CECÍLIA ROMARO

ESTUDOS EM MODELOS DE REDES CORTICAIS
STUDIES IN CORTICAL NETWORK MODELS

Ribeirão Preto
2020

CECÍLIA ROMARO

ESTUDOS EM MODELOS DE REDES CORTICAIS
STUDIES IN CORTICAL NETWORK MODELS

Tese apresentada à Faculdade de Filosofica Ciências e Letras de Ribeirão Preto da Universidade de São Paulo para obtenção do Título de Doutor em Ciências.

Ribeirão Preto
2020

CECÍLIA ROMARO

ESTUDOS EM MODELOS DE REDES CORTICAIS
STUDIES IN CORTICAL NETWORK MODELS

Tese apresentada à Faculdade de Filosofica Ciências e Letras de Ribeirão Preto da Universidade de São Paulo para obtenção do Título de Doutor em Ciências.

Área de Concentração:
Física Aplicada à Medicina e Biologia

Orientador:
Antônio Carlos Roque da Silva Filho

Co-orientador:
José Roberto Castilho Piqueira

Ribeirão Preto
2020

Romaro, Cecília
Studies in Cortical Network Models. Ribeirão Preto, 2020.
178 p. : il. ; 30 cm

Tese de Doutorado apresentada à Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto / USP. Área de concentração: Física Aplicada à Medicina e Biologia.
Orientador: Roque da Silva Filho, Antônio Carlos
Co-orientador: Castilho Piqueira, José Roberto.

1. Somatosensory cortex model. 2. Mean-field potential. 3. Metastability behavior. 4. Rescaling neuron network. 5. Phase transition.

I dedicate this work to all people who, in some way, act for a better world.

ACKNOWLEDGMENTS

The author is the recipient of PhD scholarships from the Brazilian Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES, grant number 88882.378774/2019-01).

The author is thankful to NeuroMat for the scientific and stimulating environment; Professor Piqueira, who had a contribution in completion of this work beyond his imagination; her friend, “academic and life assistant” Angela, for always telling her the truth; her friend and “husband” Julia Maria, who stood by her side through parties and tears; and the author’s sister, “big little” Cinthia, for actually being her sister.

“I’m late! I’m late! For a very important date! No time to say hello, goodbye! I’m late! I’m late! I’m late!”

--White Rabbit (Alice in Wonderland)

RESUMO

Esta tese apresenta-se em cinco capítulos-artigos resolvendo de forma simples problemas e perguntas não triviais na neurosciencia computacional.

O primeiro capítulo apresenta uma reimplementação do modelo de Potjans-Diesmann (PD) para a microcircuitaria cortical local e uma técnica de redimensionamento do número de neurônios do modelo capaz de manter as probabilidades de conexões e o comportamento da atividade da rede mesmo quando redimensionada para 1% do tamanho original.

O segundo capítulo, baseado no potencial de campo médio, explica formalmente o método de redimensionamento e apresenta um novo método de corrigir e compensar a atividade dos neuronios da borda em redes com extensão espacial sem introduzir conexões toroidais e/ou oscilaçôes.

O terceiro capítulo introduz extensão espacial ao modelo PD, soluciona o problema de borda e estuda a resolução espacial (topográfica) da atividade da rede como um reflexo da resolução estrutural.

O quarto capítulo, baseado em transição de fase e metaestabilidade, inovadoramente estuda o falso estado estacionário e o tempo de duração da atividade em redes que não recebem entrada externa forçada capaz de mantê-las ativas.

O quinto capítulo contém a caracterização da rede de neurônios do córtex somatossensorial primário do rato em termos do levantamento estatístico dos parâmetros. Em seguida, um modelo do córtex somatossensorial utilizando o neurônio estocástico de Galves-Löcherbach (GL) é construido com base nos parâmetros levantados. No fim do capítulo, apresenta-se um método para substituição de neurônios determinísticos do tipo integra-e-dispara com vazamento por neurônios GL em modelos de redes de neurônios.

Palavras-Chave – modelagem do córtex somatossensorial, potencial de campo médio, metaestabilidade, redimensionamento de rede neuronal, transição de fase.

ABSTRACT

This thesis consists of five chapter-articles, each proposing simple solutions to nontrivial questions and problems in computational neuroscience.

The first chapter presents a reimplementation of the Potjans-Diesmann (PD) model of the local cortical microcircuitry, and a rescaling method for the number of neurons in the model that is capable of maintaining both the connection probabilities and the behavior of the network activity even when rescaled to 1 % of original size.

The second chapter, based on mean field potential, formally explains the scaling method and presents a new method to correct and compensate for the activity of boundary neurons in networks with spatial extension without introducing toroidal connections and/or oscillations.

The third chapter introduces spatial extension to the PD model, solves the boundary problem, and studies the spatial (topographic) resolution of the network activity as a consequence of the structural resolution.

The fourth chapter, based on phase transition and meta-stability, innovatively studies the false steady state and activity lifetime in networks that do not receive forced external input to keep them active.

The fifth chapter contains a characterization of the primary somatosensory cortex network of the rat in terms of a statistical survey of the parameters. It also presents a model of the somatosensory cortex using the stochastic Galves-Löcherbach (GL) neuron, which was constructed based on the somatotopic parameters raised. At the end of the chapter, a method for replacing deterministic leaky integrate-and-fire neurons by GL neurons in neural network models is presented.

