{\infty} \exp\left(-\frac{(x+k)^2}{2\sigma_Y^2}\right) \sum_{l=-\infty}^{\infty} \exp\left(-\frac{(y+l)^2}{2\sigma_Y^2}\right). \end{aligned} \tag{4.45}$$

The corresponding two-dimensional Fourier transform over the space Γ_2 with $\tilde{g}_2(m,n) = \int_0^1 \int_0^1 \exp(-2\pi\,\mathrm{i}\,m\,x)\,\exp(-2\pi\,\mathrm{i}\,n\,y) g_2(x,y)\,dx\,dy$ is then

$$\begin{aligned} \tilde{g}_2(n,m;\sigma_Y) &= \exp(-2\pi^2\,m^2\,\sigma_Y^2)\,\exp(-2\pi^2\,n^2\,\sigma_Y^2) \\ &= \exp(-2\pi^2\,(n^2+m^2)\,\sigma_Y^2). \end{aligned} \tag{4.46}$$

Therefore, all previous findings can be applied to the two-dimensional model analogously. The only difference is that the homogeneous fixpoint is stable if $\boldsymbol{A}_\epsilon(m,n)$ has only eigenvalues with negative real parts for the Cartesian product of all Fourier modes m and n.

4.2 Spike Frequency Adaption

In this section we are going to extend the model by Rosenbaum and Doiron (2014) by introducing adaptation. We will further show that the previous findings are still valid. This is a new contribution and not a review of the original paper.

Many neurons show a reduction in spike firing frequency for constant stimulation. This phenomenon is called spike frequency adaptation (SFA). Experiments showed that SFA is abundant among many cell types in visual cortex (Sanchez-Vives et al., 2000), and that this effect is usually stronger in excitatory than in inhibitory neurons (Nowak et al., 2003).

SFA can be easily included in the mean-field model by adding another linear differential equation for both populations (Augustin et al., 2013). We can do so by extending equations 4.23 and 4.24:

$$\tau \frac{\partial \nu_X(x,t)}{\partial t} = -\nu_X + [\sqrt{N}\,(\omega_{XE} * \nu_E - \omega_{XI} * \nu_I + j_{XA}) - w_X]_+, \tag{4.47}$$

$$\tau_w \frac{\partial w_X(x,t)}{\partial t} = -w_X + \tau_w\, b_X\, \nu_X, \tag{4.48}$$

with w_X the adaptation value of a neuron in population $X \in \{E,I\}$ at position x, τ_w the time constant of adaptation, and b_X denoting the adaptation strength. Usually, the dynamics of adaption are much slower than the characteristic time scale of the network, i.e. $\tau_w \gg \tau$ (Womble and Moises, 1992). Moreover, we assume that adaptation is an

intrinsic phenomenon of each neuron. It is independent of the network topology and, therefore, b_X scales with $O(1)$ for increasing network size.

Assuming a solution with strictly positive firing rates, we can drop the rectification and solve for the homogeneous fixpoint. The fixpoint in the Fourier domain is given by the following equations:[3]

$$\tilde{\nu}_E^0(n) = \frac{\tau_w b_I \tilde{j}_{EA}\epsilon + \tilde{j}_{EA}\epsilon + \tilde{j}_{EA}\tilde{\omega}_{II} - \tilde{j}_{IA}\tilde{\omega}_{EI}}{\Omega(n)}, \tag{4.49}$$

$$\tilde{\nu}_I^0(n) = \frac{\tau_w b_E \tilde{j}_{IA}\epsilon + \tilde{j}_{IA}\epsilon - \tilde{j}_{IA}\tilde{\omega}_{EE} + \tilde{j}_{EA}\tilde{\omega}_{IE}}{\Omega(n)}, \tag{4.50}$$

$$\tilde{w}_E^0(n) = \frac{\tau_w b_E(\tau_w b_I \tilde{j}_{EA}\epsilon + \tilde{j}_{EA}\epsilon + \tilde{j}_{EA}\tilde{\omega}_{II} - \tilde{j}_{IA}\tilde{\omega}_{EI})}{\Omega(n)}, \tag{4.51}$$

$$\tilde{w}_I^0(n) = \frac{\tau_w b_I(\tau_w b_E \tilde{j}_{IA}\epsilon + \tilde{j}_{IA}\epsilon - \tilde{j}_{IA}\tilde{\omega}_{EE} + \tilde{j}_{EA}\tilde{\omega}_{IE})}{\Omega(n)}, \tag{4.52}$$

$$\begin{aligned}\Omega(n) = &- \tau_w b_I \tilde{\omega}_{EE}\epsilon + \tau_w b_E \tilde{\omega}_{II}\epsilon + \tau_w b_E \epsilon^2 + \tau_w b_E \tau_w b_I \epsilon^2 + \tau_w b_I \epsilon^2 \\ &- \tilde{\omega}_{EE}\epsilon + \tilde{\omega}_{II}\epsilon + \epsilon^2 + \tilde{\omega}_{EI}\tilde{\omega}_{IE} - \tilde{\omega}_{EE}\tilde{\omega}_{II},\end{aligned} \tag{4.53}$$

with finite size factor $\epsilon = 1/\sqrt{N}$. Interestingly, in the limit of infinite network sizes ($\epsilon \to 0$), the fixpoint of the rates ν_E^0 and ν_I^0 is independent of adaptation and equal to the one encountered before (c.f. equations 4.15 and 4.16). Still, adaptation remains finite and can be non-zero, but has no influence on the rates anymore:

$$\tilde{\nu}_E^0(n) = \frac{\tilde{j}_{EA}\,\tilde{\omega}_{II} - \tilde{j}_{IA}\,\tilde{\omega}_{EI}}{\tilde{\omega}_{EI}\,\tilde{\omega}_{IE} - \tilde{\omega}_{EE}\,\tilde{\omega}_{II}}, \tag{4.54}$$

$$\tilde{\nu}_I^0(n) = \frac{\tilde{j}_{EA}\,\tilde{\omega}_{IE} - \tilde{j}_{IA}\,\tilde{\omega}_{EE}}{\tilde{\omega}_{EI}\,\tilde{\omega}_{IE} - \tilde{\omega}_{EE}\,\tilde{\omega}_{II}}, \tag{4.55}$$

$$\tilde{w}_E^0(n) = \frac{\tau_w\, b_E\,(\tilde{j}_{EA}\,\tilde{\omega}_{II} - \tilde{j}_{IA}\,\tilde{\omega}_{EI})}{\tilde{\omega}_{EI}\,\tilde{\omega}_{IE} - \tilde{\omega}_{EE}\,\tilde{\omega}_{II}}, \tag{4.56}$$

$$\tilde{w}_I^0(n) = \frac{\tau_w\, b_I(\tilde{j}_{EA}\,\tilde{\omega}_{IE} - \tilde{j}_{IA}\,\tilde{\omega}_{EE})}{\tilde{\omega}_{EI}\,\tilde{\omega}_{IE} - \tilde{\omega}_{EE}\,\tilde{\omega}_{II}}. \tag{4.57}$$

Furthermore, we can linearize around the fixpoint to obtain the Jacobian

$$\boldsymbol{A}(n) = \begin{pmatrix} \frac{1}{\tau}(-1 + \sqrt{N}\,\tilde{\omega}_{EE}(n)) & -\frac{1}{\tau}\sqrt{N}\,\tilde{\omega}_{EI}(n) & -\frac{1}{\tau} & 0 \\ \frac{1}{\tau}\sqrt{N}\,\tilde{\omega}_{IE}(n) & \frac{1}{\tau}(-1 - \sqrt{N}\,\tilde{\omega}_{II}(n)) & 0 & -\frac{1}{\tau} \\ b_E & 0 & -\frac{1}{\tau_w} & 0 \\ 0 & b_I & 0 & -\frac{1}{\tau_w} \end{pmatrix}. \tag{4.58}$$

[3]The results were verified using Wolfram Alpha: `https://www.wolframalpha.com/` (accessed 23.12.2015)

Next, we use a rescaling $\tau\epsilon$ with $\epsilon = 1/\sqrt{N}$. This time we cannot get rid of the characteristic time constant τ because the ratio between τ and τ_w matters for the relative magnitudes of the real parts of the eigenvalues. Thus, we obtain

$$\boldsymbol{A}_\epsilon(n) = \begin{pmatrix} -\epsilon + \tilde{\omega}_{EE}(n)) & -\tilde{\omega}_{EI}(n) & -\epsilon & 0 \\ \tilde{\omega}_{IE}(n) & -\epsilon - \tilde{\omega}_{II}(n)) & 0 & -\epsilon \\ \epsilon\,\tau\, b_E & 0 & -\epsilon\,\frac{\tau}{\tau_w} & 0 \\ 0 & \epsilon\,\tau b_I & 0 & -\epsilon\,\frac{\tau}{\tau_w} \end{pmatrix}. \tag{4.59}$$

For $N \to \infty$ ($\epsilon \to 0$) we have

$$\boldsymbol{A}_0(n) = \begin{pmatrix} \tilde{\omega}_{EE}(n) & -\tilde{\omega}_{EI}(n) & 0 & 0 \\ \tilde{\omega}_{IE}(n) & -\tilde{\omega}_{II}(n) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{4.60}$$

For all Fourier modes n this matrix has two vanishing eigenvalues and typically two non-zero eigenvalues. These two non-zero eigenvalues are equal to the ones of the infinite system without adaptation from before (c.f. equation 4.31 with $\epsilon = 0$). This is easy to see if we solve for the eigenvalues λ via $\det(\boldsymbol{A}_0(n) - \lambda\,\boldsymbol{E}) = 0$, where $\boldsymbol{E}$ is the identity matrix, and use Laplace's formula:

$$\begin{aligned} \det(\boldsymbol{A}_0(n) - \lambda\,\boldsymbol{E}) &= \det \begin{pmatrix} \tilde{\omega}_{EE}(n) - \lambda & -\tilde{\omega}_{EI}(n) & 0 & 0 \\ \tilde{\omega}_{IE}(n) & -\tilde{\omega}_{II}(n) - \lambda & 0 & 0 \\ 0 & 0 & -\lambda & 0 \\ 0 & 0 & 0 & -\lambda \end{pmatrix} \\ &= -\lambda \det \begin{pmatrix} \tilde{\omega}_{EE}(n) - \lambda & -\tilde{\omega}_{EI}(n) & 0 \\ \tilde{\omega}_{IE}(n) & -\tilde{\omega}_{II}(n) - \lambda & 0 \\ 0 & 0 & -\lambda \end{pmatrix} \\ &= \lambda^2 \det \begin{pmatrix} \tilde{\omega}_{EE}(n) - \lambda & -\tilde{\omega}_{EI}(n) \\ \tilde{\omega}_{IE}(n) & -\tilde{\omega}_{II}(n) - \lambda \end{pmatrix}. \end{aligned} \tag{4.61}$$

The remaining determinant is the same as for the eigenvalue problem of the matrix given in equation 4.31 with $\epsilon = 0$. Hence, in the limit of an infinite number of neurons, the previously discussed findings are still valid for the extended system with adaptation.

4.2.1 Finite Networks

Also for finite networks the influence of adaptation on stability is at most docile. This holds even in case adaptation is unequal among populations. For example, small to considerably large strengths of b_E and $b_I = 0.1 b_E$ yield only minor modulations of

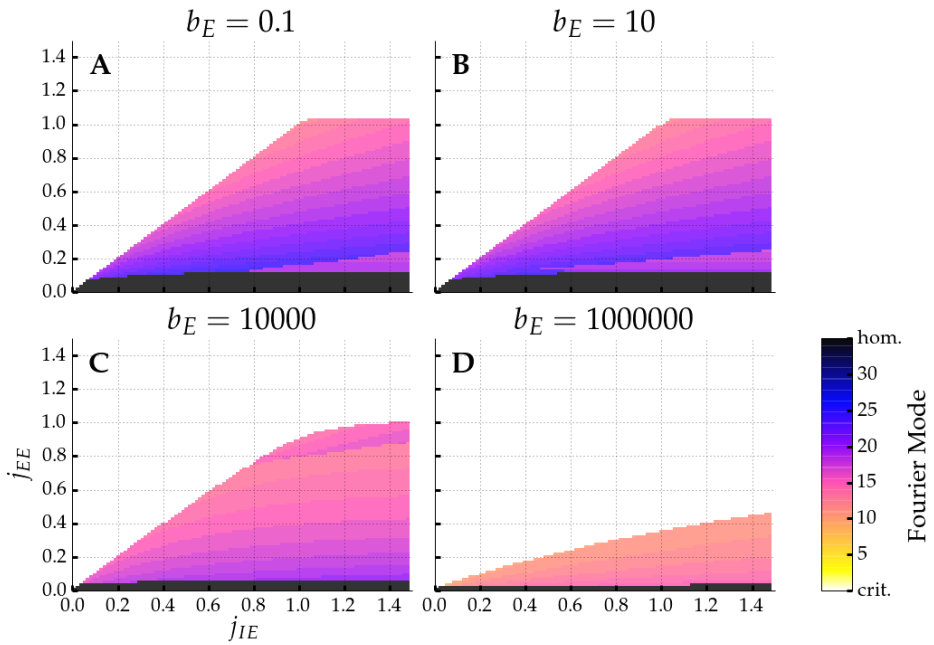

Figure 4.6: The Fourier mode $n \geq 0$ with the largest real part of all eigenvalues is color coded. Excitatory to excitatory j_{EE} and excitatory to inhibitory values j_{IE} are explored. Other parameters chosen as for the Mexican hat network in figure 4.2 except $\epsilon = 0.01$. Additional parameters are $\tau = 0.01$ and $\tau_w = 0.25$. b_E is varied across plots. Because spike frequency adaptation (SFA) is weaker in inhibitory cells (Nowak et al., 2003), we use $b_I = 0.1 b_E$.

the system (figure 4.6). Yet, for huge, biologically implausible adaption strengths, networks are less likely to be stable and $\boldsymbol{A}_\epsilon(0)$ exhibits at least one eigenvalue with positive real part for more parameter settings as shown in figure 4.6D. Of note, this pronounced area of runway activity is an effect of $b_E \neq b_I$ and vanishes for $b_E = b_I$ (data not shown). Furthermore, the influence on the eigenvalues for each Fourier mode n can be neglected for a wide range of adaption strengths too. Only huge values of b_E change the shape of the curves in figure 4.7. Similar observations can be made for parameter τ_w where only very short time scales beyond biologically plausible values show considerable effects on stability (data not shown).

This leaves us with the question: Why bother with introducing spike frequency adaptation if its influence on the homogeneous fixpoint and the fixpoint's stability are negligible? As we will see next, adaptation plays an important role in case the homogeneous fixpoint is unstable and the network settles into a spatially inhomogeneous activity state.

4.3 Inhomogeneous Activity

So far we discussed the mean-field model by Rosenbaum and Doiron (2014) in great detail. The model gives an intuition for which spiking neuron network configurations

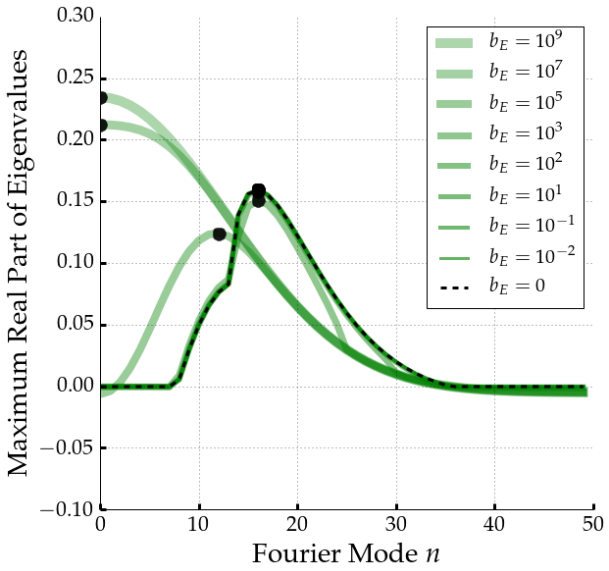

Figure 4.7: Maximum real part of all eigenvalues for all Fourier modes $n \geq 0$ up to mode 50 in a Mexican hat network with SFA. Parameters as in figure 4.6, but we fix $j_{IE} = 0.8$, $j_{IE} = 0.6$. b_E is varied across curves and we use $b_I = 0.1 b_E$. Up until $b_E = 1000$ all graphs are laying tightly on top of each other. Only for huge strengths of b_E one can observe a qualitative change in the curve in comparison to $b_E = 0$ (black dotted line). Due to SFA two eigenvalues are always very close to 0, therefore curves are not dropping considerably below zero as they do in figure 4.3.

the balance between recurrent excitation and inhibition can no longer be maintained. More precisely, for Mexican hat networks with wider inhibitory than excitatory connection spread, the fixpoint of asynchronous homogeneous activity is unstable, but firing rates may remain finite. Particular spatial Fourier modes are amplified and inhomogeneous activity may emerge. Nonetheless, the theory does not go far enough to describe the resulting inhomogeneous network activity. Therefore, in the next sections we will further reduce the model by Rosenbaum and Doiron (2014) to a single population network. Thereafter, we are going to argue that this reduced model resembles an even simpler one by Hansel and Sompolinsky (1998). The latter can be treated analytically and gives explicit descriptions of inhomogeneous activity.

4.3.1 The Single Population Model

We discussed that if assuming a homogeneous solution with strictly positive firing rates, the mean-field equations $\tau \, \partial \nu_X(x,t)/\partial t = -\nu_X + [\mu_X]_+$ for $X \in \{E, I\}$ (also refer to equations 4.23 and 4.24) can be solved analytically because the non-linear rectification $[\cdot]_+$ can be ignored. However, in case of inhomogeneous activity, some of the neurons are certainly not firing. For these silent neurons we have $\mu_X < 0$. This means the rectification plays an important role and takes care that no neuron can exhibit negative firing rates. For Gaussian coupling schemes this non-linearity makes it

difficult to solve for the inhomogeneous fixpoint to provide an analytical description of the network state.

Many studies have approached this or similar problems with various techniques, assumptions, simplifications, and various degrees of lucidity, for an overview the reader is directed to Coombes (2005). For instance, Pinto and Ermentrout (2001) assume no inhibitory to inhibitory connectivity or Guo and Chow (2005) use a special Mexican hat like coupling scheme with $\kappa(x) = A\,\exp(-a|x|) - \exp(-|x|)$ and $A > 1, a > 1$ that is termed *Wizard hat* due to its sharp cusp around the origin. Similarly, we are turning our focus to an older model by Hansel and Sompolinsky (1998) that assumes a simple sinusoidal connectivity scheme. This connectivity profile allows analytical results for spatially inhomogeneous activity. Fortunately, the model by Hansel and Sompolinsky (1998) is fathomable and one can nicely point out analogies to the one by Rosenbaum and Doiron (2014).

Besides the simpler coupling scheme, the network by Hansel and Sompolinsky (1998) violates *Dale's principle* (Dale, 1935) and makes no distinction between excitatory and inhibitory neurons. For better comparison with the model by Rosenbaum and Doiron (2014), we are going to reduce their mean-field network to a single population version too. This reduction is not part of the original paper, but a new contribution by the thesis author.

By removing one population and assuming that the remaining population forms excitatory as well as inhibitory connections with itself, we obtain a single differential equation:

$$\tau \frac{\partial \nu(x,t)}{\partial t} = -\nu(x,t) + \sqrt{N}\,[(\omega_E - \omega_I) * \nu(x,t) + j_A]_+, \tag{4.62}$$

where $*$ denotes circular convolution on $\Gamma = (0, 1]$. In addition, ν is the dynamic firing rate at position x. We neglect the subscript ν_X because we are only dealing with a single population. Moreover, j_A is the afferent input, τ is the network time constant, and N the network size. ω_E as well as ω_I denote the recurrent coupling defined similar to equations 4.11 and 4.12, but because of having only one population we do not need the population fraction factors q and $1 - q$:

$$\omega_E(x) = j_E\,\kappa_E^\Gamma(x), \tag{4.63}$$

$$\omega_I(x) = j_I\,\kappa_I^\Gamma(x). \tag{4.64}$$

Again, we can solve for the homogeneous fixpoint with strictly positive firing rates in the spatial Fourier domain:

$$\tilde{\nu}^0(n) = \frac{\tilde{j}_A(n)}{\tilde{\omega}_I(n) - \tilde{\omega}_E(n) + \epsilon}, \tag{4.65}$$

with $\epsilon = 1/\sqrt{N}$ the finite size factor.

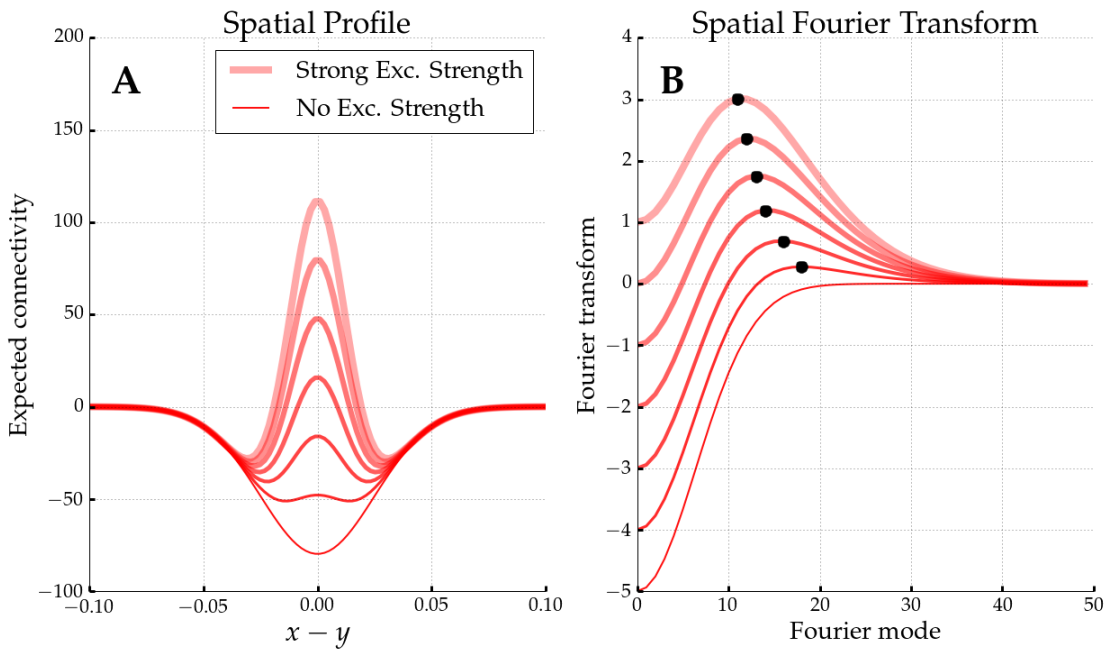

Figure 4.8: Various Mexican hat connectivity profiles $\omega_E - \omega_I$ with $\sigma_E < \sigma_I$ are shown on the left (A) for different excitatory connections strenghts j_E. The Fourier transform is shown on the right (B) for $n \geq 0$. If positive values are present, the maxima of the Fourier transforms are marked by the black dots.

Moreover, we can basically read off the condition for stability directly from the differential equation 4.62. The homogeneous fixpoint is stable if

$$\mathrm{Re}(\tilde{\omega}_E(n) - \tilde{\omega}_I(n)) - \epsilon < 0 \tag{4.66}$$

for all Fourier modes n.

The beauty of this simplification is that for Gaussian connectivity profiles $\kappa_Y^\Gamma(x) = g(x;\sigma_Y)$ (c.f. equation 4.37) we can summarize the coupling term $\omega(x) = \omega_E(x) - \omega_I(x)$ into a single expression $j_E\, g(x;\sigma_E) - j_I\, g(x;\sigma_I)$ with a Fourier transform given by

$$\tilde{\omega}(n) = \tilde{\omega}_E(n) - \tilde{\omega}_I(n) = j_E\, \exp(-2\pi^2\, n^2\, \sigma_E^2) - j_I\, \exp(-2\pi^2\, n^2\, \sigma_I^2). \tag{4.67}$$

We can plot this Fourier transform for various connection strengths and connectivity profiles as depicted in figures 4.8 and 4.9. There one can immediately see which spatial frequencies are amplified (positive parts) and which are damped (negative parts). Increasing j_E as wells σ_I shifts the maxima (black dots) of the curves slightly to the left leading to the amplification of lower spatial frequencies. Again, the 0th Fourier mode determines if there is runaway excitation ($\tilde{\omega}_E(0) - \tilde{\omega}_I(0) - \epsilon > 0$) or if firing rates have a chance to remain finite.

Nonetheless, the question remains which network activity is observed in case there is no runaway excitation, but the homogeneous fixpoint is unstable. Unfortunately, we

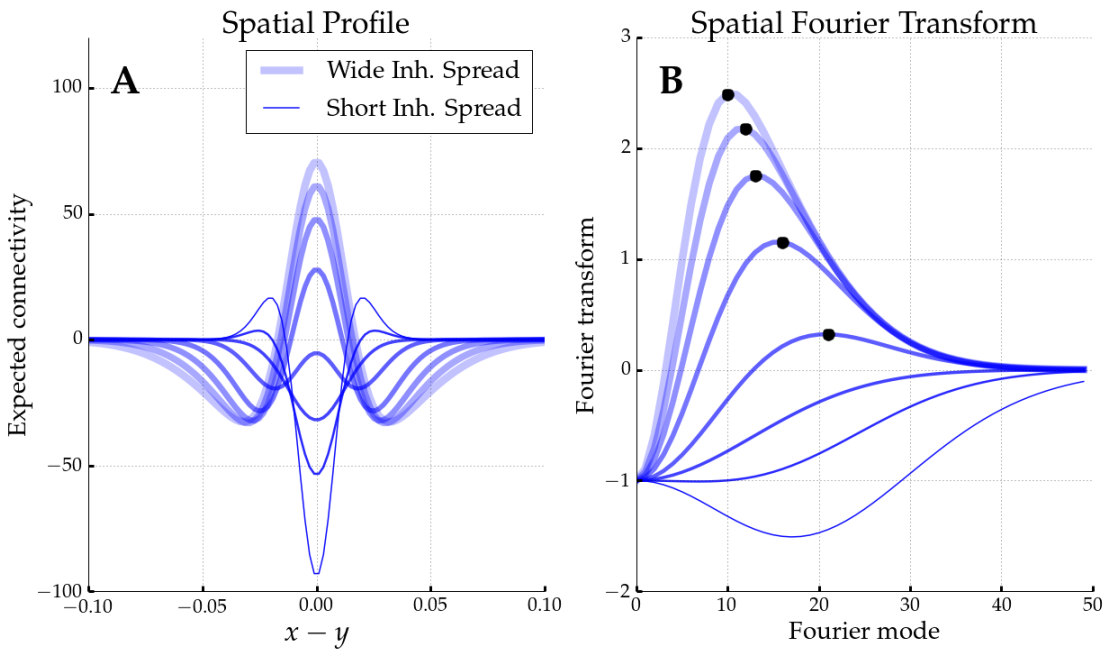

Figure 4.9: Various connectivity profiles $\omega_E - \omega_I$ from inverse hat to Mexican hat are shown on the left (A) for different inhibitory connections spreads σ_I. The Fourier transform is shown on the right (B) for $n \geq 0$. If positive values are present, the maxima of the Fourier transforms are marked by the black dots.

cannot tell due to the non-linear rectification. The least we can do is provide a self-consistent equation for any fixpoint $\partial\nu(x,t)/\partial t = 0$ in the original domain Γ using a Fourier expansion $f(x) = \tilde{f}(0) + 2\sum_{n=1}^{\infty} |\tilde{f}(n)| \cos(2\pi\, n\, (x + \arg(\tilde{f}(n))))$:

$$\nu(x,t) = \sqrt{N}\left[\tilde{\omega}(0)\,\nu_0(t) + 2\sum_{n=1}^{\infty} |\tilde{\omega}(n)|\,\nu_n(t)\,\cos(2\pi\, n\,(x + \phi_n(t))) + j_A\right]_+, \tag{4.68}$$

$$\phi_n(t) = \psi_n(t) + \arg(\tilde{\omega}(n)), \tag{4.69}$$

where $|z|$ is the absolute and $\arg(z) = \operatorname{atan2}(\operatorname{Im}(z), \operatorname{Re}(z))/(2\pi)$ the argument function. $\psi_n(t) = \arg(\tilde{\nu}(n,t))$ and $\nu_0(t) = \tilde{\nu}(0,t)$ as well as $\nu_n(t) = 2|\tilde{\nu}(n,t)|$ are the phase and amplitude coefficients of the activity, respectively. Hence, if we knew the spatial Fourier coefficients $\nu_0(t)$, $\nu_n(t)$, and phases $\psi_n(t)$, we would have a complete description of all fixpoints. Yet, due to the non-linearity $[\cdot]_+$, we cannot solve for these coefficients. However, by looking at the Fourier expansion in equation 4.69 we can recognize the system's resemblance to another model by Hansel and Sompolinsky (1998).

The mean-field model by Hansel and Sompolinsky (1998) reads

$$\tau\frac{\partial\nu(x,t)}{\partial t} = -\nu(x,t) + \sqrt{N}\,[(j_0 + j_1\cos(2\pi\, x)) * \nu(x,t) + j_A]_+, \tag{4.70}$$

where j_0 and j_1 are the recurrent coupling strength parameters. In contrast to the previous model, these can take positive as well as negative values, i.e. $j_0, j_1 \in \mathbb{R}$. The rest of

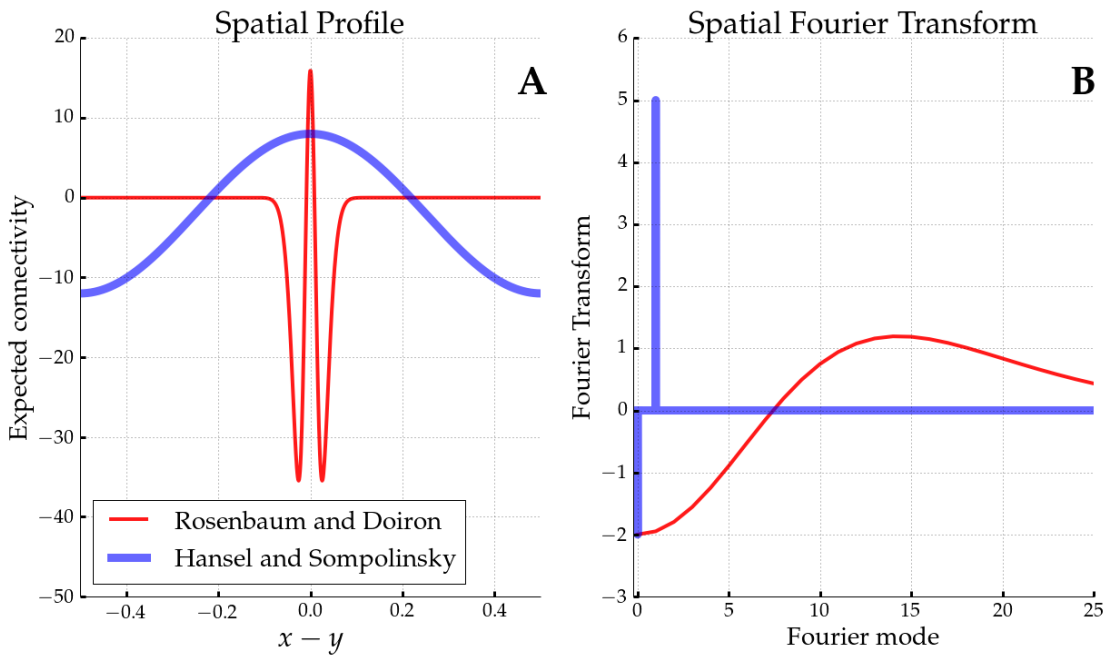

Figure 4.10: Local Gaussian connectivity profile from Rosenbaum and Doiron (2014) in red and global sinusoidal connectivity profile from Hansel and Sompolinsky (1998) in blue on the left (A). The Fourier transform is shown on the right (B) for $n \geq 0$.

the parameters are analogous to the previous model. Looking at the fixpoint equation in form of the Fourier expansion reveals the similarity to the single population model discussed before:

$$\nu(x,t) = \sqrt{N}\,[j_0\,\nu_0(t) + j_1\,\nu_1(t)\,\cos(2\pi(x+\psi(t))) + j_A]_+\,. \tag{4.71}$$

Basically, it is the same expression as in equation 4.69, but instead of an infinite sum of spatial Fourier coefficients, the series is truncated after the 1st Fourier component including only $n = 0$ and $n = 1$. The advantage is that despite the non-linear rectification $[\cdot]_+$, one can solve this analytically. Before going into detail, we want to briefly look into the differences between the two models.

A Gaussian connectivity profile instantiates a local coupling scheme where neurons are more likely to be connected with their nearest neighbors. The connection probability usually vanishes quickly with increasing distance among cell pairs. However, the scheme used by Hansel and Sompolinsky (1998) is a global sinusoidal coupling where even neurons far apart are connected. This difference can be nicely visualized. Indeed, as illustrated in figure 4.10A, the sinusoidal coupling can exhibit a topology similar to a Gaussian Mexican hat on a global scale with longer ranging inhibitory than excitatory coupling. However, the discrepancy between the two coupling schemes becomes very clear when looking at their Fourier transforms given in figure 4.10B. The transform of the Gaussian profile shows that a whole range of spatial frequencies is amplified. In contrast, the sinusoidal coupling can only exhibit sharp peaks at $n = 0$ and $n = \pm 1$.

We admit that this constitutes a substantial difference between the two systems. As a consequence, the analytical solutions for the model by Hansel and Sompolinsky (1998) cannot be directly applied to the complex Gaussian coupling. Nevertheless, the solutions can give an intuition about what kind of heterogeneous activity may emerge in the model by Rosenbaum and Doiron (2014) and how the simulation results of the next chapter can be interpreted.

4.3.2 The Sinusoidal Coupling Model

We will briefly discuss the analytical solutions for spatially inhomogeneous activity in the model by Hansel and Sompolinsky (1998) in this section. The following results are excerpts from this paper; however, some equations were taken from Roxin et al. (2006) whose expressions are sometimes more compact and to the point. We will deviate from the notation used in the original paper in order to be in line with the notation used by Rosenbaum and Doiron (2014). Moreover, Hansel and Sompolinsky (1998) assume that recurrent weights scale with $O(1/N)$ in contrast to the scalings of $O(1/\sqrt{N})$ and $O(1)$ that were previously discussed. For better comparison we will change results from Hansel and Sompolinsky (1998) to fit the scaling methods used within this chapter.

To compute the fixpoint, one can rewrite the model from equation 4.71 in terms of the Fourier coefficients $\nu_0(t)$, $\nu_1(t)$, and the phase $\psi(t)$ (Roxin et al., 2006):

$$\tau \frac{d\nu_0(t)}{dt} = -\nu_0(t) + \int_0^1 [\mu(x,t)]_+ \, dx, \tag{4.72}$$

$$\tau \frac{d\nu_1(t)}{dt} = -\nu_1(t) + 2\int_0^1 \cos(2\pi(x+\psi(t))) \, [\mu(x,t)]_+ \, dx, \tag{4.73}$$

with the mean-field current $\mu(x,t)$ defined as

$$\mu(x,t) = \sqrt{N}\,(j_0\,\nu_0(t) + j_1\,\nu_1(t)\,\cos(2\pi(x+\psi(t))) + j_A)\,. \tag{4.74}$$

This set of equations can be combined to solve for fixpoints even in case of negative currents, i.e. $\mu(x,t) < 0$ for some x.

First of all, the homogeneous fixpoint is

$$\nu^0 = \frac{j_A}{\epsilon - j_0}, \tag{4.75}$$

where ϵ is again the finite size factor with $\epsilon = 1/\sqrt{N}$. The homogeneous fixpoint is stable for $j_0 - \epsilon < 0$ and $j_1/2 - \epsilon < 0$. So if $j_0 > \epsilon$, one observes runaway activity and for $j_1 > 2\epsilon$ one encounters a Turing bifurcation (Roxin et al., 2006). In contrast to the previous model, an analytical expression for the system's behavior in case $j_0 - \epsilon < 0$ but $j_1/2 - \epsilon > 0$ can be provided. The profile is completely determined once the coefficients $\nu_0(t)$ and $\nu_1(t)$ are known. These can be computed by combining equations 4.72, 4.73,

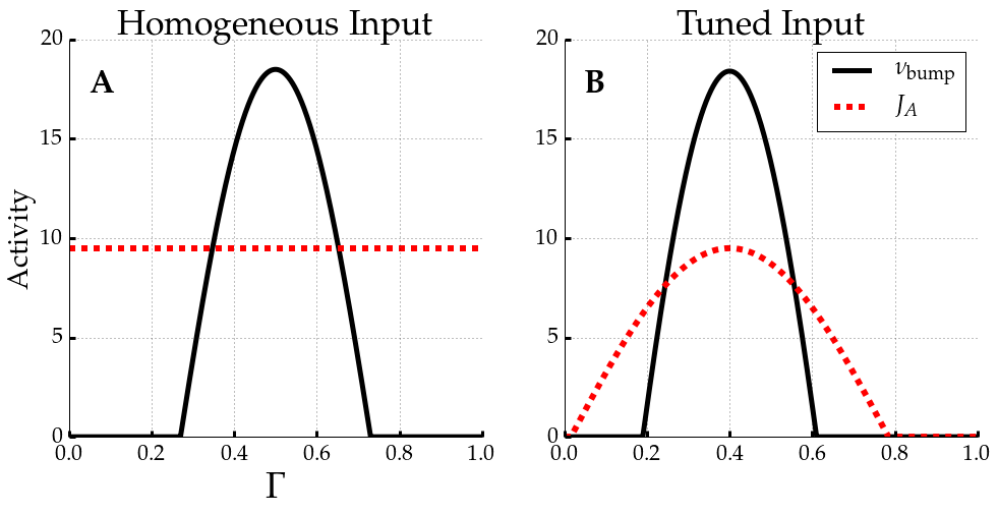

Figure 4.11: Bump solutions ν_{bump} (black lines) for two different inputs (red dotted lines, $J_A = j_A\sqrt{N}$). Solved via Euler integration. Parameters are $N = 1000$, $j_0 = -0.07$, and $j_1 = 0.15$. On the left (A) spatially constant input $j_A = 0.3$ and on the right (B) tuned input $j_A(x) = [0.3\cos(1.3\pi\,(x - 0.4))]_+$ are used. For the constant input (A) the phase is arbitrary (here 0.5). Instead of modifying the bump shape, the tuned input (B) rather selects among the potential bump solutions. The bump peak's position corresponds to the peak location of the input (here 0.4).

as well as 4.74, and some elaborate algebra (Roxin et al., 2006). In this case one observes a single stable bump solution with

$$\nu_{\text{bump}}(x) = \sqrt{N}\,[(j_0\,\nu_0 + j_1\,\nu_1\,\cos(2\,\pi\,(x + \psi_{\text{bump}})) + j_A]_+, \tag{4.76}$$

where the phase ψ_{bump} is arbitrary due to the translational symmetry along the ring Γ. This state is also called *marginal phase* (Hansel and Sompolinsky, 1998) to indicate that the system does not reach a single fixpoint but a line of potential fixpoints. The half-width x_{bump} of the active region where $\nu_{\text{bump}}(x) > 0$ can also be calculated using

$$j_1\left(x_{\text{bump}} - \frac{1}{4\pi}\sin(4\pi\,x_{\text{bump}})\right) = \epsilon, \tag{4.77}$$

with finite size factor $\epsilon = 1/\sqrt{N}$.

Moreover, the bump solution is not necessarily stable and rates may still diverge. For the bump being a stable fixpoint, one additionally requires

$$j_0 < \frac{\pi\,\epsilon}{2\pi\,x_{\text{bump}} - \tan(2\pi\,x_{\text{bump}})}. \tag{4.78}$$

Otherwise, the system undergoes an amplitude instability and the rates within the active region of the bump diverge. An example of a stable bump solution for homogeneous input is given in figure 4.11A.

Of note, the bump solution requires a finite network $\epsilon > 0$. In case $\epsilon = 0$, the only solution satisfying equation 4.77 is $x_{\text{bump}} = 0$. This would yield an impossible bump with no extent. Likewise, for decreasing ϵ or increasing the weight j_1, the half-width x_{bump} of the bump shrinks. This is true since $x_{\text{bump}} - 1/(4\pi)\sin(4\pi\, x_{\text{bump}})$ is monotonically increasing in x_{bump} because

$$\frac{d}{dx_{\text{bump}}}\left(x_{\text{bump}} - \frac{1}{4\pi}\sin(4\pi\, x_{\text{bump}})\right) = 1 - \cos(4\pi\, x_{\text{bump}}) \geq 0. \tag{4.79}$$

4.3.3 Tuned Input and Tuning Sharpening

For tuned input where $j_A(x)$ is not a constant homogeneous input, but follows a bump profile with half-width wider than the half-width of the network response for homogeneous input, one can observe that the network activity $\nu_{\text{bump}}(x)$ largely follows the original bump structure. The bump profile is only slightly modulated by the input $j_A(x)$, for details see Hansel and Sompolinsky (1998). Thus, the networks response is sharper tuned than the input. Moreover, instead of modulating the width of the network response, inhomogeneous input $j_A(x)$ rather selects a particular phase ψ_{bump} among the line attractor. An example of a response to tuned input is provided in figure 4.11B, where the input's peak determines the peak location of the bump.

4.3.4 Moving Bump Activity

For any translational asymmetry like noisy input or randomly sampled connectivity, the bump solution becomes unstable in the sense that it starts wandering around. Such perturbations along the ring are marginal, their magnitude does not decay with time (Hansel and Sompolinsky, 1998).

Moreover, we can add adaptation to the model similar to what was already discussed in previous sections. In this case we obtain the following set of equations:

$$\tau\frac{\partial\nu(x,t)}{\partial t} = -\nu(x,t) + [\sqrt{N}\,((j_0 + j_1\cos(2\,\pi\,x)) * \nu(x,t) + j_A) - w(x,t)]_+, \tag{4.80}$$

$$\tau_w\frac{\partial w(x,t)}{\partial t} = -w(x,t) + \tau_w\, b\,\nu(x,t). \tag{4.81}$$

According to Hansel and Sompolinsky (1998) the bump solution of the marginal phase starts moving along the ring in case

$$j_1 > 2\epsilon\left(1 - \frac{\tau}{\tau_w}\right) \tag{4.82}$$

and

$$b > \frac{\tau}{\tau_w^2}. \tag{4.83}$$

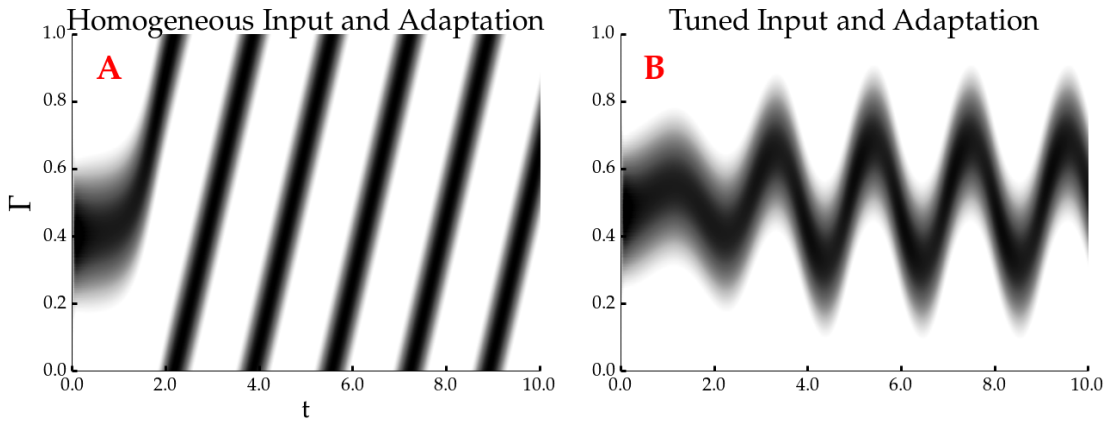

Figure 4.12: Bump solutions ν_{bump} for two different inputs in combination with adaptation. The space Γ is depicted on y-axis, time on x-axis, and activity in grey-scale. Solved via Euler integration with a stepisze $dt = 0.001$ and random initial conditions. Parameters are $N = 1000$, $j_0 = -0.07$, $j_1 = 0.15$, $b = 0.3$, $\tau = 0.01$, and $\tau_w = 0.25$. On the left (A) spatially constant input $j_A = 0.3$ and on the right (B) tuned input $j_A(x) = 0.3\cos(0.35\pi\,(x - 0.5))$ are used. For the constant input (A) the bump moves across the entire network extent with a constant velocity $v \approx 0.6$ (matching equation 4.84). The tuned input (B) confines the bump movement. The bump zig-zags around the peak location (here 0.5).

Clearly, this is not an unlikely condition because of the long time scales of adaptation, i.e. $\tau_w \gg \tau$ (Womble and Moises, 1992).

Moreover, Hansel and Sompolinsky (1998) showed that perturbations of the phases of firing rates as well as adaptation are both marginal and do not decay with time. Surprisingly, this means that the bump movement is determined only by τ, τ_w, and b. Thus, given a moving bump, its velocity v is independent of the recurrent interactions and the number of neurons:

$$v = \frac{1}{2\pi\,\tau_w}\sqrt{b\frac{\tau_w^2}{\tau} - 1}. \tag{4.84}$$

A moving bump profile with a constant velocity is depicted in figure 4.12A.

Finally, if we combine tuned input and adaptation, we observe that the movement of the bump is confined by the stimulus. We determine a zig-zagging or jitter around the peak location of the heterogeneous input. Such a limited bump wandering is shown in figure 4.12B.

4.4 Mexican Hat and Noise Correlations

We discussed that in the single population model with a simple sinusoidal coupling term $j_0 + j_1\cos(2\pi\,x)$ and finite size, one can obtain spatially inhomogeneous activity in form of a single bump with finite firing rates. For homogeneous input the center of

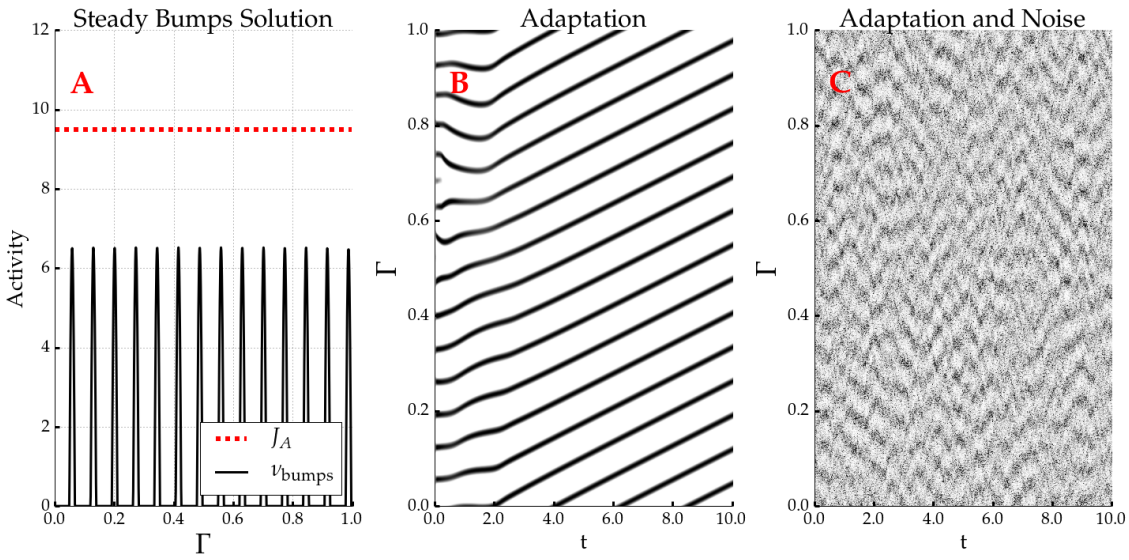

Figure 4.13: Bumps solutions of a single population Mexican hat network using Euler integration with parameters $j_A = 0.3$, $j_E = 0.5$, $j_I = 1.0$, $\sigma_E = 0.025$, $\sigma_E = 0.0125$, $\tau = 0.01$ $N = 1000$, Euler step-size $dt = 0.001$, and random initial conditions. Left (A): The emergence of multiple stable bumps can be observed. Middle (B): Adding adaptation with $b = 0.3$ and $\tau_w = 0.25$ yields moving bumps. In (B) the space Γ is depicted on y-axis, time on x-axis, and activity in grey-scale. Right (C): Model with adaptation (as in B) and Gaussian noise with intensitiy $D = 0.3$ yields zig-zagging of bumps.

the bump is located at an arbitrary position ψ_{bump}. Moreover, if noise or not too weak adaptation is added to the model, the bump starts moving across the ring.

For models with Gaussian Mexican hat connectivity we expect something similar. Nevertheless, as was shown for instance in figure 4.10, in Gaussian Mexican hat networks a range of spatial Fourier modes can be amplified instead of just the 0th or 1st mode. Accordingly, we expect not only a single but several bumps to appear. More precisely, we predict a sinusoidal modulation of active and inactive regions.

We numerically integrated the single population model with Gaussian connectivity kernel (c.f. equation 4.62) with a simple Euler method. Indeed, we observed a steady state with multiple bumps as depicted in figure 4.13A.

Moreover, we further augmented the model by introducing adaptation as well as Gaussian white noise simulating external noisy input:

$$\tau \frac{\partial \nu(x,t)}{\partial t} = -\nu + [\sqrt{N}\,((\omega_E - \omega_I) * \nu + j_A) - w + \sqrt{2D}\,\xi(x,t)]_+, \tag{4.85}$$

$$\tau_w \frac{\partial w(x,t)}{\partial t} = -w + \tau_w\, b\, \nu, \tag{4.86}$$

where $\xi(x,t)$ is a Gaussian noise process with 0 mean and variance of 1 and D denotes the noise intensity parameter. All other parameters are analogous to the previous models. With adaptation the bumps started moving across the space Γ, see figure 4.13B. In

addition, figure 4.13C illustrates that noisy input yielded a zig-zagging of bumps. Their movement directions were not necessarily aligned and could occasionally change.

For our spiking neuron network simulations in the following chapter we expect similar heterogeneous activity patterns as for the single population rate model above. Moreover, we predict that the spatially inhomogeneous activity leads to noise correlations. More precisely, bump movement can cause correlated variability in firing rates among neuron pairs. In turn, these lead to correlated spike counts, as measured by the coefficient r_{SC} (c.f. equation 3.4). If a pair of neurons is currently part of a moving active region, they are very likely to fire together. Likewise, if they are part of an inactive region, their firing is jointly reduced. Accordingly, this pair exhibits positive noise correlations. Negative noise correlations are observed among pairs of which one neuron is part of an active and the other is part of an inactive region. Moreover, the spike count correlation coefficient is measured over trials. For homogeneous network input the phases of the bumps are arbitrary and, therefore, can change across trials. Accordingly, this facilitates noise correlations besides bump movement.

Due to multiple bumps for homogeneous input, we expect distance dependent noise correlations with a sinusoidal modulation. Moreover, because of noisy input and randomly drawn connections, we further predict that there is a general decline of the amplitude of correlations with distance. In other words, active regions further apart show a reduced influence on each other's movement similar to what is depicted in figure 4.13C. In the following chapter we are going to perform rigorous parameter explorations to study the relation between connectivity strengths, Mexican hat shape, and the resulting spatial structure of noise correlations.

4.5 Summary

In this chapter we reviewed the mean-field model by Rosenbaum and Doiron (2014). We recapitulated that Mexican hat connectivity with wider inhibitory than excitatory spread can render a homogeneous fixpoint with balanced excitatory and inhibitory activity unstable. Additionally, instead of runaway activity, which corresponds to an unstable 0th Fourier mode, Mexican hat topologies may yield spatially inhomogeneous activity profiles with finite firing rates.

Moreover, augmenting the model with adaptation has a negligible effect on the stability of the homogeneous fixpoint. In particular in the limit of infinite network sizes, all results from Rosenbaum and Doiron (2014) are still valid in the extended model.

The theory by Rosenbaum and Doiron (2014) cannot give an analytical expression of the network state in case of inhomogeneous activity. Therefore, we further reduced and compared the model to work by Hansel and Sompolinsky (1998) that makes use of sinusoidal coupling. For this simplified model, one can provide an analytical description of inhomogeneous activity. Activity in form of a single bump solution can emerge. Since spatially tuned input has a negligible influence on the bump shape,

bump networks sharpen inputs with broader tuning than the bump width. Furthermore, the bump may wander along the ring in case of noise or sufficiently strong adaptation. Combining adaptation and tuned input, the bump movement is limited and jitters around the peak of the stimulus.

Numerically integrating a single population rate model with Gaussian Mexican hat coupling revealed the emergence of multiple bumps. These bumps moved across the network extent when adaptation was introduced. Additional Gaussian white noise input yielded zig-zag movements of bumps.

A similar behavior is expected for spiking neuron networks with Poisson input and Mexican hat topologies realized by sampling from Gaussian kernels. We further predict that the movement and phase changes of bumps can cause noise correlations among spiking neurons. Moreover, we expect that the correlations sinusoidally depend on the distance between cell pairs.

Correlations and Coding in Spiking Neuron Networks

5

In this chapter we are going to present and discuss our simulation results for spiking neuron networks with Gaussian connection topologies. We will investigate if recurrent connectivity can cause noise correlations, that is Mexican hat connectivity in particular. In addition, we are going to shed light on the question how such correlations affect stimulus encoding.

In the beginning we will formulate our hypotheses based on the literature and the previous theoretical considerations. Next, we will give a short overview about the network model. More detailed explanations of the model are provided in appendix A. Thirdly, we are going to look at simulation results obtained in one-dimensional ring networks. Thereby we will consider results for homogeneous as well as heterogeneous inputs. Fourthly, we are going to turn to a biologically more plausible setting of two-dimensional networks. Similarly, we will discuss results obtained for homogeneous as well as for orientation tuned inputs. Lastly, we are going to investigate how the observed correlations affect the quality of a stimulus encoding.

5.1 Hypotheses

Based on ideas reviewed in the previous two chapters, especially the simulations by Hansen et al. (2012), as well as theoretical work by Hansel and Sompolinsky (1998), and Rosenbaum and Doiron (2014), we formulate the following hypotheses:

1. A Mexican hat connectivity ($\sigma_E < \sigma_I$) can amplify certain spatial frequencies. Given homogeneous input, this causes the emergence of multiple bumps or active regions with arbitrary positions.

2. Because of adaptation and noise, the bumps move across the network extent. The changes of the bumps' positions across trials as well as their movement yield distance dependent noise correlations following a sinusoidal modulation. Besides the sinusoidal modulation, due to noise there is a general decline in amplitude with distance.

3. Heterogeneous, tuned input determines the location of the bumps. Noise correlations are still present due to the bumps' confined movement around the input's peaks.

4. Noise correlations emerging in Mexican hat topologies are detrimental to encoding quality.

5. Mexican hat topologies yielding detrimental correlations can still enhance encoding quality in comparison to other profiles by tuning sharpening.

5.2 Model Description

The model is based on a primary visual cortex network developed within our research group by Stimberg et al. (2009). Their network model leans more towards biological realism rather than providing the most simplified description. In like manner, we wanted to show that the hypotheses based on rather simplistic rate model descriptions can be supported in a biologically realistic setting. In the following section we will give a brief overview about the model. Detailed specifications including the neuronal differential equations are provided in appendix A.

The model consisted of $N = N_E + N_I$ excitatory and inhibitory adaptive exponential integrate and fire (AEIF) neurons (Brette and Gerstner, 2005). We assumed a ratio between inhibitory to excitatory neurons of 1 to 4 (Beaulieu et al., 1992). The AEIF model is common in the literature and can exhibit a very rich set of dynamics including spike frequency adaptation (SFA) (Naud et al., 2008; Touboul and Brette, 2008). We chose moderate sub-threshold and spike frequency adaptation for excitatory neurons ($a_E = 2.0\,\mathrm{nS}$, $b_E = 50\,\mathrm{pA}$). To account for the fact that adaptation is much weaker in inhibitory neurons (Nowak et al., 2003), for inhibitory cells adaptation parameters were set to one tenth of the excitatory values ($a_I = 0.2\,\mathrm{nS}$, $b_I = 5\,\mathrm{pA}$).

5.2.1 Topology and Coupling

We used a realistic synaptic coupling scheme with one type of inhibitory and two types of excitatory synapses. Inhibitory synapses were modeled as decaying exponentials following fast kinetics similar to γ-aminobutyric acid$_\mathrm{A}$ (GABA$_\mathrm{A}$) receptors. Excitatory synapses were composed of an exponential decay following fast kinetics similar to α-amino-3-hydroxy-5-methyl-4-isoxazolepropionic acid (AMPA) receptors and a bi-exponential function following slow kinetics similar to N-methyl-D-aspartic acid (NMDA) receptors. We assumed a fixed ratio between AMPA and NMDA receptors with a fraction of 70% AMPA receptors (Myme et al., 2003).

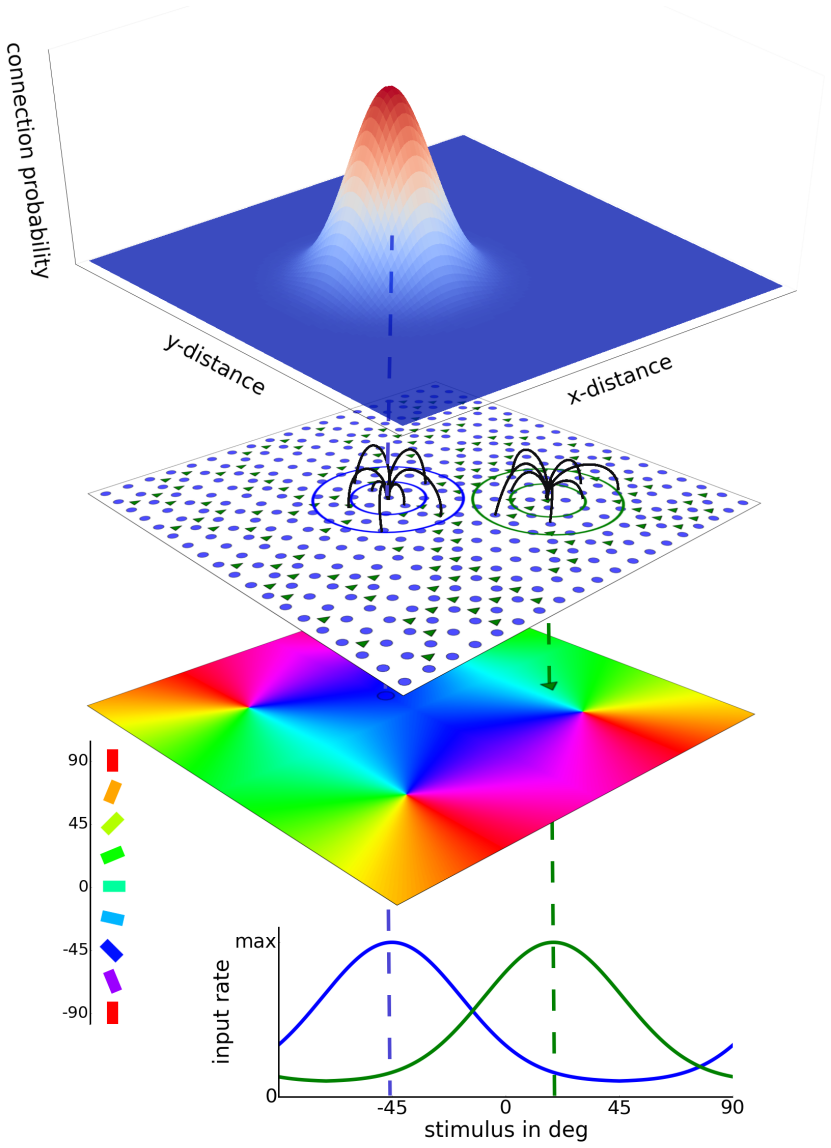

Figure 5.1: 2D Network architecture. The sketch shows the general architecture of the 2D networks with heterogeneous input. A layer of excitatory neurons (blue dots) and inhibitory neurons (green triangles) receives afferent as well as recurrent lateral input. Excitatory neurons are placed on a regular grid and inhibitory neurons are assigned random grid positions. Examples of lateral connections are depicted by the black lines. These connections are sampled from Gaussian probability kernels depending on cell distance, as depicted at the top. The preferred orientation of a cell is assigned according to its position in the artificial orientation map with 4 pinwheels (bottom colored sheet). Afferent input is realized as independent Poisson spike trains. A circular Gaussian tuning curve with a width of 27.5° determines the afferent rate for each neuron as a function of the presented input stimulus orientation. Two example input tuning curves for two different cells are shown at the bottom.

Each neuron received K_E excitatory and K_I inhibitory recurrent connections. These synaptic connections were drawn from Gaussian probability distributions defined over the Euclidean distances d_{ij} between two neurons i and j:

$$p(i,j) = \begin{cases} 0 & \text{for } i = j \text{ (no self-connections)} \\ 1/\left(\sqrt{2\pi}\sigma_Y\right)^D \exp\left(-d_{ij}^2/2\sigma_Y^2\right) & \text{otherwise} \end{cases}, \quad (5.1)$$

where $Y \in \{E, I\}$ denotes the population type of pre-synaptic neuron j, $D \in \{1, 2\}$ is the dimensionality of the model space, and σ_Y is the width or spread of the connection.

We investigated one-dimensional line networks as well as two-dimensional models where neurons were placed on a grid. Excitatory neurons were regularly and evenly spaced on the line or grid. Whereas the positions of the inhibitory neurons were chosen randomly from a uniform distribution defined over the neural space. Distances in our network are given in pixels (px) or kilo-pixels (kpx), the distance between 2 or 1001 excitatory cells, respectively. Furthermore, the two-dimensional grid map was scaled to distances matching cat primary visual cortex (Stimberg et al., 2009). We assumed a scaling of 1 px corresponding to 15 µm. Thus, a 100 px × 100 px network with 4 orientation pinwheels corresponds to a 1.5 mm × 1.5 mm piece of cat primary visual cortex. Such a two-dimensional network topology is depicted in figure 5.1.

We use the terms *configuration* or *profile* to refer to the connection topology of a network. Important parameters that we investigated are σ_E and σ_I, the connectivity spread of excitatory and inhibitory connections, respectively. For the two-dimensional networks we thoroughly varied both parameters between 5 and 25 pixels, corresponding to 75 and 375 µm in cat cortex scale. These values span a biologically plausible range. For instance, Hellwig (2000) measured a Gaussian decay of connection probability with spreads between 150 and 350 µm for pyramidal neurons in the rat primary visual cortex. Similarly, Mariño et al. (2005) reported a Gaussian connection spread of about 125 µm for local connections in cat V1.

In case we relate to the connection strengths, we use the term *regime*. Two important parameters in this regard are $\bar{g}_{EE}$ and $\bar{g}_{IE}$, the connection strength between excitatory to excitatory and excitatory to inhibitory cells. More precisely, these are the maximum conductances of the recurrent currents in case a single spike is emitted by an excitatory pre-synaptic cell. The maximum conductance was shared among AMPA and NMDA kinetics with respect to the fixed ratio between the two types.[1] The maximum conductance values were varied in rather small amplitude ranges between 0 and 1.2 nS such that the excitatory post-synaptic potential (EPSP) at resting membrane voltage (−65 mV) showed small deflections with maximum values between 0 and 1.5 mV for a single pre-synaptic spike. These values are within a biologically plausible range. For instance, Mason et al. (1991) observed single spike EPSPs ranging from 0.05 up to 2 mV with a mean size of 0.55 mV in rat primary visual cortex.

[1] $\bar{g}_{\text{AMPA},EE} = 0.7\bar{g}_{EE}$ and $\bar{g}_{\text{NMDA},EE} = 0.3\bar{g}_{EE}$, c.f. Stimberg et al. (2009)

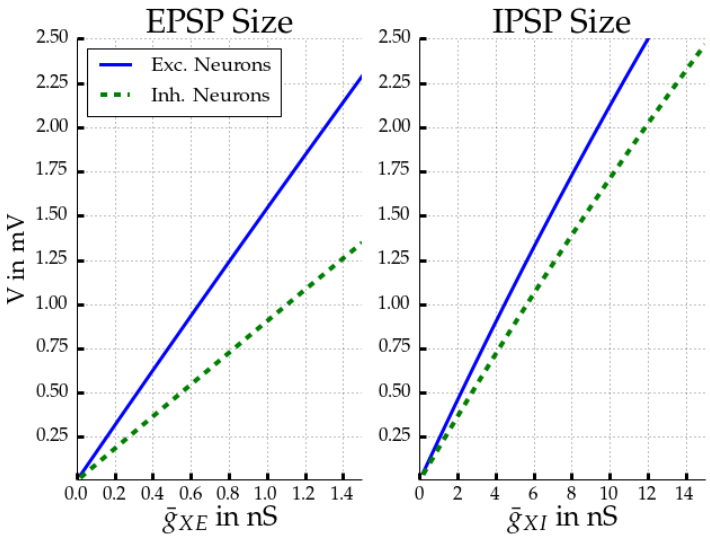

Figure 5.2: Maximum absolute deflection of membrane voltage at resting potential (-65 mV) for single pre-synaptic spikes as function of connection strength. Excitatory post-synaptic potential (EPSP) size on the left and inhibitory post-synaptic potential (IPSP) on the right; in blue for excitatory neurons and inhibitory neurons in dotted green.

Inhibitory connection strengths were not varied and kept fixed, we used $\bar{g}_{II} = \bar{g}_{EI} = 5.0\,\mathrm{nS}$. At first sight, these values seem large in comparison to the excitatory strengths. However, inhibitory synapses followed only fast $\mathrm{GABA_A}$ kinetics in contrast to the combination of fast AMPA and slow NMDA kinetics of the excitatory ones. As a consequence, an inhibitory pre-synaptic spike causes a comparably small deflection of about 1 mV of the membrane voltage of the post-synaptic cell at rest. The deflection of the membrane potential at rest as a function of connection strength is depicted in figure 5.2.

Synaptic delays depended linearly on the distance between cells. We assumed a signal velocity of 0.2 m/s (13.3 kpx/s) in the two-dimensional model. Usually values between 0.1 and 0.5 m/s are measured in experiments (Bringuier, 1999; González-Burgos et al., 2000). For further details and values of the one-dimensional networks see the appendix A.

Lastly, if not otherwise stated, boundary conditions were periodic. Hence, one-dimensional networks exhibit a ring shape, whereas two-dimensional networks instantiate a torus.

5.2.2 Network Input

Each neuron received afferent inputs modeled as K_A independent Poisson spike trains. Hence, inputs to each neuron were decorrelated and all observed correlations could only arise due to recurrent connectivity.

We distinguished between two driving modes, homogeneous, also termed a blank stimulus, and heterogeneous, also named tuned input.

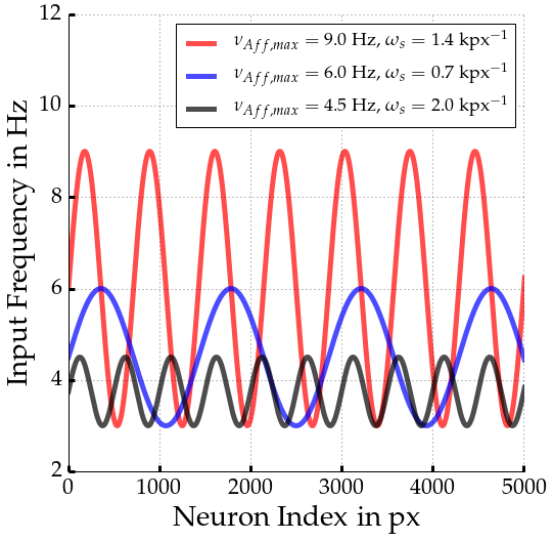

Figure 5.3: Heterogenous input to 1D network. Afferent inputs of the first 5000 neurons are depicted. Three different maximum input frequencies in combination with three different spatial frequencies are given.

In case of a blank stimulus, input was not tuned but every neuron received afferent input realized as independent Poisson spike trains with the same frequency. This mode resembles animal experiments where a blank screen stimulus is shown or spontaneous activity is measured.

In case of heterogeneous stimuli in a one-dimensional network, the frequencies of the afferent input were spatially tuned according to a sine wave. The input frequency at the kth pixel is computed as follows:

$$\nu_{\text{Aff},k} = (1 + \sin(2\pi\,\omega_s\,k))\,\bar{\nu}_{\text{Aff}} + \nu_{\text{Aff, base}}, \tag{5.2}$$

$$\bar{\nu}_{\text{Aff}} = \frac{\nu_{\text{Aff, max}} - \nu_{\text{Aff, base}}}{2}, \tag{5.3}$$

where ω_s is the spatial frequency of the sine wave per pixel, $\nu_{\text{Aff, max}}$ is the maximum firing rate and $\nu_{\text{Aff, base}}$ denotes the baseline or minimum firing rate. The input was only spatially modulated and constant in time. Example sine wave inputs are shown in figure 5.3.

If an orientation stimulus was shown to a 2D network, the afferent input was tuned according to the preferred orientation of the post-synaptic neuron. The orientation preference of each neuron is determined by its position in the pinwheel map as depicted in figure 5.1. We assumed moderate orientation tuning. Input firing was computed using a circular Gaussian tuning function with a width of $\sigma_{\text{Aff}} = 27.5°$. The input firing rate to each cell i was computed as follows:

$$\nu_{\text{Aff},i}(s) = (\nu_{\text{Aff, max}} - \nu_{\text{Aff, base}}) \exp\left(-\frac{(s - s_{\text{PO}_i})^2}{2\sigma^2_{\text{Aff}}}\right) + \nu_{\text{Aff, base}}, \tag{5.4}$$

where $\nu_{\text{Aff, max}}$ is the maximum or peak firing rate, $\nu_{\text{Aff, base}}$ is the baseline firing, s is the orientation of the stimulus ($s \in [-90°, 90°)$), and s_{PO_i} denotes the ith neuron's preferred orientation according to the orientation map. Again, inputs only varied in spatial dimensions and were constant in time.

Before any measures were applied every network was given at least 1 second of afferent stimulation to settle into a stationary state.

5.2.3 Implementation

The network was implemented in Python 2.7 and was partly compiled into C-code for efficiency using the Numba library (The Numba Development Team, 2015). The source code is available on github.[2] Moreover, some correlation experiments were also verified with smaller networks of Hodgkin-Huxley type neurons (c.f. Stimberg et al., 2009) using the BRIAN simulator package (Goodman and Brette, 2008) (data not shown). All data and parameter explorations were managed using the previously introduced simulation toolkit *pypet*.

5.2.4 Criticality

As discussed in the previous chapter, for certain coupling configurations with strong connection weights, the average amount of excitation is amplified. In the previous chapter's simple model with a rectified linear transfer function, this yields runaway excitation with divergent firing activity.

Such runaway conditions were tested in the simulations by applying network stimulation for 1 second, removing the stimulation for 1 second,[3] and checking if activity persisted. In such a case the network was classified as *critical* or *self-sustained*, viz. the network was able to sustain spiking without any external drive. The transition boundary between normal and critical activity is termed *criticality*.

This condition corresponds roughly to the runaway excitation from the previous chapter. Recurrent connectivity is strong enough to amplify activity. In the theoretic model this may translate into $\boldsymbol{A}_\epsilon(0)$ (c.f. equation 4.31) having eigenvalues with positive real parts. Therefore, firing rates diverge to infinite values. Moreover, we saw in the model by Hansel and Sompolinsky (1998) that firing rates may even diverge in case the 0th Fourier mode of the fixpoint is stable. Given a bump solution, inhibition may be too weak so that the activity within the active region of the bump grows infinitely (c.f. equation 4.78).

Still, in our simulations firing rates are always bounded due to finite time steps and refractory periods. But one can observe divergence in terms of spiking activity close to the maximum firing rates. In our simulations maximum rates were 333 Hz and 500 Hz for excitatory and inhibitory neurons, respectively.

[2] `https://github.com/SmokinCaterpillar/visualcortex`

[3] To verify this method, periods of 9 seconds were tested as well and did yield the same outcome.

In a few cases despite the absence of input, simulation activity was persistent, but firing rates remained low and did not reach values close to maximum activity. In contrast to the linear model[4] of the previous chapter, the neurons herein are highly nonlinear which may explain this observation. Although self-sustained activity with low firing rates is an intriguing phenomenon, a thorough numerical and theoretical investigation is beyond the scope of this work. Therefore, our analyses will be restricted to network regimes without self-sustained activity. Still, some minor analyses of activity patterns beyond criticality are provided in appendix B.

5.3 Noise Correlations in One-Dimensional Networks

Before investigating the more biologically realistic two-dimensional setting, we will turn to simulations of ring networks similar to the model by Hansen et al. (2012) depicted in figure 3.4 in chapter 3. Our networks consisted of $N_E = 10{,}000$ excitatory and $N_I = 2500$ inhibitory neurons placed on a ring with a length scale of 10 kpx. Due to periodic boundary conditions the maximum distance between two cells is therefore 5 kpx. Each neuron received $K_E = 200$ excitatory and $K_I = 100$ inhibitory recurrent inputs. At first, we will only consider homogeneous input of 15 Hz with $K_A = 100$ afferent connections per cell. Each of these inputs was realized as an independent Poisson process with the same rate. Thereafter, we will look at sinusoidally tuned input stimuli with spatially heterogeneous rates. We tested different spatial frequencies as well as maximum firing rates.

5.3.1 Correlated Variability and Spatial Patterns

Figure 5.4 shows spike raster plots of three different connection strengths for the same Mexican hat topology ($\sigma_E = 125\,\mathrm{px} < \sigma_E = 250\,\mathrm{px}$). From top to bottom excitatory recurrent connection strengths were chosen such that the networks operated afar (A), close to (B), and beyond criticality (C). At the bottom (C) activity was divergent and excitatory neurons exhibited a high firing rate. The top graph (A) shows asynchronous activity. In the previous chapter we discussed that for finite size networks, there exists a regime where no spatial patterns emerge despite the Mexican hat topology. The middle image (B) shows the emergence of a spatial pattern. One can observe moving bumps that may even fuse into each other or split into two.

In figure 5.4A the average noise correlation coefficient regardless of distance between cell pairs is close to 0. However, for the network yielding spatial patterns in the middle (figure 5.4B), we measured distance dependent noise correlations (data not shown).

In figure 5.5A the spike count correlation coefficient as a function of distance between cell pairs is shown for another Mexican hat network close to criticality. Each

[4] Linear in the homogeneous case where rectification can be ignored.

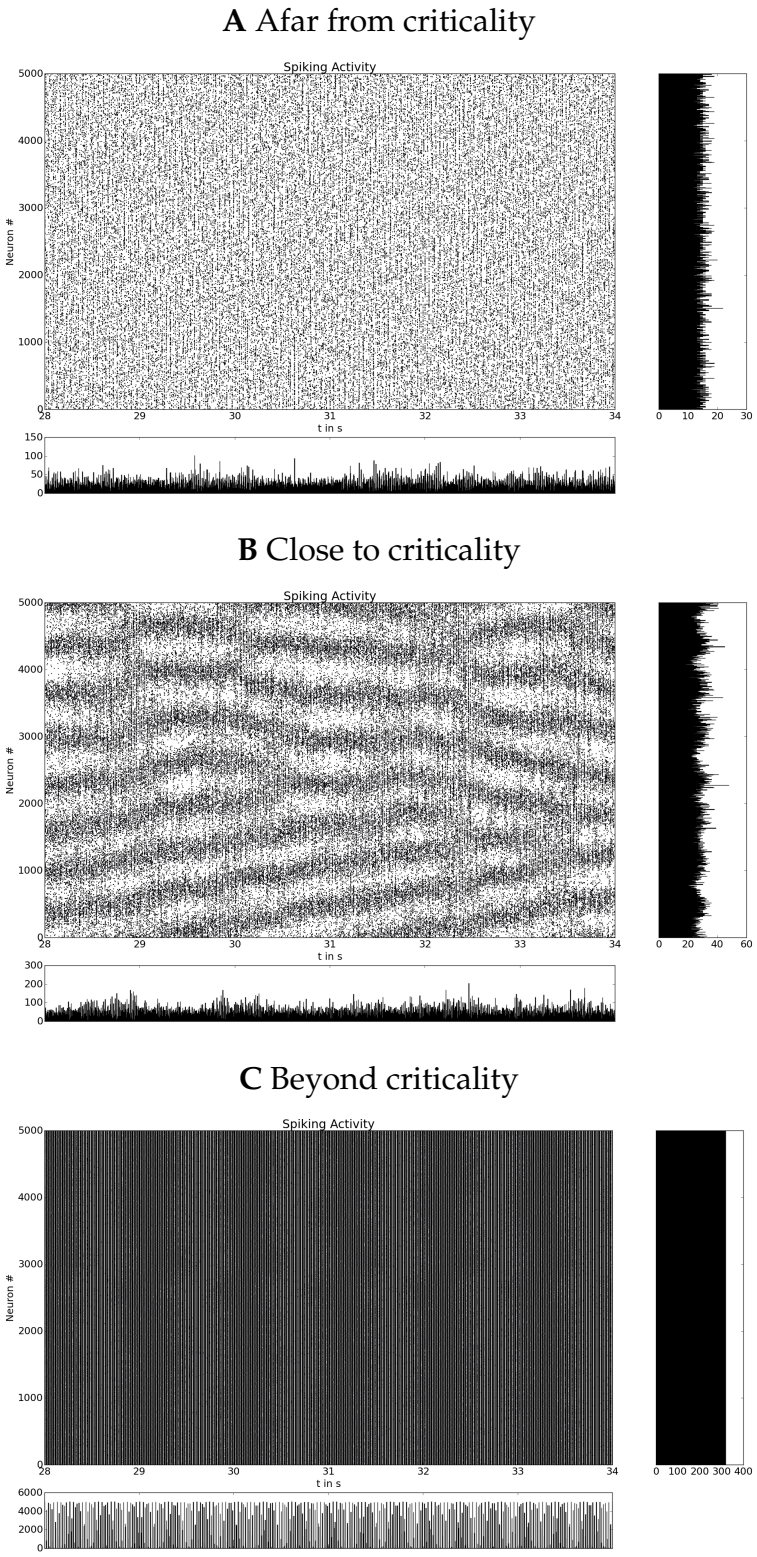

Figure 5.4: Spiking activity of a Mexican hat network ($\sigma_E = 125\,\mathrm{px} < \sigma_E = 250\,\mathrm{px}$) for three different excitatory recurrent connection strengths. A small black dot represents a spike of a particular neuron at a particular point in time. Neurons are ordered according to their position on the ring. Small plots show the activity histograms over time (horizontal) as well as over space (vertical). Top (A): Excitatory to excitatory connectivity is weak ($\bar{g}_{EE} = 0.15\,\mathrm{nS}$ and $\bar{g}_{IE} = 0.6\,\mathrm{nS}$) and the network operates far from criticality. Middle (B): Mediocre connectivity strength close to criticality ($\bar{g}_{EE} = 0.25\,\mathrm{nS}$ and $\bar{g}_{IE} = 0.5\,\mathrm{nS}$). One can observe the emergence of spatial patterns, i.e. moving bumps. Bottom (C): Strong excitatory recurrent connectivity beyond criticality ($\bar{g}_{EE} = 0.15\,\mathrm{nS}$ and $\bar{g}_{IE} = 0.1\,\mathrm{nS}$), neurons are firing with a high rate.

blue dot corresponds to the r_{SC} value for an individual cell pair calculated over 50 trials. For comparison the noise correlations of an inverse Mexican hat are shown in the small inset. No distance dependency of correlations was observed in this case. The average value for bins of 20 px is plotted in green. For the Mexican hat network this average correlation coefficient can be neatly fitted with a damped sine wave; shown on top of the green curve in red. More precisely, the corresponding function with respect to distance d is

$$\langle r_{SC} \rangle(d) = C_{r_{SC}} \exp(-|d|/\lambda_{r_{SC}}) \cos(\omega_{r_{SC}} d), \tag{5.5}$$

where $\langle r_{SC} \rangle(d)$ is expectation across neuron pairs with the same distance, $C_{r_{SC}}$ is the correlation fit for $d = 0$, $\lambda_{r_{SC}}$ is the mean decay lifetime, and $\omega_{r_{SC}}$ denotes the spatial frequency. The fits were calculated using curve fitting from SciPy's optimization library (Jones et al., 2001). This optimization was repeated 10 times with random initial values and the parameters of the best fit were chosen. In figure 5.5A for close by cells correlation is relatively large and we fitted an initial value $C_{r_{SC}} \approx 0.64$. Furthermore, the exponential decay was quite slow with $\lambda_{r_{SC}} \approx 3.4\,\mathrm{kpx}$. Lastly, the spatial frequency was fast with $\omega_{r_{SC}} \approx 1.4$ per thousand pixels.

We investigated whether the spatial scales of the noise correlations correspond to the spatial scales of the moving bumps. In order to determine the spatial scale of the moving patterns we sampled 100 windows of 10 ms activity for each trial. This yielded activity vectors of length 10,000, with one entry for each neuron ordered along the spatial axis. For each vector we computed the power spectrum using the fast Fourier transform of the SciPy package (Jones et al., 2001). The average spectrum alongside the average spatial autocorrelation across all windows is shown in figures 5.5C and D, respectively. Indeed, the highest power was measured for 1.4 cycles per thousand pixels, which nicely fits the spatial scale of the noise correlations. Moreover, the spatial power spectrum does not show a spiky peak at only 1.4 cycles, but a slight broadening around this value. This matches the observations one makes looking directly at the network activity. Usually we counted 14 bumps within the entire network of 10,000 neurons, but occasionally due to splits and merges of bumps one more or less could be found.

Moreover, with respect to the slow correlation amplitude decay $\lambda_{r_{SC}} \approx 3.4\,\mathrm{kpx}$, one may also interpret the inhomogeneous activity as a wave solution. Accordingly, the active bump regions correspond to the wave peaks. In contrast, the wave troughs are hidden by the spiking threshold. One can only observe inactive regions without firing because neurons cannot exhibit negative firing rates.

How does the spatially inhomogeneous activity relate to the amplification of spatial frequencies due to the connectivity kernel? This question is a bit more difficult to answer. As we discussed in the previous chapter, principally, for the linear rate model we can determine the spatial amplification by translating the network into the spatial

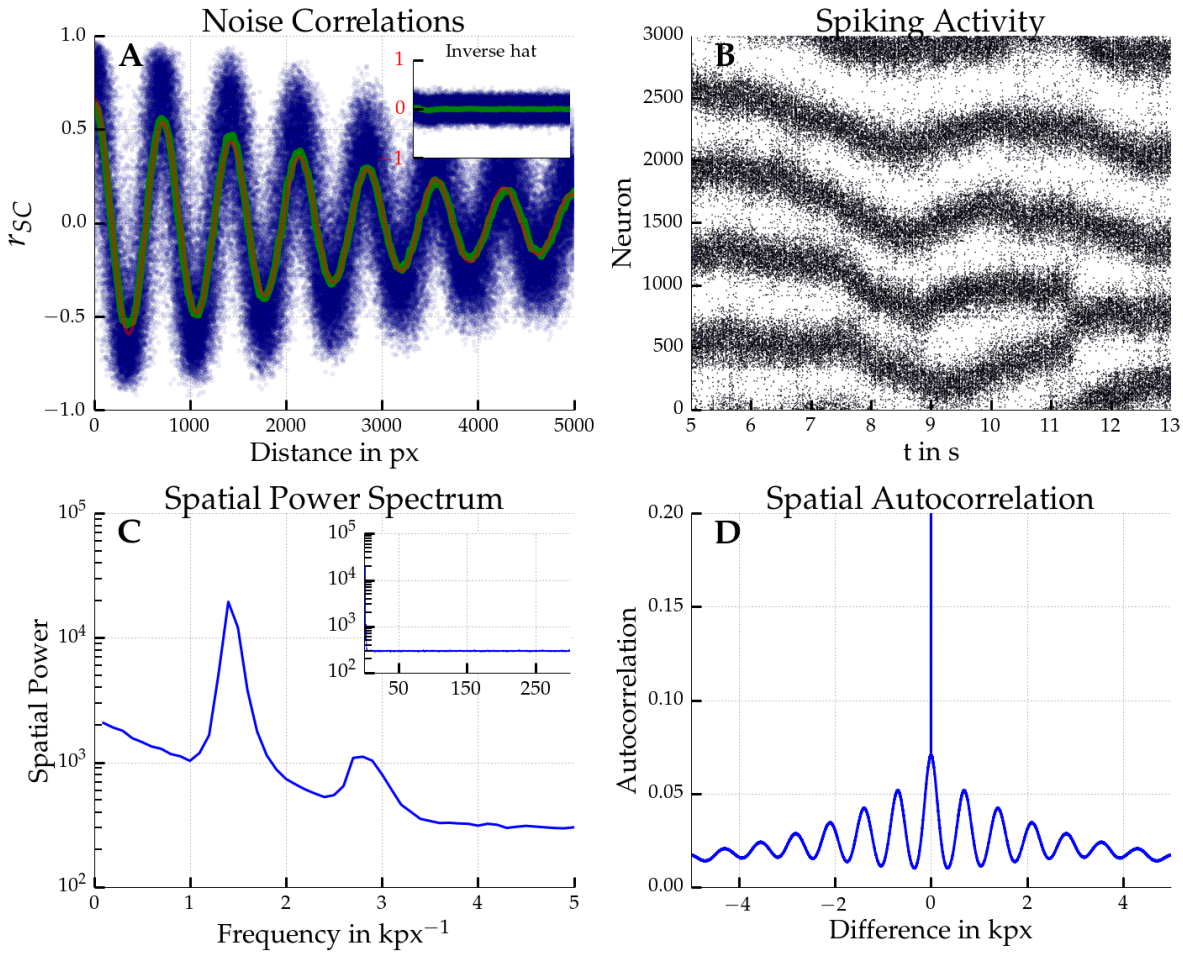

Figure 5.5: Top left (A): Noise correlations as a function of distance in a Mexican hat network ($\sigma_E = 125\,\text{px}$, $\sigma_I = 250\,\text{px}$, $\bar{g}_{EE} = 0.35\,\text{nS}$, $\bar{g}_{IE} = 0.6\,\text{nS}$). Blue dots represent r_{SC} of individual neuron pairs computed over a time window of 1 second and 50 trials. One can observe a damped oscillation which can be nicely fit by equation 5.5. The fitted function is shown in red and tightly overlays the empirical average computed from the data in green. The inset shows the correlations for an inverse Mexican hat ($\sigma_E = 250\,\text{px}$, $\sigma_I = 125\,\text{px}$) with the same recurrent strengths. Top right (B): Spike raster plot of 5000 excitatory neurons. Bottom left (C): Spatial power spectrum computed from activity snapshots of 10 ms averaged over 5000 samples. The frequency with the highest power (1.4 cycles per kpx) nicely fits the oscillations of the noise correlations with distance (A). The power spectrum for higher spatial frequencies is shown in the inset and is basically flat. Bottom right (D): Average autocorrelation of the activity snapshots.

Fourier domain. Therefore, we need to know the connectivity kernel and the weights. We can plug these values into equation 4.31 and infer which Jacobian has eigenvalues with positive real parts. However, the simple rate model from the previous chapter relies on instantaneous rate based coupling. The biologically more plausible complex model used here is coupled via conductance based synapses. Hence, post-synaptic potentials evolve over time and, additionally, depend on the membrane potential. Therefore, we do not know the effective weights between the neural populations, especially not at the unstable fixpoint with homogeneous activity. Still, we can use an approximation to the effective weights simply by looking at the average excitatory and inhibitory recurrent currents for each population at the stable inhomogeneous fixpoint. This may give us an approximation of the strength of the synaptic coupling at the unstable fixpoint. We can use the average currents as effective weights ω_{XY} in equation 4.31 and assume $\epsilon = 0$:

$$\boldsymbol{A}_0(n) = \begin{pmatrix} \tilde{\omega}_{EE}(n) & -\tilde{\omega}_{EI}(n) \\ \tilde{\omega}_{IE}(n) & -\tilde{\omega}_{II}(n) \end{pmatrix}. \tag{5.6}$$

Using a non-zero value for ϵ is a difficult choice because currents are measured in Siemens. It is unclear how this unit compares to the unitless value of the network size. Given our assumptions, the resulting spatial frequency with the highest real eigenvalue was 1.2 cycles per thousand pixels. This is close but a bit less than the spatial frequency of the noise correlations with 1.4 cycles. However, the Mexican hat profile amplifies not just a single spatial frequency but a certain range. This may explain why there were a few more bumps than expected by just looking at the frequency with the largest positive real part of the eigenvalues. Furthermore, we are not sure if the average currents in the bump attractor state are good estimates of the weights in the homogeneous state. Hence, the theoretical considerations from the previous chapter are suitable for a qualitative but not a quantitative comparison to our network models.

Furthermore, we observed an increase of correlated variability with increase in the size of the integration window. As seen in figure 5.6A, longer integration time windows yielded larger amplitudes of the damped sine wave. Moreover, increasing the integration window beyond 1 second yielded a saturation of average correlation as depicted in figure 5.6B. We recall from chapter 3 that the noise correlation coefficient is defined as the ratio between the covariance and the product of individual neurons' standard deviations of spike counts, $r_{SC} = \mathrm{Cov}(r_i, r_j)/(\mathrm{Std}(r_i)\ \mathrm{Std}(r_j))$. In this line of thought, we took a look at the relation between the mean covariance of nearby neurons and the geometric mean spike count variance of individual neurons. The results are depicted in figure 5.6D. We observed that for smaller window sizes the covariance was much smaller in comparison to the variance. Hence, much of the cells' covariation was hidden by the spiking threshold. One can also recognize this by looking at pairwise spike count distributions for different integration windows sizes as depicted in figure 5.7.