Keywords – somatosensory cortex model, mean-field potential, metastability behavior,rescaling neuron network, phase transition.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

INTRODUCTION

Understanding the brain is challenging, given both its complex mechanisms and inaccessibility. Every year new pathways, channels, proteins, and other mechanisms are discovered, but how these components interact at the system level remains a mystery (Markram et al., 2015; Yin and Wang, 2016; Lisman and Raghavachari, 2006; Diekman et al., 2013; Antunes et al., 2016). A strong mathematical research line appeared in the early XX century to explain the neuron behavior. The Hodgkin-Huxley (HH) (Hodgkin et al., 1952) and leaky integrate-and-fire (LIF) (Lapicque, 1907) models are some of the most classic and oldest mathematical models of neurons, and pools of these neurons were put together in attempts to study their collective behavior. However, it was only with the advancement of computational resources that the study of neuron networks was able to develop further (De Schutter, 2008). *In silico* simulation studies started to provide a quantitative framework for integrating disparate pieces of evidence from *in vivo* and *in vitro* experiments into coherent predictive models that can be used to investigate brain function.

Biologically detailed computational modeling became an excellent research tool to explain brain mechanisms and to propose supporting experiments *in vivo* (Markram et al., 2015; Mazza et al., 2004; Bower and Beeman, 2012). At the same time, simplified neurons became a study subject to understand the dynamics of neuron networks (Galves and Löcherbach, 2013; Potjans and Diesmann, 2014a; Brunel, 2000). These techniques are being largely adopted by research groups worldwide such as the Allen Institute for Brain Science (AIBS) (Jones et al., 2009) (`https://alleninstitute.org/what-we-do/brain-science/`), the Human Brain Project (HBP) (Markram, 2012) (`https://www.humanbrainproject.eu/en/`), the NEST initiative (Gewaltig and Diesmann, 2007) (`https://nest-initiative.org/`), and the Research, Innovation and Dissemination Center for Neuromathematics (NeuroMat) (`https://neuromat.numec.prp.usp.br/`). In this scenario, three neural simulators stand out today:

The NEURON simulator (Carnevale and Hines, 2006) (`https://neuron.yale.edu/neuron/`) allows modeling of networks of neurons with detailed neuronal morphologies, and different ion channels and neurotransmitter receptors. Developed at Yale University, NEURON is the most widely used simulator of its type, both in university labs and large research groups, including AIBS and HBP. It is an excellent software for the construction and simulation of biologically-detailed neuron networks, although it requires a large amount of machine

processing time.

The NEST simulator (Morrison et al., 2005; Gewaltig and Diesmann, 2007) (`https://nest-simulator.org/`) allows modeling of large networks of simplified neurons. It does not allow the direct introduction of morphology, ion channels and neurotransmitter receptors, but allows parallel and distributed simulation inboard build up in order to save machine time.

The Brian simulator (Goodman and Brette, 2009) (`https://briansimulator.org/`) has inboard differential equation solvers and other facilities to simulate spiking neural networks. Brian is an easy install simulator, given it is a Python package. It is not as fast as NEST, but it is more plastic. Neither Brian nor NEST allow the direct introduction of morphology or physical characteristics of neurons.

Although great advancement has been achieved, it seems that computational resources still pose a limiting factor to simulate, process, and understand detailed models of large networks (Markram et al., 2015; Towns et al., 2014). This limit is being pushed further and further away as computational resources become more powerful and cheaper, and this both supports and inspires the work done in this thesis.

This thesis starts by presenting a re-implementation of the Potjans-Diesmann (PD) model (Potjans and Diesmann, 2014a) in NetPyNE/NEURON (Dura-Bernal et al., 2019b; Carnevale and Hines, 2006) (`http://www.netpyne.org/`), a high-level Python interface to the NEURON simulator. The PD model, originally implemented in NEST (Gewaltig and Diesmann, 2007), is an eight-cell population model which reproduces the neuronal connectivity under a 1 mm^2 area of cortical surface in full scale (the model has 77,169 neurons). The PD model generates spontaneous activity with population-specific firing rates similar to those observed experimentally (de Kock and Sakmann, 2009; Sakata and Harris, 2009; Swadlow, 1989). The re-implementation of the PD model in NetPyNE/NEURON allows the use of more detailed neuron models with multicompartmental morphologies and multiple realistic biophysical ion channels. Additionally, some analyses can be provided automatically.

Due to the required memory and processing power to re-implement the PD model in NEURON (Carnevale and Hines, 2006), a rescaling method was developed to scale down the model to variable levels, down to the lowest level of 1%. The method allows to decrease the number of connections to 10^{-4} of the original number still maintaining characteristics of network activity behavior compatible with experimental data: mean population firing rate, synchrony and irregularity (Romaro et al., 2018, 2020a).

The re-scaling method introduced here can be applied not only to the PD model but to random neuron network models in general, e.g. the Brunel model (Brunel, 2000). This method

is further explored in chapter 2, and both a detailed mathematical explanation based on mean field potential and a list of the method limitations are presented there. Based on the re-scaling method, networks with spatial extension are considered in chapter 2 and a boundary solution method is introduced. The boundary solution method is able to correct and compensate the lack (or excess) of connections at the boundary in order to reestablish the correct activity of boundary-neurons and core-neurons.