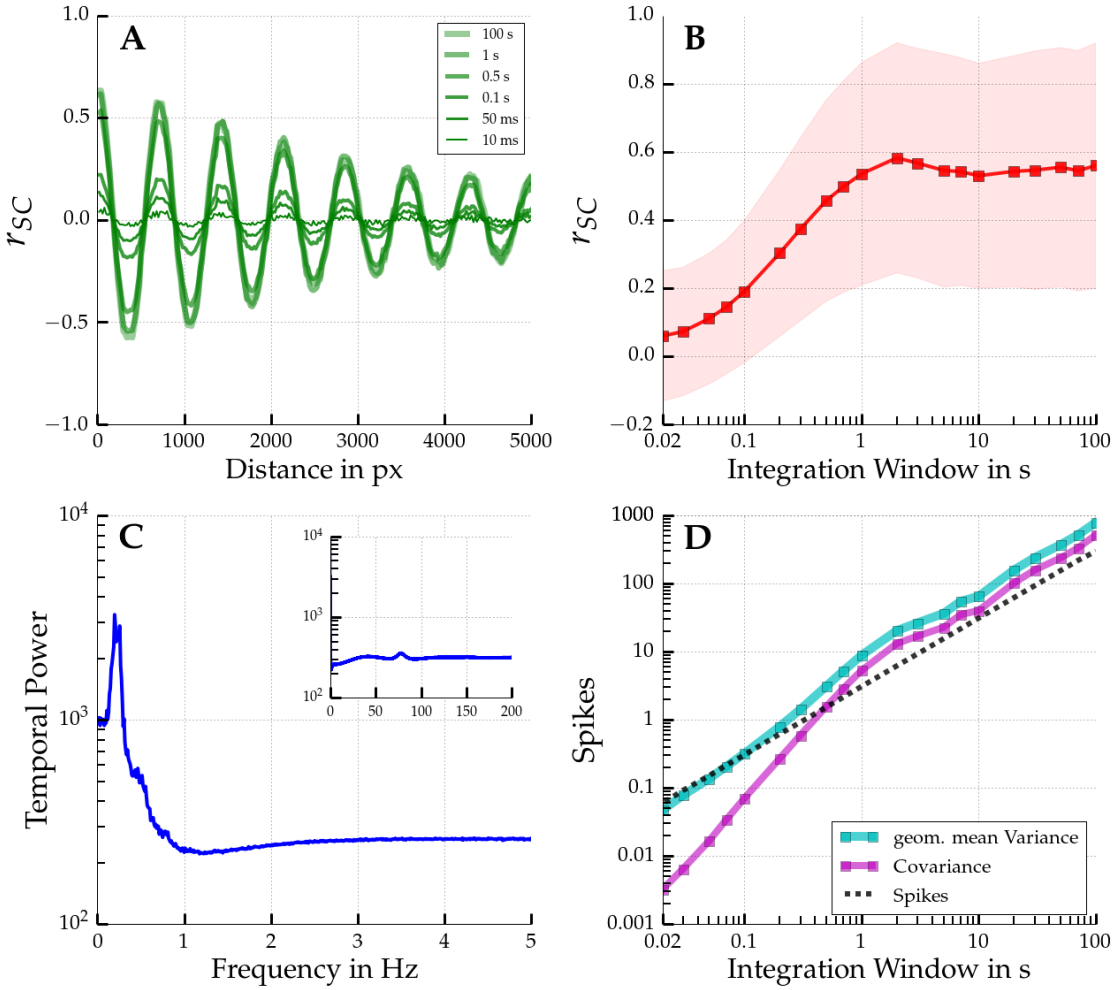

Figure 5.6: Temporal Scales of Noise correlations. Network parameters as in figure 5.5. Top left (A): Average noise correlation as a function of cell distance computed over 50 repeated trials. Different integration time window sizes are shown with increasing thickness of the lines. The smallest window we use is 20 ms because for smaller windows only a tiny minority of cells exhibits at least a single spike within 50 trials, which strongly biases our calculation of the spike count correlation coefficient. Increasing window size increases the amplitude of the damped wave. Top right (B): Average noise correlation for cells at most 100 px apart as a function of integration window size. This corresponds to less than the first fourth of a cycle of the damped sine waves one the top left. The envelope shows the standard deviation among all cell pairs. Increasing the window size beyond 1 second yields a saturation effect. Bottom left (C): Average temporal power spectrum. Movement of bumps happens on slow time scales as the peak and pronounced power on very low frequencies suggest. Additionally, there is another shallow peak in the gamma frequency band, as shown in the inset. Bottom right (D): Contribution of spike count variance and covariance to the average r_{SC} shown above. For smaller window sizes the covariance is much smaller than the neuron pairs' gemoetric mean variance of spike counts. For increasing window size the difference is reduced and both lines become parallel. The expected number of spikes per neuron is shown as a black dotted line.

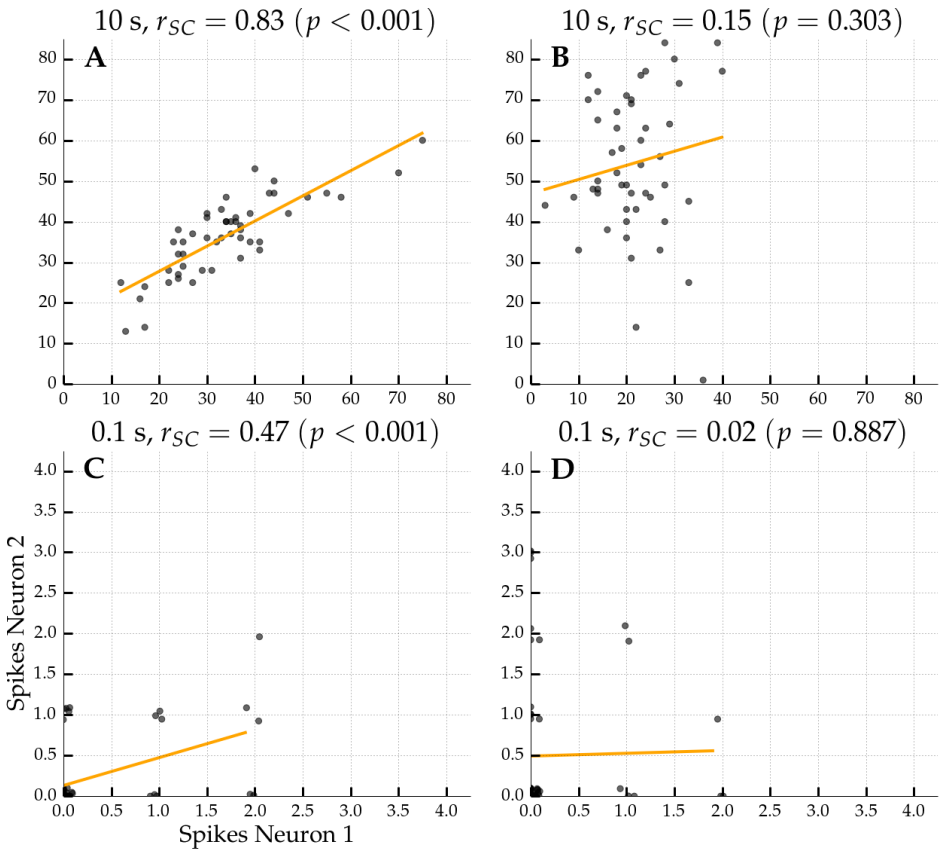

Figure 5.7: Distributions of spike counts for two different cell pairs for long 10 s (A, B) and short 0.1 s (C, D) integration sizes. Network parameters as in figure 5.5. For better visibility we added a small random jitter to each dot in the bottom two plots. For shorter time scales for both cells one observes rarely any spikes in a single trial. As a consequence, covariation of spikes is greatly reduced yielding a reduction in the spike count correlation coefficient r_{SC} as listed in the figure captions (including the p-value). The orange line depicts the corresponding linear fit.

Moreover, as shown in the temporal power spectrum (figure 5.6C), the movement of the bumps was rather slow. This is indicated by the pronounced power in the low frequency band. The peak of the temporal power spectrum was found for 0.2 Hz. Furthermore, another maximum was observed in the gamma spectrum around 80 Hz (inset). But in comparison to the lower frequencies the power was rather weak.

We also asked whether the cell type has an influence on the correlations. Figure 5.8 depicts the average spike count correlation between excitatory, inhibitory and mixed pairs. There, no qualitative difference between the three groups can be determined just a quantitative one. Pairs involving at least one inhibitory neuron exhibited a larger amplitude of distance depended noise correlations. However, the overall shape of the curves remained the same. In the rest of this thesis we will limit our analysis to excitatory neurons only. GABAergic inhibitory interneurons are not known to project information from early visual areas downstream to other cortical areas (Seriès et al., 2004; Schmolesky, 2007). Thus, in terms of stimulus processing, correlated variability

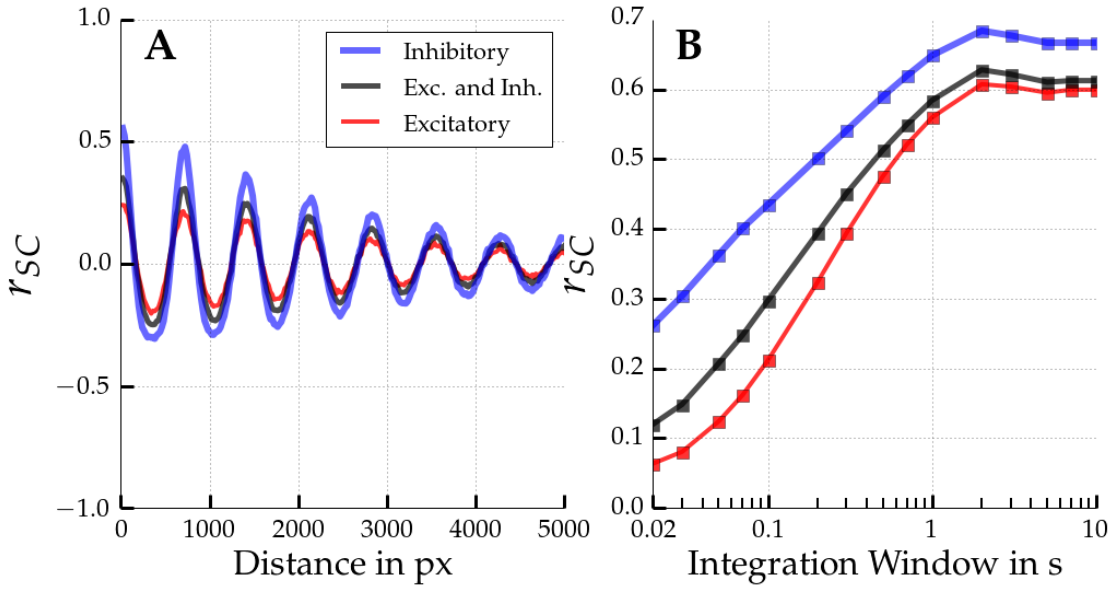

Figure 5.8: Left (A): Distance dependence of correlations for three different types of cell pairings. Network parameters as in figure 5.5. Average r_{SC} for a window of 1 second among pairs of only excitatory neurons (red), only inhibitory neurons (blue), and mixed pairs (black). Right (B): Average correlations among cell pairs at most 100 pixel apart as a function of integration window size for the different pairings.

among pairs of excitatory neurons is of stronger importance than correlations among inhibitory or mixed pairs.

5.3.2 Parameter Explorations

We explored the dependence of the fitted parameters from equation 5.5 as a function of different connection strengths $\bar{g}_{EE}$ as well as different Mexican hat sizes σ_I. The fitted parameters as well as some other basic statistics such as the average firing rate, the coefficient of variation (CV), and the average noise correlation coefficient (r_{SC}) for cells at most 100 px apart are shown in figure 5.9.

As one can see in figure 5.9A, the average firing rate increased slowly with increasing excitatory to excitatory strength until criticality is reached as depicted by the orange border. Beyond the border the average firing rate quickly diverged. However, slightly beyond criticality ($\bar{g}_{EE} = 0.575\,\mathrm{nS}$) the average firing was still below 10 Hz; as were also the individual average firing rates of every cell in the networks. So no neuron showed divergent firing activity. Nonetheless, we observed persistent activity in the absence of afferent input (as criticality is defined in this thesis).

Therefore, we investigated the effect of stronger input firing rates before and beyond criticality. For higher input firing rates and a regime beyond criticality, we observed the divergence of firing rates for at least some of the networks' cells. Figure 5.10 shows the spiking activity measured in two Mexican hat configurations if input firing rate was increased to 60 Hz instead of the default 15 Hz. In contrast to a network

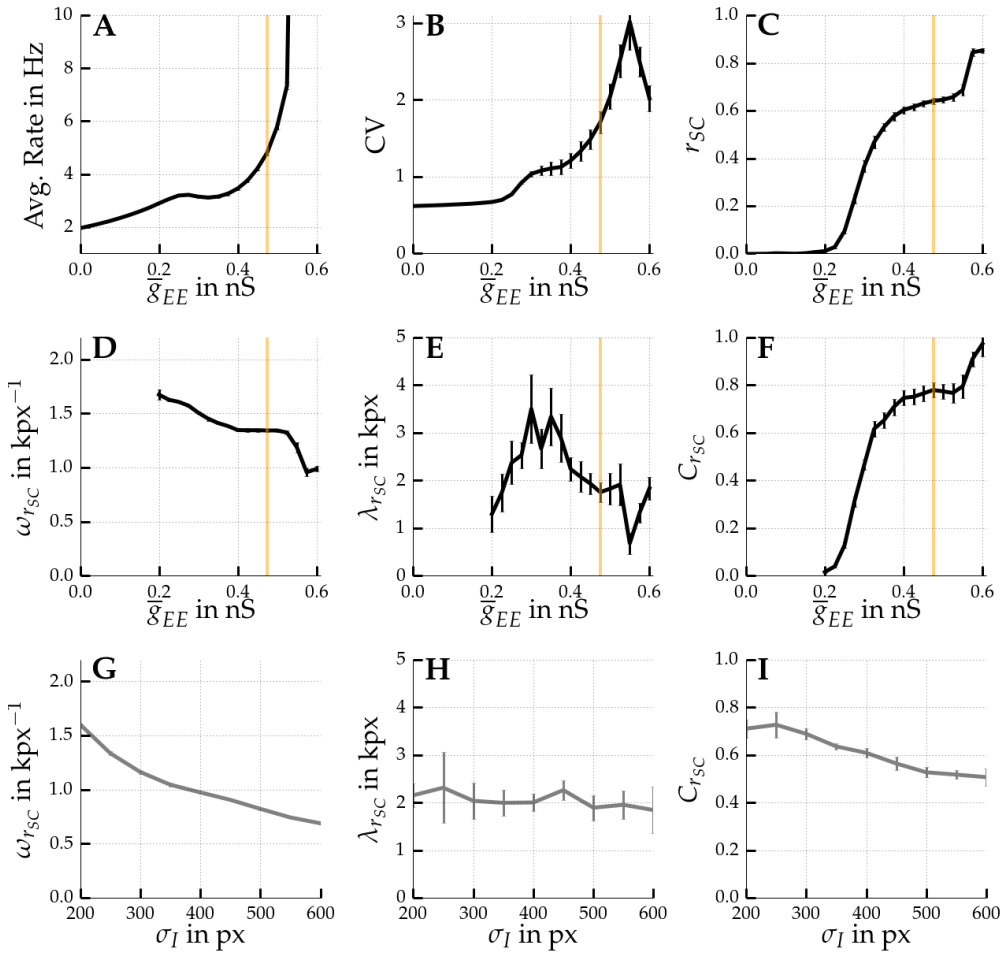

Figure 5.9: Top two rows (A-D) show parameter explorations of different excitatory to excitatory connection strengths. $\bar{g}_{EE}$ is explored using a Mexican hat topology with $\sigma_E = 125\,\text{px}$, $\sigma_I = 250\,\text{px}$, and $\bar{g}_{IE} = 0.6\,\text{nS}$. The orange line marks the bifurcation to criticality. All values are computed from 50 trials and averaged across 10 network realizations. Values are computed from excitatory neurons only. Error bars mark the standard deviation across network realizations. The top row shows from left to right: (A) the average firing rate as a function of connection strength, followed by (B) the average coefficient of variation (CV), and (C) the average noise correlation coefficient (r_{SC}) for cells with at most 100 px distance between them. The r_{SC} is computed over an integration window of 1 second and based on 50 trials. Middle row shows the curve parameters from fitting distance dependend noise correlation with equation 5.5. From left to right: (D) Average revolutions per thousand pixels ($\omega_{r_{SC}}$), (E) decay constant ($\lambda_{r_{SC}}$), and (F) maximum average noise correlation ($C_{r_{SC}}$). Values were computed only for $\bar{g}_{EE} \geq 0.2\,\text{nS}$ because for smaller values no distance dependent noise correlations were observed. Bottom row (G-I) same as in the middle but for exploring inhibitory connection spread σ_I using $\bar{g}_{EE} = 0.4\,\text{nS}$ and $\sigma_E = 125\,\text{px}$.

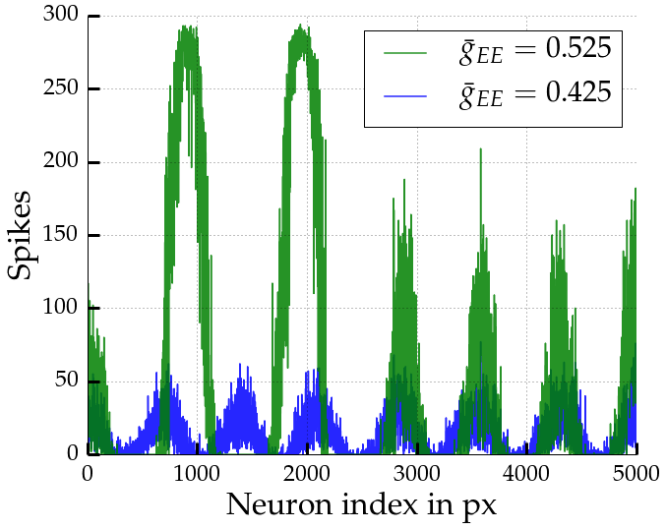

Figure 5.10: Number of spikes emitted by every neuron within a 1 second time window for two Mexican hat networks ($\sigma_E = 125\,\text{px}$, $\sigma_I = 250\,\text{px}$, $\bar{g}_{IE} = 0.6\,\text{nS}$). Both networks are stimulated with strong homogenous input using a frequency of 60 Hz instead of the default 15 Hz. In blue a network close to criticality and in green beyond criticality. Neurons are ordered along their location on the ring topology (only first 5000 neurons are shown). For the network regime beyond criticality in green, some of the neurons are firing almost with the maximum firing rate.

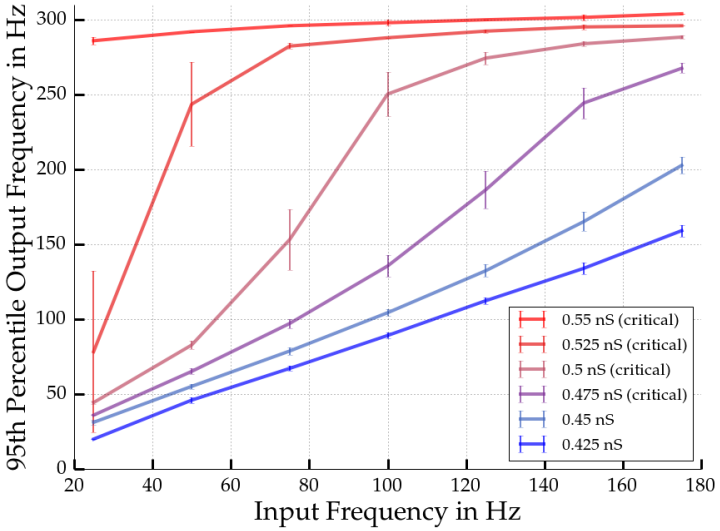

Figure 5.11: 95th percentile of excitatory neurons' firing frequencies at criticality as a function of input frequency. Mexican hat network parameters as in figure 5.10. Values are computed from single trials of 5 seconds duration after an initial period of 1 second. Firing rates were determined from consecutive windows of 250 ms. Error bars mark the standard deviation across 10 network realizations. This gives an impression of the top firing rates of all neurons in each network. For networks before criticality (dark and light blue) the relation between input and top output frequency is roughly linear. Whereas for networks beyond criticality there is a non-linear increase and neuron firing rates rise to 250 Hz or more for stronger input.

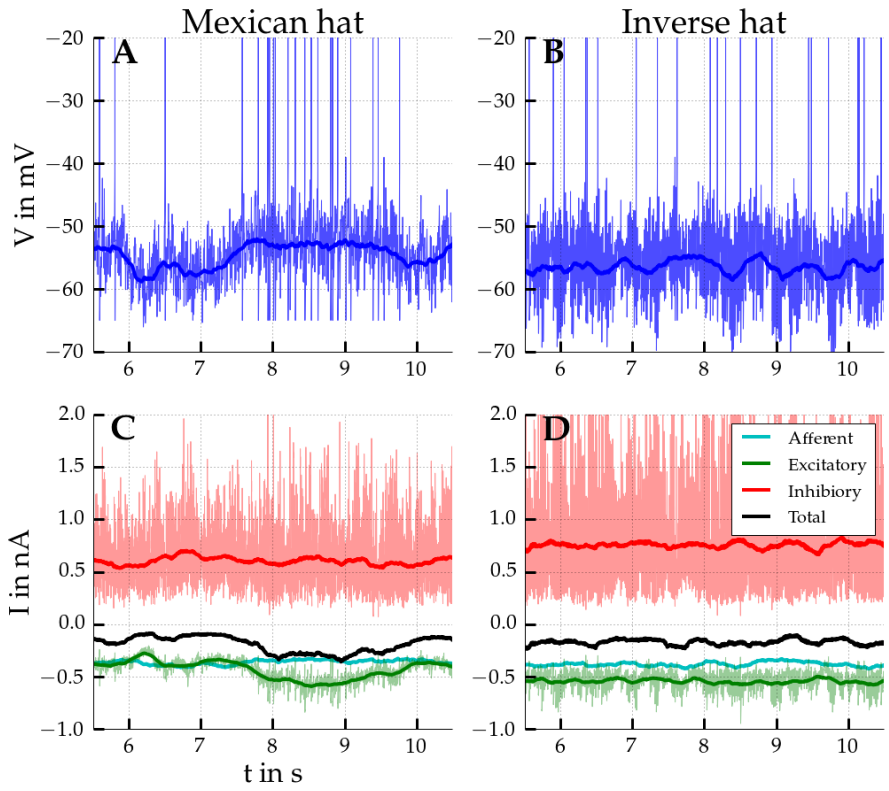

Figure 5.12: Voltage and current traces for a neuron from a Mexican hat network (A, C) and one from an inverse Mexican hat network (B, D). Strengths in both networks are $\bar{g}_{EE} = 0.35\,\mathrm{nS}$ and $\bar{g}_{IE} = 0.6\,\mathrm{nS}$, spreads are $\sigma_E = 125\,\mathrm{px}$, $\sigma_I = 250\,\mathrm{px}$ and $\sigma_E = 250\,\mathrm{px}$, $\sigma_I = 125\,\mathrm{px}$. Thick lines are moving averages with a window size of 300 ms. Top left (A): Voltage trace of a neuron in a Mexican hat network. The movement of bumps yields bursting activity on the individual cell level. Top right (B): Voltage trace of an inhibitory neuron which spikes irregularly. Bottom row (C, D): Current traces where the excitatory afferent current is given in cyan, recurrent excitatory in green, recurrent inhibitory in red, and total current in black. For the afferent and total currents only the moving averages are displayed.

regime that is just close to the bifurcation to criticality ($\bar{g}_{EE} = 0.425\,\mathrm{nS}$, blue line), one can see that for $\bar{g}_{EE} = 0.575\,\mathrm{nS}$ some of the neurons were firing almost with the maximum firing rate. These neurons participated in comparably broad bump structures. Likewise, figure 5.11 shows that for networks beyond criticality, there is a non-linear relation between input frequency and output frequency. At some point firing rates increased non-linearly and the most active 5% of cells showed increased firing of 250 Hz or more. Again, when we switched off input activity, the critical networks settled into states of persistent activity (data not shown).

Moreover, figure 5.9B shows an increase of the coefficient of variation (CV) averaged across all excitatory cells with an increase of $\bar{g}_{EE}$. The CV is based on the inter-spike interval (ISI) distribution, i.e. the distribution over time intervals between two

consecutive spikes of the same cell. The coefficient of variation of an individual neuron is defined as the standard deviation of the cell's ISI distribution over its mean:

$$CV_i = \frac{\mathrm{Std}(ISI_i)}{\langle ISI_i \rangle}. \tag{5.7}$$

As figure 5.9B illustrates with increasing strength the average CV exhibited a rather sudden jump beyond 1 after which a seemingly exponential growth happened. Of note, the value of 1 is a standard reference point because it is also the CV for any homogeneous Poisson process. Hence, the spiking activity of the cells was more random and irregular than Poisson spiking. CVs larger than 1 are usually indicative of bursting activity (Christodoulou and Clarkson, 1995). Bursting neurons show periods of no activity followed by periods with strong repetitive firing. In other words, they emit a couple of action potentials in so called bursts alternating with intervals of inactivity. Bursting can be identified by looking at voltage traces of individual neurons. Figure 5.12 depicts the voltage and current traces of a Mexican hat network neuron and a cell from a network with an inverse topology. For the former we observed that action potentials happened in bursts. Accordingly, the absolute value of the excitatory current increased for a period of time which was not counterbalanced by inhibition leading to increased spiking activity. Likewise, by looking at the raster plots of neural activity, as in figure 5.4, we can see that such bursting behavior was indeed present for each individual cell. Yet, if one takes the joint neural activity into account, one can understand that the bursts are due to the moving activity bumps. Thus, cells may be participating in a bump, i.e. bursting, or are silent if they are part of an inactive region.

Figure 5.9C shows the noise correlation coefficient r_{SC} over 50 trials using a time window of 1 second and averaged over cell pairs being at most 100 px apart. For lower connection strength $\bar{g}_{EE}$, the average noise correlation was essentially zero. For values $\bar{g}_{EE} \geq 0.2\,\mathrm{nS}$ there was a rapid increase followed by a plateau around 0.4 and a further increase beyond criticality. When inspecting the raster plots for small values of $\bar{g}_{EE} < 0.2\,\mathrm{nS}$, we observed no bump activity. From the theoretical discussion in the previous chapter we know that the homogeneous fixpoint with asynchronous activity can still be stable even in case of Mexican hat coupling due to a finite network size. Indeed, this was the case here. We tested different network sizes and scaled the number of afferent and recurrent connections and the connection spread proportionally. By contrast, we kept connection strength fixed because the post-synaptic potentials were already in a biologically plausible range and we did not want to render these too small. More simply, we changed the density of the network assuming that a multiple of excitatory neurons, i.e. $k \cdot 10{,}000$, occupies the same space as the original network with $10{,}000$ neurons. Using low values of $\bar{g}_{EE} = 0.15\,\mathrm{nS}$ and $0.2\,\mathrm{nS}$, we observed an increase in noise correlations between neurons that are at most $k \cdot 100\,\mathrm{px}$ apart for larger network sizes, as depicted in figure 5.13. Similarly, figure 5.14 shows that the bumps became more pronounced with an increase in network size. In contrast to what is predicted by

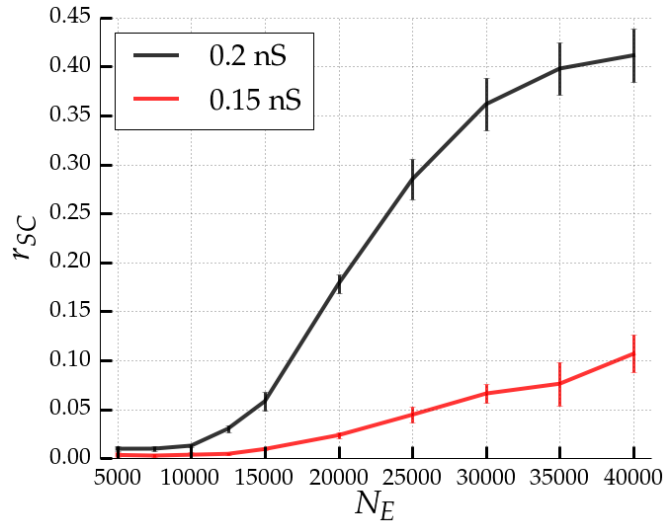

Figure 5.13: Average noise correlation as a function of network size. Correlations are computed and averaged from 50 trials between neurons that are at most $k100\,\mathrm{px}$ apart with k the size factor, $k = N_E/10{,}000$. The connection strength is kept fixed $\bar{g}_{EE} = 0.15\,\mathrm{nS}$ or $0.2\,\mathrm{nS}$ and $\bar{g}_{IE} = 0.6\,\mathrm{nS}$. Spreads are $\sigma_E = 125\,\mathrm{px}$ and $\sigma_I = 250\,\mathrm{px}$. The number of inhibitory neurons (N_I) as well as the number of connections (K_A, K_E, and K_I) and the spread of connections (σ_E and σ_I) scale linearly with the number of excitatory neurons (N_E).

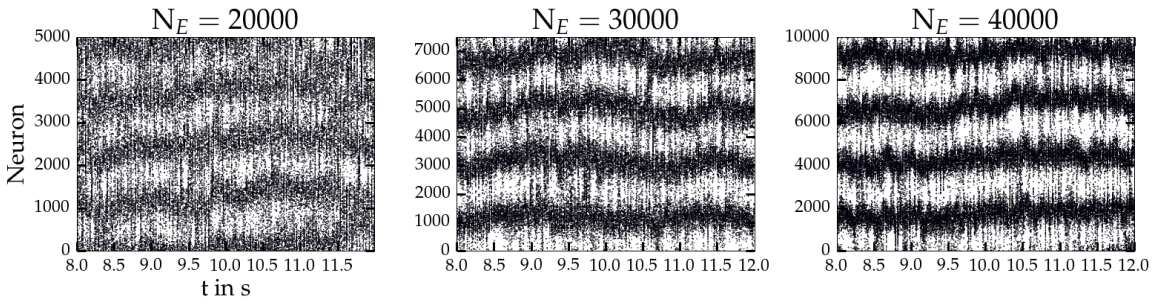

Figure 5.14: Raster plots of excitatory network activity in Mexican hat networks for different network sizes (20,000, 30,000, and 40,000 excitatory neurons; 5000, 7500, and 10,000 inhibitory neurons). Parameters as in figure 5.13.

Hansel and Sompolinsky (1998) (c.f. equation 4.77), we did not observe that the size of the active regions shrinked; not even for huge networks with 50,000 neurons (40,000 excitatory and 10,000 inhibitory).

The middle row (D-F) of figure 5.9 shows the fitted parameters for equation 5.5 for different connection strengths. As a reminder, the function nicely fitting the average noise correlations is the following damped oscillation:

$$\langle r_{SC} \rangle(d) = C_{r_{SC}} \exp(-|d|/\lambda_{r_{SC}}) \cos(\omega_{r_{SC}} d). \qquad \text{(5.5 revisited)}$$

The parameters were only fitted for $\bar{g}_{EE} \geq 0.2\,\mathrm{nS}$.

With increase in connection strength, the frequency of the noise correlations $\omega_{r_{SC}}$ decreased slightly in figure 5.9D. Likewise, by looking at raster plots of the corresponding runs, we could observe a decrease in bump frequency too (data not shown). This is in line with the theoretical considerations from the previous chapter. We saw for the simple case of a single population model that increasing the excitatory connectivity strength affects the spatial Fourier transform of the expected connectivity kernel. As was shown in figure 4.8, the maximum of the spatial Fourier transform increased, but also shifted to the left which yielded an amplification of lower spatial frequencies.

Furthermore, the relation between $\bar{g}_{EE}$ and $\lambda_{r_{SC}}$ was non-monotonic. First, in figure 5.9E we identified an increase in $\lambda_{r_{SC}}$ which means a slower decay of average correlations, followed by a decrease towards instability. Overall the exponential decay of correlations with distance was rather slow, similar to what was observed in figure 5.5A. Hence, there were even considerable positive and negative correlations between neurons with wide distances of 4000 or 5000 pixels apart.

As expected, the shape of the curve for the fitted initial correlations $C_{r_{SC}}$ in figure 5.9F strongly resembles the one above (figure 5.9C) showing the empirical average noise correlation coefficient r_{SC}. Since the empirical one is based on cell pairs at most 100 px apart, its magnitude is a bit lower than compared to $C_{r_{SC}}$. As a reminder, $C_{r_{SC}}$ fits the hypothetical average noise correlation value for cell pairs with no distance in between them.

The bottom row (G-I) of figure 5.9 displays the parameter fits as a function of inhibitory spread σ_I for a fixed connection strength of $\bar{g}_{EE} = 0.4\,\mathrm{nS}$. Similar to increasing connection strength, widening the inhibitory connection spread yielded a decrease in the spatial frequency of noise correlations (G). As before, this was expected from theory discussed in the previous chapter. Increasing the width of inhibitory coupling in the single population model shifted the maximum of the Fourier transform to the left. This in turn amplified slower spatial frequencies (c.f. figure 4.9). Surprisingly, there was no effect of inhibitory connection spread on the decay $\lambda_{r_{SC}}$ (H) and only a minor one on the amplitude parameter $C_{r_{SC}}$ (I). We observed a weak decrease in $C_{r_{SC}}$ for wider spreads.

5.3.3 Boundary Conditions

Clearly, a network's boundary conditions are important with respect to the theoretical considerations of the previous chapter. In order to translate the rate model's activity into the spatial Fourier domain, circular boundary conditions are necessary. Still, we wanted to know if the theoretical findings also apply to less strict assumptions. Do Mexican hat networks yield spatial activity patterns for other boundary conditions as well?

We ran simulations with absorbing as well as reflecting boundary conditions. In the former case, neurons close to one of the boundaries received less inputs from their neighbors because outgoing connections beyond the boundary were simply cut off. In the latter case connectivity at the boundaries was denser than in the center of the network. Connections that were supposed to reach over the boundary were mirrored and projected back into the network.

Figure 5.15 shows network activity for these two different boundary conditions. As for circular boundary conditions, we observed the emergence of spatial patterns in Mexican hat networks close to criticality. In turn, these spatial patterns yielded noise correlations (data not shown). Likewise, no patterns and correlations were observed for balanced and inverse spreads (data not shown).

However, using reflecting or absorbing boundary conditions introduces inhomogeneities into the spatial topology. As a consequence, analyses about distance dependencies of variables become difficult. Besides the cell distance, the location of each cell relative to the network boundaries is important. Therefore, throughout the rest of the thesis we will restrict all simulations and analyses to circular boundary conditions.

5.3.4 Sinusoidal Stimuli

So far we investigated the correlated variability for homogeneous inputs where each cell in the network was driven by Poisson input with the same frequency. We could discern the emergence of moving bumps, i.e. we observed a bifurcation to a spatial pattern similar to the marginal phase predicted by Hansel and Sompolinsky (1998). We further measured correlated variability that followed the same spatial scales as the bumps.

What happens if the input is tuned according to some particular function? In the following we will investigate sinusoidally tuned input as previously defined in equation 5.2 with different spatial frequencies ω_s as well as different maximum input strengths $\nu_{\text{Aff, max}}$. As a reminder, the tuning function of choice is

$$\nu_{\text{Aff},k}(t) = (1 + \sin(2\pi\,\omega_s\,k))\,\bar{\nu}_{\text{Aff}} + \nu_{\text{Aff, base}}, \qquad \text{(5.2 revisited)}$$

$$\bar{\nu}_{\text{Aff}} = \frac{\nu_{\text{Aff, max}} - \nu_{\text{Aff, base}}}{2}. \qquad \text{(5.3 revisited)}$$

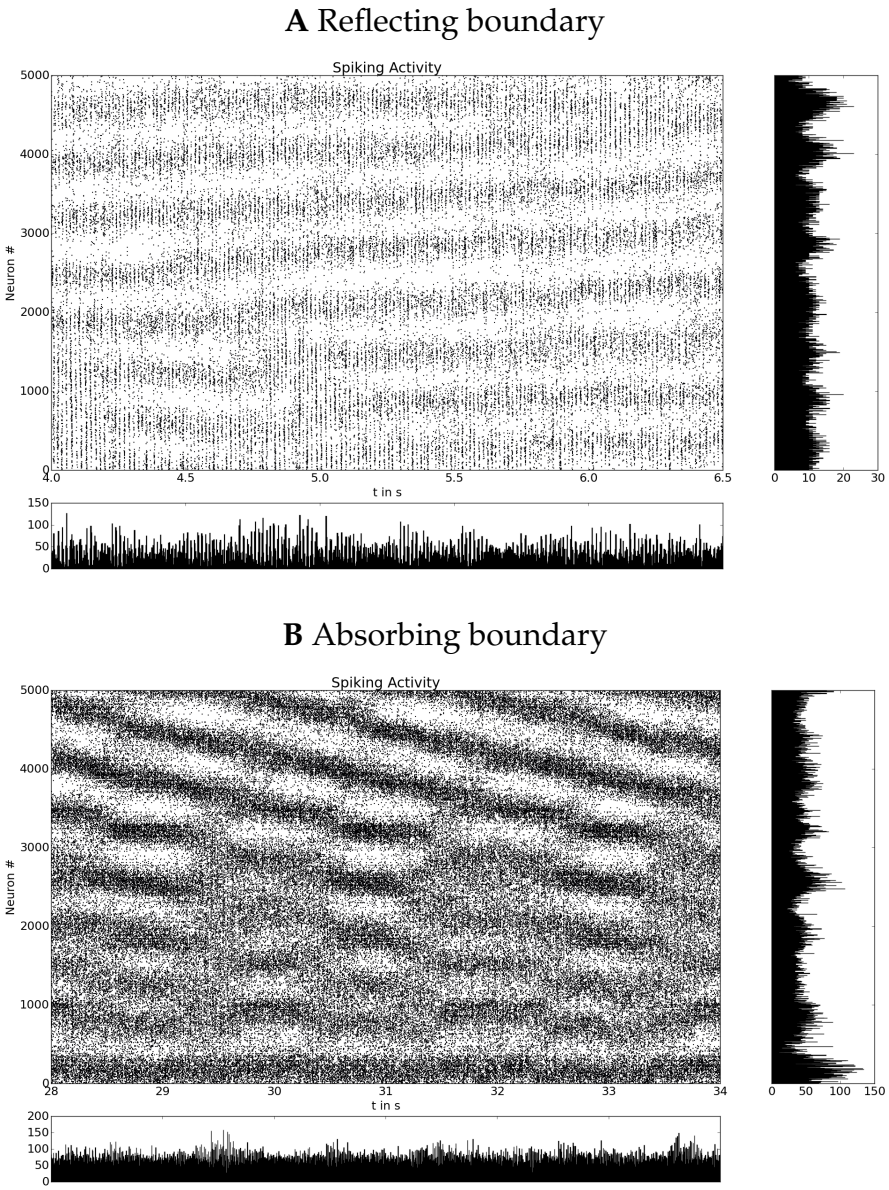

Figure 5.15: Spiking activity of a Mexican hat network ($\sigma_E = 125\,\mathrm{px} < \sigma_E = 250\,\mathrm{px}$, $\bar{g}_{EE} = 0.3\,\mathrm{nS}$, $\bar{g}_{IE} = 0.6\,\mathrm{nS}$) for reflecting (top) as well as absorbing (bottom) boundary conditions. A small black dot represents a spike of a particular neuron at a particular point in time. Neurons are ordered according to their position on the ring. Small plots show the activity histograms over time (horizontal) as well as over space (vertical). As for circular boundary conditions spatial patterns are observed.

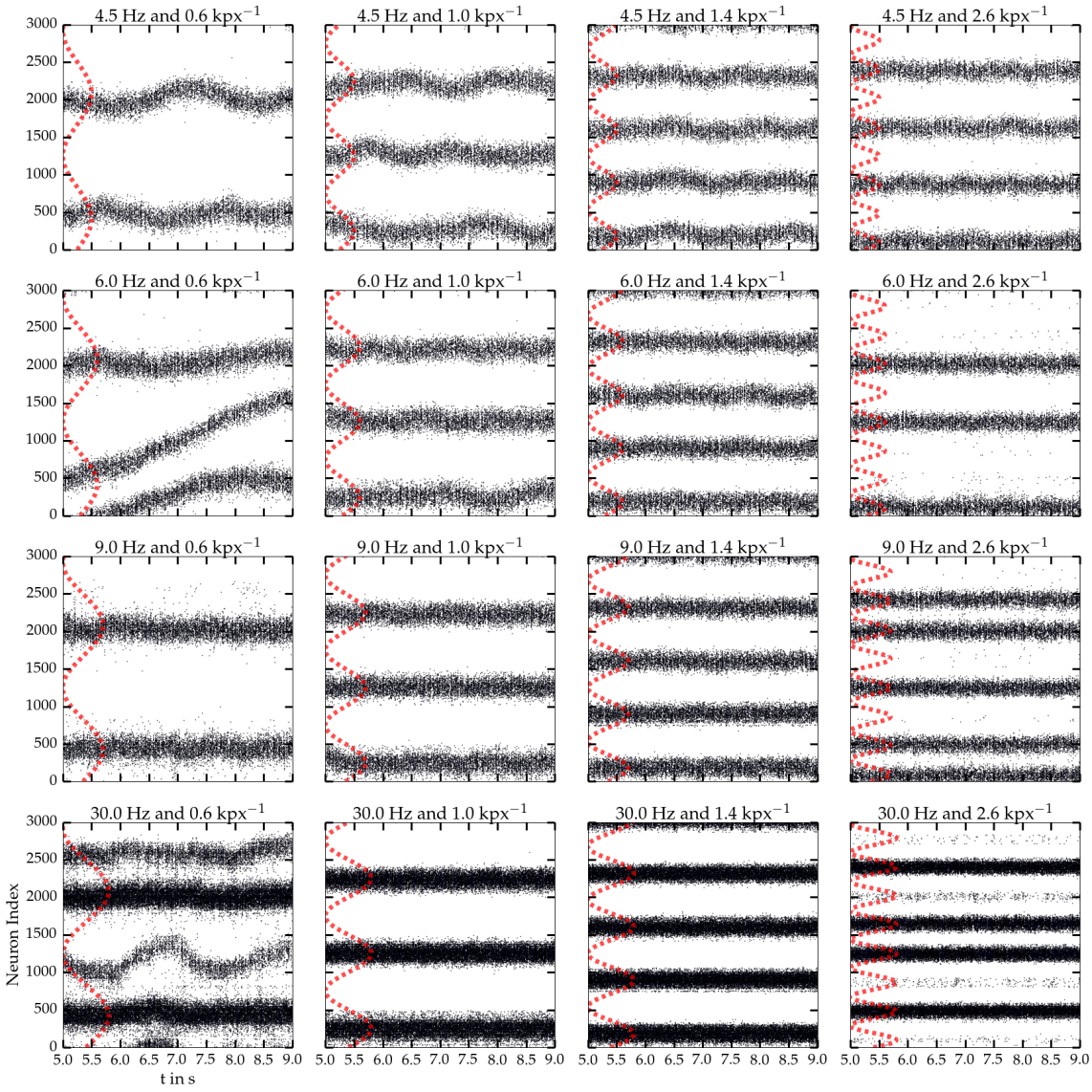

Figure 5.16: Raster plots for different spatially heterogeneous inputs to a Mexican hat network ($\sigma_E = 125\,\text{px}$, $\sigma_I = 250\,\text{px}$) close to criticality ($\bar{g}_{EE} = 0.4\,\text{nS}, \bar{g}_{IE} = 0.6\,\text{nS}$). From left to right there is an increase in spatial frequency of the input ω_s, whereas top to bottom the maximum frequency $\nu_{\text{Aff, max}}$ is increased. The red sinusoid sketches the spatially modulated input.

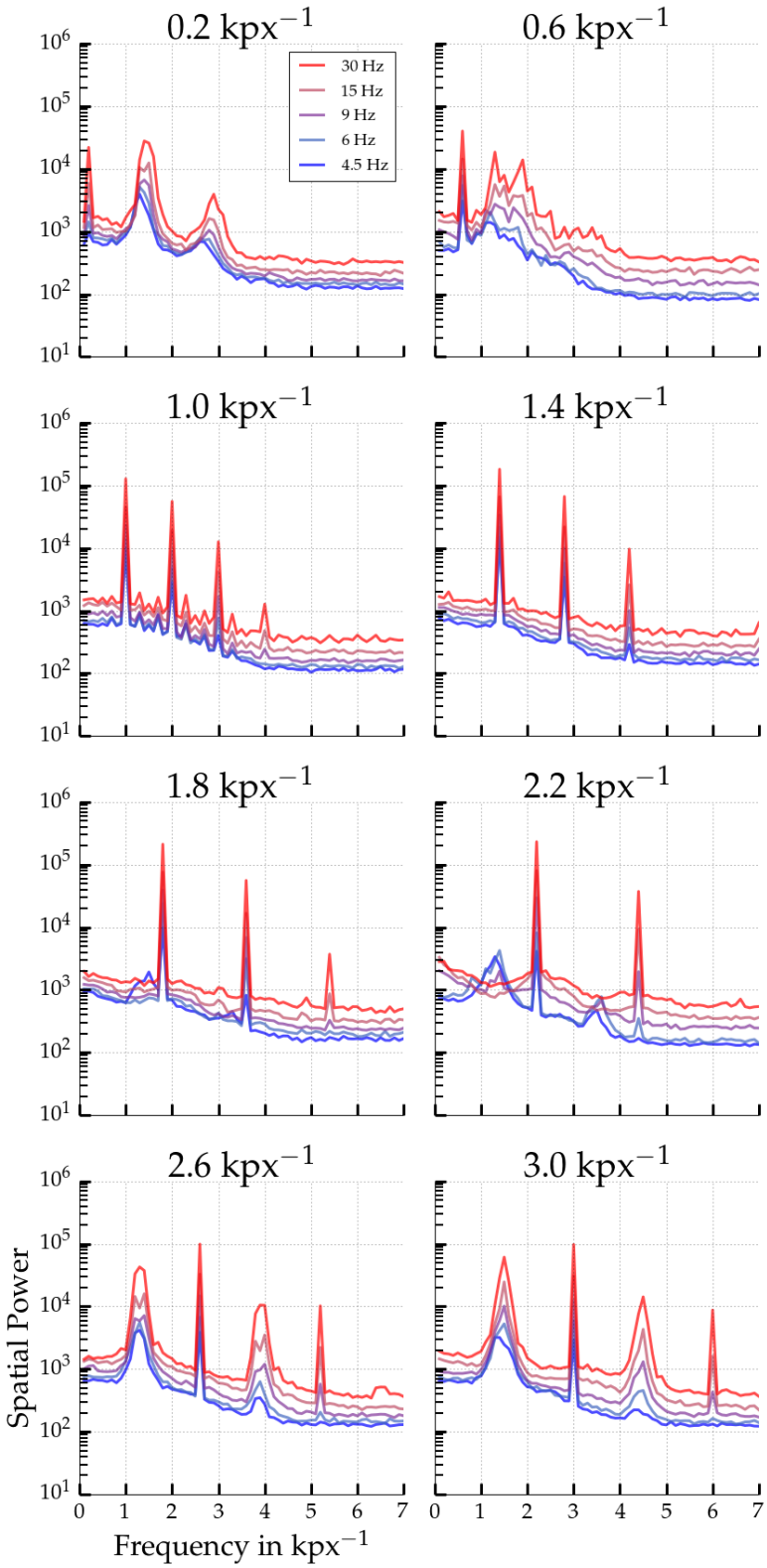

Figure 5.17: Spatial Power Spectrum of Mexican hat networks for different spatial driving frequencies ω_s and maximum input $\nu_{\text{Aff, max}}$. The spatial driving frequencies ω_s are given in the captions, maximum input frequencies $\nu_{\text{Aff, max}}$ are depicted by the differently colored lines. Network parameters as in figure 5.16.

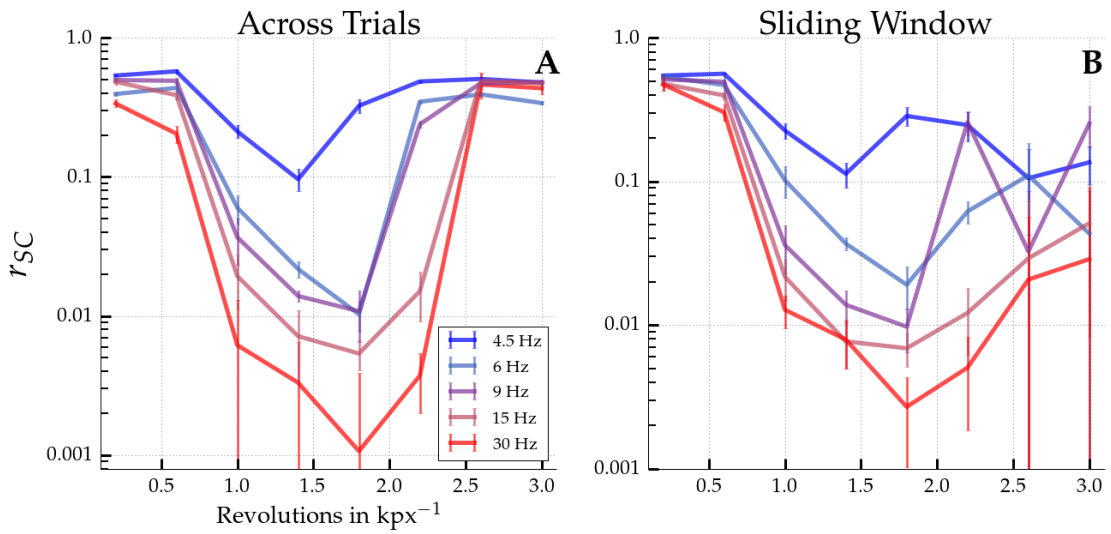

Figure 5.18: Average noise correlations among pairs at most 100 px apart within Mexican hat networks as a function of spatial driving frequency ω_s are shown. Different maximum frequencies $\nu_{\text{Aff, max}}$ are depicted by the differently colored lines. On the left (A) correlations are computed for 1 second of simulation across 50 trials. On the right (B) correlations are computed from a sliding window of 1 second size across a single long trial of 30 seconds. Error bars mark standard deviations across 5 network realizations. Network parameters as in figure 5.16.

We expected an interaction between the spatially inhomogeneous attractor for homogeneous input and the shape of the tuned input. This is indeed what we observed. For lower spatial frequencies ω_s (0.2 kpx^{-1}, 0.6 kpx^{-1}) we recognized freely moving bumps, whereas for intermediate frequencies that matched or were close to the spatial frequency of the bumps for homogeneous input (1.4 kpx^{-1}, 1.8 kpx^{-1}), the bumps were tightly locked to the input frequency and showed almost no movement. For faster spatial frequencies (2.6 kpx^{-1}, 3.0 kpx^{-1}) jitter among the locked bumps was slightly increased. Moreover, the location of the bumps varied across trials and we observed two different widths of the active regions. Thick bumps alternating with rather thin activity that is still phase locked to the input frequency could be recognized (figure 5.16). In addition, for intermediate input frequencies ω_s that match the bump frequency of the network, increasing the amplitude or maximum frequency $\nu_{\text{Aff, max}}$ of the input strengthened the bump locking to the spatial maxima of the input. Hence, for weaker maximum frequencies like $\nu_{\text{Aff, max}} = 4.5$ Hz or $\nu_{\text{Aff, max}} = 6$ Hz bumps showed a slight movement around the peaks of the input, whereas this jitter vanished for larger frequencies $\nu_{\text{Aff, max}}$.

Furthermore, the locking to the driving frequency was also reflected in the observed spatial power spectra of network activity as depicted in figure 5.17. For the lowest spatial input frequency $\omega_s = 0.2$ kpx^{-1}, the spatial power spectrum resembles the one for the homogeneous input as in figure 5.5C. Increasing the frequency ω_s to intermediate values, however, yielded sharp peaks at the driving input frequencies and the higher

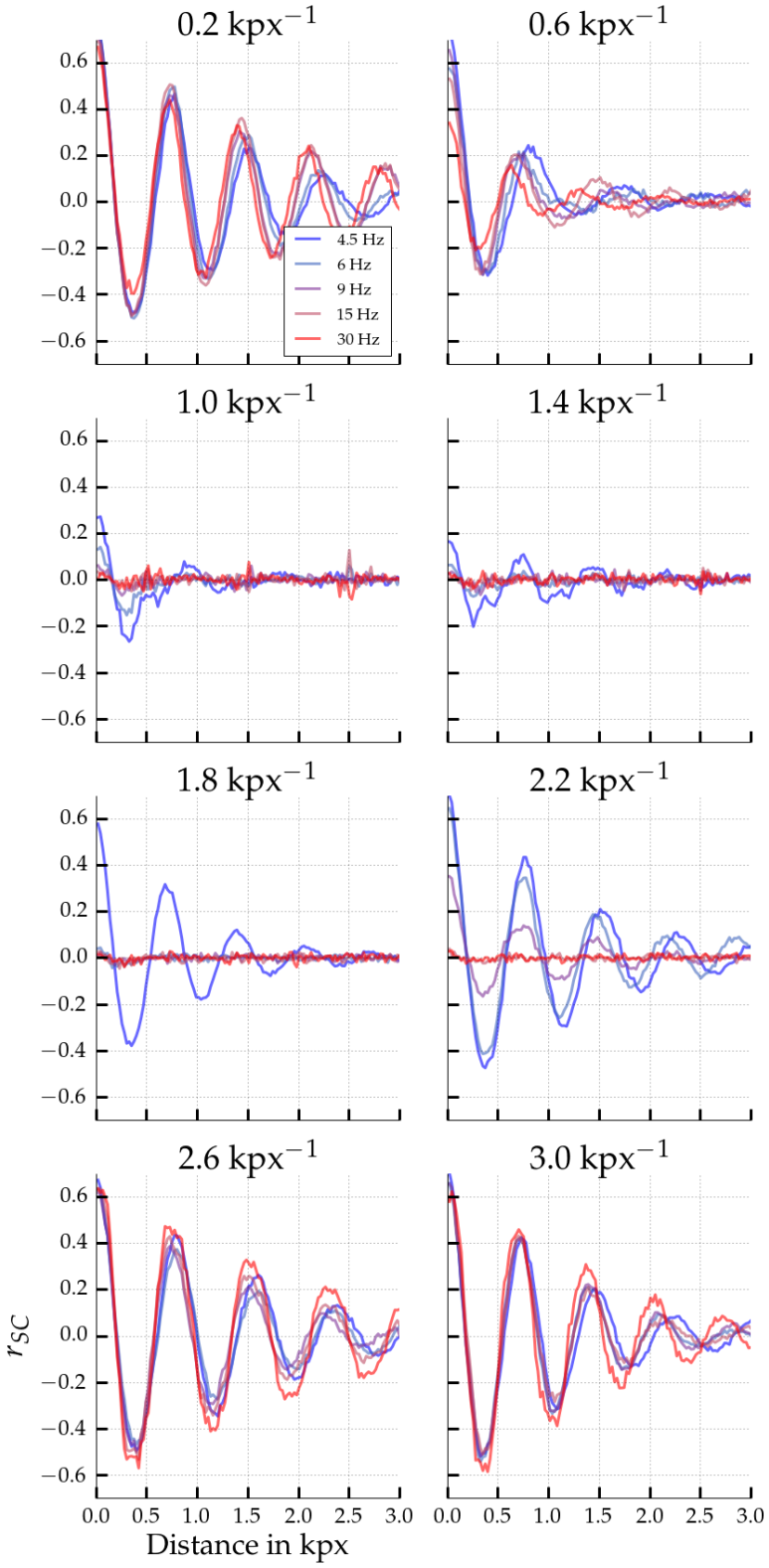

Figure 5.19: Distance dependency of noise correlations in a Mexican hat network for spatially heterogeneous input. Subplots show distance dependency for different spatial input frequencies ω_s, colored lines indicate the maximum driving frequency $\nu_{\text{Aff, max}}$. The average empirical noise correlation in bins of 20 px of distance between pairs of cells is given. Network parameters as in figure 5.16.

harmonics. These sharp peaks also followed the driving frequency for higher spatial frequencies of 2.2 kpx^{-1} and more. Yet, for such high spatial frequencies, an additional broader peak was observed at the frequency band of spatial patterns for the homogeneous drive at 1.4 kpx^{-1}.

Similar effects could be found for the measured noise correlations. Figure 5.18 shows the average noise correlations over an integration window of 1 second for pairs at most 100 px apart. Correlations were weakest for stimuli which spatial frequency ω_s resemble the inherent frequency of the bump activity. In this setting, increasing the maximum frequency $\nu_{\text{Aff, max}}$ decreased correlations further. Moreover, there is a discrepancy for higher spatial input frequencies ω_s between noise correlations measured over trials, as depicted in figure 5.18A, and measured as a sliding window over a single long trial in figure part B. As mentioned before, for such higher spatial input frequencies, the location of the bumps changed over trials, but was rather stable within a trial. Consequently, measuring noise correlations over a single trial yielded weaker values.

Next, we will take a closer look at the distance dependency of noise correlations for inhomogeneous input (figure 5.19). As before for the homogeneous input, we measured on average a damped sine wave of noise correlations for a wide range of frequencies and input strengths. Yet, if spatial input frequency ω_s was similar to the inherent frequencies of the bumps around 1.4 kpx^{-1}, we identified a strong weakening of the amplitude of noise correlations. Likewise, increasing the maximum input frequency $\nu_{\text{Aff, max}}$ further decreased the amplitude of noise correlations. The wave characterizing the distance dependency of correlations almost vanished completely for a maximum frequency of $\nu_{\text{Aff, max}} = 30$ Hz.

5.4 Noise Correlations in Two-Dimensional Networks

In the previous sections we looked at simulations of one-dimensional ring networks. We observed that Mexican hat networks close to criticality undergo a bifurcation to a spatially inhomogeneous state similar to the marginal phase discussed by Hansel and Sompolinsky (1998). Heterogeneous activity emerged showing a pattern of bumps which moved across the network. These bumps, in turn, yielded distance dependent noise correlations and the dependency followed the spatial spectrum of the bumps. Moreover, using spatially tuned afferent input, we observed an interaction between the frequency of the tuning and the inherent frequency of the bump activity.

In the following sections we will discuss repetitions of some of the experiments in a biologically more plausible setting with two-dimensional networks. These correspond to sheets of neurons from one layer of visual cortex. Our model is based on a previous study by Stimberg et al. (2009) where the distances within the network were scaled to match primary visual cortex of a cat. Consequently, one pixel corresponds to 15 µm in cortical dimensions. We tested models of different size using either 100 × 100 excitatory neurons or a 200 px × 200 px grid with 2500 and 10,000 inhibitory neurons, respectively.

Moreover, every neuron received $K_A = 100$ excitatory afferent inputs with 10 Hz input firing rate as well as $K_E = 400$ recurrent excitatory and $K_I = 200$ recurrent inhibitory connections. For the smaller 100 px $\times$ 100 px networks we also simulated responses to heterogeneous, orientation tuned input using an orientation map consisting of 4 pinwheels as depicted in figure 5.1. The tuned input will be discussed later after an analysis of homogeneous stimuli in the following sections. As before, we used periodic boundary conditions which means the two-dimensional spatial topology is a torus.

5.4.1 Correlated Variability and Spatial Patterns

Similar to a one-dimensional network, we observed a distance dependency of noise correlations for a large 2D Mexican hat network ($N_E = 200 \times 200$, $\sigma_E = 10\,\mathrm{px} < \sigma_I = 15\,\mathrm{px}$) close to criticality ($\bar{g}_{EE} = 0.4\,\mathrm{nS}$, $\bar{g}_{IE} = 0.6\,\mathrm{nS}$) that followed a damped wave, as depicted in figure 5.20A. We did not make such an observation for balanced (data not shown) or inverse networks (inset). In terms of biological scales, the decay of average correlation (green line) happened rather quickly and there was already a considerable decrease in the amplitude within the first 100 pixels, which corresponds to 1.5 millimeters scaled to cat cortex. Hence, it is more suitable to interpret the activity as the emergence of individual bumps with minor influence on each other's movement instead of a wave solution. Moreover, the oscillation of the average r_{SC} was fast too. It was about 2.2 cycles per 100 pixels.

Furthermore, two example distribution of spike counts for a pair close by and a pair of cells far apart are given at the bottom in figure 5.20. Moreover, the joint spiking activity underlying the distance dependent correlations revealed bump or stripe like patterns, as shown in figure 5.21. Thus, the set of observed heterogeneous activity became richer in comparison to the simple moving bumps in 1D networks. Round bumps may not only move and fuse together, but also eventually appear, disappear, or turn into flickering stripes. These stripes can be stable over a few seconds and may turn back into moving circular bumps. These spatial inhomogeneities were also reflected by the spatial autocorrelation shown in figure 5.22. The figure shows an almost radial-symmetric damped oscillation similar to our observations of the decay of noise correlations with distance in figure 5.20A.

Moreover, similar to the observations made for a one-dimensional network, we found a saturation of the distance dependent correlations when increasing the integration time window. Figure 5.20B shows the average correlation among cell pairs at most 13.3 px (0.2 mm in cat scale) apart. Of note, the thin black line depicts the average r_{SC} estimated from a sliding window using only the first trial of 30 seconds. Thereby, we could correctly estimate noise correlation with much less data at least for integration windows smaller than 3 seconds. For larger windows, the sliding window method underestimated the trial based spike count correlation considerably. A single trial of 30 seconds did not provide a sufficient amount of data for correct estimation. Increasing the length of the trial to 120 seconds did in fact improve the estimate for longer

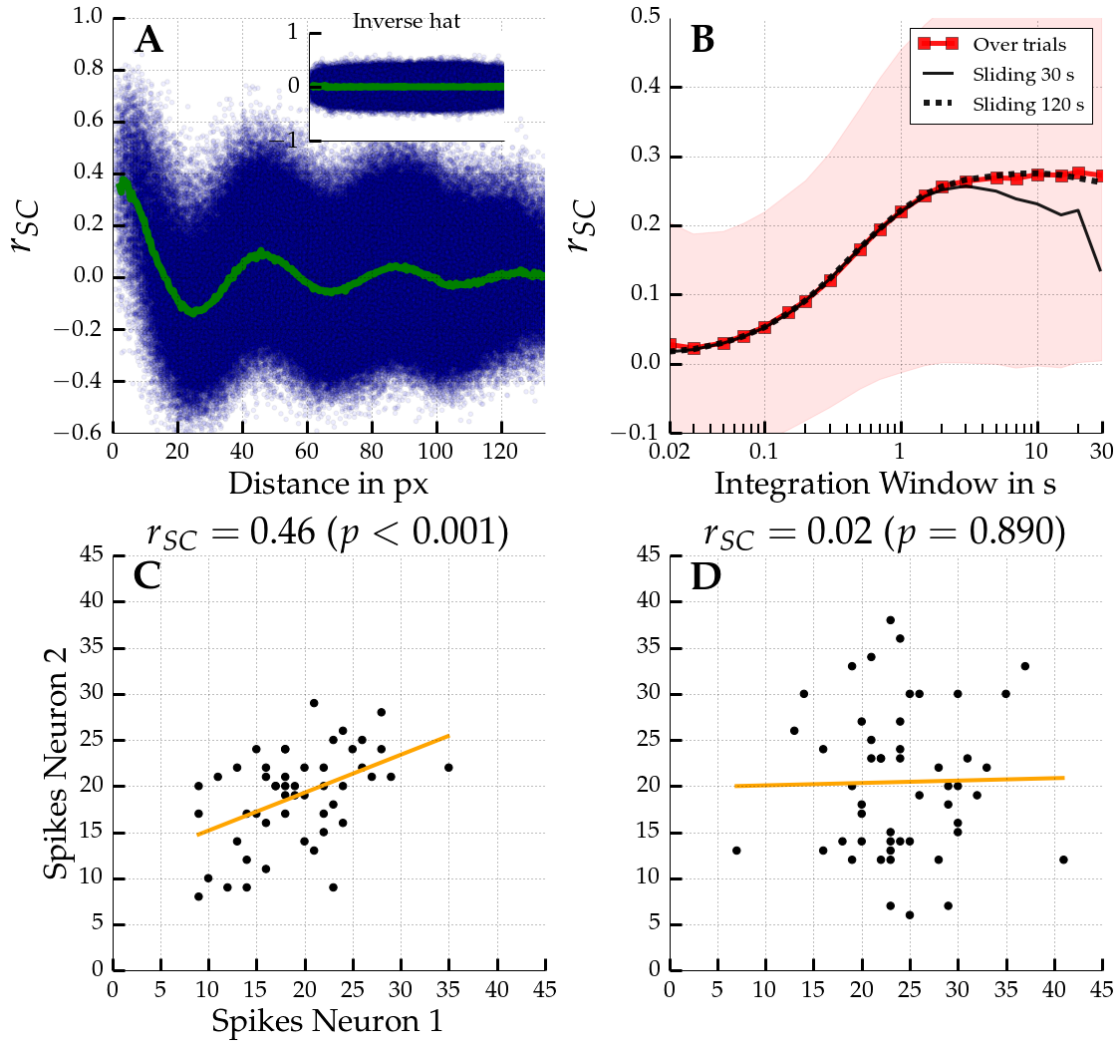

Figure 5.20: Top left (A): Noise correlations for pairs of cells according to distance computed across 50 trials using a 1 second integration window. Results are shown for a large 200 px × 200 px Mexican hat ($\sigma_E = 10\,\text{px} < \sigma_I = 15\,\text{px}$, $\bar{g}_{EE} = 0.4\,\text{nS}$, $\bar{g}_{IE} = 0.6\,\text{nS}$) network. The average correlation is shown in green. As for the 1D network, a decaying oscillation of average correlations can be observed. The small inset shows correlations for an inverse Mexican hat ($\sigma_E = 15\,\text{px} > \sigma_I = 10\,\text{px}$) with the same recurrent strengths where no distance dependency is observed. Top right (B): Temporal scales of correlations averaged across all cells at most 13.3 px (which corresponds to 0.2 mm in the scaling to cat cortex) apart. Increasing the window size beyond 1 yields a saturation of the curve. The envelope shows the standard deviation of noise correlation among all cell pairs. The thin black line shows the average noise correlation if estimated from sliding windows over a single experimental run of 30 s length. Estimating from a single run yields similar results than over trials for most window sizes, but deviates for larger time scales of 3 seconds and more. Using a longer trial of 120 s (dotted black line) gives correct estimates even for larger window sizes. Bottom row (C, D): Distribution of spike counts given a 10 second integration time window for two pairs of cells. Left: Two neighboring neurons. Right: Neurons far apart (133 px). The corresponding correlation coefficients are listed above together with the corresponding p-value. The linear fits are shown as orange lines.

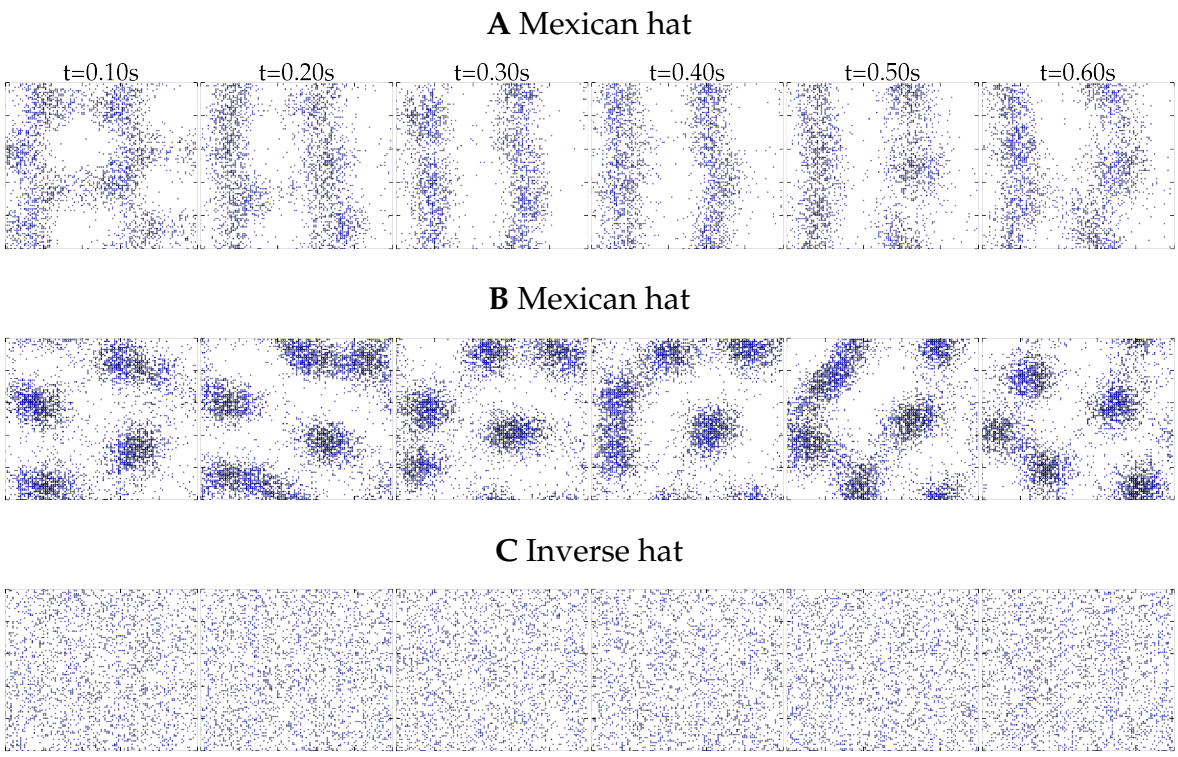

Figure 5.21: Consecutive snapshots of 100 ms length of network activity for 100 px × 100 px networks. A blue dot corresponds to an excitatory spike within the time interval. At the top the activity pattern of a Mexican hat network ($\sigma_E = 10\,\text{px} < \sigma_I = 15\,\text{px}$) close to criticality ($\bar{g}_{EE} = 0.4\,\text{nS}$ and $\bar{g}_{IE} = 0.6\,\text{nS}$) is shown. An operating point even a little closer to criticality ($\bar{g}_{EE} = 0.32\,\text{nS}$ and $\bar{g}_{IE} = 0.4\,\text{nS}$) is given in the middle. One can observe activity bumps that may appear, disappear, wander around, or flickering stripes that eventually turn into wandering bumps and vice versa. Activity of an inverse Mexican hat network close to criticality ($\sigma_E = 15\,\text{px} > \sigma_I = 10\,\text{px}$, $\bar{g}_{EE} = 0.4\,\text{nS}$, and $\bar{g}_{IE} = 0.6\,\text{nS}$) is shown at the bottom. There one only observes homogeneous asynchronous activity.

integration time windows (dotted black line). Since using a single and short trial of 30 seconds is sufficient to estimate the correlation for small integration windows, we used this method and a smaller network consisting of 100 × 100 excitatory neurons for extensive parameter explorations. These explorations will be discussed in the next section.

5.4.2 Parameter Explorations

To investigate if the emergence of noise correlations holds for wide parameter ranges, we varied the excitatory connections strength parameters $\bar{g}_{EE}$ and $\bar{g}_{IE}$ for networks with 100 × 100 excitatory neurons.

Figure 5.23 compares such an extensive parameter exploration of a Mexican hat and an inverse network topology. We can see that beyond criticality (thick dotted red line)

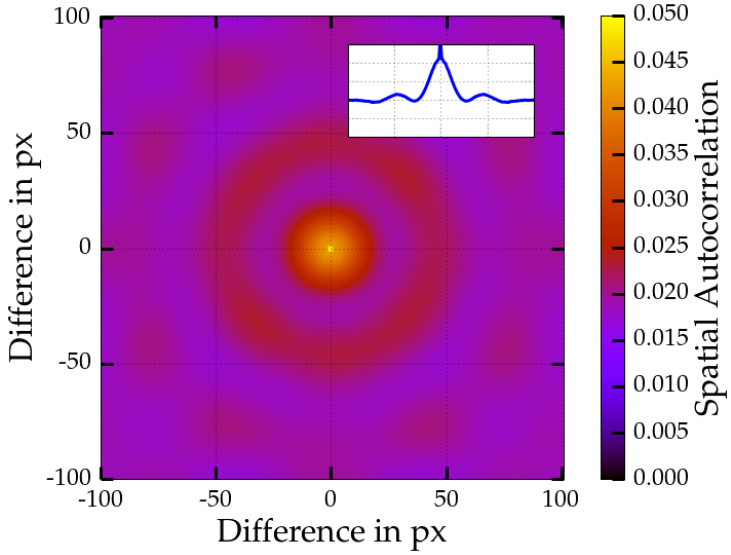

Figure 5.22: Spatial Autocorrelation in a Mexican hat network close to criticality. Network parameters as in figure 5.20. Autocorrelation is averaged across 5000 snapshots of 10 ms length. Inset shows autocorrelation along the horizontal axis at 0.

firing rates quickly diverged for both networks (A, B). Close to criticality we observed noise correlations for the Mexican hat network (E) but not for the inverse one (F). Correlations were averaged across all cell pairs at most 13.3 pixels (200 µm) apart. Configurations yielding noise correlations also exhibited average coefficients of variation (CV) larger than 1 (C). Hence, similar to the observations made in the one-dimensional networks, the emergence of bump and stripe patterns lead to bursting activity at the single neuron level.

The region where noise correlations emerged is rather thin and the Mexican hat network needs to operate in a recurrent regime with synaptic coupling strengths close to criticality. This is indicative of a strong finite size effect. Increasing the width of the Mexican hat, i.e. scaling σ_I, we measured strong correlations for much wider ranges, as depicted in figure 5.24.

Moreover, we performed an extensive parameter exploration of σ_E and σ_I while keeping the synaptic strengths fixed at values close to criticality ($\bar{g}_{EE} = 0.4\,\mathrm{nS}$ and $\bar{g}_{IE} = 0.6\,\mathrm{nS}$). As expected for Mexican hat but no other configurations we measured noise correlations, as shown in figure 5.25.

5.4.3 Orientation Stimuli

As for the one-dimensional networks, we tested spatially tuned afferent input in the two-dimensional models too. However, instead of simple sinusoidal input, we used a more realistic input scheme with a two-dimensional orientation map consisting of 4 pinwheels, as depicted in figure 5.1. Input frequencies were determined from Gaussian tuning functions. The Gaussian tuning was defined over the difference between the

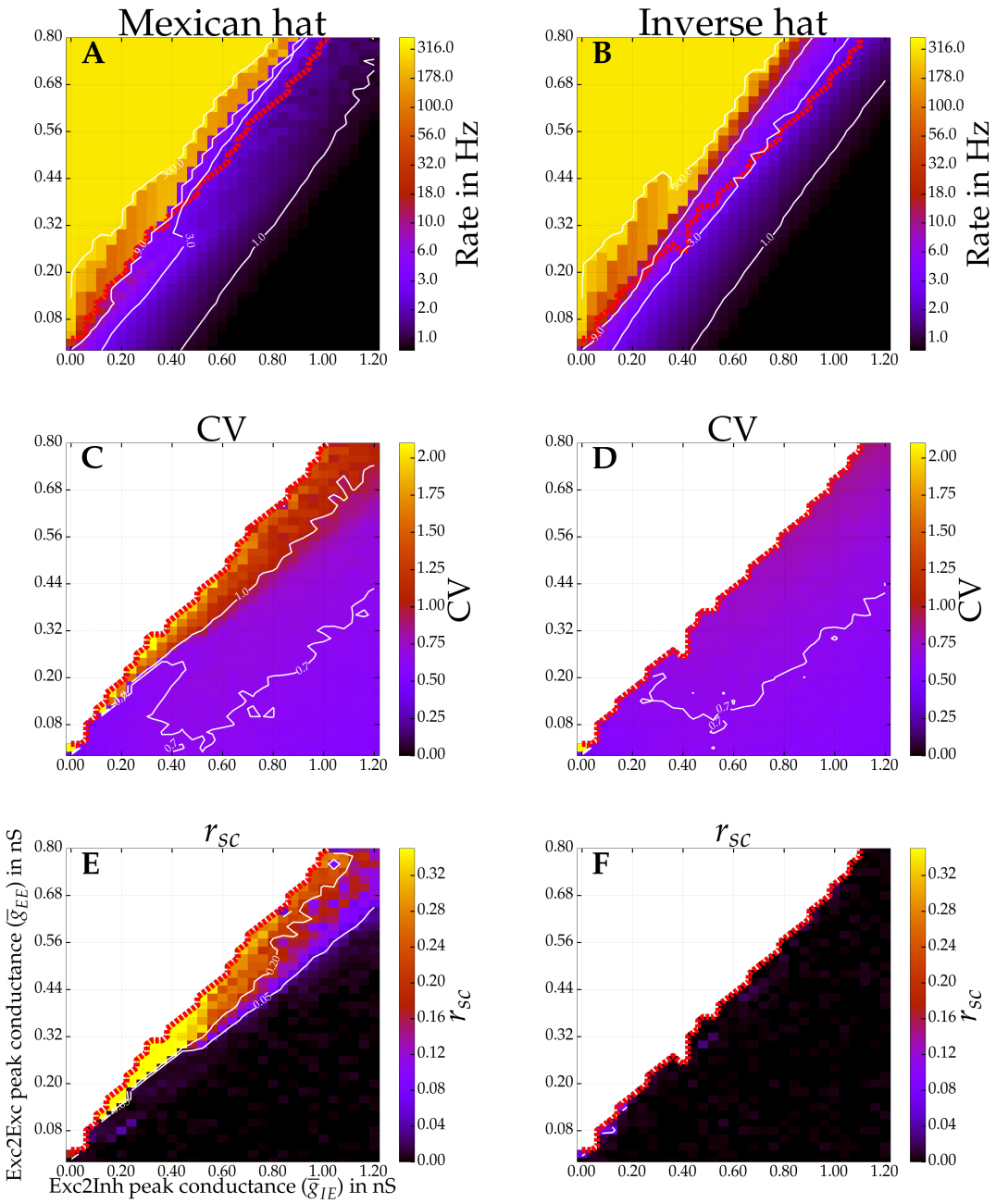

Figure 5.23: Parameter exploration of $\bar{g}_{EE}$ and $\bar{g}_{IE}$ for a Mexican hat network on the left ($\sigma_E = 10\,\text{px} < \sigma_I = 15\,\text{px}$) and an inverse network on the right side ($\sigma_E = 15\,\text{px} > \sigma_I = 10\,\text{px}$). Top row (A, B) shows the average firing rate (color coded) of all neurons in the network. The thick dotted red line marks the bifurcation to criticality. Middle row (C, D) displays the average coefficient of variation (CV). Third row (E, F) gives the average r_{SC} among close by cell pairs (at most 13.3 px apart) estimated from a single trail of 30 seconds and a sliding window of 1 second. For better visibility values beyond criticality are filtered and not shown. The unfiltered images are provided in appendix B.

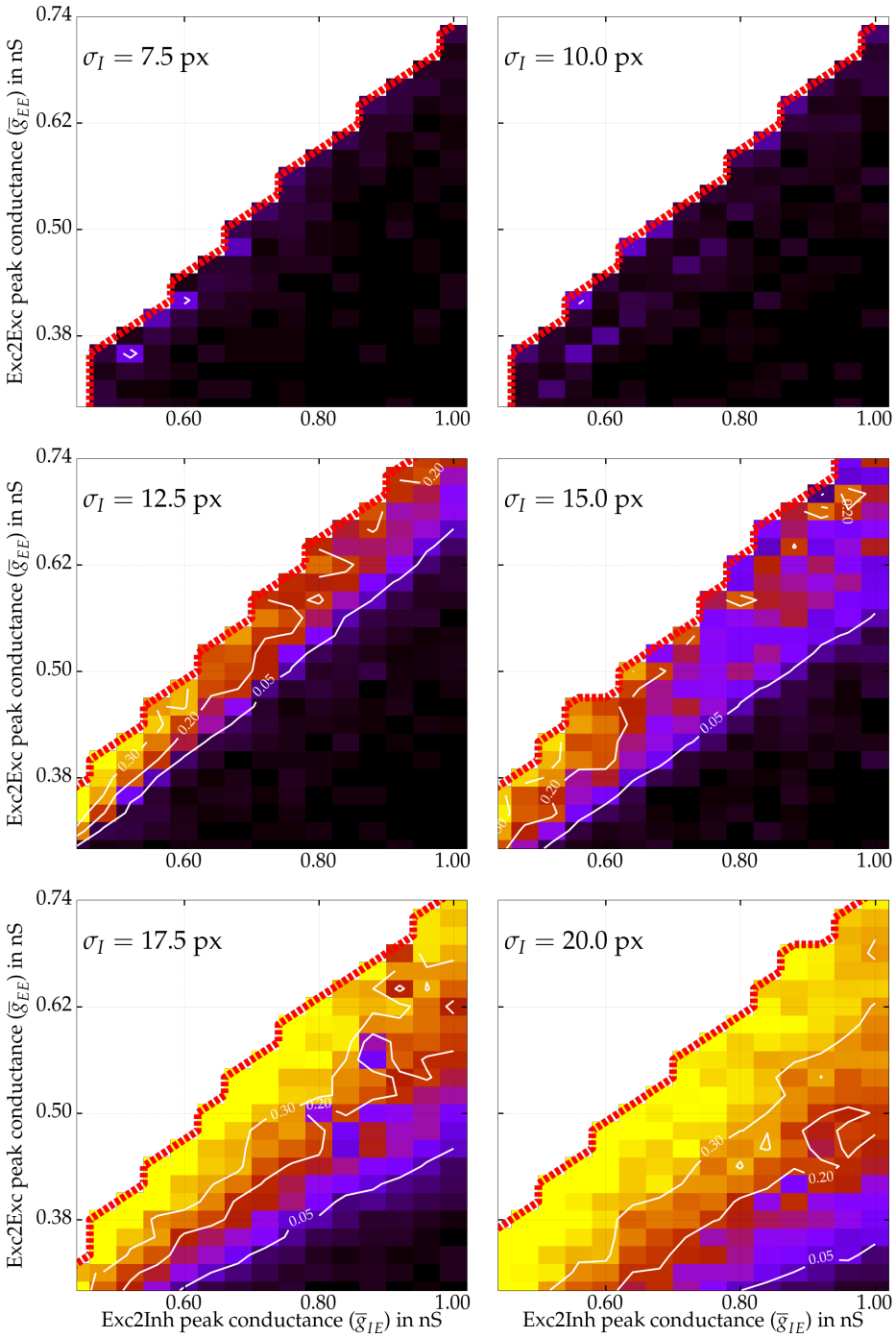

Figure 5.24: Average noise correlations (color coded, scale as in figures 5.23E and F) of cell pairs at most 13.3 px apart estimated using a sliding window of 1 second over one trail of 30 seconds for various network topologies exploring excitatory synaptic coupling strengths $\bar{g}_{EE}$ and $\bar{g}_{IE}$. The thick dotted red line marks the bifurcation to criticality. Excitatory spread σ_E is fixed to 10 px. Hence, the top row shows correlations for an inverse and balanced spread, whereas below correlations for Mexican hat networks are shown. The wider σ_I becomes, the wider the range where noise correlations are observed gets.

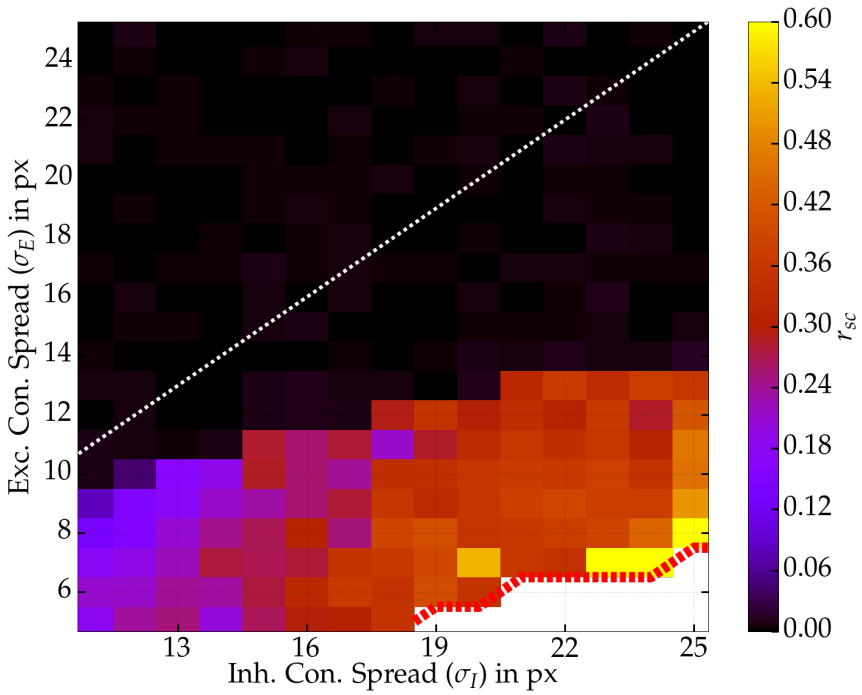

Figure 5.25: Extensive parameter exploration of connection spreads σ_E and σ_I for fixed recurrent regime close to criticality ($\bar{g}_{EE} = 0.4\,\mathrm{nS}$, $\bar{g}_{IE} = 0.6\,\mathrm{nS}$). Average noise correlation of close by cells maximally 13.3 px apart for an integration time window of 1 second over one trial of 30 seconds are color coded. White dotted line marks equality $\sigma_E = \sigma_I$. Thick red dottet line indicates bifurcation to criticality. Only for Mexican hat profiles the emergence of correlated variability can be observed.

stimulus orientation and the cell's preferred orientation according to the orientation map. As a reminder the tuning function is

$$\nu_{\mathrm{Aff},i}(s) = (\nu_{\mathrm{Aff,\,max}} - \nu_{\mathrm{Aff,\,base}}) \exp\left(-\frac{(s - s_{\mathrm{PO}_i})^2}{2\sigma^2_{\mathrm{Aff}}}\right) + \nu_{\mathrm{Aff,\,base}}. \qquad \text{(5.4 revisited)}$$

We observed an interaction between the tuned afferent drive and the network's bump structure in case of homogeneous input. Figure 5.26 shows the distance dependency of correlations using tuned afferent drive of different maximum strength in a Mexican hat network close to criticality. Correlations were averaged across 6 evenly spaced orientation stimuli and computed from 30 trials per stimulus. The strong sinusoidal fluctuations of noise correlations almost vanished completely. We observed only a minor dip below zero of the average noise correlation as a function of distance. Instead, the distance dependency looks like a linear decay within the first 25 pixels, corresponding to about 0.4 mm scaled to cat cortex. Additionally, as expected, no distance dependent correlations were observed for an inverse Mexican hat network with the same recurrent coupling strengths (inset of figure 5.26). Moreover, increasing the maximum input frequency $\nu_{\mathrm{Aff,\,max}}$ from 15 Hz (figure 5.26A) to 30 Hz (B) slightly decreased the magnitude of the correlations.

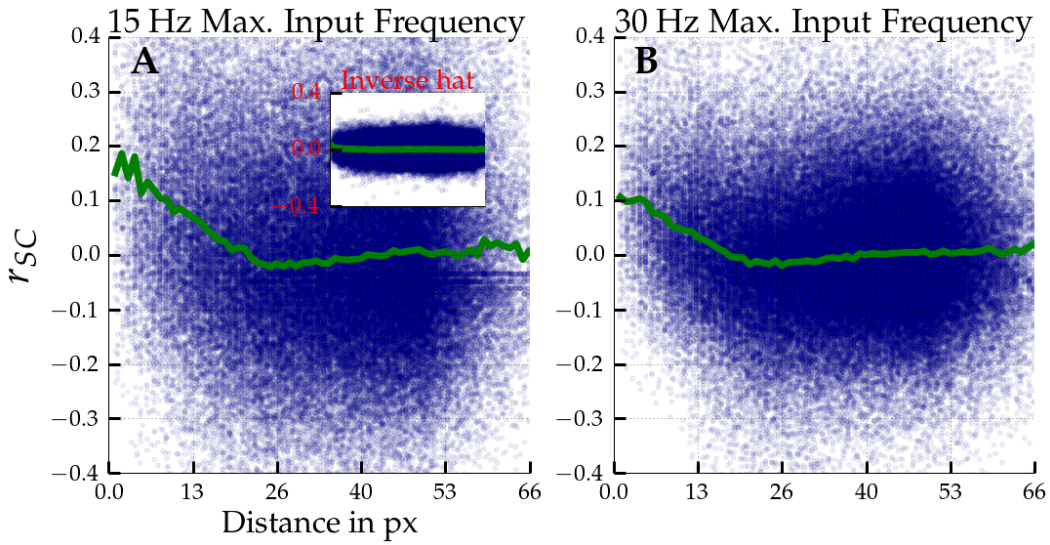

Figure 5.26: Distance dependent noise correlation in a Mexican hat network ($\sigma_E = 10\,\text{px} < \sigma_I = 15\,\text{px}$) close to criticality ($\bar{g}_{EE} = 0.4\,\text{nS}, \bar{g}_{IE} = 0.6\,\text{nS}$) driven by tuned input. Inset shows results for an inverse Mexican hat ($\sigma_E = 15\,\text{px} > \sigma_I = 10\,\text{px}$) with the same recurrent coupling strengths. Correlation coefficients were averaged across 6 stimulus orientations ($-89°, -59°, -29°, 1°, 31°, 61°$) with 30 trials per orientation and 1 second of simulation. On the left (A) noise correlations for a maximum input frequency $\nu_{\text{Aff, max}}$ of 15 Hz is shown and 30 Hz on the right (B).