The boundary solution method was developed because it was needed in the construction of a model of the primary somatossensory cortex exhibiting somatotopy, the main subject of Chapter 3.

Somatotopy is the topographic organization of somatic sensory pathways – touch and proprioception – in the primary somatosensory cortex (S1, located in the post-central gyrus) (Bear et al., 2020). That is, the mapping of neighboring areas of the skin to neighboring areas in the cortex, e.g. two sensors in the skin close together activate in the cortex two groups of neurons that are close together. Many models have been proposed to reproduce and explain the mechanisms and effects of the topographic organization in S1, including the re-organization properties following some lesion (Ramachandran, 1998; Bear et al., 2020). A property of the somatotopic maps is the direct proportionality between the cortical area allocated to represent a body surface and the sensitivity of this surface. In other words, the higher the sensitivity of a body area, the higher the cortical area allocated to process it (Bear et al., 2020). However, none of these models explored the influence of the connectome on the resolution of the topographic map in S1.

In chapter 3, adjustments are introduced in the PD model in order to adapt it to describe the somatosensory cortex and allow a study of the role of network structure and activity on the topographic organization and spatial resolution of inputs. These adjustments include the introduction of spatial locations for the neurons and consideration of distance-dependent connectivity to integrate anatomical and physiological data. This leads to a model that accounts for the topographic pattern of connections. The objective is to shed some light on the question of how the parameters of the topographic connections relate to the local activity patterns in specific cortical layers.

Work on cortical network models raised other questions that were studied during the period of this doctorate and resulted in chapters 4 and 5.

One of these questions is how to characterize the lifetime of active states in a network in the absence of external input. In the context of the Galves-Löcherbach (GL) model (Galves and Löcherbach, 2013), André has recently proved (André, 2019b) that for finite one-dimensional

lattices of continuous GL neurons with hard threshold rate function (Ferrari et al., 2018), in the sub-critical regime where the leak rate γ is smaller than a critical value γ_c the lifetime of the active state is finite but the exit time from this state is exponentially distributed. In chapter 4, a computational version of the finite lattice model considered by André is described together with extensions of the model to two- and three-dimensional square lattices and linear and sigmoidal rate functions. The simulation studies confirm the rigorous results obtained by André and provide evidence that they also hold in the extended settings considered (Romaro et al., 2019).

Another question is related to the common practice of using data from a mix of different cortical areas and animals in the construction of a model for a specific cortical area from a given animal. For example, the PD model uses data from the primary somatossensory, motor and visual cortices of rat and rabbit (Potjans and Diesmann, 2014a). To address this problem, in chapter 5 a statistical survey of parameters from the connectome of the primary somatosensory cortex of the rat is made. This allowed the construction of a model of the primary somatosensory cortex of the rat with GL neurons.

References are presented at the end of each chapter, except for introduction references, which are presented in the References section. Chapter 1 was developed in collaboration with Fernando Najman and Salvador Dura-Bernal. Chapter 4 was developed in collaboration with Fernando Najman and Morgan Andre.

PART I

1 NETPYNE IMPLEMENTATION AND RESCALING OF THE POTJANS-DIESMANN CORTICAL MICROCIRCUIT MODEL

Abstract

The Potjans-Diesmann cortical microcircuit model is a widely used model originally implemented in NEST. Here, we re-implemented the model using NetPyNE, a high-level Python interface to the NEURON simulator, and reproduced the findings of the original publication. We also implemented a method for rescaling the network size which preserves first and second order statistics, building on existing work on network theory. The new implementation enables using more detailed neuron models with multicompartment morphologies and multiple biophysically realistic channels. This opens the model to new research, including the study of dendritic processing, the influence of individual channel parameters, and generally multiscale interactions in the network. The rescaling method provides flexibility to increase or decrease the network size if required when running these more realistic simulations. Finally, NetPyNE facilitates modifying or extending the model using its declarative language; optimizing model parameters; running efficient large-scale parallelized simulations; and analyzing the model through built-in methods, including local field potential calculation and information flow measures.

Keywords: Somatosensory cortex, Modeling, Microcircuit, Rescaling

1.1 Introduction

The Potjans-Diesmann cortical microcircuit (PDCM) model (Potjans and Diesmann, 2014a) reproduces the cortical network under a 1 mm^2 surface area of early sensory cortex. The model generates spontaneous activity with layer-specific firing rates similar to those observed experimentally (de Kock and Sakmann, 2009; Sakata and Harris, 2009; Swadlow, 1989).

The PDCM model was the first one to reproduce the connectivity structure of the cortical layers with statistical fidelity to the biological data observed experimentally (Thomson et al., 2002; West et al., 2005). This model is broadly used to study the emergence of macroscopic cortical patterns, such as layer specific oscillations (Van Albada et al., 2015; Bos et al., 2016) or effects on cortical functionality resulting from inter-layer or inter-columns communication (Cain et al., 2016; Schwalger et al., 2017; Schmidt et al., 2018). Some examples of use of this model are the study of the influence of the microconnectome on the activity through the network layers (Schuecker et al., 2017), modeling of spatial attention in the visual cortex (Wagatsuma et al., 2013) and modeling the effects of inhibitory connections in contextual visual processing (Lee et al., 2017) and in different cortical microcircuitry regions (Beul and Hilgetag, 2015).