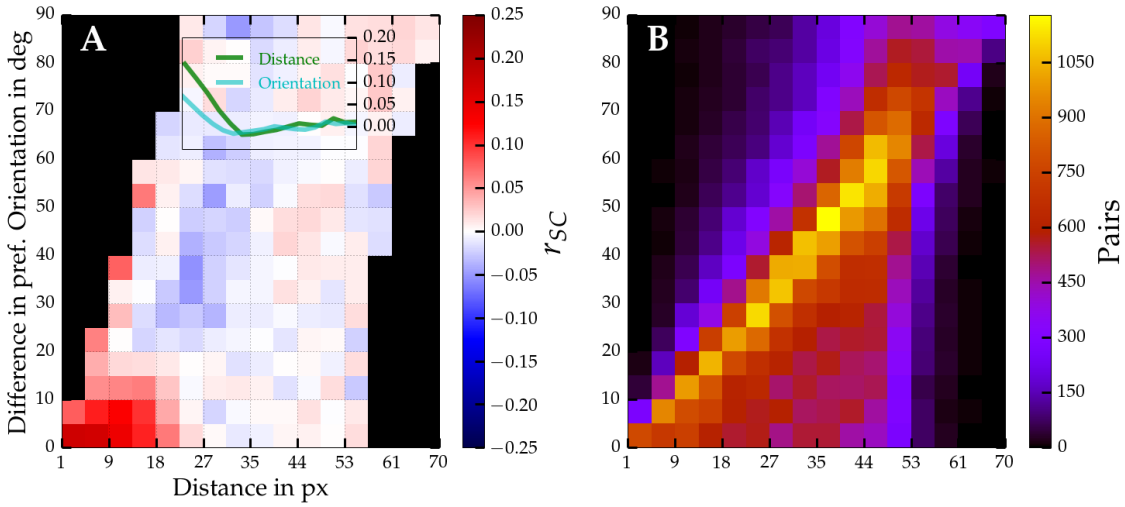

Figure 5.27: On the left (A): Distance (x-axis) and orientation (y-axis) dependent noise correlation in a Mexican hat network close to criticality driven by tuned input with a maximum frequency of 15 Hz. Other network parameters as in figure 5.26. Correlation coefficients were averaged across 6 stimulus orientations. Average noise correlations are color coded. In black two-dimensional bins that were excluded because of containing less than 50 pairs. Inset shows averages along each axis, in green noise correlations as a function of distance (c.f. figure 5.26A) and in cyan as a function of preferred orientation difference. On the right (B): Histogram of number of cell pairs in each bin according to cell distance and preferred orientation difference.

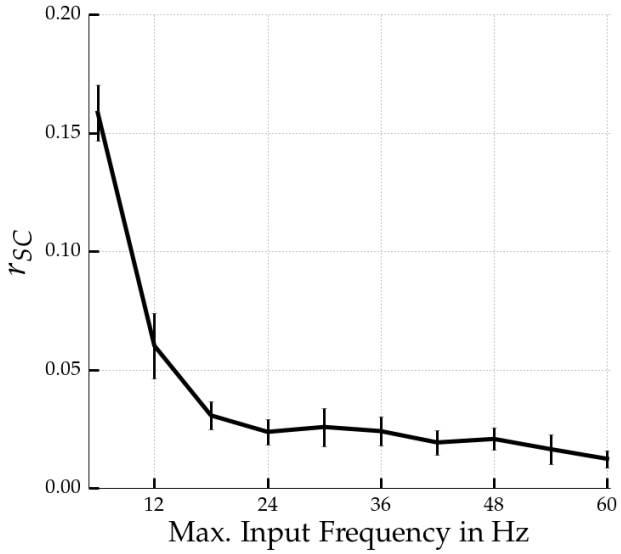

Figure 5.28: Noise correlations of cells at most 13.3 px apart in Mexican hat networks operating close to criticality ($\sigma_E = 10\,\mathrm{px} < \sigma_I = 15\,\mathrm{px}$, $\bar{g}_{EE} = 0.4\,\mathrm{nS}, \bar{g}_{IE} = 0.6\,\mathrm{nS}$) are shown as a function of maximum input frequency $\nu_{\mathrm{Aff,\,max}}$. Correlation coefficients were averaged across 6 stimulus orientations ($-89°, -59°, -29°, 1°, 31°, 61°$) with 30 trials per orientation and 1 second per trail. Errorbars mark standard devations over 5 network realizations. Increasing the maximum input frequency reduces correlations.

We further studied the dependency on difference in preferred orientations. Figure 5.27A depicts the average r_{SC} value as a two-dimensional function of distance as well as difference in orientations. The corresponding number of pairs, the spike count correlation coefficient was averaged across, is given in figure 5.27B. Clearly, strongest correlations were measured for close by neurons with similar preferred orientations. A gradient was observed in both directions. The r_{SC} as a function of difference in orientation only (cyan curve in inset) exhibits a similar shape as the spike count correlation coefficient as a function of distance (green curve in inset and c.f. figure 5.26A), but the overall magnitude is reduced. This is not surprising, because due to the map layout (figure 5.1) there is already a strong correlation between short distances and orientation difference. Still, because the map contains 4 pinwheels and activity between pinwheels was pretty much decorrelated, noise correlations — solely as a function of orientation difference — are averaged across pinwheels and thereby the magnitude is reduced.

In addition, we systematically investigated the influence of the maximum frequency $\nu_{\mathrm{Aff,\,max}}$ of the tuned input on correlated variability. Again similar to the observations in the one-dimensional case, an increase in maximum frequency lead to a decrease in noise correlations (c.f. figure 5.28). As a reminder, for the one-dimensional networks we observed a reduction in correlation most strongly for spatial input frequencies that matched the spatial frequency of the bumps. Here the situation for the

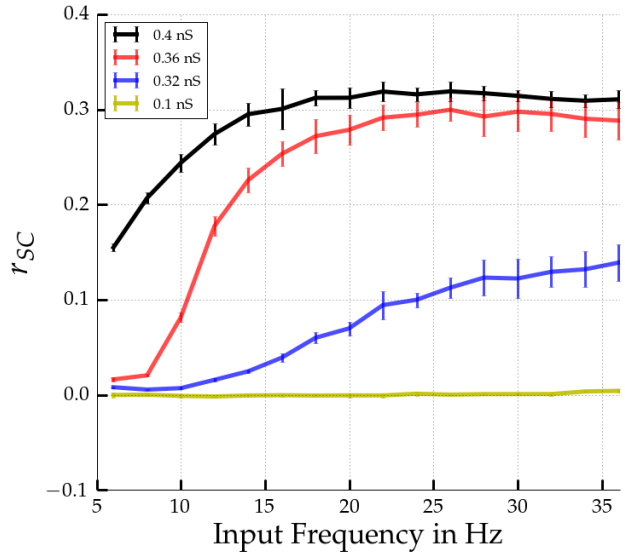

Figure 5.29: Average noise correlations as a function of input strength using a **homogeneous** stimulus presented in 50 trials and 1 second per trail. Plots depict average noise correlation of close by cells (not more than 13.3 px apart) within Mexican hat networks ($\sigma_E = 10\,\mathrm{px} < \sigma_I = 15\,\mathrm{px}$) for various excitatory to excitatory coupling strengths $\bar{g}_{EE}$. Excitatory to inhibitory strenght is kept fixed with $\bar{g}_{IE} = 0.6\,\mathrm{nS}$. Error bars mark standard deviations calculated across 5 network realizations. We can observe a saturation of average correlations for an increase in input strength. For regimes closer to criticality the correlations are stronger and saturation happens faster.

two-dimensional networks is similar. We measured a spatial frequency of noise correlations of about 2.2 cycles per 100 pixels for homogeneous input. This is close to the spatial frequency of the orientation map with 4 pinwheels in the 100×100 grid.

Moreover, the spatial tuning of the input is indeed very important for a reduction of correlations (figure 5.28). Increasing the firing rate of a blank stimulus had the opposite effect. As one can see in figure 5.29, scaling the homogeneous input firing rate yielded an increase followed by saturation of correlated variability. Of note, the black line corresponds to the recurrent excitatory strength also used in figure 5.28.

Furthermore, in the one-dimensional case we saw that an increase in maximum firing rate locked bump activity to the stimulus drive. Stronger maximum input meant less jitter in the activity bumps. Indeed, this is the case for the two-dimensional network too. In figure 5.30 network activity constantly clusters around 4 spots. These spots correspond to the orientation of the driving stimulus in the orientation map with 4 pinwheels. Nonetheless, just by looking at these snapshots of activity by eye, it is impossible to make out the relation between jitter and different maximum frequencies $\nu_{\mathrm{Aff,\,max}}$. To make this effect visible we looked at the movement of the centers of these activity clusters over time. In figure 5.31 the trajectory of the activity centers (black lines) at one of the network's pinwheels is depicted. For weak maximum frequencies like 6 Hz we identified considerable amount of movement in the plane. Increasing the

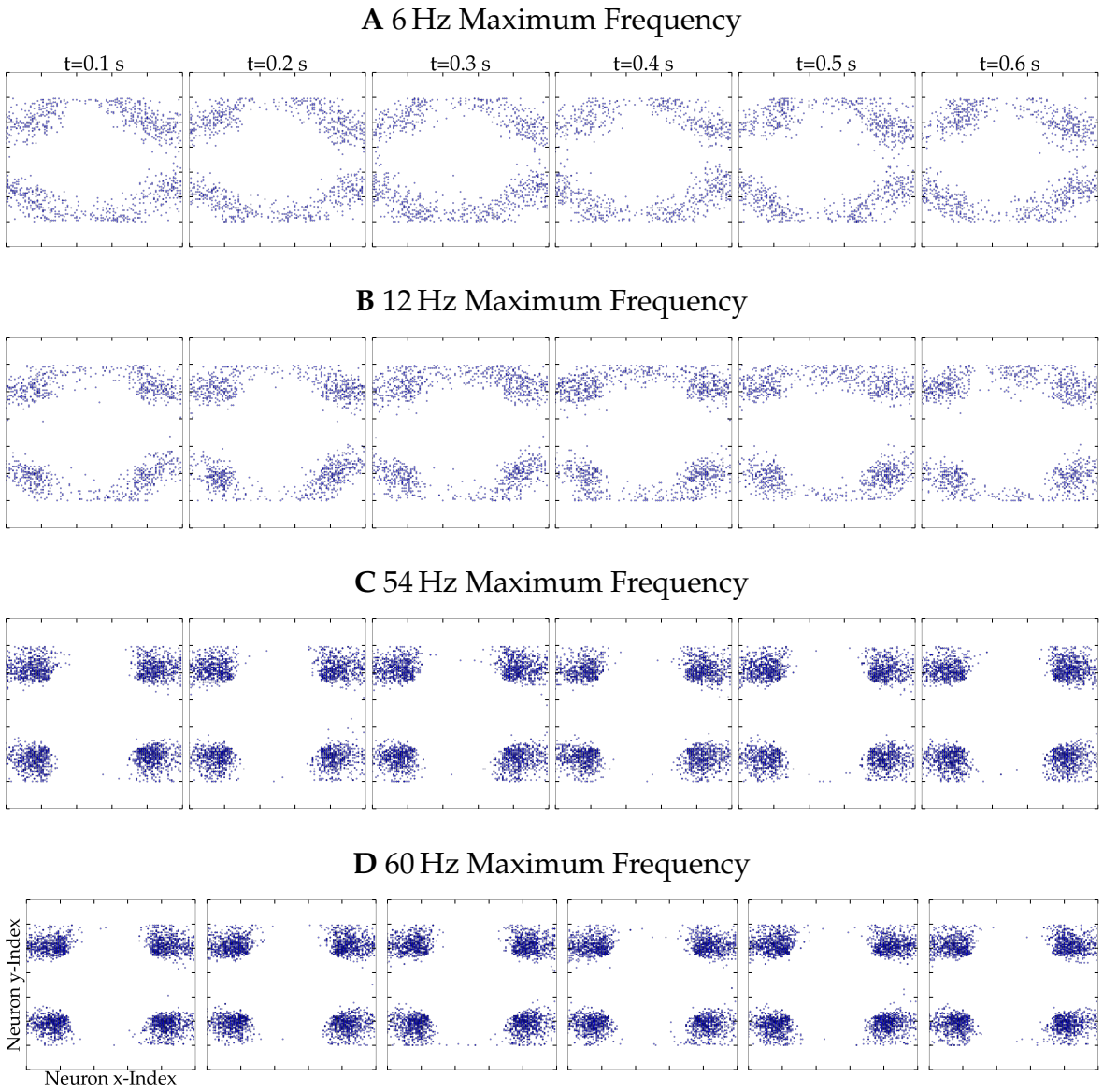

Figure 5.30: Consecutive snapshots of firing activity for a Mexican hat network close to criticality ($\sigma_E = 10\,\mathrm{px} < \sigma_I = 15\,\mathrm{px}$, $\bar{g}_{EE} = 0.4\,\mathrm{nS}, \bar{g}_{IE} = 0.6\,\mathrm{nS}$) driven by a stimulus of 1°. Each blue dot corresponds to a spike observed in a time interval of 100 ms. From top to bottom maximum input frequency $\nu_{\mathrm{Aff,\,max}}$ is increased from 6 up to 60 Hz.

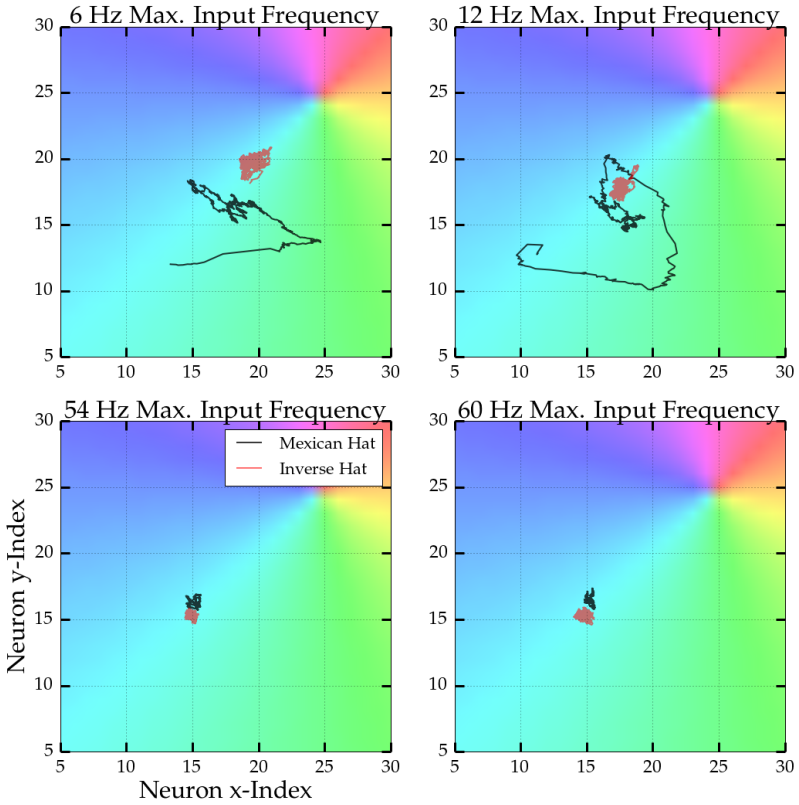

Figure 5.31: Trajectories of the center of activity at one pinwheel for four different maxium driving frequencies. Black line for a Mecican hat network (parameters as in figure 5.30) and a stimulus of 1° and in red for an inverse Mexican hat network ($\sigma_E = 15\,\mathrm{px} > \sigma_I = 10\,\mathrm{px}$) with the same connections strengths and a stimulus of 0.5°. Trajectories are computed from the activity within a sliding window of 250 ms length. Every neuron's position was interpreted as a vector in the 2D plane and the trajectory is the average vector weighted by the neurons activity within the sliding window. The preferred orientations of the neurons are color coded.

maximum frequency focused the trajectory on small spots. In comparison, the red line shows the same trajectories for an inverse Mexican hat driven by a slightly different stimulus. There, the center of activity was stable even for weaker maximum frequencies.

5.5 Discussion I

We demonstrated that recurrent connectivity can yield correlated variability. As we hypothesized, in networks of adaptive exponential integrate and fire neurons, Mexican hat coupling with wider inhibitory than excitatory connectivity spread could lead to spatially inhomogeneous activity in terms of patterns of moving bumps. The inhomogeneous activity was reminiscent of the marginal phase attractor for the simple sinusoidal coupling as discovered by Hansel and Sompolinsky (1998). Because the

Mexican hat coupling amplifies a range of spatial frequencies, we did not observe a single but a multitude of bumps. We further showed that this holds for a wide range of parameter settings in one-dimensional as well as two-dimensional networks.

Due to the moving patterns, we measured distance dependent noise correlations. These correlations were sinusoidally modulated and the amplitude degraded exponentially with the distance between pairs of cells. In one-dimensional networks this dependency of correlated variability can be described by a damped sine wave of the form

$$\langle r_{SC} \rangle(d) = C_{r_{SC}} \exp(-|d|/\lambda_{r_{SC}}) \cos(\omega_{r_{SC}} d). \qquad \text{(5.5 revisited)}$$

We demonstrated that increasing excitatory to excitatory connection strength $\bar{g}_{EE}$ as well as the width of the Mexican hat σ_I reduced the spatial frequency $\omega_{r_{SC}}$. This is in line with the predictions made in the previous chapter that increasing analogous parameters in the mean-field model amplified lower spatial frequencies (c.f. figures 4.8 and 4.9).

For a wide range of parameters the decay $\lambda_{r_{SC}}$ was quite slow with values around 1000 to 4000 pixels (c.f. figure 5.9). Hence, even bump structures quite far apart were influencing each other's movements. Thus, it is also appropriate to characterize the activity as a wave solution that holds across the entire network. Similar observations were made for two-dimensional networks but with a stronger amplitude decay. Thus, bump structures far enough apart appeared, disappeared, and moved independently of each other. Consequently, in a 2D network it is more suitable to speak of individual bumps rather than a wave solution.

Moreover, we showed that noise correlations of similar magnitude could also be observed among inhibitory as well as mixed pairs (c.f. figure 5.8). Hence, inhibition still tried to track excitation as suggested by Renart et al. (2010). Clearly, positive correlations among mixed pairs lead to negative correlations among inhibitory and excitatory currents due to the opposite signs. Yet, inhibition was not strong enough to eradicate the moving bumps and spatial inhomogeneities (c.f. figure 5.12). Accordingly, increasing network size did not decorrelate neural responses. This contrasts with results obtained for densely but uniformly coupled networks by Renart et al. (2010). With growing size of Mexican hat networks, noise correlations increased by reducing the finite size effect and destabilizing homogeneous activity (c.f. figures 4.4 and 5.13).

In contrast to what was predicted by Hansel and Sompolinsky (1998) (c.f. equation 4.77), we did not observe that the width of the bumps shrank with an increase in size of the spiking neuron networks. Either the shrinkage is an artifact of the sinusoidal coupling scheme in the mean-field model with global inhibition, or the change in bump width for a Gaussian topology happens rather slowly. It may only be observed for massive network sizes beyond our simulation capabilities and beyond neuron densities of biologically plausible scales.

We further demonstrated that noise correlations persisted for heterogeneous inputs. The activity patterns and spatial dependency were modulated in comparison to homogeneous inputs. In 2D networks we observed an almost linear decay with distance. In contrast to the homogeneous input, the average r_{SC} did not reach values considerably below 0. Furthermore, the magnitude was usually reduced for stronger stimuli as well as stimuli that matched the inherent frequency of the spatial pattern for homogeneous input. Moreover, spike count correlations as a function of difference in preferred orientations exhibited a similar pattern as the r_{SC} as a function of cell distance. Hence, overall correlations were strongest among close by cells with similar orientation preference. Furthermore, inhomogeneous input basically locked bump activity at certain locations corresponding to the peaks of the afferent input. Noise correlations are a product of small movements or jitter around the peak locations.

5.5.1 Comparison to other Modeling Studies

Similar results to ours were obtained by Keane and Gong (2015) with a spiking neural network model where not the connection probability depended on distance between cells but the coupling strength. In this study the coupling strength among excitatory neurons exhibited a Gaussian profile similar to our connectivity kernel. Keane and Gong (2015) observed activity of moving bumps as well as traveling wave fronts and noise correlations following a damped sine wave as a function of cell pair distance. Trivially, the theory by Rosenbaum and Doiron (2014), as discussed in chapter 4, can be applied to their model without any modifications. For the mean-field model only the expected connectivity strength $J_{XY}\,\kappa_{XY}$ is of importance. Hence, it makes no difference if the connection probability κ_{XY} follows a Gaussian profile and the strength J_{XY} is fixed or vice versa.

Furthermore, Litwin-Kumar and Doiron (2012) developed a spiking neuron network model with clustered connectivity among excitatory neurons and global inhibition. Increasing the clustering, in terms of rewiring random connections to target more neurons of a local cluster, lead to a similar loss of balance as observed in our models. Instead of moving bumps, clusters showed switching between high and low activity states. When Litwin-Kumar and Doiron (2012) replaced the distinct clusters by a ring topology, they observed bumps wandering across the ring, too.

Yger et al. (2011) made an investigation on the influence of Gaussian topology in terms of inhibitory σ_I and excitatory connection width σ_E on correlated variability. Surprisingly, they only observed the emergence of bump patterns for very narrow excitatory spreads ($\sigma_E \ll \sigma_I$). Yet, they kept recurrent weights fixed and used a rather low coupling strength. Therefore, it is likely that their network size was too small for the given recurrent weights and the finite size effect kept network activity stable.

Furthermore, a ring network with parametrization beyond a bifurcation to multistability can be a suitable model for phenomena observed in the MT area in experiments with awake monkeys. Ponce-Alvarez et al. (2013) measured noise correlations

in direction sensitive MT neurons via repeated presentations of moving gratings. They reported a modulation of correlations with movement direction and difference to the preferred directions of the recorded neurons. Moreover, the authors studied a ring network model consisting of non-linear rate units. The rate units were connected via sinusoidal coupling defined over the units' preferred movement directions. For particular parameter regimes Ponce-Alvarez et al. (2013) observed bump patterns similar to the marginal phase discussed in the previous chapter. In fact, they demonstrated that a network in such a state can explain the experimentally observed directional tuning of noise correlations in MT neurons well.

Moreover, bump attractor networks are prominent models of working memory (Laing et al., 2003; Wimmer et al., 2014). For instance, Kilpatrick (2013) studied neural fields with sustained bump activity. Instead of viewing bumps as a source of correlated variability, they investigated their behavior under uncorrelated and correlated noisy inputs. The authors demonstrated that connecting several areas with bump attractors can stabilize the bumps' locations and hinder their wandering. If variability across areas was correlated, however, the activity profiles lost stability and the authors observed a diffusion of bumps.

Other network structures that are capable of pattern formation are reaction diffusion systems (Vanag and Epstein, 2007). Accordingly, local reactors are combined with spatial signal or compound diffusion. One prominent local reactor is the FitzHugh-Nagumo model (FitzHugh, 1961; Nagumo et al., 1962). It consists of two coupled differential equations and exhibits dynamics akin to spiking. Reaction diffusion systems based on FitzHugh-Nagumo models can produce complex bump and wave patterns similar to what we observed (Hagberg and Meron, 1994; Postnov et al., 2012; Dahlem and Isele, 2013). However, Laing and Longtin (2001) demonstrated that diffusive coupling lacks one characteristic property found in our simulations. The theory by Hansel and Sompolinsky (1998) predicts that given a deterministic rate model with adaptation, a bump pattern moves with a constant velocity into a particular direction determined by the initial conditions. Yet, because our systems were driven by noise in terms of Poisson input, we observed bump structures with consistent, stable shapes whose course of movement could change. The bumps could move into opposite or similar directions eventually fusing into each other (c.f. figure 5.5). According to Laing and Longtin (2001) this is characteristic for Mexican hat coupling. They observed that movement modulation is even more pronounced for growing noise intensities. On the contrary, using diffusively coupled FitzHugh-Nagumo models, Laing and Longtin (2001) reported that the noise intensity did not affect the bump velocity and direction up to a certain critical value. If the intensity reached this critical value, the noise had a catastrophic influence and bump propagation broke down entirely. In this case the authors observed scattered, appearing, and disappearing bumps of different sizes. Yet, Postnov et al. (2009) demonstrated that with a more complex diffusive coupling — involving an additional, third dynamical variable — moving bump patterns appear

under certain conditions. For strong coupling in combination with high noise levels, deformed clusters of activity of variable sizes moved within the two spatial dimensions of their reaction diffusion system.

Of course, FitzHugh-Nagumo models are not necessarily diffusively coupled, but they can be connected using a Mexican hat topology too. For example, Bachmair and Schöll (2014) demonstrated that a FitzHugh-Nagumo model in combination with Mexican hat connectivity can undergo a Turing bifurcation and their system produced wandering bump solutions.

Mexican hat profiles defined over some cortical or neural space are not the only recurrent coupling schemes that can yield correlated variability. Along these lines, the recurrent model by Hansen et al. (2012) has no notion of cortical spatial scales and the authors used a Mexican hat topology over orientation space. Therefore, they argued that their system models horizontal long range connections spanning several columns in visual cortex. Due to the Mexican hat topology over orientations, excitatory long range connections were more orientation specific, i.e. it is more likely that excitatory cells with similar preferred orientations are connected. Likewise, inhibitory long range connections were rather unspecific. However, in comparison to excitatory axons, long range connections originating from inhibitory neurons are rarely observed in visual cortex. Stepanyants et al. (2009) showed in an experimental tracer study that only about 25% of excitatory axons terminated within a distance of 500 µm from the pre-synaptic cell, but almost 90% of inhibitory connections did.

However, excitatory long range connection alone can be sufficient for pattern formation and noise correlations. For instance, using excitatory patchy long range connections but only local inhibitory connections, Voges and Perrinet (2012) demonstrated the emergence of spatially inhomogeneous activity patterns for a wide range of parameter settings. Similarly, we obtained preliminary results which show that orientation specific excitatory long range connections can amplify spatial frequencies. Moreover, spatial patterns emerged in simulations of spiking neuron networks with such long range profiles. As a consequence, we observed correlated variability comparable to a local Mexican hat topology. This preliminary data is briefly discussed in appendix B. Hence, Mexican hat networks are of course not the only coupling scheme that can cause noise correlations. Any topology where the 0th spatial Fourier mode is stable, but some higher frequencies are amplified, is a potential candidate that may lead to pattern formation and, in turn, to noise correlations.

5.5.2 Biological Plausibility of the Model

Our model simulations aim at biological realism rather than at developing the simplest spiking neuron model validating the theoretical findings of the mean-field approach discussed in chapter 4. Yet, our networks still constitute a considerable simplification of biology. First of all, using point neuron models is without doubt a substantial loss of biological realism. Real neurons can exhibit highly non-linear input transformations

due to the arborisation of their dendritic tree. These nonlinearities are much better captured by compartmental models (Poirazi, 2009). However, the model complexity and costs associated with their simulation make networks of compartmental models infeasible. To perform large parameter explorations with multiple network realizations, point-neuron models are a good alternative. Moreover, the choice of the adaptive exponential integrate and fire neuron (AEIF) is a compromise between computational efficiency and a more biologically motivated description of intracellular currents as in Hodgkin-Huxley (HH) neurons (Hodgkin and Huxley, 1952). In fact, Brette and Gerstner (2005) demonstrated that AEIF neurons predict spike timing of HH neurons well. Likewise, Jolivet et al. (2008) showed that AEIF neurons are a suitable model of real pyramidal neurons under noisy current injections.

The input statistics of our model are a major drawback. Our assumption that every neuron received afferent input in terms of independent Poisson spike trains is far from biologically realistic LGN input. For instance, Lin et al. (2012) demonstrated that more realistic LGN models can change response characteristics such as tuning sharpening of post-synaptic V1 neurons. Moreover, common or shared input noise can yield oscillations in neural networks (Doiron et al., 2004; Lindner et al., 2005). However, we chose the simple input statistics for a reason. We wanted to exclude any source of correlated variability except recurrent connectivity. Even so, it is unlikely that more complex inputs change our results considerably apart from adding another potential source of correlated variability. For instance, Hansen et al. (2012) used a thorough LGN model where V1 neurons even shared common afferent input and, like us, observed noise correlations for Mexican hat topologies. Still, afferent input may yield correlations for balanced or inverse Mexican hat profiles, too. We discussed that in order for the balanced fixpoint of the model by Rosenbaum and Doiron (2014) to exist, afferent input must be more broadly tuned than the recurrent coupling (c.f. equation 4.20). Thus, too sharply tuned input or highly convergent afferent connections may render the fixpoint invalid and, thereby, may yield spatial inhomogeneities and noise correlations.

Furthermore, we assumed homogeneous populations of neurons. For example, all cells received the same number of inputs. However, theoretical studies have shown that heterogeneity in the number of connections can have a substantial effect on a network's stability and success of pattern formation and propagation (Sonnenschein et al., 2013; Kouvaris et al., 2014). Moreover, in our model all cells exhibited equal parameter values like time constants or spiking thresholds. The only distinction was made between excitatory and inhibitory neurons that differed in leak conductance and refractory periods. Even input peak conductances were equal among all connections originating from and terminating in equal populations. In contrast, in biology one can observe a huge variation among parameters like membrane and adaptation time constants (Sanchez-Vives et al., 2000) or amplitudes of spike triggered post-synaptic potentials (Mason et al., 1991). A modeling study by Mejias and Longtin (2012) suggests that such heterogeneities can even enhance information processing in neural networks.

Nonetheless, we expect that increased heterogeneity among neurons will not change our results by much. One of the key assumptions by Rosenbaum and Doiron (2014) for their mean-field approach is that neurons are operating in a regime with strong mean input. In this regime fluctuations in input currents can be ignored. In fact, in our simulations noise correlations were largest in conditions with strong mean input, i.e. for strong excitatory recurrent weights and strong untuned afferent inputs (c.f. figure 5.29). Analogously, we assume that if heterogeneities are small, their impact may be weak for large networks and in the light of strong inputs the effects average out. For instance, Chelaru and Dragoi (2008) introduced heterogeneity into a relatively small network model with less than 2000 neurons by drawing recurrent connection strengths from Gaussian distributions. They showed that correlations were reduced, but after all were still present for considerable heterogeneity. Even if they used a wide distribution with a variance of 35% of the mean connection strength, they observed correlated variability.

Furthermore, in our 2D networks with 100×100 and 200×200 excitatory neurons corresponding to $1.5\,\text{mm} \times 1.5\,\text{mm}$ or $3\,\text{mm} \times 3\,\text{mm}$ of cortex area, respectively, we obtain a density of about 5555 neurons per square millimeter (including inhibitory neurons). This is an undersampling of neurons in a particular layer in cat primary visual cortex with densities of up to 15,000 neurons (Beaulieu and Colonnier, 1983). Yet, the theory by Rosenbaum and Doiron (2014) states that increasing the network size even fosters inhomogeneous activity and stable Mexican hat networks may lose balance due to a vanishing finite size effect. Of course, this requires particular scaling assumptions larger than $O(1/N)$ for the recurrent weights. However, we already chose biologically plausible conductances. So we can assume $O(1)$ for increasing N. Accordingly, we could demonstrate that larger neuron densities yielded correlated variability for even more parameter settings (c.f. figure 5.13).

Many modeling studies assume recurrent coupling in form of Mexican hats in order to achieve tuning sharpening or pattern formation (Ernst et al., 2001; Kang et al., 2003; Bressloff, 2012). Moreover, theoretical work by Carreira-Perpiñán and Goodhill (2004) showed that the structure and distribution of orientation preference and ocular dominance maps in primary visual cortex support the hypothesis of Mexican hat coupling. Yet, it is unclear if this coupling scheme is really implemented in the brain. Experiments by Levy and Reyes (2012) revealed that the auditory cortex of mice relies on inverse Mexican hat coupling. Mariño et al. (2005) used retrograde tracers to visualize connections among neurons in cat visual cortex and found a balanced spread of excitatory and inhibitory connections. Data by Stepanyants et al. (2008) suggest that at least in the fourth layer of cat visual cortex, inhibitory connections are less broad than excitatory ones. However, Buzas et al. (2001) found inhibitory cells in layer 3 projecting further than in layer 4. This is in line with work by Hansen et al. (2012) as well as Smith et al. (2013) who measured noise correlations only in layers above and below the fourth layer.

Finally, besides a peak in low frequencies due to the bump movement, the temporal power spectrum of the network activity revealed a second peak in the gamma frequency band (c.f. inset of figure 5.6). This is in agreement with many studies that report gamma rhythms in sensory areas of various species (Liu and Newsome, 2006; Jia and Kohn, 2011; Xing et al., 2012).

5.5.3 Biological Relevance of Results

We demonstrated that noise correlation can be caused by moving patterns in forms of bumps. On a single cell level this yielded alternating phases of spiking activity and silence. In the former case neurons are participating in an active region of a bump and, in the latter, they are part of an inactive region. Similarly, Mochol et al. (2015) showed experimentally that noise correlations are dominated by phases of coinactivation in the auditory cortex of anesthetized rats. During spontaneous activity, the authors observed periods where all neurons in a local network jointly stopped firing. Reduction of these silent periods by external stimulation with click sounds decreased the magnitude of the correlations.

Likewise, our simulation showed that increasing the strength of an orientation tuned stimulus lead to a decrease in correlated variability in a 2D network. Similar results were obtained in the one-dimensional model using a sinusoidally tuned stimulus with a frequency close to the inherent bump frequency. We can relate these paradigms to experimental settings with different stimulus contrasts or different levels of attention. Kohn and Smith (2005) reported weaker correlations for strong contrasts of drifting orientation gratings in experiments with Macaque monkeys. Studies by Cohen and Maunsell (2009) and Mitchell et al. (2009) showed that visual attention can reduce correlated variability among neuron pairs in visual cortex.

Furthermore, in our simulations noise correlations depended on the size of the integration time window. We observed an amplitude saturation for windows larger than 1 s. Similar values were reported by Reich (2001) in experiments with monkeys. Others reported slightly faster saturation like Bair et al. (2001) or Smith and Kohn (2008) (c.f. figure 3.3D). In these studies maximum correlations were already observed for window sizes slightly below 1 second.

Additionally, using heterogeneous stimuli we further demonstrated that besides a modulation with distance, correlations weakened with an increase in difference in preferred orientation among cell pairs (c.f. figure 5.27). Similar results were reported by Hansen et al. (2012) experimentally for the primary visual cortex of anesthetized macaque monkeys. However, they did not measure a dependency on distance. Yet, they used a laminar electrode. Such electrodes penetrate the layers in the visual cortex rather orthogonally. This leads to measuring distances across but not within layers. In contrast, our two-dimensional networks correspond to a single layer and distances are defined within the neural sheet.

Smith and Kohn (2008) measured correlations in visual cortex of monkeys alongside the neural tissue using a micro-electrode array (MEA). They reported a distance dependency of noise correlations. However, the spatial scale is very different from what is observed in our model (c.f. figure 3.3A). Given orientation gratings, the authors reported a linear decay with distance spanning over several millimeters. In our model we did measure a seemingly linear decay for tuned input as well, but correlations vanished after a few ten pixels which corresponds to ca. 400 µm in our scaling to cat visual cortex. Still, Solomon et al. (2014) reported a fast decay of correlations with distance between cell pairs in the marmoset monkey's middle temporal visual array (MT) comparable to our simulation results (c.f. figure 3.3B). Moreover, using a blank grey screen stimulus, they observed a sinusoidal modulation of correlations. Admittedly, this modulation is not significant. An explanation might be the rather coarse spatial resolution of MEAs. They used a 10×10 an electrode array with a spacing of 0.4 mm in between electrodes. Fortunately, a much finer resolution can be achieved with two-photon imaging. For example, Ch'ng and Reid (2010) observed a significant sinusoidal modulation of spontaneous activity correlations with distance in the visual cortex of rats. They did not report such a modulation for data obtained from cat, though. In cat Ch'ng and Reid (2010) only identified a seemingly exponential decay. However, their technique is limited in spatial scale. They could measure noise correlations for a maximum pair distance of only 400 µm. Our model simulations imply that only beyond this distance one should expect an increase in correlations due to a sinusoidal modulation (c.f. figure 5.20A). Therefore, Ch'ng and Reid (2010) might have measured only the first quarter of a cycle of the sinusoid. Thus, our model suggests that future experimental work may aim at measuring noise correlations over a large scale of more than 1 mm with a high spatial resolution.

In fact, the emergence of spatially inhomogeneous patterns in spontaneous activity on several hundred micrometer scales has been long known (Arieli et al., 1995). For instance, Kenet et al. (2003) reported bump shaped spontaneous activity in the visual cortex of anesthetized cat using voltage sensitive dyes. More important, the spatial scales of the patterns matched the preferred orientation map. This is also the case in our simulations. In our 2D network model we measured $\omega_{r_{SC}} \approx 2.2$ cycles per 100 neurons, which agrees with the frequency of 4 pinwheels per 100×100 cells. Similar observations with spontaneous patterns resembling evoked activity have been made in experiments with rats (Han et al., 2008) and monkeys (Muller et al., 2014).

Alternative hypotheses for correlated variability were already discussed in chapter 3. For instance, Ecker et al. (2010) argued that confounds from the experimental setup like anesthesia or badly implanted electrodes can cause large values of r_{SC}. Schulz et al. (2015) pointed out the substantial influence of spike sorting on measured noise correlations. Therefore, we do not propose that Mexican hat topology is the primary cause of correlated variability in cortex. We argue that it can be one among many others. This may explain why on average the spike count correlation is positive over

a span of several millimeters in biological experiments (c.f figure 3.3B and Solomon et al., 2014), but we report a clear sinusoidal modulation with a total average close to zero across all distances. The bump activity correlations could additively lie on top of other shared variability. For instance, Ecker et al. (2014) reported average strong positive noise correlations in primary visual cortex of anesthetized macaque monkey even for cells more than 3 millimeters apart. During wakefulness, however, correlations were by an order of magnitude weaker and some non-significant sinusoidal modulation was observed.

Moreover, the spike count correlation coefficient r_{SC} is conceptually easy to grasp, but also a very coarse statistic of neural activity. It can help to identify potentially interesting phenomena in cortex, but it is difficult to discern underlying principles just by measuring correlation coefficients. As pointed out before, future experimental work may focus on high-resolution recordings of joint-activity to shed more light on the origins of correlated variability.

Next, we are going to take a closer look at the consequences of correlated variability for information processing. To be precise, we will investigate the effect of noise correlations on encoding quality in terms of Fisher information.

5.6 Information Processing

So far we discussed recurrent connectivity as a potential source of noise correlations. We analyzed how distance dependent correlations can emerge for homogeneous as well as heterogeneous inputs. In this section we are going to investigate how correlations influence stimulus processing. We will discuss the effects of correlations on encoding quality in terms of the three information measures introduced earlier. These are Fisher, shuffled, and diagonal information.

Fisher information I_{LOLE} was estimated using the approach by Seriès et al. (2004) as introduced in chapter 3. As a reminder, the locally optimal linear estimator (LOLE) is based on the following linear regression (c.f. equation 3.19):

$$\hat{s} = \boldsymbol{w}\,\boldsymbol{r} + w_0. \qquad \text{(3.19 revisited)}$$

Fisher information is calculated using two rather similar stimuli s_1 and s_2 via

$$I_{\text{LOLE}} = \frac{\left(\frac{\langle \hat{s}_2 \rangle - \langle \hat{s}_1 \rangle}{s_2 - s_1}\right)^2}{\frac{1}{2}(\text{Var}(\hat{s}_2) + \text{Var}(\hat{s}_1))}. \qquad \text{(3.20 revisited)}$$

60% of the simulation data were used to train the LOLE to predict the stimulus orientation $\hat{s}$ from the spiking activity. Moreover, 20% of the data were taken as a validation and another 20% as a test set. The parameters $\boldsymbol{w}$ and w_0 were optimized via stochastic

gradient descent using the scikit-learn Python library (Pedregosa et al., 2011). To prevent overfitting training was stopped if validation performance deteriorated for 500 consecutive iterations. The LOLE parameters with the best validation performance were chosen for testing. Performance was evaluated in terms of the mean squared error of the estimated and real stimuli:

$$\langle(\hat{s}_i - s_i)^2\rangle, \tag{5.8}$$

with $i \in \{1, 2\}$. I_{shuff} was computed analogously by shuffling trial data and I_{diag} by training on the shuffled but testing on the original data.

Estimating these quantities for a full network requires a lot of data and a lot of repeated trials. To reduce simulation time considerably we based our calculations on random subsamples of neurons. Cells were sampled evenly among preferred orientations. All values were averaged over 10 sampling realizations. The information experiments were based on repeated trials of orientation stimuli presentation of $s_1 = -1^\circ$ and $s_2 = 1^\circ$. After an initial phase of 1 second, 3 seconds of stimulus presentation were taken into account to calculate Fisher information. If not stated otherwise, the number of trials was 500 per orientation stimulus. Hence, we estimated the information measures from 500 neurons. For all analyses we only considered excitatory neurons because GABAergic inhibitory interneurons are not known to project out of primary visual cortex (Seriès et al., 2004; Schmolesky, 2007).

5.6.1 Fisher Information and Tuning

We compared Fisher information between realizations of a two-dimensional Mexican hat topology, a balanced configuration, and an inverse Mexican hat profile. The Mexican hat profile exhibited strong sinusoidally modulated distance dependent correlations in the homogeneous input case. In addition, we measured positive, roughly linearly decaying noise correlations if stimulated with an orientation tuned input. The balanced profile, where excitatory and inhibitory spread ranges equally wide, and the inverse Mexican hat profile showed average correlations close to 0 regardless of the stimulus type and distance between cell pairs.

Figure 5.32 displays the information measures for all three network configurations for stimuli with a maximum frequency of $\nu_{\text{Aff, max}} = 15\,\text{Hz}$. As expected from the literature, average weak positive correlations — as measured for the Mexican hat network for nearby cells — decreased the stimulus encoding quality in comparison to the shuffled data (left blue and cyan bars in figure 5.32). Furthermore, the diagonal information I_{diag} (left magenta bar) is significantly smaller than I_{LOLE} (left cyan bar). Thus, the correlations themselves carried information. A decoder cannot safely ignore correlated variability without facing a penalty in performance. Interestingly, the total information that could be recovered from the Mexican hat configuration was significantly larger

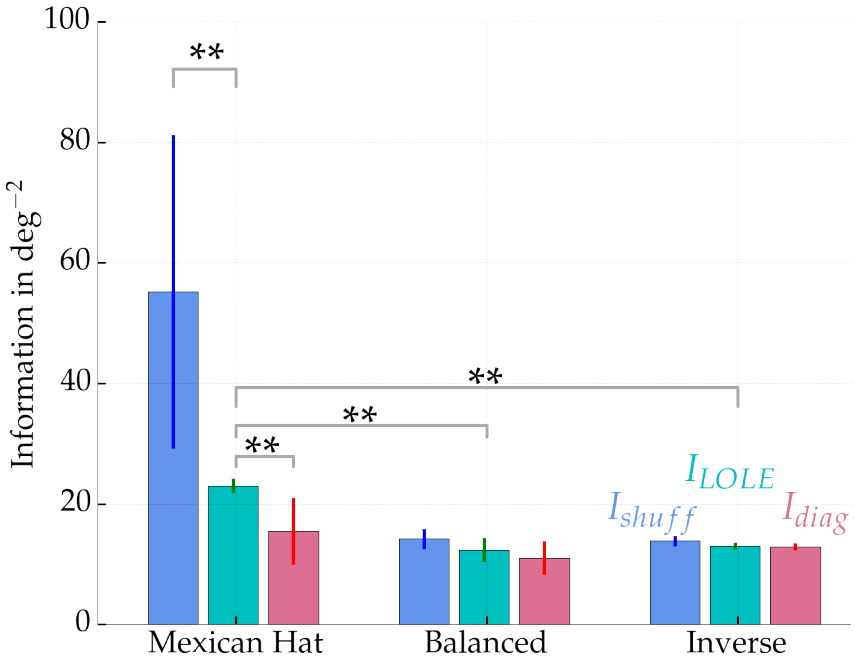

Figure 5.32: Fisher information measures I_{shuff}, I_{LOLE}, and I_{diag} of three network configurations with operating points close to criticality based on a sample of 500 neurons per network. Values are averaged over 10 network realizations, error bars show standard deviations. A maximum firing rate of $\nu_{\text{Aff, max}} = 15\,\text{Hz}$ is used. Parameters are for Mexican hat $\sigma_E = 10\,\text{px} < \sigma_I = 15\,\text{px}$, balanced $\sigma_E = 10\,\text{px} = \sigma_I = 10\,\text{px}$ and inverse Mexican hat $\sigma_E = 15\,\text{px} > \sigma_I = 10\,\text{px}$. Strengths are chosen equally for all networks with $\bar{g}_{IE} = 0.6\,\text{nS}$ and $\bar{g}_{EE} = 0.4\,\text{nS}$. Information is estimated using an approach by Seriès et al. (2004). To estimate information 500 trials each of a -1 and 1 degree orientation stimulus are used. Blue bars show information of shuffled data I_{shuff}, the information extracted from the real data I_{LOLE} is given in cyan, and light red displays the diagonal Fisher information I_{diag}. As expected for the Mexican hat configuration with positive average noise correlations, information is significantly overestimated if correlations are removed (left blue bar vs. left cyan bar). The correlations themselves carry a small but significant amount of information (left cyan bar vs. light red bar). Surprisingly, the amount of information I_{LOLE} (cyan) that can be read out from a Mexican hat network is significantly larger than for the other two network configurations (comparison of all cyan bars). Significance values are based on a non-parametric Wilcoxon rank-sum test for comparisons of I_{LOLE} between different topologies ($p < 0.001$, 10 networks samples) and Wilcoxon signed-rank test ($p < 0.001$, 10 network samples) for comparisons within the Mexican hat topology.

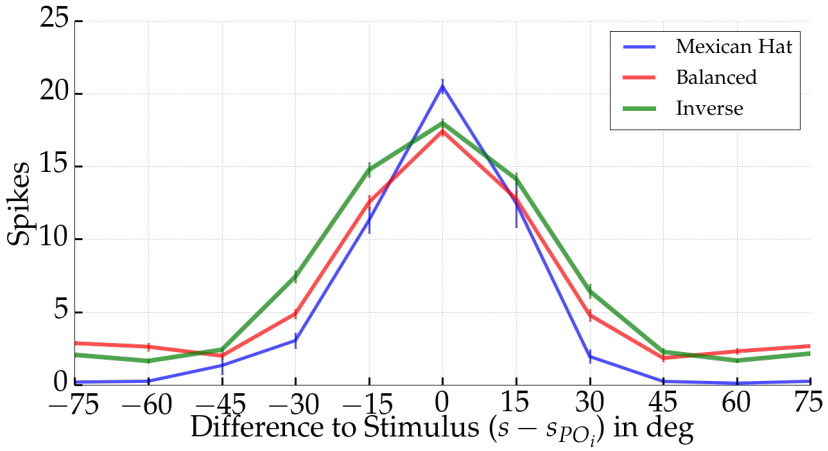

Figure 5.33: Tuning curves of a Mexican hat, balanced, and inverse network close to criticality. A range of stimuli with 10 trials and 3 seconds per stimulus were presented to the networks and responses are averaged across trials and all neurons with similar preferred orientations in the range of $[13°, 16°]$. Error bars mark standard deviations across 10 network realizations. Network parameters as in figure 5.32. One can observe that the tuning of the Mexican hat networks is sharper than among the other two topologies.

than in the other two networks (I_{LOLE}, cyan bars) despite the presence of noise correlations.

Tuning sharpening is a potential explanation for this phenomenon. In theory, as shown by Dayan and Abbott (2005), sharper tuning can enhance stimulus encoding. Thus, despite the presence of correlations, a Mexican hat profile may better encode the stimulus by sharpening of responses. Individual neurons fire more selective to orientation stimuli and thereby make discrimination between stimuli easier. This is indeed what we observed when we looked at the networks' tuning curves. In order to determine the tuning curves we analyzed the response of neurons with equal or very similar preferred orientations to a range of stimuli. As seen in figure 5.33, the Mexican hat networks exhibited a sharper tuning than the other two networks. The average response of neurons with similar preferred orientation showed a narrower and larger peak around the preferred orientation. Whether the balanced architecture gives a sharper response than the inverse or vice versa is hard to discern from the figure. On the one hand, the balanced networks' curve appears more narrow than the tuning function of the inverse hat. On the other hand, the ratio between peak and baseline response is larger for the inverse topology.

In order to quantify the sharpening of the tuning we applied a measure called *Orientation Selectivity Index* (OSI) (Swindale, 1998):

$$\text{OSI} = \sqrt{\left(\sum_{s_i} \langle r(s_i)\rangle \cos(2s_i)\right)^2 + \left(\sum_{s_i} \langle r(s_K)\rangle \sin(2s_i)\right)^2} \Big/ \sum_{s_i} \langle r(s_i)\rangle, \tag{5.9}$$

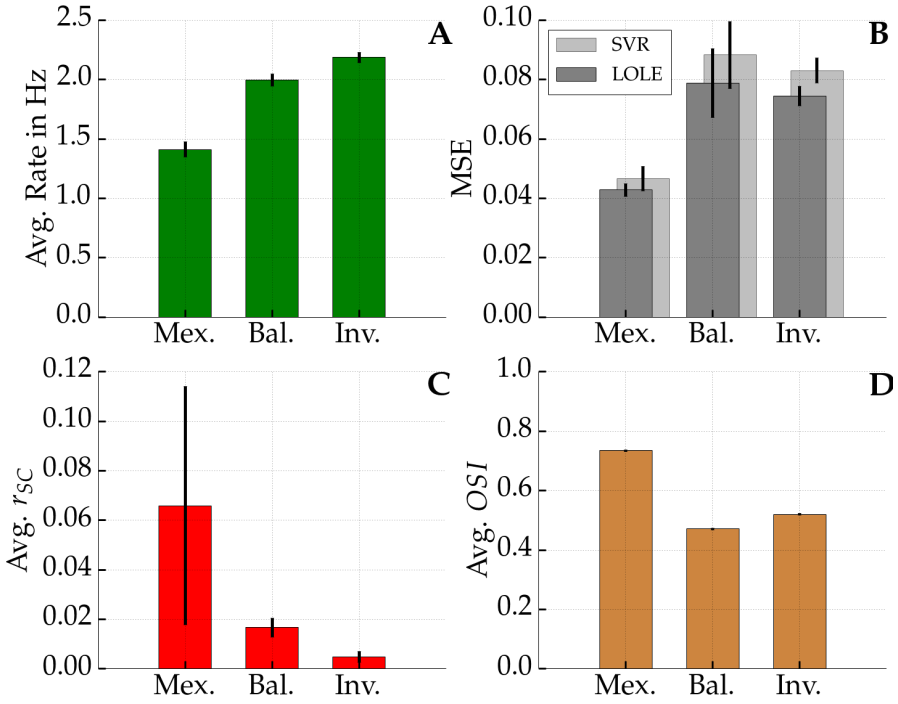

Figure 5.34: Top left (A): Average network firing rate averaged over all excitatory neurons. Black bars show the standard deviation across 10 network realizations per profile. From Mexican hat (left) to the inverse case (right) firing rate increases. Top right (B): In dark grey the average mean squared error (MSE) of the linear decoder is shown whereas in light grey one sees the average MSE of the best Support Vector Regression (SVR). Bottom left (C): Average noise correlation among pairs of neurons at most 13.3 px apart for a 3 second time window. In the Mexican hat networks one observes the strongest correlations. Bottom right (D): Average Orientation Selectivity Index (OSI) among all neurons in the networks. The sharpest tuning is observed for Mexican hat networks. Standard deviations are so low that the black bars are not visible. Network parameters as in figure 5.32.

where $\langle r(s_i) \rangle$ is the average spike response of a neuron for a particular stimulus s_i. We chose twelve evenly spaced orientations, $s_i \in \{-90°-75°, -60°, .., 75°\}$. $\langle r(s_i) \rangle$ was averaged over 10 trials per orientation. The OSI is a measure of tuning sharpness that ranges from 0, not selective, to 1.0, perfectly selective to orientations. We computed the average OSI across the whole excitatory neuron population in each network.

Figure 5.34D shows the average OSI of the different networks. The Mexican hat networks exhibited the sharpest tuning. The value of about 0.734 (± 0.003) is larger compared to the others. In addition, the inverse configuration showed the second sharpest tuning with 0.520 (± 0.003) followed by the balanced network with 0.471 (± 0.003). All these differences are significant (Wilcoxon rank-sum test, all $p < 0.01$, 10 sample networks). Notably, the coding was not simply improved due to the availability of more spikes. In figure 5.34A one sees that the average network firing rate was even less for the Mexican hat networks. Sharper tuning means less neurons in total are responding

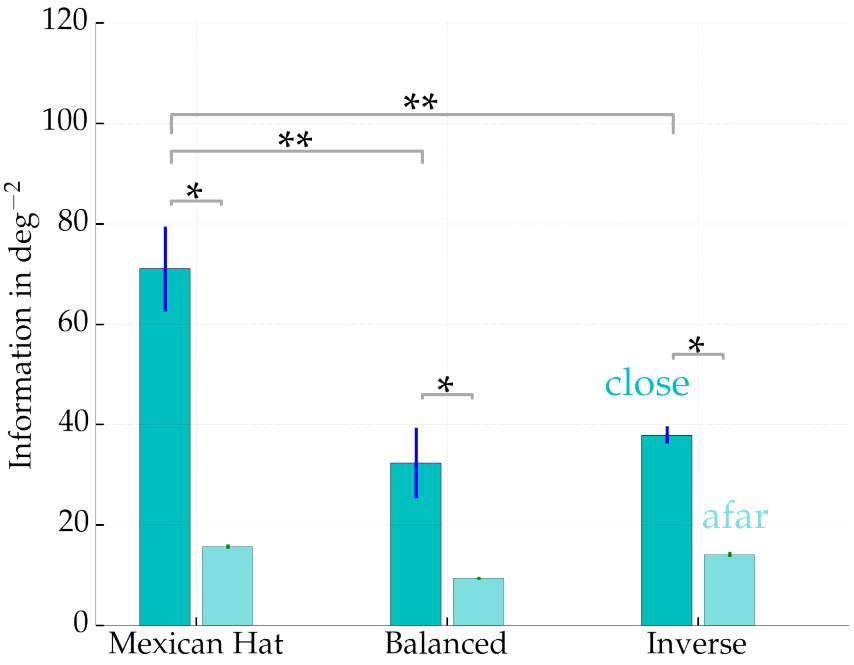

Figure 5.35: Fisher information I_{LOLE} of three network configurations ($\sigma_E = 10$ or $15\,\mathrm{px}$, $\sigma_I = 10$ or $15\,\mathrm{px}$) with operating points close to ($\bar{g}_{EE} = 0.4\,\mathrm{nS}$) and afar ($\bar{g}_{EE} = 0.2\,\mathrm{nS}$) from criticality based on a sample of 500 neurons per network. Values are averaged over 5 network realizations, error bars show standard deviations. A maximum firing rate of $\nu_{\mathrm{Aff,\,max}} = 30\,\mathrm{Hz}$ is used. Other parameters as in figure 5.32. Mexican hat networks show significantly better encoding quality than the other two; Wilcoxon rank-sum test, $p < 0.01$, 5 network samples. For all networks encoding performance is significantly better close to criticality than afar; Wilcoxon signed-rank test, $p < 0.05$, 5 network samples.

to an input stimulus. This in turn decreases the average network firing rate. In summary, despite stronger noise correlations for Mexican hat networks (figure 5.34C), the coding quality improved alongside a sparser spiking code.

We further tested if performance could be improved over LOLE by using a non-linear support vector regression (SVR) with radial basis function (RBF) kernels. We rescaled the training data to zero mean and variance of one. Furthermore, we optimized two SVR hyper-parameters with a grid search. We explored the SVR's error penalty function parameter ($C_{SVR} \in \{0.01, 0.1, 1, 10\}$) as well as the RBF kernel width coefficient ($\gamma_{SVR} \in \{0.0001, 0.001, 0.01, 0.1\}$) using 5-fold cross-validation on the training data before benchmarking with the test data. However, we could not obtain an improvement in performance. On average the SVR's mean squared error (MSE) was slightly worse than the error of the linear decoder, as shown in figure 5.34B. This suggests that the linear decoder is close to optimality and it is unlikely that there is information that could only be obtained by using non-linear methods. Hence, the lower bound on Fisher information given by I_{LOLE} is much likely very tight and gives a good estimate of the real information in the system.

Moreover, we also tested if the phenomenon of better encoding performance by Mexican hat networks still holds for a stronger maximum input frequency $\nu_{\text{Aff, max}} = 30\,\text{Hz}$. Indeed, just increasing input strength did not change the effect, as seen in figure 5.35 (darker cyan bars). Not surprisingly, for stronger inputs where correlations were reduced (c.f. figure 5.28A), Mexican hat networks still showed the best stimulus encoding performance.

An interesting observation was made regarding the operating regime of all three networks. Decreasing the recurrent excitatory to excitatory coupling strength (light cyan bars in figure 5.35), i.e. moving the operating regime away from criticality, was highly detrimental to information processing in all networks. Networks operating afar from criticality are dominated by recurrent inhibition which reduces the average spike rate considerably (c.f. figure 5.23A and B). Accordingly, much less information could be recovered from the weaker firing activity.

5.6.2 Fisher Information and Network Topology

What is the relation between different Mexican hat topologies and the increased information? To answer this question, we explored different widths of the inhibitory connections (σ_I) and increased the spread of the Mexican hat.

Although increasing connection spread also increased noise correlations (red dotted line in figure 5.36B), stimuli could be better reconstructed up to a certain width, see all information measures in figure 5.36A. Increasing the inhibitory spread also sharpened the network tuning (dotted orange line). Average tuning of excitatory neurons grew beyond 0.9 and reached a plateau for wide inhibitory spreads. As before, information was not increased due to an increase in spike rate. Instead the number of spikes dropped constantly with the increase of inhibitory connection spread. It decreased from about 60,000 spikes per stimulus trial (i.e. average firing rate of 2 Hz) for the tight inhibitory spread to about half of the number of spikes for the widest inhibitory spreads. Hence, for intermediate values of σ_I not only a gain in the total information was obtained, but also the information per spike grew.

Yet, increasing inhibitory spread has its limitations. Information drastically decreased for too wide spreads although tuning is still very sharp. This happened for scales of $\sigma_I \approx 22.5\,\text{px}$. Accordingly, for such wide connection spreads inhibitory connections from one pinwheel leap into the others. Moreover, noise correlations increased rigorously. Additionally, the decoder performance dropped quickly and the mean squared error of the estimated stimulus rose considerably (black dotted line in figure 5.36).

Furthermore, we tested whether the performance gain is really a product of the Mexican hat topology or rather an effect of the width of the inhibitory connection spread only. Therefore, we explored different widths of both connection spreads (σ_I, σ_E) and kept recurrent strengths fixed. To reduce simulation time we ran 125 trials

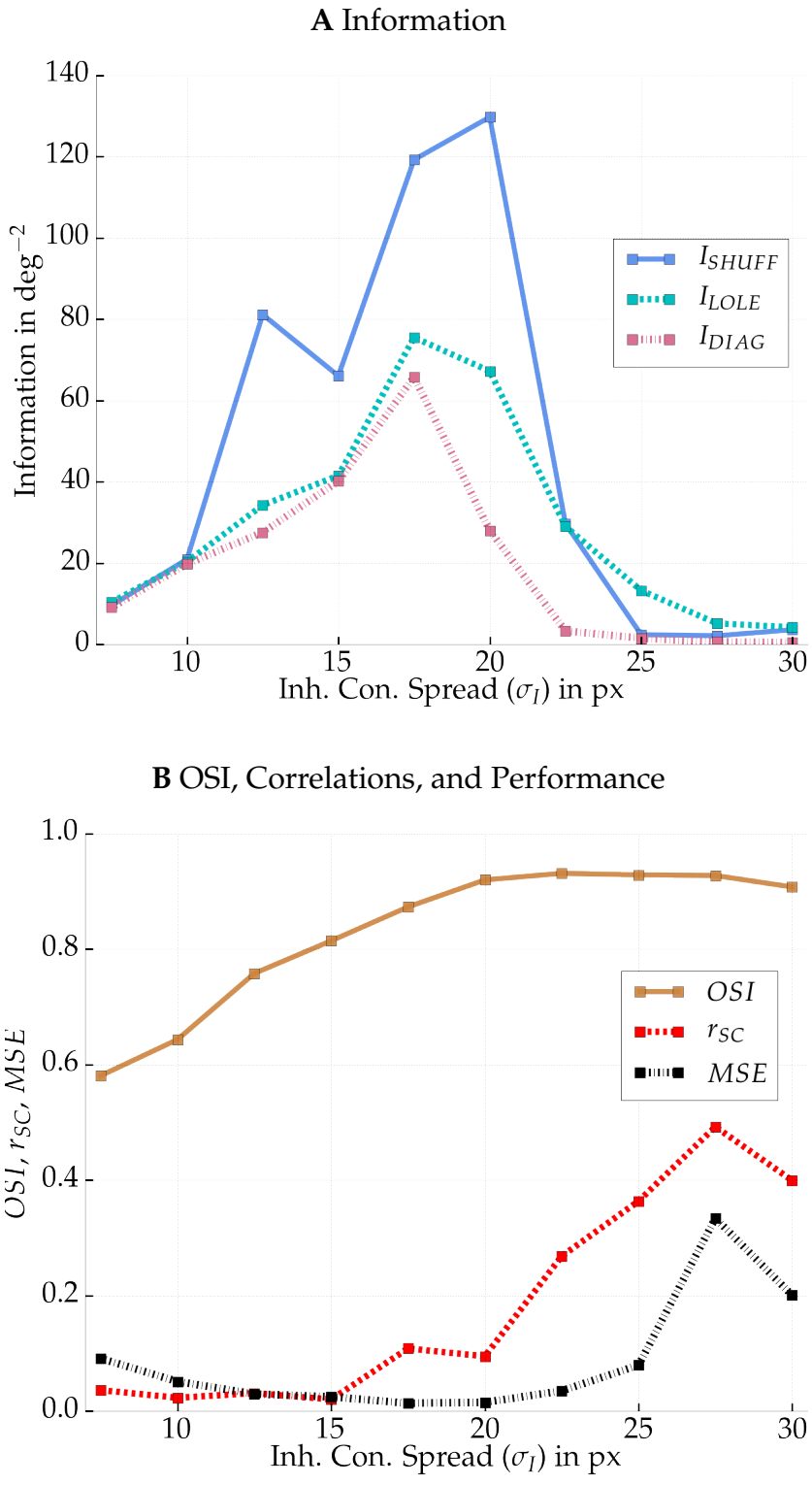

Figure 5.36: Top: Fisher information measures I_{shuff}, I_{LOLE}, and I_{diag} extracted from 500 sampled neurons and a maximum input frequency of 30 Hz. Bottom: Orientation selectivity index (OSI) and average noise correlations r_{SC} ($d \leq 13.3\,\text{px}$, $T = 3\,\text{s}$), and mean squared error (MSE) as a function of inhibitory spread close to criticality (for each spread a single network realization, $\sigma_E = 10\,\text{px}$, $\bar{g}_{IE} = 0.6\,\text{nS}$, $\bar{g}_{EE} = 0.4\,\text{nS}$). In general, average r_{SC} increases with span of the Mexican hat. Similarly, the OSI shows a trend to increase with inhibitory spread and reaches a plateau for large spreads. There is a sweet-spot for the information that can be extracted from 500 neurons at about $\sigma_I = 17$ to $20\,\text{px}$ after which information decreases drastically. Likewise as information decreases, the MSE of the LOLE decoder rises.

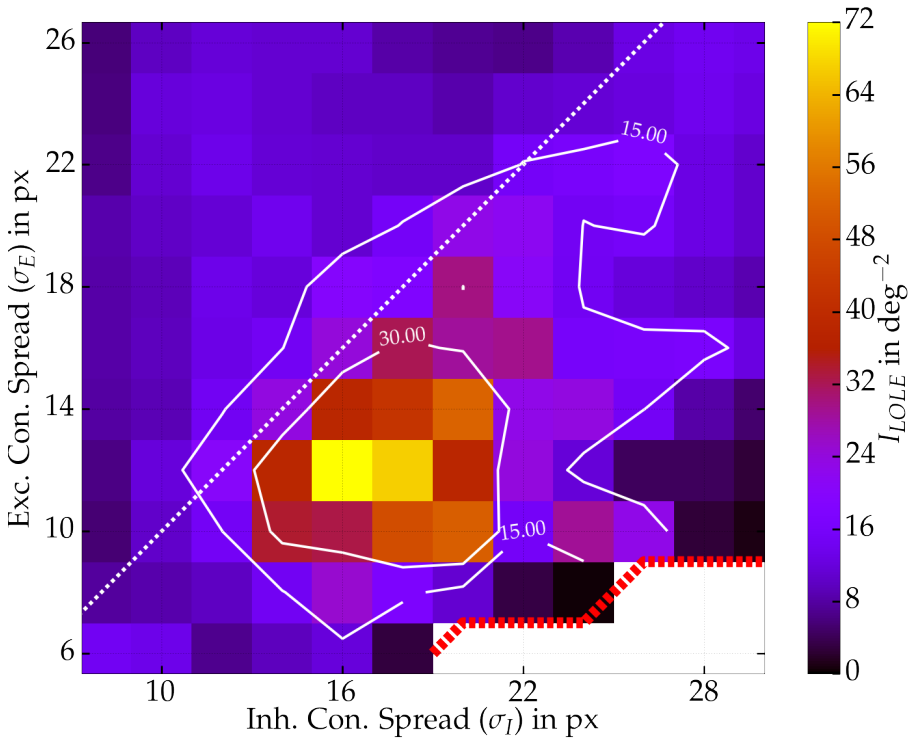

Figure 5.37: Fisher information I_{LOLE} as a function of both connection widths σ_I and σ_E. All network topologies are close to criticality ($\bar{g}_{EE} = 0.4\,\text{nS}, \bar{g}_{IE} = 0.6\,\text{nS}$). Some Mexican hat networks with very narrow σ_E led to self-sustained activity (white area, red dotted border) and were excluded from the analysis. Values are computed from 125 excitatory neurons and 125 trials per stimulus ($s \in \{-1°, 1°\}$). Maximum input firing rate is 30 Hz. Best encoding performance is measured for Mexican hat networks.

per input stimulus (60% training, 20% validation, and 20% test data) to estimate Fisher information. Consequently, we sampled activity from only 125 excitatory neurons.

Clearly, as depicted in figure 5.37, the phenomenon is related to the Mexican hat topology. Best stimulus encoding performances were achieved by network topologies with shorter excitatory than inhibitory connection spread ($\sigma_E < \sigma_I$).

5.6.3 Fisher Information and Sample Size

We further wanted to know how this difference in information between the network types depends on sampling. As Averbeck et al. (2006) argued, it is important how information content in neural population saturates with network size. Thus, it might be worth investigating if there are a differences in the information saturation of the Mexican hat and the inverse profile. We repeated the previous experiments but with 7000 trials per stimulus (80% training, 10% validation, and 10% test trials) and estimated I_{LOLE} again for different number of sampled excitatory neurons.

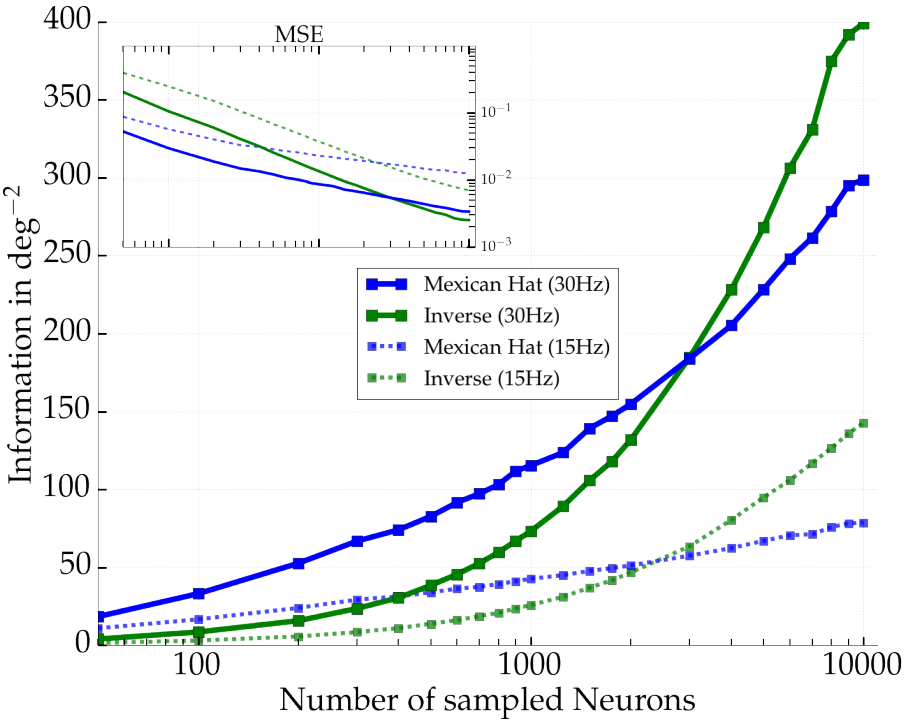

Figure 5.38: Amount of Fisher information as a function of samplesize for two network realizations, Mexican hat ($\sigma_E = 10\,\mathrm{px}$, $\sigma_I = 15\,\mathrm{px}$) in blue and inverse Mexican hat ($\sigma_E = 15\,\mathrm{px}$, $\sigma_I = 10\,\mathrm{px}$) in green, both operating close to criticality ($\bar{g}_{IE} = 0.6$ and $\bar{g}_{EE} = 0.4$). Solid lines for a strong maximum input frequency of $\nu_{\mathrm{Aff,\,max}} = 30\,\mathrm{Hz}$ and dashed for weaker input of 15 Hz. The inset shows the corresponding mean squared error (MSE) of the LOLE decoder. As long as the samplesize is small (i.e. up to one third of the whole network) more information can be extracted from the Mexican hat network.

Figure 5.38 shows I_{LOLE} as a function of the number of sampled neurons for two input strengths with 15 and 30 Hz. As long as roughly fewer than one third of the whole networks' neurons are considered for estimating the stimulus, a better performance was observed for the Mexican hat network. However, if more neurons were read out to reconstruct the stimulus, the inverse Mexican hat provided more information. Hence, for larger samples of neurons, already weak correlations lead to a considerable amount of redundant information. Accordingly, the detrimental nature of the noise correlations was more pronounced and eradicated the advantage of sharper tuning in Mexican hat networks.

5.7 Discussion II

As we hypothesized, noise correlations observed for Mexican hat topologies are indeed detrimental to stimulus processing. Information due to correlated responses was reduced relative to information estimated in shuffled trial data. However, we also found

support for our tuning sharpening hypothesis. In comparison to other topologies like balanced or inverse Mexican hat, more information about the stimulus could be extracted even if correlations were present (c.f. figures 5.32 and 5.37). Still, this effect is limited to samples of neurons. Increasing the number of neurons from which information was extracted to more than one third of the network size, better decoding performance was achieved by an inverse Mexican hat topology (c.f. figure 5.38).

5.7.1 Limitations of Tuning Sharpening

The sharpening of the network tuning due to the Mexican hat topology may explain this relative gain in information in comparison to other topologies. Yet, as sample size increased, the deteriorating effect of correlation worsened and eliminated the advantage of tuning sharpening. Moreover, our stimulus space is one-dimensional, only encompassing an orientation dimension. It is unclear if the effect of information gain for sub-sampling still holds for higher dimensional stimuli. For instance, one may think of a multi-dimensional stimulus like an orientation grating with a particular movement direction and velocity. For an ensemble of independent Poisson neurons with Gaussian tuning curves of width σ_s arranged evenly along the stimulus dimensions, Fisher information can be computed directly via (Dayan and Abbott, 2005)

$$I_{\text{Fisher}} = T \frac{(2\pi)^{D/2} \rho \sigma_s^D \nu_{\max}}{D \sigma_s^2}, \tag{5.10}$$

where $\nu_{\max}$ is the maximum firing rate or peak of the response tuning curve and ρ constitutes the density of neurons on a stimulus unit length. Additionally, T is the trial length in seconds and D denotes the dimensionality of the stimulus. As long as $D = 1$, decreasing σ_s, i.e. sharpening of the tuning, increases I_{Fisher}. For $D = 2$ the tuning width σ_s is irrelevant and, in particular, for $D > 2$ decreasing the tuning width actually reduces Fisher information. Although the cells in our simulation showed variability beyond Poisson spiking and responses were correlated, analogously to equation 5.10 tuning sharpening might be futile for higher dimensional stimuli. Yet, tuning sharpening could still provide the advantage of a sparser code and improve the information transmitted per spike.

Furthermore, all of our networks were essentially leaking most of the available afferent stimulus information. Every neuron received 100 afferent inputs in form of independent Poisson spike trains with firing rates of a similar order as the network output. Hence, these 100 independent spike trains were condensed into a single output spike train per network neuron. Thereby the networks reduced the available information by a factor on the order of 100. Arguably, one may interpret our networks as performing compression rather than being information relay nodes. A more realistic setting encompasses afferent inputs of an amount similar to the network size. Accordingly, in order to retain the number of afferent connections per neuron, many afferent inputs

may be shared by individual network neurons. However, this would inevitably yield correlations in the afferent input which we wanted to avoid in our study.

5.7.2 Comparison to other Modeling Studies

It is unclear whether it is beneficial for the cortex to perform tuning sharpening by recurrent rather than by afferent connections. Similar to our study, Seriès et al. (2004) investigated the influence of noise correlations and tuning sharpening on stimulus information. They used two different models. In the first model tuning was sharpened by strong recurrent Mexican hat connectivity much like in our model. In the second one the tuning of the afferent input was already sharp and no excitatory recurrent connections were present. The afferent inputs were chosen such that the output tuning curves of the two network models matched. In this setting, the Mexican hat network performed much worse than the model with afferent tuning. Besides a strong reduction in information due to noise correlations, Mexican hat networks showed already a much lower value of shuffled information. However, Seriès et al. (2004) used a Mexican hat defined over orientation space with a very large inhibitory spread of $\sigma_I = 60°$ compared to a very narrow excitatory width of $\sigma_E = 7.5°$. In our simulations we discovered that too wide inhibitory spread can have devastating effects on encoding quality. Information was enhanced only if σ_I was at most twice as wide as σ_E (c.f. figure 5.36A). Thus, the very wide inhibitory spread used by Seriès et al. (2004) might explain why already shuffled information was low in their Mexican hat network.

Likewise, Hansen et al. (2012) developed a model with a Mexican hat defined over orientation space. However, their ratio between excitatory and inhibitory widths was less vast with $\sigma_E = 15°$ and $\sigma_I = 40°$. Accordingly, in comparison to the work by Seriès et al. (2004), less information was lost in Mexican hat networks relative to topologies with wider excitatory spread.

Moreover, Toyoizumi et al. (2006) investigated a spike based Fisher information measure that takes precise spike timings into account. They proved that for decoding of spatio-temporal inputs and identifying stimulus onset, the optimal coupling scheme resembles a Mexican hat topology. This optimal topology exhibits global inhibition and local excitatory connections.

Furthermore, Eyherabide and Samengo (2013) argued that one needs to be careful when assessing the impact of noise correlations on information with decoders. Researchers may under- or overestimate the role of correlations by using non-optimal decoding methods. For example, a linear decoder like ours (c.f. equation 3.19) obviously fails to decode the simple XOR-problem.[5] Similarly, one may consider a neuron that represents one stimulus with both low and high activity whereas a second stimulus elicits medium activity. A linear decoder fails to tell apart the first from the second

[5]The name XOR stems from the expression *either or*.

stimulus because it cannot combine the low and high activity representation into a single class with the second stimulus class in between. Accordingly, linear decoders are not suitable to estimate the amount of information in activity of neural populations representing stimuli in non-linear fashion; neither in the shuffled nor in the original data. As a consequence, the estimated impact of correlations based on these information measures is wrong. However, we showed that using non-linear support vector regression did not yield an improvement in decoding performance in comparison to the LOLE decoder. Thus, we are confident that our linear method yielded a good estimation of the real information and impact of noise correlations in our networks.

Moreno-Bote et al. (2014) demonstrated analytically and numerically with leaky-integrate and fire neurons that a particular type of correlations is detrimental to stimulus encoding. They termed this shared variability *differential correlations*. The authors showed that the noise covariance matrix $\boldsymbol{Q}$ of neural responses to a stimulus s can be decomposed as

$$\boldsymbol{Q}(s) = \boldsymbol{Q}_0(s) + \varepsilon\, \boldsymbol{f}'(s)\, \boldsymbol{f}'^T(s), \tag{5.11}$$

where $\boldsymbol{Q}_0$ represents noise that is not harmful to encoding, whereas correlations that are detrimental can take up the form $\varepsilon\, \boldsymbol{f}'(s)\, \boldsymbol{f}'^T(s)$. ε is a potentially small coefficient, $\boldsymbol{f}'$ is the derivative of the neural tuning curve vector with respect to stimulus s, and $\cdot^T$ denotes the transpose operation. Hence, correlations are limiting encoding quality if they shift joint neural responses tangentially along the stimulus manifold in the neural response space. More simply, assuming a one-dimensional stimulus, like orientation, and a network response in form of a Gaussian curve or a bump profile defined over the stimulus space, the following holds: If noise moves the response curve back and forth across the stimulus space, this yields differential correlations and one cannot discriminate the noise from the actual stimulus. Indeed, we made similar observations in our network model with heterogeneous stimuli. The heterogeneous input locked bump activity to a particular location, but we still observed small jitter around the maximum input (c.f. figures 5.16 and 5.31). Consequently, in our 2D networks featuring an orientation map, small movements of the bump responses defined over the two-dimensional neural space simultaneously imply jitter of network responses in the stimulus space. Similarly, the spatial profile of the differential correlations found by Moreno-Bote et al. (2014) are reminiscent of a sinusoidal modulation whose amplitude decays with distance in stimulus space akin to our results (c.f. figures 5.5A and 5.20A). Moreno-Bote et al. (2014) argued that differential correlations are usually very small ($\varepsilon \ll 1$) and hidden underneath other correlations ($\boldsymbol{Q}_0$). This makes them notoriously difficult to identify by simply measuring correlation coefficients. In order to detect differential correlations the authors suggested to use the decoder approach by Seriès et al. (2004) to estimate Fisher information. This is what we did in our study and indeed observed that noise

correlations from recurrent connectivity reduced Fisher information in comparison to shuffled trials (c.f. figure 5.32).

Kanitscheider et al. (2015) developed a generative model of differential correlations based on convergent feed-forward projections in a primary visual cortex model. In this model noise correlations emerged due to shared noise from receptor and LGN input to V1. In comparison, one can interpret our findings as a model of differential correlations originating from recurrent connections instead of afferent input.

5.7.3 Computing at the Edge of Chaos

We pointed out that recurrent connection strength operating closer at the bifurcation to criticality was beneficial for stimulus encoding regardless of the network topology (c.f. figure 5.35). Similarly, Stimberg et al. (2009) identified this operating regime as the one most likely reproducing tuning phenomena from an experimental study by Mariño et al. (2005). This is in line with the prominent notion of *computation at the edge of chaos* (Langton, 1990). Other theoretical studies have shown that networks can accomplish more challenging tasks near the transition from ordered to chaotic dynamics (Bertschinger and Natschläger, 2004), or that pattern recognition can be performed best close to a chaotic state (Rossello et al., 2014). Analysis of multi-array electrode recordings in cat, monkey, and human cortex revealed statistics and signal propagation patterns indicative of criticality (Dehghani et al., 2012). In like manner, criticality might be abundant among many stages in sensory processing. For instance, Mora et al. (2015) demonstrated that activity in the rat retina is balanced near a critical point.

5.7.4 Biological Relevance of Results

A modulation of information due to noise correlations has been reported frequently in visual cortex experiments (Gu et al., 2011; Chelaru and Dragoi, 2014). Similar to our observations that noise correlations are less disadvantageous if neurons within a network were sub-sampled (c.f. figure 5.38), Montijn et al. (2014) measured a saturation of decoding performance with sample size using a variety of decoders in mouse visual cortex. Similar observations were made by Freiwald et al. (2002) reconstructing stimuli with a Bayesian decoder from data recorded in rat primary visual cortex. Comparable to our Mexican hat networks (c.f. figure 5.32), Graf et al. (2011) discovered that correlations among neurons in macaque primary visual cortex carry a significant amount of information. They reported that decoding accuracy could drop by more than five percent if a decoder ignored correlations.

Hansen et al. (2012) as well as Smith et al. (2013) found that the magnitude of correlations in monkey visual cortex are laminar dependent. For the input layer, often referred to a the granular (Hansen et al., 2012) or middle layer (Smith et al., 2013), the measured average r_{SC} was almost 0. Whereas in the deep or infra-granular (IG) as

well as the superficial or supra-granular (SG) layers the experimenters measured significant correlations on the order of 0.1 to 0.2 for cells up to 300 µm apart. This can give rise to the interpretation that correlations are predominant in layers that are projecting downstream to higher cortical areas. In contrast, the middle layer, receiving input from the LGN, provides a very accurate and unmodified representation of a stimulus. Accordingly, precise discrimination between similar stimuli can be achieved at the cost of requiring spiking activity from many neurons. Since projections originating from the middle layer terminate predominantly within the primary visual cortex, this is not an unreasonable assumption. However, the output layers (IG and SG) provide a compact stimulus representation that yields correlated variability as a side effect. This representation allows downstream areas, presumably relying on costly long range connections, to read out much information about a stimulus from sampling only a few neurons. There, a Mexican hat coupling would be beneficial because the detrimental effect of correlations is negligible for small sample sizes, but the stimulus representation is sparser and Fisher information is increased in comparison to other topologies.

Conclusion

6

We developed a novel framework for managing of numerical experiments called *pypet*. The Python library supports massive parameter exploration of simulations not only in Computational Neuroscience but in all disciplines relying on numerical experiments. The novel tool facilitates reproducible research by tightly linking parameters and results and disentangling core simulation code from administrative tasks like scheduling and data serialization. The program is well-tested, documented, and freely available. Moreover, *pypet* helped performing all experiments of this thesis and guided the extensive parameter explorations presented in the previous chapter.

Furthermore, we discussed the phenomenon of noise correlations in the brain, especially in visual cortex. For repeated presentation of the same stimulus, cells can exhibit correlated variability around an expected mean response. Experimentalist often measured a modulation of noise correlations with increasing distance between cell pairs (Smith and Kohn, 2008; Solomon et al., 2014). In general, shared variability is positive on the order of 0.1 to 0.4 for nearby cells with similar tuning properties (Zohary et al., 1994; Huang and Lisberger, 2009). The magnitude and spatial structure of correlations are particularly important because they have an impact on how well a stimulus is represented by a neural response. Surprisingly, correlations in cortex are often detrimental to stimulus information (Gutnisky and Dragoi, 2008; Hansen et al., 2012). Why should the brain hinder its own stimulus discrimination? Why are noise correlations present, especially if they are detrimental to correlations?

Hitherto, there are no clear answers to these questions. We discussed many competing hypotheses about what is causing the correlated variability in cortex. Ecker et al. (2010) argued that correlations are merely a byproduct or artifact of experimental setups like anesthesia or erroneous spike sorting. In contrast, Hansen et al. (2012) fostered the hypothesis that recurrent connectivity is a source of correlated variability. They suggested that a so called Mexican hat profile can yield noise correlations.

We further investigated this hypothesis analytically by looking at recent work of Rosenbaum and Doiron (2014). They conveyed that Mexican hat connectivity with

wider excitatory than inhibitory spread can amplify spatial frequencies in neural activity. We further demonstrated that this finding still holds if one augments the model with adaptation. However, one cannot give an analytical expression of the network state in case the homogeneous fixpoint is unstable. Therefore, we simplified the model and compared it to work by Hansel and Sompolinsky (1998). Their model is based on a simpler sinusoidal coupling scheme that can only amplify a single spatial frequency. Accordingly, the analytical expression comprises a bump solution, also called marginal phase. Instead of a single fixpoint, a line of potential fixpoints emerges. Hence, for homogeneous input the position of the bump is arbitrary. Moreover, the activity bump may wander along the spatial dimension of the network in case of noise or adaptation. As a consequence, the wandering bump leads to coactivation of neighboring neurons. For spiking neuron networks with the Gaussian Mexican hat coupling we predicted a similar behavior but with the occurrence of multiple bumps. We further predicted that bump phase changes across trials and the joint movement due to noise and adaptation lead to sinusoidally modulated noise correlations.

Indeed, we could support this hypothesis with simulations of spiking neuron networks. We demonstrated that the findings are robust and can be obtained for wide parameter ranges in one-dimensional as well as tow-dimensional networks. Moreover, we showed that using heterogeneous tuned input, an interaction between the inherent bump structure and the input stimulation was observed. For 2D networks, noise correlations were modulated fairly linearly instead of a strong sinusoidal distance dependency. Similar observations were made regarding preferred orientation dependency. Hence, strongest correlations were measured among nearby neurons with similar response properties. This matches reports from the experimental literature (Lee et al., 1998; Huang and Lisberger, 2009; Schulz et al., 2015).

Furthermore, we demonstrated that the noise correlations caused by Mexican hat topology were indeed detrimental to stimulus encoding. Yet, results were twofold. Besides the detrimental correlations, the encoding quality based on small samples of neurons in Mexican hat networks was higher than compared to other topologies. Accordingly, Mexican hat networks sharpened tuned inputs. This lead to a compact representation of the stimulus. Nonetheless, with increasing sample size, noise correlations eradicated this advantage. Such a saturation of information and decoding performance with sample size was also observed in experimental studies (Freiwald et al., 2002; Montijn et al., 2014). This led us to the speculation that shared variability is a price that cortex is willing to pay in output layers of sensory areas. Consequently, downstream neurons relying on long range inter-areal connections need to contact fewer neurons in order to obtain a good estimate of the stimulus. This is in line with recent findings that researchers located noise correlations only in output layers of primary visual cortex (Hansen et al., 2012; Smith et al., 2013).

Still, there remain many open questions. Does the brain really rely on Mexican hat coupling? Up to know the biological data is inconclusive and points in opposing

directions (Buzas et al., 2001; Mariño et al., 2005; Levy and Reyes, 2012). Moreover, what are other sources of noise correlations? We argued that recurrent connectivity is not the primary source of shared variability, but may be one among many others. Thus, noise correlations measured in animal studies might be a compound of experimental factors as anesthesia, feed-back modulations like attention, and recurrent and shared inputs. The spike count correlation coefficient can identify correlated variability in cortex, but it is a rather poor statistic to discern potential factors contributing to the correlations. Accordingly, we suggested that future experimental research should aim at multi-cell recordings with a high spatial resolution. Researchers need to analyze the recorded network dynamics and spiking activity in its entirety and not just condense it to summarizing statistics such as r_{SC}.

Of course, there is also more work for theoreticians. Finally, we are going to sketch two potential future lines of research for network modeling in the following section.

6.1 Future Work

Besides experimental factors entailing noise correlations, there might be other intrinsic network properties. For example, in our study we completely ignored the contribution of afferent inputs. All afferent stimulation used independent Poisson spike trains. Accordingly, we neither included correlations in the input signal nor the sharing of afferent connections by several post-synaptic cells. Renart et al. (2010) demonstrated how recurrent connectivity may even decorrelate network activity despite afferent correlations. Yet, we do not know the outcome of combining correlated afferent inputs and a recurrent topology like Mexican hat. As this connectivity scheme yields noise correlations by itself, we could investigate how these two sources of shared variability collate. Do correlations simply add linearly? Or do we observe non-linear interactions between afferent and recurrent noise correlations?

What other recurrent profiles can cause noise correlations? Indeed, there can be coupling schemes — besides Gaussian Mexican hat and the simple sinusoidal one by Hansel and Sompolinsky (1998) — that amplify certain spatial frequencies, but at the same time damp the 0th Fourier mode. For instance, our preliminary data provided in the appendix B suggests that patchy excitatory long range connections between neurons with similar orientation preference are a suitable candidate. Future work may elaborate on this phenomenon and investigate the influence of long range connections on noise correlation thoroughly. Indeed, the existence of long ranging horizontal connections in the visual cortex has been known for quite some time (Gilbert and Wiesel, 1989; Das and Gilbert, 1995). Patchy projections were also taken up by the modeling community (Voges and Perrinet, 2010; Voges et al., 2010). For instance, Voges and Perrinet (2012) reported the emergence of complex spatial patterns in such networks. Hence, it might be fruitful to investigate the link between these spatial patterns, shared variability, and their role in stimulus processing.

Appendix: Spiking Neuron Network Details

A

In the following section we are going to explain the spiking neuron network model in detail. We will list all differential equations and give all parameter settings in a compact table at the end of the appendix.