In this work, we converted the PDCM model from NEST to NetPyNE (Dura-Bernal et al., 2019; Lytton et al., 2016) (www.netpyne.org). NetPyNE provides a high-level interface to the NEURON simulator (Carnevale and Hines, 2006) that facilitates the development, parallel simulation and analysis of biological neuronal networks. NetPyNE provides a high-level declarative format that clearly separates the model parameters from the underlying implementation, making the PDCM model easier to understand, share and manipulate. NetPyNE enables efficient parallel simulation of the model with a single function call, and provides a wide array of built-in analysis functions to further explore the model.

Our NetPyNE implementation enables employing more detailed cell models as alternatives to the original leaky integrate-and-fire (LIF) neurons. NetPyNE makes it possible to readily use PDCM model connection topology for more complex simulations by swapping in multi-compartmental neuron models with arbitrarily detailed features: conductance-based channels, more complex synaptic models, (Hines et al., 2004) and reaction-diffusion processes (McDougal et al., 2013; Ranjan et al., 2011; Newton et al., 2018). This allows a new array of possible studies, such as investigating the interaction between network topology and dendritic morphology or channel-specific parameters (Bezaire et al., 2016; Dura-Bernal et al., 2018; Neymotin et al., 2016).

More detailed simulations require considerable additional computational resources. To make these more detailed simulations computationally feasible, it may be necessary to reduce the number of neurons in the network. Given the increasing availability of supercomputing resources (Towns et al., 2014; Sivagnanam et al., 2013), researchers may also wish to switch back and forth across different network sizes (Schwalger et al., 2017; Schmidt et al., 2018; Bezaire et al., 2016). However, rescaling the network to decrease or increase its size while maintaining its dynamical properties is a challenging process. For example, as we reduce the number of neurons we need to increase the number of connections or the synaptic weight to balance the external inputs. However, this can lead to an undesired spiking synchrony and regularity (Brunel, 2000). To address this issue we implemented a rescaling method, adapted from the original model, to resize the number of network neurons, connections and external inputs as well as the synaptic weights, while keeping the matrix of connection probabilities and the proportions of cells per population fixed.

Our implementation is able to generate NEURON-based network models of different sizes with layer-specific average firing rates, synchrony and irregularity features, similar to those in the original PDCM model (Potjans and Diesmann, 2014a). This will allow researchers to modify both the level of detail and size of the PDCM network to adapt it to their computational resources and research objectives.

1.2 Methods

1.2.1 Original NEST PDCM model

The network consists of around 80,000 leaky integrate-and-fire (LIF) (Lapicque, 1907) neurons divided in eight cell populations representing excitatory and inhibitory neurons in cortical layers 2/3, 4, 5 and 6 (these populations will be referred to here by L2e, L2i, L4e, L4i, L5e, L5i, L6e and L6i). External input is provided by thalamic and cortico-cortical afferents.

The network, originally built in NEST (Gewaltig and Diesmann, 2007), specifies fixed numbers of excitatory and inhibitory neurons per layer, fixed number and strength of connections between these neuronal populations and fixed number of external inputs to each cell population. These numbers are based on experimental data (Thomson et al., 2002; West et al., 2005). The connectivity of the model corresponds to the one of a cortical slab under a surface area of 1 mm^2.

1.2.2 NetPyNE implementation of the PDCM model

NetPyNE employs a declarative language to specify the network parameters. Informally, declarative languages allow the user to describe 'what' they want, in contrast to imperative languages which specify 'how' to get there. In NetPyNE, this means the user only needs to provide the biological parameters at the different modeled scales, but is not required to implement all the low-level details. We therefore extracted the model parameters from the original PDCM publication (Potjans and Diesmann, 2014a) and from the NEST source code available at OSB (Potjans and Diesmann, 2014b). More specifically, the NetPyNE model specification required defining the parameters of 8 cell populations, 8 populations of spike generators (NetStims) that served as background inputs, and 68 connectivity rules. Since NetPyNE models require spatial dimensions, even if not explicitly used, we embedded the model into a cylinder of $1470\,\mu\mathrm{m}$ depth and $300\,\mu\mathrm{m}$ diameter, and set the cortical depth range (layer boundaries) for each population based on macaque V1 data (Schmidt et al., 2018). Connectivity rules included the pre- and post-synaptic population, a fixed divergence value, and a weight and delay that followed a parameterized normal distribution. The model parameters were specified programmatically using NetPyNE's high-level declarative language, and could later be explored interactively via command line or NetPyNE's graphical user interface (GUI).

To reproduce the PDCM model, a new NEURON LIF neuron model was required since the built-in LIF models do not allow setting the membrane time constant higher than the synaptic decay time constant. This feature was required to reproduce the original PDCM LIF model. We therefore implemented a new LIF point process neuron model using the NMODL (.mod) language.