A.1 Model Details

We used adaptive exponential integrate and fire neurons (AEIF) (Brette and Gerstner, 2005). Each neuron of population $X \in \{E, I\}$ indexed by i comprised two differential equations describing the development of the membrane potential $V_{X,i}$ and an adaptation current $w_{X,i}$ over time:

$$C\frac{dV_{X,i}(t)}{dt} = -g_{L,X}(V_{X,i} - E_L) + g_{L,X}\Delta_T \exp\left(\frac{V_{X,i} - V_T}{\Delta_T}\right) - w_{X,i} - I_{X,i}(t), \quad \text{(A.1)}$$

$$I_{X,i}(t) = I_{XA,i}(t) + I_{XE,i}(t) + I_{XI,i}(t), \quad \text{(A.2)}$$

$$\tau_w\frac{dw_{X,i}(t)}{dt} = a_X(V_i - E_L) - w_{X,i}, \quad \text{(A.3)}$$

with $-g_{L,X}(V_{X,i} - E_L)$ defining the leak term and $g_{L,X}\Delta_T \exp\left(\frac{V_{X,i}-V_T}{\Delta_T}\right)$ being an approximation to exponential rise of the sodium (Na^+) current of an action potential, assuming that the activation of Na^+-channels is instantaneous and inactivation can be neglected (Fourcaud-Trocme et al., 2003). Moreover, C denotes the capacitance, $g_{L,X}$ is the leak conductance of population X, E_L the leak reversal potential, V_T the threshold parameter, Δ_T the slope factor, a_X the sub-threshold adaptation of population X, τ_w the adaptation time constant, and $I_{XY,i}$ denotes the afferent and recurrent currents. Moreover, each neuron exhibited a threshold condition:

$$\begin{aligned} \text{if } V_{X,i} \geq V_{\text{cut}}: \quad & V_{X,i} \to E_L \quad (\text{with clamping for } \tau_{\text{refr},X}), \\ & w_{X,i} \to w_{X,i} + b_X, \end{aligned} \quad \text{(A.4)}$$

i.e. each neuron was reset to the leakage potential in case the membrane potential rose beyond the cut off point V_{cut}. In addition, this crossing was considered to be the particular point in time a neuron spiked. Then $w_{X,i}$ was increased by a constant term b_X,

modeling spike-triggered adaptation. Moreover, the dynamic of $V_{X,i}$ was clamped for a period of $\tau_{\text{refr},X}$ to E_L after the crossing to simulate an absolute refractory period.

Note that we used different refractory periods $\tau_{\text{refr},X}$ for excitatory and inhibitory neurons. We assumed a shorter period for fast spiking inhibitory neurons of 2 ms in comparison to the excitatory cells with 3 ms. Similarly, we used different leak conductances $g_{L,X}$ among the two populations to account for the experimental observation that the conductance is about twice as large in inhibitory cells (Karayannis et al., 2007).

Furthermore, the currents $I_{XY,i}$ with $Y \in \{A, E, I\}$ were modeled as conductance based:

$$I_{XY,i}(t) = g_{XY,i}(t)\,(V_{X,i}(t) - E_Y). \tag{A.5}$$

Afferent currents were mediated by fast exponentially decaying *AMPA*-like synapses. Recurrent input consisted of excitatory as well as an inhibitory part. The excitatory current depended on a mixture of fast *AMPA* and slow bi-exponential *NMDA*-like synapses, whereas the inhibitory current was based on fast *GABA$_A$*-like synapses:

$$\tau_{\text{AMPA}} \frac{dg_{XA,i}(t)}{dt} = -g_{XA,i}(t) + \bar{g}_{XA} \sum_{t_j^k} \delta(t_j^k - t), \tag{A.6}$$

$$g_{XE,i}(t) = g_{XE_{\text{AMPA}},i}(t) + g_{XE_{\text{NMDA}},i}(t), \tag{A.7}$$

$$\tau_{\text{AMPA}} \frac{dg_{XE_{\text{AMPA}},i}(t)}{dt} = -g_{XE_{\text{AMPA}},i}(t) + \alpha\,\bar{g}_{XE} \sum_{t_j^k} \delta(t_j^k - t + \Delta_{ij}), \tag{A.8}$$

$$\tau_{\text{decay}} \frac{dg_{XE_{\text{NMDA}},i}(t)}{dt} = -g_{XE_{\text{NMDA}},i}(t) + \beta\, h_{XE_{\text{NMDA}},i}(t), \tag{A.9}$$

$$\tau_{\text{rise}} \frac{dh_{XE_{\text{NMDA}},i}(t)}{dt} = -h_{XE_{\text{NMDA}},i}(t) + (1-\alpha)\bar{g}_{XE} \sum_{t_j^k} \delta(t_j^k - t + \Delta_{ij}), \tag{A.10}$$

$$\beta = \left(\frac{\tau_{\text{rise}}}{\tau_{\text{decay}}}\right)^{\tau_{\text{decay}}/(\tau_{\text{rise}}-\tau_{\text{decay}})}, \tag{A.11}$$

$$\tau_{\text{GABA}_\text{A}} \frac{dg_{XI,i}(t)}{dt} = -g_{XI,i}(t) + \bar{g}_{XI} \sum_{t_j^k} \delta(t_j^k - t + \Delta_{ij}), \tag{A.12}$$

where $\alpha \in [0, 1]$ determines the ratio between *AMPA* and *NMDA* receptors and t_j^k is the time point of the kth spike of pre-synaptic neuron j. In addition, $\bar{g}_{XY}$ denotes the maximum conductance, i.e. the synaptic coupling strength, $\tau_{...}$ are time constants, E_Y is the synaptic reversal potential, and Δ_{ij} is the synaptic delay between neuron j and i as given below.

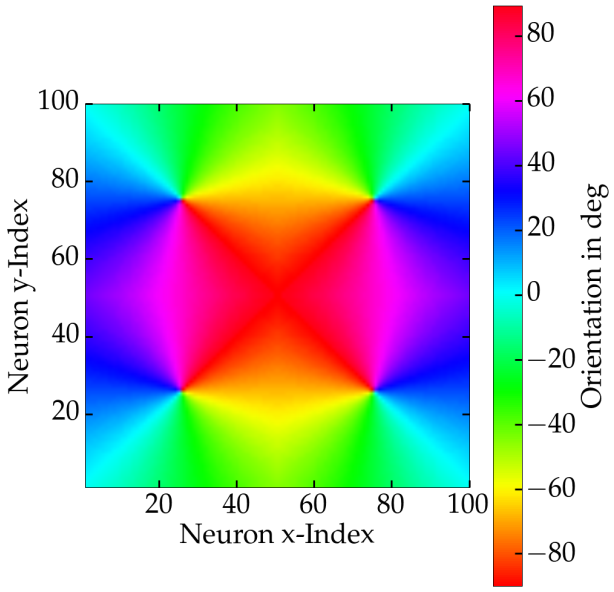

Figure A.1: Pinwheel Map

A.1.1 Synaptic Delays

Synaptic delays Δ_{ij} were distance dependent and computed as

$$\Delta_{ij} = \underline{\Delta}_{ij} + \frac{d_{ij}}{v}, \tag{A.13}$$

with a basic time delay $\underline{\Delta}_{ij}$ that was sampled uniformly from the interval $[0.2, 1.2]$ ms. Moreover, v denotes the signal velocity and d_{ij} is the distance between cells i and j. We assumed a signal velocity of $v = 200\,\mathrm{kpx/s}$ for the one-dimensional networks and $v = 13.3\,\mathrm{kpx/s}$ in the two-dimensional setting which corresponds to 0.2 m/s in our cat cortex scaling.

A.1.2 Numerical Simulation

We integrated all equations using a simple Euler scheme with a particular step size dt. The runtime was optimized by pre-compiling Python code into faster C code using the Numba package (The Numba Development Team, 2015).

To allow fast computation of the exponential upstroke as in $g_{L,X}\Delta_T \exp\left(\frac{V_i(t)-V_T}{\Delta_T}\right)$, we precomputed the function. We used 10^6 sampling points and linearly sampled membrane voltages in a range between 92.5 mV and V_{cut}. We did not interpolate in between points. For membrane voltages falling in between two sampling points, we used the result of the closest point. For values smaller than the lower bound, we simply took a voltage of 92.5 mV.

Randomly sampled membrane potentials $V_{X,i}$ from a uniform distribution over the interval $[E_L, V_T]$ were used as initial conditions. All other dynamic parameters were

set to 0. Moreover, before any measures or statistics were computed a simulation was run for at least 1 second to allow the network to settle into a stationary state.

The source code is publicly available and hosted on github (`https://github.com/SmokinCaterpillar/visualcortex`).

A.2 Parameter Settings

In here we provide all parameter settings, see table A.1. The pinwheel map of the heterogeneous input for the 2D networks is also depicted in figure A.1, showing the preferred orientations s_{PO_i} of 100×100 neurons.

Parameter	Description	Default Value	varied
Simulation			
dt	Euler step size	0.1 ms	
Network Topology			
N_E	Number of excitatory neurons	10,000	yes
N_I	Number of inhibitory neurons	2500	yes
K_E	Number of excitatory connections	200 (1D) or 400 (2D)	yes
K_I	Number of inhibitory connections	100 (1D) or 200 (2D)	yes
K_A	Number of afferent connections	100	yes
D	Map dimensionality	1 or 2	
σ_E	Excitatory connection spread	125 (1D) or 10 px (2D)	yes
σ_I	Inhibitory connection spread	250 (1D) or 15 px (2D)	yes
$\underline{\Delta}_{ij}$	Basic synaptic delay	[0.2, 1.2] ms	
v	Signal velocity	200 (1D) or 13.3 kpx/s (2D)	
Neuron Properties			
C	Capacitance	200 pF	
$g_{L,E}$	Excitatory leak conductance	10 nS	
$g_{L,I}$	Inhibitory leak conductance	20 nS	
E_L	Leak reversal potential	−65 mV	
V_T	Threshold parameter	−50 mV	
Δ_T	Slope factor	2 mV	
V_{cut}	Spike cut-off	−30 mV	
$\tau_{\text{refr},E}$	Excitatory refractory period	3 ms	
$\tau_{\text{refr},I}$	Inhibitory refractory period	2 ms	
τ_w	Adaptation time constant	250 ms	
a_E	Exc. sub-threshold adaptation	2 nS	
a_I	Inh. sub-threshold adaptation	0.2 nS	
b_E	Excitatory SFA strength	50 pA	in appendix
b_I	Inhibitory SFA strength	5 pA	in appendix

Parameter	Description	Default Value	varied
Synaptic Coupling			
$\bar{g}_{EE}$	Exc. to exc. coupling strength	0.4 nS	yes
$\bar{g}_{IE}$	Exc. to inh. coupling strength	0.6 nS	yes
$\bar{g}_{EI}$	Inh. to exc. coupling strength	5.0 nS	
$\bar{g}_{II}$	Inh. to inh. coupling strength	5.0 nS	
$\bar{g}_{EA}$	Aff. to exc. coupling strength	1.5 nS	
$\bar{g}_{IA}$	Aff. to inh. coupling strength	0.5 nS	
E_E	Excitatory reversal potential	0 mV	
E_I	Inhibitory reversal potential	−80 mV	
E_A	Afferent reversal potential	0 mV	
α	Fraction of AMPA receptors	0.7	
τ_{AMPA}	AMPA decay time constant	3 ms	
τ_{rise}	NMDA rise time constant	5 ms	
τ_{decay}	NMDA decay time constant	80 ms	
$\tau_{\text{GABA}_\text{A}}$	GABA_A decay time constant	5 ms	
Homogeneous Input			
$\nu_{\text{Aff,i}}$	Afferent firing rate	15 Hz (1D) or 10 Hz (2D)	yes
Heterogeneous Input 1D			
$\nu_{\text{Aff, max}}$	Maximum afferent firing rate	30 Hz	yes
$\nu_{\text{Aff, base}}$	Baseline afferent firing rate	3 Hz	
ω_s	Spatial frequency	1.4 kpx^{-1}	yes
Heterogeneous Input 2D			
$\nu_{\text{Aff, max}}$	Maximum afferent firing rate	30 Hz	yes
$\nu_{\text{Aff, base}}$	Baseline afferent firing rate	3 Hz	
σ_{Aff}	Input tuning width	27.5°	
s	Orientation stimulus	1°	yes
s_{PO_i}	Preferred orientation	See figure A.1	

Table A.1: Model Parameters. All parameters that were systematically varied at some point during the thesis are marked as such in the last column.

Appendix: Preliminary Results

B

In the following sections we are going to present some preliminary results. First, we will briefly discuss correlated variability and pattern formation in regimes beyond criticality. Afterwards, we will shortly look at the influence of adaptation on correlations. Finally, we are going to investigate the influence of excitatory patchy long range connections on shared variability.

B.1 Pattern Formation Beyond Criticality

Figure B.1 shows the average firing rate, CV, and noise correlations for a Mexican hat as well as inverse Mexican hat configuration. These are the same networks as in figure 5.23, but the CV and r_{SC} beyond criticality are depicted as well.

We identified that even for the inverse Mexican hat there existed a very small region beyond criticality (red dotted line) where the CV is larger than 1, rates remained small, and average noise correlations were on the order of 0.1 to 0.3. Looking at the joint spiking activity, one can make out the emergence of spatial patterns. These were less well formed than the bumps of Mexican hat networks, see figure B.2.

This behavior was not expected from the theoretical considerations in chapter 4. However, the theory assumes rectified, linear rate models with instantaneous coupling. Further thorough investigations including an extension of the model by using a nonlinear transfer function may explain this phenomenon. Alternatively, it might be helpful to incorporate the conductance based coupling of the spiking neurons. For instance, Rudolph-Lilith et al. (2012) demonstrated that rate models with conductance based dynamics can exhibit complex activity patterns and are well suited reductions of spiking neurons.

B.2 Noise Correlations and Adaptation

Clearly, adaptation plays an important role in the emergence of noise correlations. Adaptation yields movement of the bump solutions. Accordingly, removing adaptation from the spiking neurons reduced the magnitude of noise correlations for nearby neurons considerably (figure B.3).

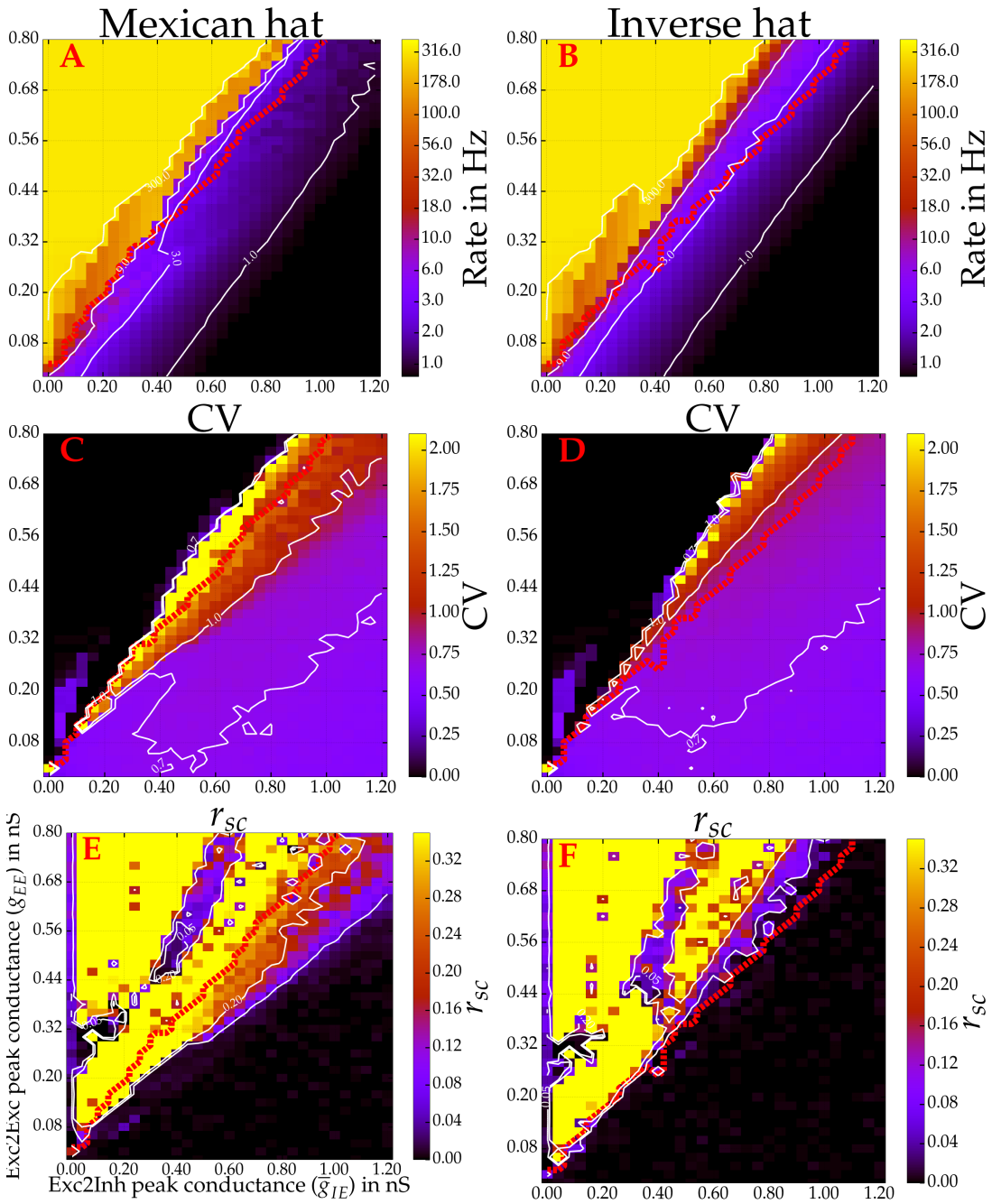

Figure B.1: Parameter exploration of $\bar{g}_{EE}$ and $\bar{g}_{EI}$ for a Mexican hat network on the left ($\sigma_E = 10\,\text{px} < \sigma_I = 15\,\text{px}$) and an inverse network on the right side ($\sigma_E = 15\,\text{px} < \sigma_I = 10\,\text{px}$). Top row (A, B) shows the average firing rate (color coded) of all neurons in the network. The thick dotted red line marks the bifurcation to criticality. Middle row (C, D) displays the average coefficient of variation (CV). The third row (E, F) gives the average r_{SC} among close by cell pairs (at most 13.3 px apart) estimated from a single trail of 30 seconds and a sliding window of 1 second.

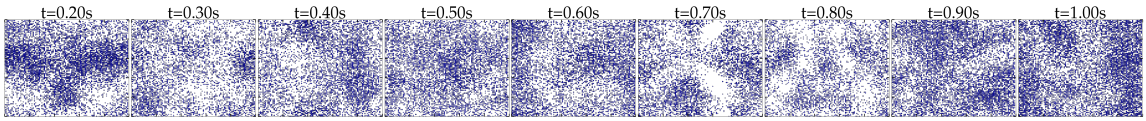

Figure B.2: Consecutive snapshots of 100 ms length of network activity. A blue dot corresponds to an excitatory spike within the time interval. The activity pattern of an inverse Mexican hat network ($\sigma_E = 15\,\mathrm{px} > \sigma_I = 10\,\mathrm{px}$) beyond criticality ($\bar{g}_{EE} = 0.7\,\mathrm{nS}$ and $\bar{g}_{IE} = 0.8\,\mathrm{nS}$) is shown.

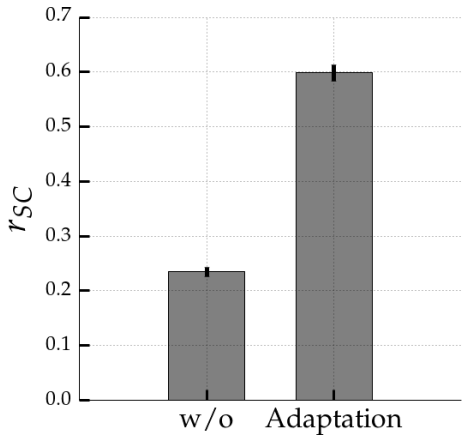

Figure B.3: Average noise correlation within a Mexican hat network close to criticality ($\sigma_E = 125\,\mathrm{px} < \sigma_E = 250\,\mathrm{px}$, $\bar{g}_{EE} = 0.4\,\mathrm{nS}$, $\bar{g}_{IE} = 0.6\,\mathrm{nS}$). Values are averaged across pairs at most 100 px apart over 50 trials of 1 second each with homogeneous input. Error bars mark standard deviations across 10 network realizations. On the left without adaptation $a_X = b_X = 0$ and on the right with adaptation $a_E = 10a_I = 2\,\mathrm{nS}$ and $b_E = 10b_I = 50\,\mathrm{pA}$. Removing adaptation significantly reduces correlations (Wilcoxon rank-sum test, $p < 0.001$, 10 network samples).

Furthermore, we extensively varied the spike frequency adaptation (SFA) strength b_E while keeping the ratio $b_E = 10b_I$. Accordingly, figure B.4 depicts the average r_{SC} as a function of b_E for varying integration window lengths. For stronger magnitudes of spike frequency adaption $b_E > 50\,\mathrm{pA}$, we observed a reduction in the r_{SC} for longer time windows. More precisely, the average r_{SC} as a function of adaptation strength exhibited a slight sinusoidal modulation for longer integration window lengths.

Stronger adaptation in the simple sinusoidal rate model by Hansel and Sompolinsky (1998), as discussed in chapter 4, increases the bump velocity. We observed a similar behavior for the Gaussian Mexican hat coupling in the spiking neural networks. For the weak default SFA with $b_E = 50\,\mathrm{pA}$, we identified slow bump movements that eventually change directions. Whereas stronger adaptations like 100 or 200 pA yielded coordinated unidirectional bump movement as depicted in figure B.5. Unidirectional movement leads to regular oscillations on the single neuron level. Stronger values of b_E increased the bump velocity and thereby increased the oscillation frequency. The phases of the oscillations may fit multiple times into a large integration time window.

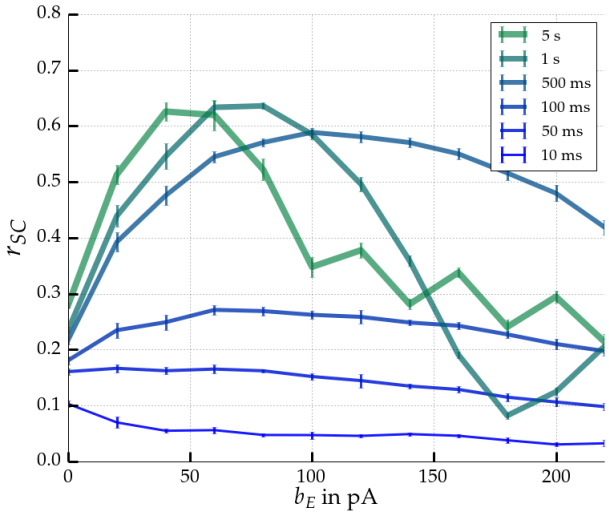

Figure B.4: Average noise correlation within Mexican hat network close to criticality ($\sigma_E = 125\,\text{px} < \sigma_E = 250\,\text{px}$, $\bar{g}_{EE} = 0.4\,\text{nS}$, $\bar{g}_{IE} = 0.6\,\text{nS}$) for various integration window sizes. Values are averaged across pairs at most 100 px apart over 50 trials of 1 second each with homogeneous input. Error bars mark standard deviations across 5 network realizations. We vary b_E and $b_I = 0.1 b_E$ and keep all other parameters fixed.

This in turn may explain the sinusoidal modulation of the average r_{SC} as function of b_E for larger integration time windows in figure B.4.

B.3 Long Range Connections

Herein we are going to discuss long-range connections as a potential source of correlated variability. At first, we will show that in line with the findings from chapter 4, long range connection can amplify some spatial Fourier mode $n \neq 0$ while the fixpoint is still stable for $n = 0$. Next, we are going to present a simulation with long range connections in a network of spiking neurons.

B.3.1 Theoretical Motivation

In contrast to previous coupling schemes, we assume that besides a local coupling, excitatory connections can exhibit patchy long range connections. Moreover, for simplicity we will use a one-dimensional model. Let $\kappa^{\Gamma}_{XE}(x) = g_{\text{LLR}}(x; p, \sigma_E, K_{LR}, d_{LR}, \sigma_{\text{LR}})$, where g_{LLR} is a compound of local (L) and long range (LR) connectivity:

$$g_{\text{LLR}}(x; p, \sigma_E, K_{\text{LR}}, d_{\text{LR}}, \sigma_{\text{LR}}) = p\, g(x; \sigma_E) + (1 - p)\, g_{\text{LR}}(x; K_{\text{LR}}, d_{\text{LR}}, \sigma_{\text{LR}}), \tag{B.1}$$

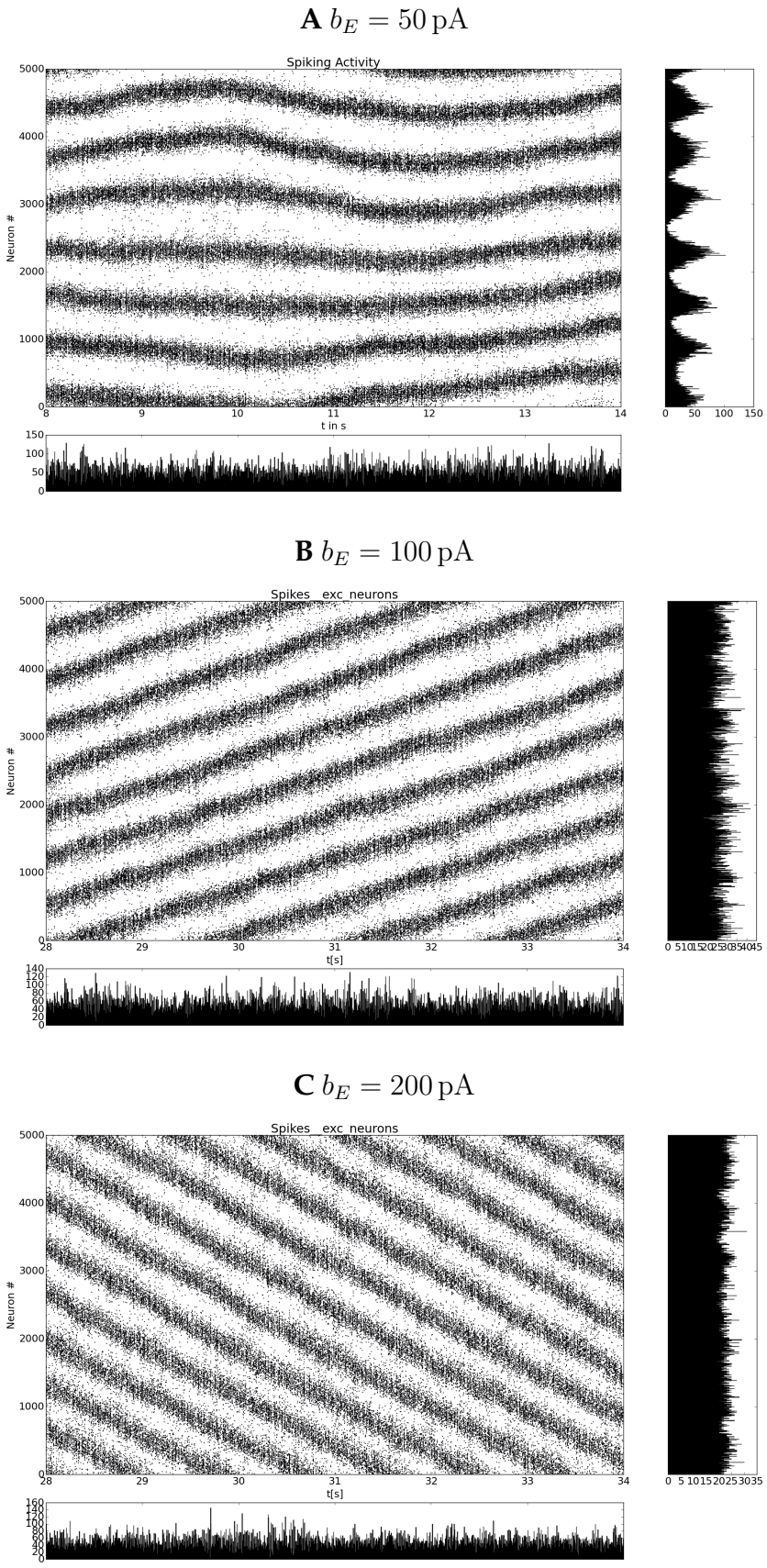

Figure B.5: Spiking activity of a Mexican hat network ($\sigma_E = 125\,\mathrm{px} < \sigma_E = 250\,\mathrm{px}$, $\bar{g}_{EE} = 0.4\,\mathrm{nS}$, $\bar{g}_{IE} = 0.6\,\mathrm{nS}$) for three different SFA strengths b_E. A small black dot represents a spike of a particular neuron at a particular point in time. Neurons are ordered according to their position on the ring. Small plots show the activity histograms over time (horizontal) as well as over space (vertical).

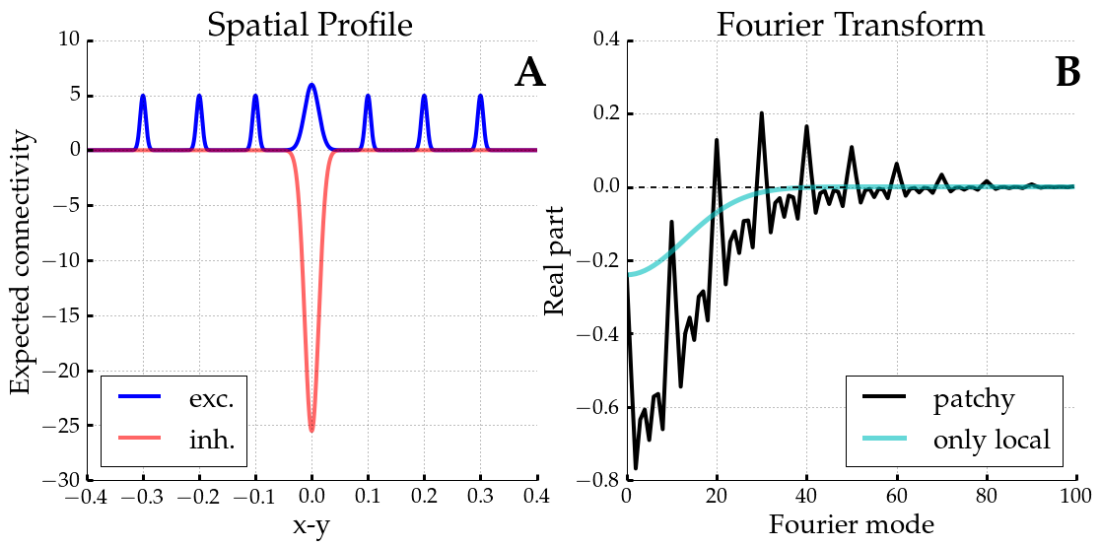

Figure B.6: Left (A): Spatial profile of excitatory patchy long rang connections (blue) and local inhibitory connections (red). For better visibility inhibition is depicted with a negative sign. Parameters are $\sigma_E = \sigma_I = 0.0125$, $\sigma_{\text{LR}} = 0.005$, $K_{\text{LR}} = 3$, $d_{\text{LR}} = 0.1$, $p = 1/3$, $q = 0.8$, $j_E = 0.7$, and $j_I = 4$. Right (B): Real part of Fourier transform as black curve assuming a simple single population rate model and a coupling as in equation B.4. Some spatial frequencies are amplified. In cyan the spatial Fourier transform of the local coupling only is shown, there all modes are stable.

with p being the fraction of short range connections, $g(x;\sigma_E)$ is the standard Gaussian local coupling from equation 4.37, and $g_{\text{LR}}(x;K_{\text{LR}},d_{\text{LR}},\sigma_{\text{LR}})$ models patchy long range connection as a sum of wrapped Gaussians:

$$g_{\text{LR}}(x;K_{\text{LR}},d_{\text{LR}},\sigma_{\text{LR}}) = \frac{1}{\sqrt{8\pi}\sigma_{\text{LR}}\,K_{\text{LR}}} \sum_{\substack{k\neq 0,\\ k=-K_{\text{LR}}}}^{K_{\text{LR}}} \sum_{l=-\infty}^{\infty} \exp\left(-\frac{(x+l-k\,d_{\text{LR}})^2}{2\sigma_{LR}^2}\right), \tag{B.2}$$

where K_{LR} denotes the number of patchy centers on each side of the post-synaptic neuron, d_{LR} is the distance between the patchy centers, and σ_{LR} is the width of the patchy Gaussian distributions. The Fourier transform of the patchy coupling can be computed by summing the Fourier transform of the individual Gaussians. For a wrapped Gaussian with non-zero mean μ, the circular Fourier transform is (Rosenbaum and Doiron, 2014)

$$\tilde{g}(n;\mu,\sigma) = \exp(-2\pi^2\,n^2\,\sigma^2 - 2\pi\,\mathrm{i}\,n\,\mu). \tag{B.3}$$

We assume locally balanced profile and that patchy long range projections are sharp and specifically targeted connections; hence, $\sigma_{LR} < \sigma_E = \sigma_I$. This setup is depicted in

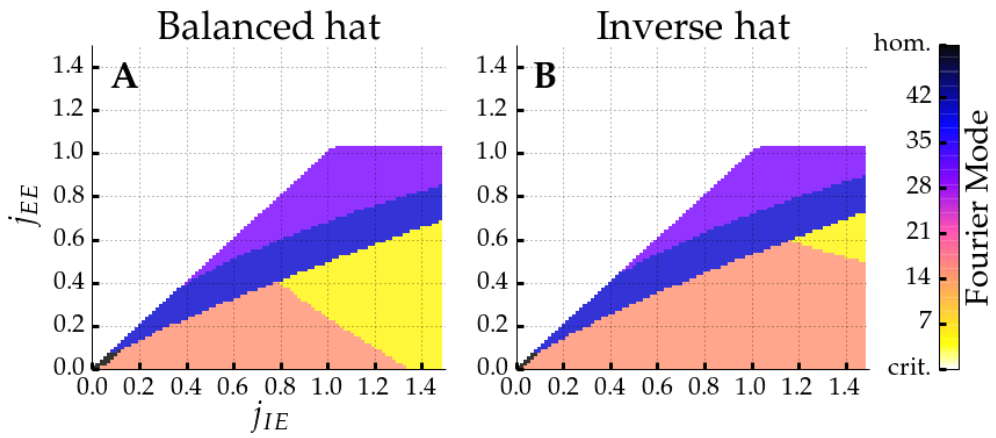

Figure B.7: The Fourier mode $n \geq 0$ with the largest real part of all eigenvalues is color coded for two different local coupling schemes combined with patchy projections. Excitatory to excitatory j_{EE} and excitatory to inhibitory values j_{IE} are explored. Inhibitory connection strengths are fixed with $j_{II} = j_{EI} = 4.0$ and we set $q = 0.8$, $p = 1/3$, $K_{\text{LR}} = 3$, $d_{\text{LR}} = 0.1$, and $\epsilon = 0.01$. For the balanced case it holds $\sigma_{\text{LR}} = 0.005 < \sigma_E = \sigma_I = 0.0125$, whereas for the inverse Mexican hat we use $\sigma_{\text{LR}} = 0.005 < \sigma_I = 0.0125 < \sigma_E = 0.025$. Critical runaway excitation where already Fourier mode $n = 0$ is unstable is depicted in white. Inhomogeneous non-runaway activity can be observed for both networks with patchy long range projections.

figure B.6A. If we use a single population mean-field model (as in equation 4.62) and a patchy coupling of the form

$$\omega_E(x) - \omega_I(x) = j_E \, g_{\text{LLR}}(x; p, \sigma_E, K_{LR}, d_{LR}, \sigma_{\text{LR}}) - j_I \, g(x; \sigma_I), \tag{B.4}$$

with $\sigma_{\text{LR}} < \sigma_E = \sigma_I$, we can see a very spiky Fourier transform in figure B.6B. The patchy connectivity amplifies certain spatial frequencies (parts of the black line above 0). Still, for $n = 0$ the system is stable. If the patchy long range connections $g_{\text{LR}}(x; K_{\text{LR}}, d_{\text{LR}}, \sigma_{\text{LR}})$ are removed, all Fourier modes are stable again (cyan curve in figure B.6B).

If we use a rate model with two populations including excitatory neurons and inhibitory cells (c.f. equations 4.23 and 4.24), we can make a similar observation. Figure B.7 depicts the strongest amplified spatial frequency for different recurrent connection strengths. In other words, the Fourier modes with the eigenvalue with the largest real part of the Jacobian matrices $\boldsymbol{A}_\epsilon(n)$ from equation 4.31 are color coded. Runaway excitation is shown in white. Even if local coupling has the form of an inverse Mexican hat (figure B.7B), patchy projections amplify some spatial frequencies. Next, we are going to present simulation results from a two-dimensional spiking neural network with patchy projections.

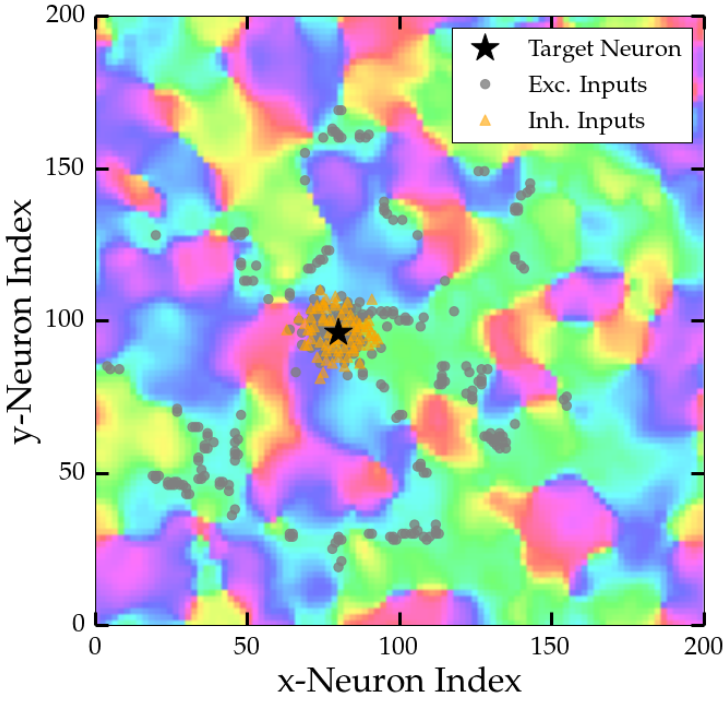

Figure B.8: Orientation map created with an approach by Macke et al. (2009). The pre-synaptic inputs of one target neuron (black star) are depicted as grey dots (excitatory) and orange triangles (inhibitory).

B.3.2 Network Simulations

We simulated a network of 200×200 excitatory adaptive exponential integrate and fire neurons arranged on a regular grid and 10,000 randomly placed inhibitory neurons. Each neuron received $K_A = 100$ afferent inputs, $K_I = 75$ local inhibitory inputs, $K_E = 100$ local excitatory inputs, and $K_{\mathrm{LR}} = 200$ patchy excitatory long range connections.

Each neuron is characterized by a particular preferred orientation according to the orientation map that is depicted in figure B.8. We created the artificial orientation map based on an approach by Macke et al. (2009). We convolved a two-dimensional 200×200 image of Gaussian white noise with a filter $f(x, y)$ consisting of the difference of two Gaussians:

$$f(x,y) = \frac{\alpha_1}{2\pi\,\sigma_1^2} \exp\left(-\frac{x^2+y^2}{2\sigma_1^2}\right) - \frac{\alpha_2}{2\pi\,\sigma_2^2} \exp\left(-\frac{x^2+y^2}{2\sigma_2^2}\right), \tag{B.5}$$

with $\alpha_1 = \alpha_2 = 1$, $\sigma_1 = 5$, and $\sigma_2 = 10$.

Moreover, local connections were sampled as for the simulations in chapter 5 using equation 5.1 with a balanced profile $\sigma_E = \sigma_I = 6\,\mathrm{px}$. Patchy long range connections were sampled independent of distance except that we did not allow projections further

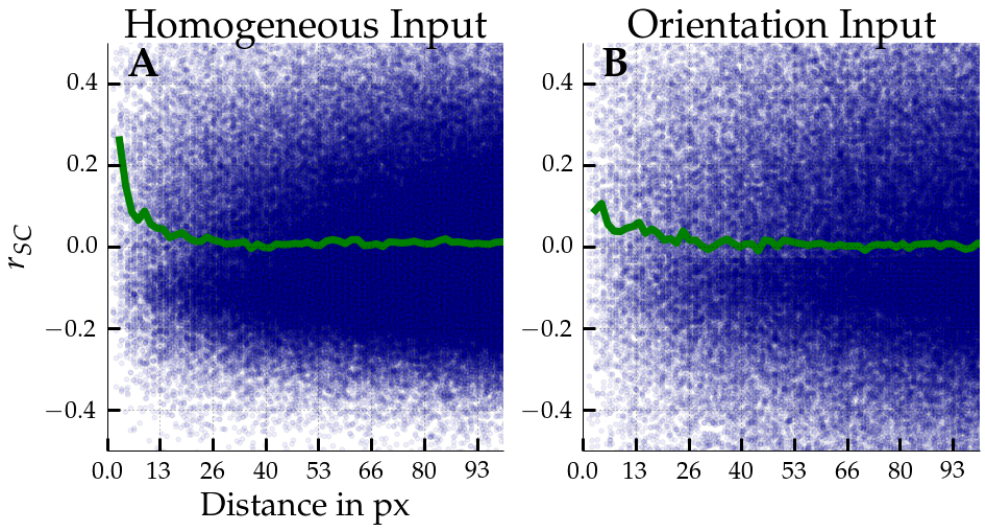

Figure B.9: Noise correlations as a function of distance in a network with patchy excitatory long range connections. Blue dots represent r_{SC} of individual neuron pairs computed over a time window of 1 second. Green depicts the empirical average for bins of 1 px. (A) Noise correlations for homogeneous input calculated over 50 trials and (B) shared variability for orientation tuned input averaged across 6 evenly spaced orientations $\{-89°, -59°, -29°, 1°, 31°, 61°\}$ with 30 trials each.

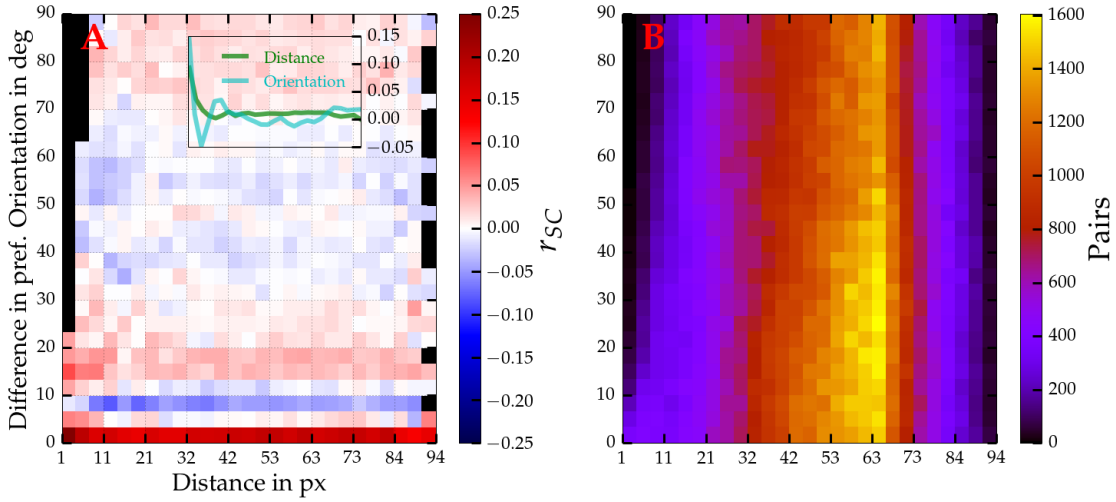

Figure B.10: Left (A): Average noise correlations as a function of distance and orientation difference between cell pairs in a network with patchy excitatory long range connections. In black excluded bins with less than 50 pairs. Noise correlations are computed from 50 trials of 1 second with homogeneous stimulation. Inset shows averages along each axis, in green noise correlations as a function of distance (c.f. figure B.9A) and in cyan as a function of preferred orientation difference. Right (B): Corresponding number of cell pairs in each bin.

than 80 pixels. Connection probability was defined only over the difference in preferred orientation between the pre- and post-synaptic cell j and i:

$$p_{\mathrm{LR}}(i,j) = \frac{1}{\sqrt{2\pi}\sigma_{\mathrm{PO}}} \exp\left(-\frac{{\Delta s_{\mathrm{PO}i,j}}^2}{2\sigma_{\mathrm{PO}}^2}\right), \tag{B.6}$$

where σ_{PO} is the width of the Gaussian over the preferred orientation space and $\Delta s_{\mathrm{PO}i,j}$ is the difference between the preferred orientations of the pre-synaptic neuron j and post-synaptic neuron i. We assumed very targeted, orientation specific connections with $\sigma_{\mathrm{PO}} = 1.5°$. A realization of patchy long range connections for one post-synaptic cell is depicted in figure B.8. All other parameters were chosen as in table A.1 except $\bar{g}_{EE} = 0.25\,\mathrm{nS}$ and $\bar{g}_{IE} = 0.8\,\mathrm{nS}$.

We computed the spike count correlation coefficient as a function of distance between cell pairs for 50 trials lasting 1 second, as depicted in figure B.9A. We observed a decline of average shared variability (green) with distance (A). In contrast to Mexican hat coupling, we did not obtain a sinusoidal modulation but a seemingly exponential decay. Furthermore, if we used tuned input instead of homogeneous stimulation (B), the magnitude was decreased, but the shape of the curve remained the same.