The initial membrane potential for each neuron was set randomly from a Gaussian distribution with a mean of 58 mV and a standard deviation of 10 mV, as in the original article. However, we did not consider the initial transient phase of the first 100 ms of network activity in our analysis and took into account only the stationary condition of the network.

As in the original article, we implemented three different conditions in terms of the external inputs to the network (Potjans and Diesmann, 2014a):

1- Poisson and balanced: inputs follow a Poisson distribution and the number of external inputs to each population is balanced to generate a network behavior similar to that observed in biology.

2- Direct current (DC) input and balanced: inputs are replaced with an equivalent DC injection, and are balanced as in case 1.

3- Poisson and unbalanced: inputs follow a Poisson distribution but each population receives the same number of inputs (unbalanced) resulting in non-biological firing rates, including absence of layer 6 excitatory activity.

The source code for the NetPyNE model, including the network Python code and the LIF neuron NMODL (.mod) code, are publicly available from GitHub (`https://github.com/ceciliaromaro/PD_in_NetPyNE`).

1.2.3 Network rescaling

The rescaling method implemented in our model was developed based on previous theoretical work (Van Albada et al., 2015; Vreeswijk and Sompolinsky, 1998) and on the rescaling implementation of the original NEST PDCM model. The rescaling implementation of the original model is available as source code from the Open Source Brain (OSB) platform (Potjans and Diesmann, 2014b), but was not described or addressed in the original article (Potjans and Diesmann, 2014a). Our rescaling implementation was simplified and adapted in order to guarantee the conservation of the first and second order statistics of network activity for all possible rescalings while being easy to implement in NetPyNE/NEURON. It is dependent on a single scaling parameter in the interval [0, 1], which is used to resize the number of network neurons, connections and external inputs as well as the synaptic weights, while keeping the matrix of connection probabilities and the proportions of cells per population fixed.

Since the original model article and source code did not include the rescaling option, we followed the rescaling implementation available in the OSB PDCM model version (Potjans and Diesmann, 2014b). However, since the implementation methods were not described, we had to "scavenge" the source code to obtain the necessary information to understand the rescaling options and theoretical foundations (Van Albada et al., 2015; Vreeswijk and Sompolinsky, 1998). Our approach consisted in running the simulation with different network sizes, adding breakpoints if necessary, in order to characterize all the relevant functions and parameters used for the rescaling mechanism. Their implementation allowed for different ways to rescale the network, most of them resulting in an alteration of the network statistics. We constrained our rescaling implementation to allow only for the specific cases in which the first and second order statistics are preserved. Therefore, we developed a simplified and consolidated rescaling function with a single scaling factor performing the following operations (see also Table 1):

1. Decrease the number of neurons and external inputs per neuron — by multiplying them by the scale factor — while keeping the proportions of cells per population fixed;

2. Decrease the number of connections per population — by multiplying them by the square of the scale factor — while keeping the probabilities of connections between populations unchanged;

3. Increase the synaptic weights — by dividing them by the square root of the scale factor;

4. Provide each cell with an additional DC input current with a value corresponding to the total input lost due to rescaling.

The first three steps maintain the original network proportions across layers, whereas the fourth step maintains the original statistics of network activity across layers. Therefore, this method is able to produce the same layer-specific average firing rates, synchrony and irregularity features in networks of smaller or larger size.

	Full-scale Network	**Resized Network**
Number of neurons	N	kN
Number of external inputs per neuron	I	kI
Probability of connection	p	p
Total connections between two populations	$C_{i,j}$	$k^2 C_{i,j}$
Synaptic weight	W	$\frac{W}{\sqrt{k}}$
Inner input per neuron	$pN_j w f_{j\text{mean}}$	$\sqrt{k} pN_j w f_{j\text{mean}}$
External input per neuron	$IW f_{\text{external}}$	$\sqrt{k} IW f_{\text{external}}$
DC input equivalence	X	$X + (1-\sqrt{k}) pN_j W f_{j\text{mean}}$ $+(1-\sqrt{k}) IW f_{\text{external}}$

Table 1: Rescaling description. Transformation of the parameters of the full-scale network of size N to a rescaled network of size kN. k is the scale factor, $f_{j-\text{mean}}$ is the mean firing rate of the pre-synaptic population and f_{external} is the mean fire rate of the external population.

It is important to point out that to accurately reproduce the layer-specific average firing rates of the original model, it is fundamental to calculate the exact number of synapses and avoid using approximations (Shimoura et al., 2018).

To compare the raster plot of spiking activity across the scaled networks we plotted approximately the same number of neurons as in the original publication, even though the total number of simulated neurons differed. Since the estimation of irregularity and synchrony may depend on the number of neurons included, we decided to always perform these calculations on a sample with the same number of neurons, despite comparing networks of different sizes. We utilized the same bin width (3 ms) as in the original article. The influence of the number of neurons is further assessed in the Results and Discussion sections.

Each population irregularity is estimated using a irregularity metric defined as the coefficient of variation (the estimated standard deviation divided by the mean) of the interspike interval (CV ISI), that is

$$CV = \frac{\frac{1}{N}\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2}}{\frac{1}{N}\sum_{i=1}^{N} x_i},$$

where the sequence x_i are the time intervals between the consecutive spikes of a fixed neuron (ISI). Synchrony per population is estimated as the variance of the spike count histogram normalized by its mean.