We further discriminated cell pairs not only by distance but also by difference in their preferred orientation. Figure B.10A shows the average r_{SC} as a function of cell distance as well as orientation difference. Strongest noise correlations were observed for nearby cells with similar orientation preference. In contrast to the distance dependency, one can discern a sinusoidal modulation of noise correlations with difference in orientation for rather similar preferred orientations (cyan curve in inset). In addition, we identified that shared variability is positive among distant cells of very similar and very opposite preferred orientations. Similar observations were made by Smith and Kohn (2008) in the visual cortex of monkeys. They reported that noise correlations were positive and stronger among distant cells with negative (close to -1) or positive (close to 1) signal correlations than among distant neurons with signal correlations around 0.

List of Figures

List of Tables

List of Abbreviations

ACG	Auto-Correlogram
AEIF	Adaptive Exponential Integrate and Fire neuron
AI	Asynchronous Irregular activity
AMPA	α-Amino-3-Hydroxyl-5-Methyl-4-Isoxazole-Propionate Acid
API	Application Programming Interface
CCG	Cross-Correlogram
CV	Coefficient of Variation
EPSP	Excitatory Post-Synaptic Potential
GABA	γ-Aminobutyric Acid
GUI	Graphical User-Interface
HDF5	Hierarchical Data Format 5
HH	Hodgkin-Huxley neuron
HPC	High-Performance Computing
IG	Infragranular layers
LGN	Lateral Geniculate Nucleus
LIF	Leaky Integrate and Fire neuron
LOLE	Locally Optimal Linear Estimator
IPSP	Inhibitory Post-Synaptic Potential
ISI	Inter-Spike Interval
MT	Middle Temporal visual area
MEA	Multi-Electrode Array
MSE	Mean Squared Error
NMDA	N-Methyl-D-Aspartic Acid
OOP	Object Oriented Programming
PyPI	Python Package Index
RBF	Radial Basis Function
SEM	Standard Error of the Mean
SFA	Spike Frequency Adaptation
SG	Supragranular layers
SGE	Sun Grid Engine
SHA	Secure Hash Algorithm
SVR	Support Vector Regression
TCP	Transmission Control Protocol
V1	Primary Visual Cortex
V2	Secondary Visual Cortex

List of Symbols

Symbol	Name	Unit
$\boldsymbol{A}$	Jacobian matrix	unitless
a	Sub-threshold adaptation parameter	S
b	Spike frequency adaptation strength	A or unitless
$C_{r_{SC}}$	Fit for initial decay of noise correlations	unitless
d	Distance between cells	px (pixels)
$\bar{g}$	Maximum conductance, i.e. connection strength	S
I	Membrane current	A or unitless
I_{LOLE}	Fisher information approximated by linear estimator	deg^{-2}
I_{shuff}	Information after shuffling trials	deg^{-2}
I_{diag}	Information based on a decoder ignoring correlations	deg^{-2}
J	Coupling strength	unitless
j	Rescaled coupling strength independent of network size	unitless
N	Number of neurons	unitless
q	Fraction of excitatory neurons $N_E = q\,N$, $N_I = (1-q)\,N$	unitless
$\boldsymbol{r}$	Spike response vector, $\boldsymbol{r} = (r_1, r_2, ..., r_N)^T$	unitless
r_{CCG}	Noise correlation coefficient based on correlogram	unitless
r_{SC}	Spike count correlation coefficient	unitless
s	Input stimulus	deg or unitless
s_{PO}	Preferred orientation of a neuron	deg
V	Membrane voltage	V or unitless
w	Adaptation current	A or unitless
ϵ	Finite size factor $\epsilon = 1/\sqrt{N}$ or $\epsilon = (N\,\eta(N))^{-1}$	unitless
η	Network scaling function, default is $\eta(N) = 1/\sqrt{N}$	unitless
κ	Connectivity kernel	unitless
$\lambda_{r_{SC}}$	Exponential distance decay constant of correlation fit	px
μ	Current in the mean-field limit	unitless
ν	Instantaneous firing rate	Hz or unitless
$\nu_{\text{Aff, max}}$	Maximum afferent input firing rate	Hz
σ	Spread or width of Gaussian connection probability	px or unitless
τ	Network time constant	unitless
τ_w	Adaptation time constant	Hz or unitless
ω	Effective coupling strength	unitless
$\omega_{r_{SC}}$	Spatial frequency of correlation fit	px^{-1}
ω_s	Spatial frequency of 1D input stimulus	unitless

Bibliography

Abbott, L. F. and P. Dayan (1999). "The effect of correlated variability on the accuracy of a population code." In: *Neural Comput.* 11.1, pp. 91–101.

Aertsen, A. M., G. L. Gerstein, M. K. Habib, and G. Palm (1989). "Dynamics of neuronal firing correlation: Modulation of effective connectivity". In: *J. Neurophysiol.* 61.5, pp. 900–917.

Alted, F., I. Vilata, and Others (2002). *PyTables: Hierarchical Datasets in Python*. URL: `http://www.pytables.org/`.

Antolík, J. and A. P. Davison (2013). "Integrated workflows for spiking neuronal network simulations." In: *Front. Neuroinform.* 7.December.

Arieli, A., D Shoham, R Hildesheim, and A Grinvald (1995). "Coherent spatiotemporal patterns of ongoing activity revealed by real-time optical imaging coupled with single-unit recording in the cat visual cortex." In: *J. Neurophysiol.* 73.5, pp. 2072–2093.

Augustin, M., J. Ladenbauer, and K. Obermayer (2013). "How Adaptation Shapes Spike Rate Oscillations in Recurrent Neuronal Networks". In: *Front. Comput. Neurosci.* 7.9.

Averbeck, B. B. and D. Lee (2004). "Coding and transmission of information by neural ensembles". In: *Trends Neurosci.* 27.4, pp. 225–230.

Averbeck, B. B. and D. Lee (2006). "Effects of noise correlations on information encoding and decoding." In: *J. Neurophysiol.* 95.6, pp. 3633–3644.

Averbeck, B. B., P. E. Latham, and A. Pouget (2006). "Neural correlations, population coding and computation." In: *Nat. Rev. Neurosci.* 7.5, pp. 358–66.

Bachmair, C. A. and E. Schöll (2014). "Nonlocal control of pulse propagation in excitable media". In: *Eur. Phys. J. B* 87.276.

Bair, W., E. Zohary, and W. T. Newsome (2001). "Correlated firing in macaque visual area MT: Time scales and relationship to behavior." In: *J. Neurosci.* 21.5, pp. 1676–97.

Bavoil, L., S. P. Callahan, P. J. Crossno, J. Freire, C. E. Scheidegger, T Silva, and H. T. Vo (2005). "VisTrails : Enabling Interactive Multiple-View Visualizations". In: *Proc. IEEE Vis.* Dx, pp. 135–142.

Beaulieu, C. and M. Colonnier (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. 337–44.

Beaulieu, C., Z. F. Kisvárday, P. Somogyi, M. Cynader, and A. Cowey (1992). "Quantitative distribution of GABA-immunopositive and -immunonegative neurons and synapses in the monkey striate cortex (area 17)". In: *Cereb. Cortex* 2.Area 17, pp. 295–309.

Berkes, P., F. Wood, and J. Pillow (2008). "Characterizing neural dependencies with copula models". In: *Adv. Neural Inf. Process. Syst. 21*. Ed. by D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou. Curran Associates, Inc., pp. 129–136.

Bertschinger, N. and T. Natschläger (2004). "Real-Time Computation at the Edge of Chaos in Recurrent Neural Networks". In: *Neural Comput.* 16.7, pp. 1413–1436.

Blumenfeld, B., D. Bibitchkov, and M. Tsodyks (2006). "Neural network model of the primary visual cortex: From functional architecture to lateral connectivity and back". In: *J. Comput. Neurosci.* 20.2, pp. 219–41.

Braitenberg, V. and A. Schüz (1991). *Anatomy of the Cortex: Statistics and Geometry*. 1st ed. Berlin, Heidelberg: Springer.

Bressloff, P. C. (2012). "Spatiotemporal dynamics of continuum neural fields". In: *J. Phys. A Math. Theor.* 45.3.

Brette, R. and W. Gerstner (2005). "Adaptive exponential integrate-and-fire model as an effective description of neuronal activity." In: *J. Neurophysiol.* 94.5, pp. 3637–3642.

Bringuier, V. (1999). "Horizontal propagation of visual activity in the synaptic integration field of area 17 neurons". In: *Science (80-.).* 283.5402, pp. 695–699.

Brunel, N. (2000). "Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons." In: *J. Comput. Neurosci.* 8.3, pp. 183–208.

Buzas, P., U. T. Eysel, P. Adorjan, and Z. F. Kisvarday (2001). "Axonal topography of cortical basket cells in relation to orientation, direction, and ocular dominance maps". In: *J. Comp. Neurol.* 437.3, pp. 259–285.

Carnevale, N. T. and M. L. Hines (2006). *The NEURON Book*. New York, NY, USA: Cambridge University Press.

Carreira-Perpiñán, M. A. and G. J. Goodhill (2004). "Influence of lateral connections on the structure of cortical maps". In: *J. Neurophysiol.* 92.5, pp. 2947–2959.

Chater, N., J. B. Tenenbaum, and A. Yuille (2006). "Probabilistic models of cognition: where next?" In: *Trends Cogn. Sci.* 10.7, pp. 292–293.

Chelaru, M. I. and V. Dragoi (2008). "Efficient coding in heterogeneous neuronal populations." In: *Proc. Natl. Acad. Sci.* 105.42, pp. 16344–9.

Chelaru, M. I. and V. Dragoi (2014). "Negative correlations in visual cortical networks". In: *Cereb. Cortex* 26.1, pp. 246–256.

Ch'ng, Y. H. and R. C. Reid (2010). "Cellular imaging of visual cortex reveals the spatial and functional organization of spontaneous activity". In: *Front. Integr. Neurosci.* 4.August.

Christodoulou, C. and T. Clarkson (1995). "A review on the stochastic firing behaviour of real neurons and how it can be modelled". In: *From Nat. to Artif. Neural Comput.*

SE - 31. Ed. by J. Mira and F. Sandoval. Vol. 930. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 223–230.

Coelho, L. P. (2013). *Why Python is Better than Matlab for Scientific Software*. URL: `https://metarabbit.wordpress.com/2013/10/18/why-python-is-better-than-matlab-for-scientific-software/` (visited on Feb. 17, 2016).

Cohen, M. R. and A. Kohn (2011). "Measuring and interpreting neuronal correlations." In: *Nat. Neurosci.* 14.7, pp. 811–9.

Cohen, M. R. and J. H. R. Maunsell (2009). "Attention improves performance primarily by reducing interneuronal correlations". In: *Nat Neurosci* 12.12, pp. 1594–1600.

Constantinidis, C. and P. S. Goldman-Rakic (2002). "Correlated discharges among putative pyramidal neurons and interneurons in the primate prefrontal cortex". In: *J. Neurophysiol.* 88.6, pp. 3487–3497.

Coombes, S. (2005). "Waves, bumps, and patterns in neural field theories". In: *Biol. Cybern.* 93.2, pp. 91–108.

Dahlem, M. A. and T. M. Isele (2013). "Transient localized wave patterns and their application to migraine". In: *J. Math. Neurosci.* 3.1.

Dale, H. (1935). "Pharmacology and nerve-endings (Walter Ernest Dixon memorial lecture)". In: *Proc. R. Soc. Med.* 28, pp. 319–332.

Das, A. and C. D. Gilbert (1995). "Long-range horizontal connections and their role in cortical reorganization revealed by optical recording of cat primary visual cortex". In: *Nature* 375.6534, pp. 780–784.

Davison, A. P. (2008). "PyNN: a common interface for neuronal network simulators". In: *Front. Neuroinform.* 2.January.

Davison, A. P. (2012). "Automated capture of experiment context for easier reproducibility in computational research". In: *Comput. Sci. Eng.* 14.4, pp. 48–56.

Dayan, P. and L. F. Abbott (2005). *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. The MIT Press.

Dehghani, N., N. G. Hatsopoulos, Z. D. Haga, R. A. Parker, B. Greger, E. Halgren, S. S. Cash, and A. Destexhe (2012). "Avalanche analysis from multielectrode ensemble recordings in cat, monkey, and human cerebral cortex during wakefulness and sleep". In: *Front. Physiol.* 3.

Deneve, S., P. E. Latham, and A. Pouget (1999). "Reading population codes: a neural implementation of ideal observers". In: *Nat. Neurosci.* 2.8, pp. 740–745.

Doiron, B., B. Lindner, A. Longtin, L. Maler, and J. Bastian (2004). "Oscillatory activity in electrosensory neurons increases with the spatial correlation of the stochastic input stimulus". In: *Phys. Rev. Lett.* 93.4.

Dorn, J. D. and D. L. Ringach (2003). "Estimating membrane voltage correlations from extracellular spike trains". In: *J. Neurophysiol.* 89.4, pp. 2271–8.

Doya, K., S. Ishii, A. Pouget, and R. P. N. Rao (2007). *Bayesian Brain: Probabilistic Approaches to Neural Coding*. Cambridge, MA: MIT Press.

Ecker, A. S., P. Berens, G. A. Keliris, M. Bethge, N. K. Logothetis, and A. S. Tolias (2010). "Decorrelated neuronal firing in cortical microcircuits". In: *Science* 327.5965, pp. 584–7.

Ecker, A. S., P. Berens, R. J. Cotton, M. Subramaniyan, G. H. Denfield, C. R. Cadwell, S. M. Smirnakis, M. Bethge, and A. S. Tolias (2014). "State dependence of noise correlations in macaque primary visual cortex". In: *Neuron* 82.1, pp. 235–48.

Ernst, U. A., K. R. Pawelzik, C. Sahar-Pikielny, and M. V. Tsodyks (2001). "Intracortical origin of visual maps". In: *Nat. Neurosci.* 4.4, pp. 431–6.

Eyherabide, H. G. and I. Samengo (2013). "When and why noise correlations are important in neural decoding". In: *J. Neurosci.* 33.45, pp. 17921–17936.

Fejtl, M., A. Stett, W. Nisch, K. Boven, and A Möller (2006). "On micro-electrode array revival: Its development, sophistication of recording, and stimulation". In: *Adv. Netw. Electrophysiol. Using Multi-Electrode Arrays*. Ed. by M. Taketani. Boston, MA: Springer US, pp. 24–37.

FitzHugh, R. (1961). "Impulses and physiological states in theoretical models of nerve membrane". In: *Biophys. J.* 1.6, pp. 445–466.

Fortin, F.-A., F.-M. De Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné (2012). "DEAP: Evolutionary algorithms made easy". In: *J. Mach. Learn. Res.* 13, pp. 2171–2175.

Fourcaud-Trocme, N., D. Hansel, C. V. Vreeswijk, and N. Brunel (2003). "How spike generation mechanisms determine the neuronal response to fluctuating inputs". In: *J. Neurosci.* 23.37, pp. 11628–11640.

Franke, F., R. Pröpper, H. Alle, P. Meier, J. R. P. Geiger, K. Obermayer, and M. H. J. Munk (2015). "Spike sorting of synchronous spikes from local neuron ensembles". In: *J. Neurophysiol.* 114.4, pp. 2535–2549.

Freiwald, W. A., H. Stemman, A Wannig, A. K. Kreiter, U. G. Hofmann, M. D. Hills, G. T. A. Kovacs, D. T. Kewley, J. M. Bower, A. Etzold, S. D. Wilke, and C. W. Eurich (2002). "Stimulus representation in rat primary visual cortex: Multi-electrode recordings and estimation theory". In: *Neurocomputing* 44-46, pp. 407–416.

Gabbiani, F. and S. J. Cox (2010). "Chapter 26: Population codes". In: *Mathematics for Neuroscientists*. London: Academic Press, pp. 367–379.

Ganmor, E., R. Segev, and E. Schneidman (2011). "Sparse low-order interaction network underlies a highly correlated and learnable neural population code". In: *Proc. Natl. Acad. Sci.* 108.23, pp. 9679–84.

Gawne, T. J., T. W. Kjaer, J. a. Hertz, and B. J. Richmond (1996). "Adjacent visual cortical complex cells share about 20% of their stimulus-related information". In: *Cereb. Cortex* 6.3, pp. 482–489.

Gewaltig, M.-O. and R. Cannon (2014). "Current practice in software development for computational neuroscience and how to improve it". In: *PLoS Comput. Biol.* 10.1.

Gilbert, C. D. and T. N. Wiesel (1989). "Columnar specificity of intrinsic horizontal and corticocortical connections in cat visual cortex". In: *J. Neurosci.* 9.7, pp. 2432–2442.

Golledge, H. D. R., S. Panzeri, F. Zheng, G. Pola, J. W. Scannell, D. V. Giannikopoulos, R. J. Mason, M. J. Tovée, and M. P. Young (2003). "Correlations, feature-binding and population coding in primary visual cortex". In: *Neuroreport* 14.7, pp. 1045–50.

González-Burgos, G., G. Barrionuevo, and D. A. Lewis (2000). "Horizontal synaptic connections in monkey prefrontal cortex: An in vitro electrophysiological study". In: *Cereb. Cortex* 10.1, pp. 82–92.

Goodman, D. and R. Brette (2008). "BRIAN: A simulator for spiking neural networks in Python". In: *Front. Neuroinform.* 2.November.

Graf, A. B. A., A. Kohn, M. Jazayeri, and J. A. Movshon (2011). "Decoding the activity of neuronal populations in macaque primary visual cortex". In: *Nat Neurosci* 14.2, pp. 239–245.

Gu, Y., S. Liu, C. R. Fetsch, Y. Yang, S. Fok, A. Sunkara, G. C. DeAngelis, and D. E. Angelaki (2011). "Perceptual learning reduces interneuronal correlations in macaque visual cortex". In: *Neuron* 71.4, pp. 750–761.

Guo, Y. and C. Chow (2005). "Existence and stability of standing pulses in neural networks: I. Existence". In: *SIAM J. Appl. Dyn. Syst.* 4.2, pp. 217–248.

Gutnisky, D. A. and V. Dragoi (2008). "Adaptive coding of visual information in neural populations". In: *Nature* 452.7184, pp. 220–4.

Hagberg, A. and E. Meron (1994). "Pattern formation in non-gradient reaction-diffusion systems: The effects of front bifurcations". In: *Nonlinearity* 7.3, pp. 805–835.

Han, F., N. Caporale, and Y. Dan (2008). "Reverberation of recent visual experience in spontaneous cortical waves". In: *Neuron* 60.2, pp. 321–327.

Hansel, D. and H. Sompolinsky (1998). "Modeling feature selectivity in local cortical circuits". In: *Methods Neuronal Model. From Ions to Networks*, pp. 499–567.

Hansen, B. J., M. I. Chelaru, and V. Dragoi (2012). "Correlated variability in laminar cortical circuits". In: *Neuron* 76.3, pp. 590–602.

Harris, K. D., D. A. Henze, J. Csicsvari, H. Hirase, and G. Buzsáki (2000). "Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements". In: *J. Neurophysiol.* 84.1, pp. 401–414.

Hellwig, B. (2000). "A quantitative analysis of the local connectivity between pyramidal neurons in layers 2/3 of the rat visual cortex". In: *Biol. Cybern.* 82.2, pp. 111–121.

Hellwig, B., A. Schuz, and A. Aertsen (1994). "Synapses on axon collaterals of pyramidal cells are spaced at random intervals: A Golgi study in the mouse cerebral cortex". In: *Biol. Cybern.* 71.1, pp. 1–12.

Hodgkin, A. L. and A. F. Huxley (1952). "A quantitative description of membrane current and its application to conduction and excitation in nerve". In: *J. Physiol.* 117.4, pp. 500–544.

Hold-Geoffroy, Y., O. Gagnon, and M. Parizeau (2014). "Once you SCOOP, no need to fork". In: *Proc. 2014 Annu. Conf. Extrem. Sci. Eng. Discov. Environ.* XSEDE '14. New York, NY, USA: ACM, 60:1–60:8.

Huang, X. and S. G. Lisberger (2009). "Noise correlations in cortical area MT and their potential impact on trial-by-trial variation in the direction and speed of smooth-pursuit eye movements". In: *J. Neurophysiol.* 101.6, pp. 3012–3030.

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.

Ince, D. C., L. Hatton, and J. Graham-Cumming (2012). "The case for open computer programs". In: *Nature* 482.7386, pp. 485–8.

Izhikevich, E. M. (2007). "Chapter 8: Simple models". In: *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting*. Cambridge, Mass.: MIT Press, pp. 267–323.

Jia, X. and A. Kohn (2011). "Gamma rhythms in the brain". In: *PLoS Biol.* 9.4.

Jolivet, R., R. Kobayashi, A. Rauch, R. Naud, S. Shinomoto, and W. Gerstner (2008). "A benchmark test for a quantitative assessment of simple neuron models". In: *J. Neurosci. Methods* 169.2, pp. 417–424.

Jones, E., T. Oliphant, P. Peterson, and Others (2001). *SciPy: Open Source Scientific Tools for Python*. URL: `http://www.scipy.org/`.

Kandel, E. R., J. H. Schwartz, and T. M. Jessell (2000). "Part V: Perception". In: *Principles of Neural Science*. 4th. McGraw-Hill Medical, pp. 411–652.

Kang, K., M. Shelley, and H. Sompolinsky (2003). "Mexican hats and pinwheels in visual cortex". In: *Proc. Natl. Acad. Sci.* 100.5, pp. 2848–53.

Kanitscheider, I., R. Coen-Cagli, and A. Pouget (2015). "Origin of information-limiting noise correlations". In: *Proc. Natl. Acad. Sci.* 112.50, pp. 6973–82.

Karayannis, T., I. Huerta-Ocampo, and M. Capogna (2007). "GABAergic and pyramidal neurons of deep cortical layers directly receive and differently integrate callosal input". In: *Cereb. Cortex* 17.5, pp. 1213–1226.

Kazama, H. and R. I. Wilson (2009). "Origins of correlated activity in an olfactory circuit". In: *Nat Neurosci* 12.9, pp. 1136–1144.

Keane, A. and P. Gong (2015). "Propagating waves can explain irregular neural dynamics". In: *J. Neurosci.* 35.4, pp. 1591–1605.

Kenet, T., D. Bibitchkov, M. Tsodyks, A. Grinvald, and A. Arieli (2003). "Spontaneously emerging cortical representations of visual attributes". In: *Nature* 425.6961, pp. 954–956.

Kilpatrick, Z. P. (2013). "Interareal coupling reduces encoding variability in multi-area models of spatial working memory". In: *Front. Comput. Neurosci.* 7.July.

Knill, D. C. and A. Pouget (2004). "The Bayesian brain: The role of uncertainty in neural coding and computation". In: *Trends Neurosci.* 27.12, pp. 712–719.

Koepke, H. (2010). *Why Python Rocks for Research*. URL: `https://www.stat.washington.edu/~hoytak/blog/whypython.html` (visited on Feb. 17, 2016).

Kohn, A. and M. a. Smith (2005). "Stimulus dependence of neuronal correlation in primary visual cortex of the macaque". In: *J. Neurosci.* 25.14, pp. 3661–73.

Körding, K. and D. Wolpert (2004). "Bayesian integration in sensorimotor learning". In: *Nature* 427.6971, pp. 244–247.

Kouvaris, N. E., T. Isele, A. S. Mikhailov, and E. Schöll (2014). "Propagation failure of excitation waves on trees and random networks". In: *Europhys. Lett.* 106.6.

Kumar, A., S. Schrader, A. Aertsen, and S. Rotter (2008). "The high-conductance state of cortical networks". In: *Neural Comput.* 20.1, pp. 1–43.

Laing, C. R. and A. Longtin (2001). "Noise-induced stabilization of bumps in systems with long-range spatial coupling". In: *Phys. D Nonlinear Phenom.* 160.3-4, pp. 149–172.

Laing, C. R., W. C. Troy, B. Gutkin, and G. B. Ermentrout (2003). "Multiple bumps in a neuronal model of working memory". In: 63.1, pp. 62–97.

Langton, C. G. (1990). "Computation at the edge of chaos: Phase transitions and emergent computation". In: *Phys. D Nonlinear Phenom.* 42.1-3, pp. 12–37.

Latham, P. E. and Y. Roudi (2011). "Role of correlations in population coding". In: *arXiv*.

Latham, P. E., S. Deneve, and A. Pouget (2003). "Optimal computation with attractor networks". In: *J. Physiol. Paris* 97.4-6, pp. 683–694.

Lee, D., N. L. Port, W. Kruse, and A. P. Georgopoulos (1998). "Variability and correlated noise in the discharge of neurons in motor and parietal areas of the primate cortex". In: *J. Neurosci.* 18.3, pp. 1161–1170.

Levy, R. B. and A. D. Reyes (2012). "Spatial profile of excitatory and inhibitory synaptic connectivity in mouse primary auditory cortex". In: *J. Neurosci.* 32.16, pp. 5609–19.

Lewicki, M. S. (1998). "A review of methods for spike sorting: the detection and classification of neural action potentials". In: *Netw. Comput. Neural Syst.* 9.4, pp. 53–78.

Lin, I.-C., D. Xing, and R. Shapley (2012). "Integrate-and-fire vs Poisson models of LGN input to V1 cortex: Noisier inputs reduce orientation selectivity". In: *J. Comput. Neurosci.* 33.3, pp. 559–572.

Lindner, B., B. Doiron, and A. Longtin (2005). "Theory of oscillatory firing induced by spatially correlated noise and delayed inhibitory feedback". In: *Phys. Rev. E* 72.6.

Litwin-Kumar, A. and B. Doiron (2012). "Slow dynamics and high variability in balanced cortical networks with clustered connections". In: *Nat. Neurosci.* 15.11, pp. 1498–1505.

Liu, J. and W. T. Newsome (2006). "Local field potential in cortical area MT: Stimulus tuning and behavioral correlations". In: *J. Neurosci.* 26.30, pp. 7779–7790.

Ma, W. J., V. Navalpakkam, J. M. Beck, R. van den Berg, and A. Pouget (2011). "Behavior and neural basis of near-optimal visual search". In: *Nat Neurosci* 14.6, pp. 783–790.

Macke, J., S. Gerwinn, L. White, M. Kaschube, and M. Bethge (2009). "Bayesian estimation of orientation preference maps". In: *Adv. Neural Inf. Process. Syst. 22*. Ed. by Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta. Vol. 3. Curran Associates, Inc., pp. 1195–1203.

Mariño, J., J. Schummers, D. C. Lyon, L. Schwabe, O. Beck, P. Wiesing, K. Obermayer, and M. Sur (2005). "Invariant computations in local cortical networks with balanced excitation and inhibition". In: *Nat. Neurosci.* 8.2, pp. 194–201.

Markov, N. T., P. Misery, A. Falchier, C. Lamy, J. Vezoli, R. Quilodran, M. A. Gariel, P. Giroud, M. Ercsey-Ravasz, L. J. Pilaz, C. Huissoud, P. Barone, C. Dehay, Z. Toroczkai, D. C. Van Essen, H. Kennedy, and K. Knoblauch (2011). "Weight consistency specifies regularities of macaque cortical networks". In: *Cereb. Cortex* 21.6, pp. 1254–1272.

Marre, O., V. Botella-Soler, K. D. Simmons, T. Mora, G. Tkačik, and M. J. Berry (2015). "High accuracy decoding of dynamical motion from a large retinal population". In: *PLoS Comput. Biol.* 11.7.

Martin, K. A. and S. Schröder (2013). "Functional heterogeneity in neighboring neurons of cat primary visual cortex in response to both artificial and natural stimuli". In: *J Neurosci* 33.17, pp. 7325–7344.

Mason, A., A. Nicoll, and K. Stratford (1991). "Synaptic transmission between individual pyramidal neurons of the rat visual cortex in vitro". In: *J. Neurosci.* 11.January, pp. 72–84.

McKinney, W. (2011). "pandas: A foundational Python library for data analysis and statistics". In: *Python High Perform. Sci. Comput.*

Mejias, J. F. and A. Longtin (2012). "Optimal heterogeneity for coding in spiking neural networks". In: *Phys. Rev. Lett.* 108.22.

Merzky, A., O. Weidner, and S. Jha (2015). "SAGA: A standardized access layer to heterogeneous distributed computing infrastructure". In: *SoftwareX* 1, pp. 3–8.

Mitchell, J. F., K. A. Sundberg, and J. H. Reynolds (2009). "Spatial attention decorrelates intrinsic activity fluctuations in macaque area V4". In: *Neuron* 63.6, pp. 879–888.

Mochol, G., A. Hermoso-Mendizabal, S. Sakata, K. D. Harris, and J. de la Rocha (2015). "Stochastic transitions into silence cause noise correlations in cortical circuits". In: *Proc. Natl. Acad. Sci.* 112.11.

Montijn, J. S., M. Vinck, and C. M. A. Pennartz (2014). "Population coding in mouse visual cortex: Response reliability and dissociability of stimulus tuning and noise correlation". In: *Front. Comput. Neurosci.* 8.June.

Mora, T., S. Deny, and O. Marre (2015). "Dynamical criticality in the collective activity of a population of retinal neurons". In: *Phys. Rev. Lett.* 114.7.

Moreno-Bote, R., J. Beck, I. Kanitscheider, X. Pitkow, P. Latham, and A. Pouget (2014). "Information-limiting correlations". In: *Nat. Neurosci.* 17.10, pp. 1410–1417.

Muenchen, R. A. (2012). *The Popularity of Data Analysis Software*. URL: `http://r4stats.com/articles/popularity/` (visited on Feb. 17, 2016).

Muller, L., A. Reynaud, F. Chavane, and A. Destexhe (2014). "The stimulus-evoked population response in visual cortex of awake monkey is a propagating wave". In: *Nat. Commun.* 5.

Myme, C. I. O., K. Sugino, G. G. Turrigiano, and S. B. Nelson (2003). "The NMDA-to-AMPA ratio at synapses onto layer 2/3 pyramidal neurons is conserved across prefrontal and visual cortices". In: *J. Neurophysiol.* 90.2, pp. 771–779.

Nagumo, J., S. Arimoto, and S. Yoshizawa (1962). "An active pulse transmission line simulating nerve axon". In: *Proc. IRE* 50, pp. 2061–2070.

Naud, R., N. Marcille, C. Clopath, and W. Gerstner (2008). "Firing patterns in the adaptive exponential integrate-and-fire model". In: *Biol. Cybern.* 99.4-5, pp. 335–347.

Nirenberg, S. and P. E. Latham (2003). "Decoding neuronal spike trains: How important are correlations?" In: *Proc. Natl. Acad. Sci.* 100.12, pp. 7348–53.

Nowak, L. G., R. Azouz, M. V. Sanchez-Vives, C. M. Gray, and D. A. McCormick (2003). "Electrophysiological classes of cat primary visual cortical neurons in vivo as revealed by quantitative analyses". In: *J. Neurophysiol.* 89.3, pp. 1541–1566.

Ohiorhenuan, I. E., F. Mechler, K. P. Purpura, A. M. Schmid, Q. Hu, and J. D. Victor (2010). "Sparse coding and high-order correlations in fine-scale cortical networks". In: *Nature* 466.7306, pp. 617–21.

Oliveira, S. C. de, A. Thiele, and K. P. Hoffmann (1997). "Synchronization of neuronal activity during stimulus expectation in a direction discrimination task". In: *J. Neurosci.* 17.23, pp. 9248–9260.

Onken, A., V. Dragoi, and K. Obermayer (2012). "A maximum entropy test for evaluating higher-order correlations in spike counts". In: *PLoS Comput. Biol.* 8.6.

Paradiso, M. A. (1988). "A theory for the use of visual orientation information which exploits the columnar structure of striate cortex". In: *Biol. Cybern.* 58.1, pp. 35–49.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011). "Scikit-learn: Machine learning in Python". In: *J. Mach. Learn. Res.* 12, pp. 2825–2830.

Perez, F. and B. E. Granger (2007). "IPython: A system for interactive scientific computing". In: *Comput. Sci. Eng.* 9.3, pp. 21–29.

Pinto, D. J. and G. B. Ermentrout (2001). "Spatially structured activity in synaptically coupled neuronal networks: II. Lateral inhibition and standing pulses". In: *SIAM J. Appl. Math.* 62.1, pp. 226–243.

Poirazi, P. (2009). "Information processing in single cells and small networks: Insights from compartmental models". In: *AIP Conf. Proc.* 1108.February, pp. 158–167.

Ponce-Alvarez, A., A. Thiele, T. D. Albright, G. R. Stoner, and G. Deco (2013). "Stimulus-dependent variability and noise correlations in cortical MT neurons". In: *Proc. Natl. Acad. Sci. U. S. A.* 110.32, pp. 13162–7.

Postnov, D. E., F. Müller, R. B. Schuppner, and L. Schimansky-Geier (2009). "Dynamical structures in binary media of potassium-driven neurons". In: *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.* 80.3.

Postnov, D. E., D. D. Postnov, and L. Schimansky-Geier (2012). "Self-terminating wave patterns and self-organized pacemakers in a phenomenological model of spreading depression". In: *Brain Res.* 1434, pp. 200–211.

Potjans, T. C. and M. Diesmann (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.

Pouget, A., P. Dayan, and R. Zemel (2000). "Information processing with population codes". In: *Nat. Rev. Neurosci.* 1.2, pp. 125–32.

Reich, D. S. (2001). "Independent and redundant information in nearby cortical neurons". In: *Science (80-.).* 294.5551, pp. 2566–2568.

Reimann, M., C. Anastassiou, R. Perin, S. L. Hill, H. Markram, and C. Koch (2013). "A biophysically detailed model of neocortical local field potentials predicts the critical role of active membrane currents". In: *Neuron* 79.2, pp. 375–390.

Renart, A., J. de la Rocha, P. Bartho, L. Hollender, N. Parga, A. Reyes, and K. D. Harris (2010). "The asynchronous state in cortical circuits". In: *Science* 327.5965, pp. 587–90.

Rieke, F., D. Warland, R. de Ruyter van Steveninck, and W. Bialek (1997). "Chapter 3: Quantifying information transmission". In: *Spikes: Exploring the Neural Code*. MIT Press, pp. 101–188.

Rocha, J. de la, B. Doiron, E. Shea-Brown, K. Josić, and A. Reyes (2007). "Correlation between neural spike trains increases with firing rate". In: *Nature* 448.7155, pp. 802–806.

Rosenbaum, R. and B. Doiron (2014). "Balanced Networks of Spiking Neurons with Spatially Dependent Recurrent Connections". In: *Phys. Rev. X* 4.2.

Rossello, J. L., V. Canals, A. Oliver, and A. Morro (2014). "Stochastic spiking neural networks at the edge of chaos". In: *Neural Networks (IJCNN), 2014 Int. Jt. Conf.* Pp. 2399–2406.

Roxin, A., N. Brunel, and D. Hansel (2006). "Rate models with delays and the dynamics of large networks of spiking neurons". In: *Prog. Theor. Phys. Suppl.* 161, pp. 68–85.

Rudolph-Lilith, M., M. Dubois, and A. Destexhe (2012). "Analytical integrate-and-fire neuron models with conductance-based dynamics and realistic postsynaptic potential time course for event-driven simulation strategies". In: *Neural Comput.* 24.6, pp. 1426–61.

Rullen, R. van and S. J. Thorpe (2001). "Rate coding versus temporal order coding: what the retinal ganglion cells tell the visual cortex". In: *Neural Comput.* 13.6, pp. 1255–83.

Sanchez-Vives, M. V., L. G. Nowak, and D. A. McCormick (2000). "Cellular mechanisms of long-lasting adaptation in visual cortical neurons in vitro". In: *J. Neurosci.* 20.11, pp. 4286–4299.

Schmolesky, M. (2007). *Webvision: The Primary Visual Cortex*. Ed. by H. Kolb, E. Fernandez, and R. Nelson. Salt Lake City. URL: `http://webvision.med.utah.edu/book/part-ix-psychophysics-of-vision/the-primary-visual-cortex/` (visited on Mar. 20, 2016).

Schneidman, E., M. J. Berry, R. Segev, and W. Bialek (2006). "Weak pairwise correlations imply strongly correlated network states in a neural population". In: *Nature* 440.7087, pp. 1007–12.

Schulz, D. P. a., M. Sahani, and M. Carandini (2015). "Five key factors determining pairwise correlations in visual cortex". In: *J. Neurophysiol.* 114.2, pp. 1022–1033.

Seriès, P., P. E. Latham, and A. Pouget (2004). "Tuning curve sharpening for orientation selectivity: coding efficiency and the impact of correlations". In: *Nat. Neurosci.* 7.10, pp. 1129–35.

Seung, H. S. and H Sompolinsky (1993). "Simple models for reading neuronal population codes". In: *Proc. Natl. Acad. Sci.* 90.22, pp. 10749–10753.

Shadlen, M. N. and W. T. Newsome (1998). "The variable discharge of cortical neurons: implications for connectivity, computation, and information coding". In: *J. Neurosci.* 18.10, pp. 3870–96.

Shamir, M. and H. Sompolinsky (2004). "Nonlinear population codes". In: *Neural Comput.* 16.6, pp. 1105–1136.

Shannon, C. E. (1948). "A mathematical theory of communication". In: *Bell Syst. Tech. J.* 27.July 1928, pp. 379–423.

Shlens, J., G. D. Field, J. L. Gauthier, M. I. Grivich, D. Petrusca, A. Sher, A. M. Litke, and E. J. Chichilnisky (2006). "The structure of multi-neuron firing patterns in primate retina". In: *J. Neurosci.* 26.32, pp. 8254–66.

Smith, M. A. and A. Kohn (2008). "Spatial and temporal scales of neuronal correlation in primary visual cortex". In: *J. Neurosci.* 28.48, pp. 12591–603.

Smith, M. A., X. Jia, A. Zandvakili, and A. Kohn (2013). "Laminar dependence of neuronal correlations in visual cortex". In: *J. Neurophysiol.* 109.4, pp. 940–7.

Solomon, S. S., S. C. Chen, J. W. Morley, and S. G. Solomon (2014). "Local and global correlations between neurons in the middle temporal area of primate visual cortex". In: *Cereb. Cortex* September, pp. 3182–3196.

Sompolinsky, H., H. Yoon, K. Kang, and M. Shamir (2001). "Population coding in neuronal systems with correlated noise". In: *Phys. Rev. E* 64.5.

Sonnenschein, B., M. A. Zaks, A. B. Neiman, and L. Schimansky-Geier (2013). "Excitable elements controlled by noise and network structure". In: *Eur. Phys. J. Spec. Top.* 222.10, pp. 2517–2529.

Stepanyants, A., J. A. Hirsch, L. M. Martinez, Z. F. Kisvárday, A. S. Ferecskó, and D. B. Chklovskii (2008). "Local potential connectivity in cat primary visual cortex". In: *Cereb. Cortex* 18.1, pp. 13–28.

Stepanyants, A., L. M. Martinez, A. S. Ferecskó, and Z. F. Kisvárday (2009). "The fractions of short- and long-range connections in the visual cortex". In: *Proc. Natl. Acad. Sci.* 106.9, pp. 3555–60.

Stevens, J.-L. R., M. Elver, and J. A. Bednar (2013). "An automated and reproducible workflow for running and analyzing neural simulations using Lancet and IPython Notebook". In: *Front. Neuroinform.* 7.December 2013.

Stevens, J.-L. R., M. Elver, J. A. Bednar, and P. Rudiger (2016). *Lancet Online Documentation*. URL: http://ioam.github.io/lancet/ (visited on Feb. 13, 2016).

Stevenson, I. H. and K. P. Kording (2011). "How advances in neural recording affect data analysis". In: *Nat Neurosci* 14.2, pp. 139–142.

Stimberg, M., K. Wimmer, R. Martin, L. Schwabe, J. Mariño, J. Schummers, D. C. Lyon, M. Sur, and K. Obermayer (2009). "The operating regime of local computations in primary visual cortex". In: *Cereb. Cortex* 19.9, pp. 2166–80.

Stimberg, M., D. F. M. Goodman, V. Benichoux, and R. Brette (2014). "Equation-oriented specification of neural models for simulations." In: *Front. Neuroinform.* 8.February.

Stockton, D. B. and F. Santamaria (2015). "NeuroManager: A workflow analysis based simulation management engine for computational neuroscience". In: *Front. Neuroinform.* 9.October.

Stodden, V. (2011). "Trust your science? Open your data and code". In: *AMSTATNEWS* 409, pp. 21–22.

Swindale, N. V. (1998). "Orientation tuning curves: Empirical description and estimation of parameters". In: *Biol. Cybern.* 78.1, pp. 45–56.

Tang, A., D. Jackson, J. Hobbs, W. Chen, J. L. Smith, H. Patel, A. Prieto, D. Petrusca, M. I. Grivich, A. Sher, P. Hottowy, W. Dabrowski, A. M. Litke, and J. M. Beggs (2008). "A maximum entropy model applied to spatial and temporal correlations from cortical networks in vitro". In: *J. Neurosci.* 28.2, pp. 505–18.

The HDF Group (1996). *Hierarchical Data Format, version 5*. URL: https://www.hdfgroup.org/HDF5/.

The Numba Development Team (2015). *Numba Version 0.22.1*. URL: http://numba.pydata.org.

Topalidou, M., A. Leblois, T. Boraud, and N. P. Rougier (2015). "A long journey into reproducible computational neuroscience". In: *Front. Comput. Neurosci.* 9.30.

Touboul, J. and R. Brette (2008). "Dynamics and bifurcations of the adaptive exponential integrate-and-fire model". In: *Biol. Cybern.* 99.4-5, pp. 319–34.

Toyoizumi, T., K. Aihara, and S. I. Amari (2006). "Fisher information for spike-based population decoding". In: *Phys. Rev. Lett.* 97.9.

Trier, M., S. Thiel, and Others (2016). *GitPython*. URL: https://gitpython.readthedocs.org/ (visited on Mar. 28, 2016).

Vanag, V. K. and I. R. Epstein (2007). "Localized patterns in reaction-diffusion systems". In: *Chaos* 17.3.

Vogels, T. P., K Rajan, and L. F. Abbott (2005). "Neural network dynamics". In: *Annu. Rev. Neurosci.* 28.1, pp. 357–376.

Voges, N. and L. Perrinet (2010). "Phase space analysis of networks based on biologically realistic parameters". In: *J. Physiol. Paris* 104.1-2, pp. 51–60.

Voges, N. and L. Perrinet (2012). "Complex dynamics in recurrent cortical networks based on spatially realistic connectivities". In: *Front. Comput. Neurosci.* 6.July.

Voges, N., A. Schüz, A. Aertsen, and S. Rotter (2010). "A modeler's view on the spatial structure of intrinsic horizontal connectivity in the neocortex". In: *Prog. Neurobiol.* 92.3, pp. 277–92.

Vreeswijk, C. van and H. Sompolinsky (1996). "Chaos in neuronal networks with balanced excitatory and inhibitory activity". In: *Science* 274.5293, pp. 1724–6.

Vreeswijk, C. van and H. Sompolinsky (1998). "Chaotic balanced state in a model of cortical circuits". In: *Neural Comput.* 10.6, pp. 1321–71.

Walt, S. van der, S. C. Colbert, and G. Varoquaux (2011). "The NumPy array: A structure for efficient numerical computation". In: *Comput. Sci. Eng.* 13.2.

Wang, X.-J. (2010). "Neurophysiological and computational principles of cortical rhythms in cognition". In: *Physiol. Rev.* 90.3, pp. 1195–1268.

Wilke, S. D. and C. W. Eurich (2002). "On the functional role of noise correlations in the nervous system". In: *Neurocomputing* 44-46, pp. 1023–1028.

Wilson, H. R. and J. D. Cowan (1972). "Excitatory and inhibitory interactions in localized populations of model neurons". In: *Biophys. J.* 12.1, pp. 1–24.

Wimmer, K., D. Q. Nykamp, C. Constantinidis, and A. Compte (2014). "Bump attractor dynamics in prefrontal cortex explains behavioral precision in spatial working memory". In: *Nat. Neurosci.* 17.3, pp. 431–9.

Womble, M. D. and H. C. Moises (1992). "Muscarinic inhibition of M-current and a potassium leak conductance in neurones of the rat basolateral amygdala". In: *J. Physiol.* 457, pp. 93–114.

Xing, D., C.-I. Yeh, S. Burns, and R. M. Shapley (2012). "Laminar analysis of visually evoked activity in the primary visual cortex". In: *Proc. Natl. Acad. Sci.* 109.34, pp. 13871–13876.

Yarrow, S., E. Challis, and P. Series (2012). "Fisher and Shannon information in finite neural populations". In: *Neural Comput.* 24, pp. 1740–80.

Yger, P., S. El Boustani, A. Destexhe, and Y. Frégnac (2011). "Topologically invariant macroscopic statistics in balanced networks of conductance-based integrate-and-fire neurons". In: *J. Comput. Neurosci.* 31.2, pp. 229–45.

Yu, S., H. Yang, H. Nakahara, G. S. Santos, D. Nikolic, and D. Plenz (2011). "Higher-order interactions characterized in cortical activity". In: *J. Neurosci.* 31.48, pp. 17514–17526.

Zohary, E, M. N. Shadlen, and W. T. Newsome (1994). "Correlated neuronal discharge rate and its implications for psychophysical performance". In: *Nature* 370, pp. 140–144.