All the new model results and analyses were obtained using the NetPyNE tool, except for the synchrony statistic, which was calculated using the equation described in the original paper (Potjans and Diesmann, 2014a).

1.3 Results

1.3.1 Reproduction of Potjans-Diesmann (PDCM) model results

We start showing results from the NetPyNE implementation of the full scalle PDCM model with different sampling sizes and different external input conditions. The purpose is to compare the NetPyNe and NEST implementations and show that they are similar.

Figure 1 shows results from the NetPyNE implementation reproducing the raster plot and firing rate, irregularity and synchrony statistics for the balanced Poisson inputs condition (Figure 6 of the original article (Potjans and Diesmann, 2014a)). Although the results are not identical, due to the random components of the model (see Discussion), the major characteristics of the original model were reproduced: the raster plot included 1862 neurons and showed apparent asynchronous activity (but see below); L2e and L6e exhibited the lowest firing rates with a mean around 1 Hz; L4e fired around 4 Hz, and L5e presented the higher excitatory firing rate, around 7 Hz. As in the original article, the irregularity of all populations was around 0.8, with the lowest irregularity (just under 0.8) was for L5i and L6i. The synchrony measure also closely matched the pattern across populations exhibited in the original model, with L5e showing the highest value, followed by L2e and L4e, and L5i and L6i displaying the lowest values.

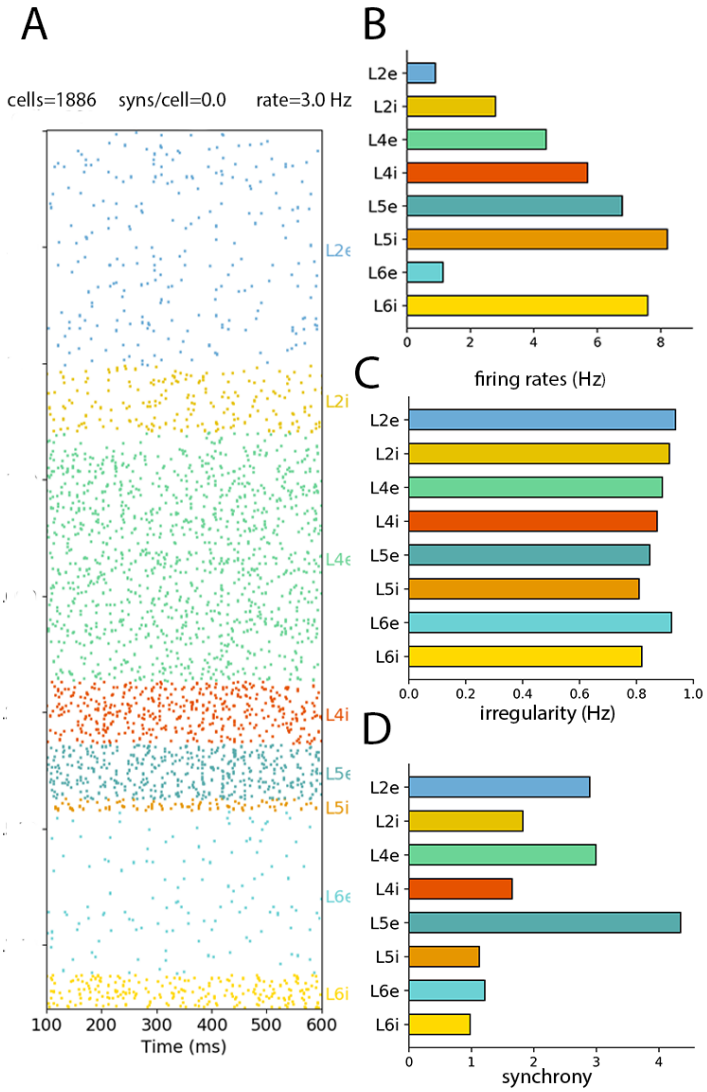


Figure 1: NetPyNE implementation results reproducing Figure 6 of the original article (Potjans and Diesmann, 2014a) (balanced Poisson inputs): (A) Raster plot of the 8 neural populations with 1862 excitatory and inhibitory neurons distributed across layers 2, 4, 5 and 6 for 500 ms. Only 2.3% of the neurons in each populations are shown. (B) Mean firing rates of each cell population over 60 sec. (C) Irregularity per population estimated as the coefficient of variation of the interspike interval (CV ISI) over 60 sec. (D) Synchrony per population estimated as the variance of the spike count histogram normalized by its mean over 5 sec. Statistics in B, C and D were based on calculations with a fixed sample of 8000 neurons as explained in the Methods section.

A comparison of the mean population firing rates of the NetPyNE implementation with the original implementation (taken from Table 6 of the original article, which shows data only for the excitatory populations) is given in Table 2. The mean rates of the excitatory neurons of the NetPyNE implementation fall within the standard deviation ranges of their respective counterparts in the NEST implementation.

Next, we show that the ongoing spiking activity of the network and the corresponding synchrony measure depend on the number of neurons sampled. In Figure 2 we show results of the same full scale NetPyNE implementation of the PDCM model as in Figure 1 but now with a sample of all neurons (77,169) in the network. The synchronous activity is visually obvious and the synchrony measure is strongly changed for all cell populations. On the other hand, the mean firing rate and the irregularity per cell population remained approximately the same.

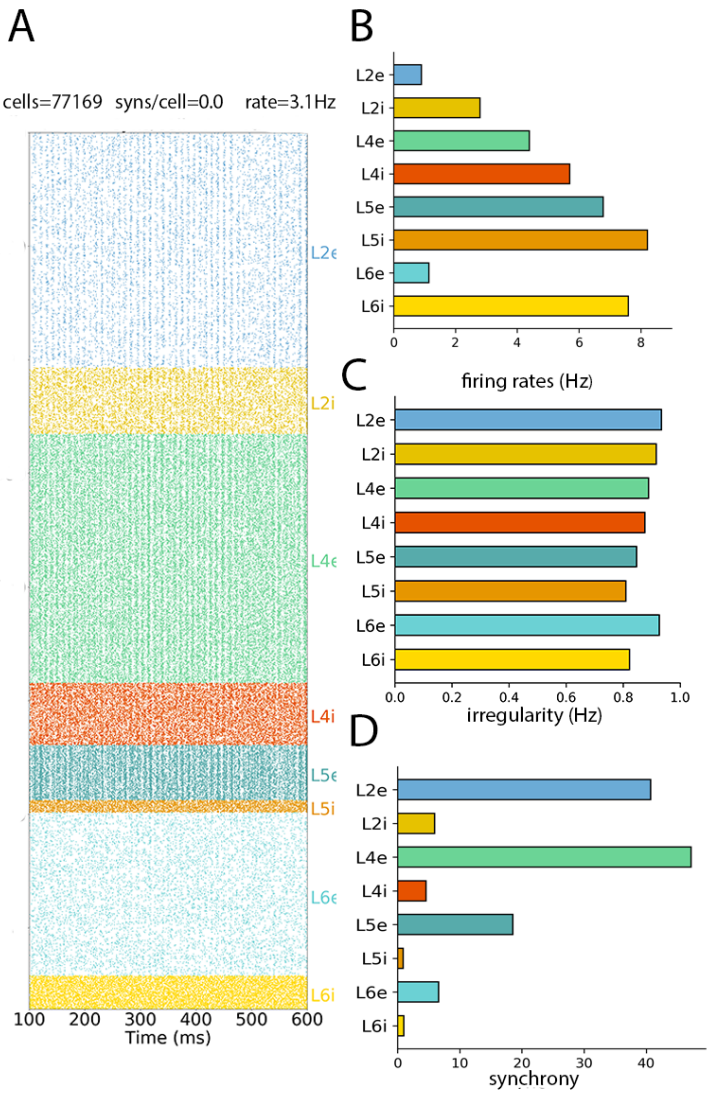


Figure 2: NetPyNE implementation of the PDCM model with full size sampling (balanced Poisson inputs). (A) Spike raster plot of the approximately 80k neurons distributed across layers 2, 4, 5 and 6 for 600ms. (B) Mean firing rates of each cell population over 60 sec. (C) Population irregularities, estimated as the coefficient of variation of the interspike interval, over a 60 sec simulation. (D) Synchrony per population estimated as the variance of the spike count histogram normalized by its mean over 5 sec. Statistics in B, C and D were based on calculations with the full number of neurons in the network.

Figure 3 reproduces the raster plot and mean firing rates for the DC current and unbalanced Poisson input conditions (panels A1, A2, B1 and B2 from Figure 7 in the original article (Potjans and Diesmann, 2014a)). The raster plot was for a sample of 1862 neurons as in Figure 1. In similar fashion to the original article, replacing the balanced Poisson inputs with DC current did not affect the irregular firing displayed in the raster plot nor the population average firing rate properties. However, replacing them with unbalanced Poisson inputs resulted in no activity in L6e and modified the average firing rates across populations.

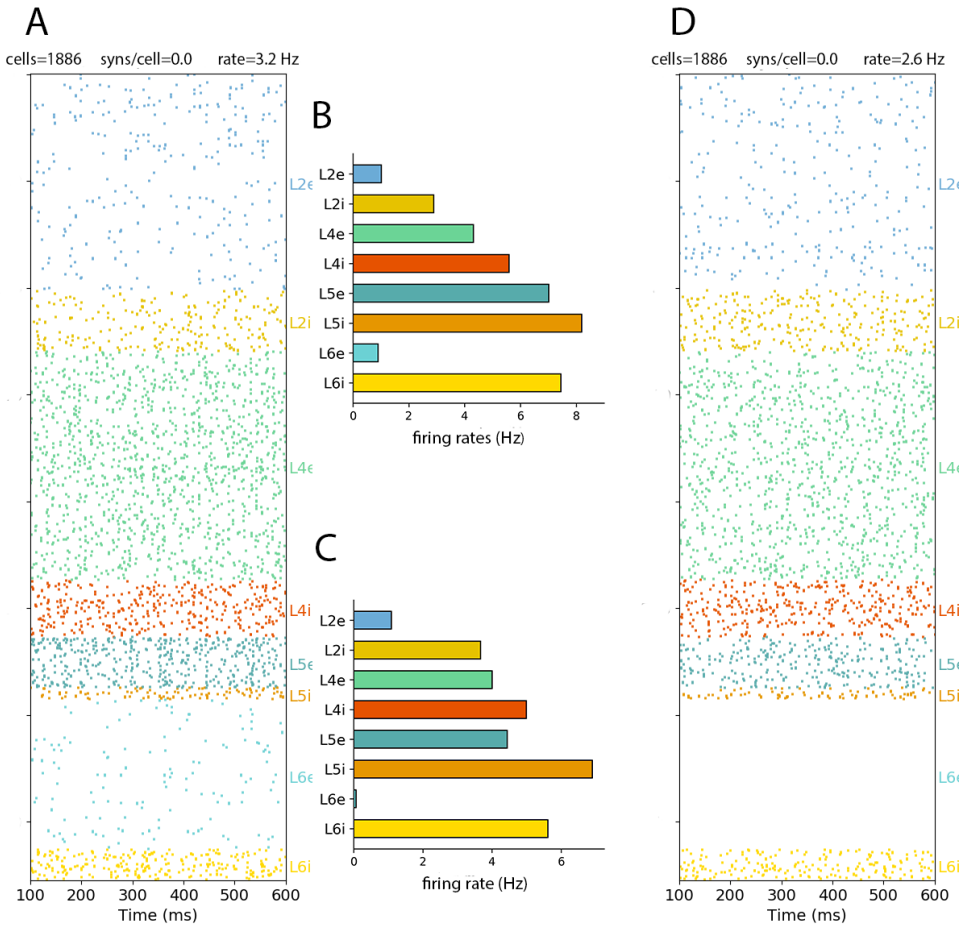


Figure 3: NetPyNE implementation results reproducing Figure 7 of the original article (Potjans and Diesmann, 2014a): (A) Raster plot of the 8 neural populations with 1862 excitatory and inhibitory neurons distributed across layers 2, 4, 5 and 6 for 500 ms with balanced DC inputs. Only 2.3% of the neurons in each populations are shown. (B) Bar chart of single unit firing rates per population over 60 sec with balanced DC inputs. (C) Bar chart of single unit firing rates per population over 60 sec with unbalanced Poisson inputs. (D) Raster plot of the 8 neural populations with 1852 excitatory and inhibitory neurons distributed across layers 2, 4, 5 and 6 for 500 ms with unbalanced Poisson inputs. Only 2.3% of the neurons in each populations are shown. Statistics in B, C and D were based on calculations with a fixed sample of 8000 neurons as explained in the Methods section.

Platform	L2e	L2i	L4e	L4i	L5e	L5i	L6e	L6i
NEST	0.85	-	4.45	-	7.59	-	1.09	-
NetPyNE	0.90	2.80	4.39	5.70	6.79	8.21	1.14	7.60
NEST-100 trials	1.11 ± 0.8	-	4.8 ± 1.1	-	11 ± 6.1	-	0.56 ± 0.9	-

Table 2: Layer-specific firing rates (Hz) for the full version of the PDCM model implemented in NEST and NetPyNE. The third row shows mean and standard deviation firing rates from 100 runs of the NEST implementation with random numbers of external inputs to each layer (see (Potjans and Diesmann, 2014a) for details).

1.3.2 Network rescaling

Now that we have compared the full scale versions of the NetPyNE and NEST implementations with different sampling sizes, we will proceed to compare the rescaled NetPyNE implementations with the full scale NEST implementation.

Figures 4-6 show raster plots and statistics for scaled down NetPyNE versions of the original PDCM network, with either Poisson or DC external inputs. As in the original article (Potjans and Diesmann, 2014a), raster plots show 1862 cells and the statistical measures were calculated using a fixed number of 8000 neurons. The raster plots exhibit similar firing patterns, and mean firing rate, irregularity and synchrony per layer as the full scale raster plot (Figure 1), both when using Poisson inputs (panels A-D of Figures 4, 5 and 6) and DC inputs (panels E-H of Figures 4 and 5). The raster plot and synchrony for the case of 10% rescaling with DC external inputs differed from the results in Figure 1, as they exhibited a visually perceptible synchrony (Figure 6E), and the synchrony values measured (Figure 6H) were considerably higher.

To show again that the sampling size matters, Figure 7 shows the results for the 30% rescaled network but includes all the neurons (23,147) in the raster plot and statistics calculations (compare to Figure 5). In a similar fashion to what was seen in the full scale network simulation (Figure 2), spike synchrony can be observed visually in the raster plots and the population synchrony values are significantly altered.

Extended results for the behavior of the cell populations mean firing rate, irregularity and synchrony as a function of the degree of rescaling and external input type for the NetPyNE implementation of the PDCM model are shown in Figure 8. A more complete set of data is given in Supplementary Tables 4 (mean firing rates, Poisson input), 5 (mean firing rate, DC input), 6 (irregularity, Poisson input), 7 (irregularity, DC input), 8 (synchrony, Poisson input), and 9 (synchrony, DC input). They allow a comparison of the different rescaled NetPyNE implementations of the PDCM model with the original NEST implementation. They also allow a comparison of the NetPyNE implementations among themselves.

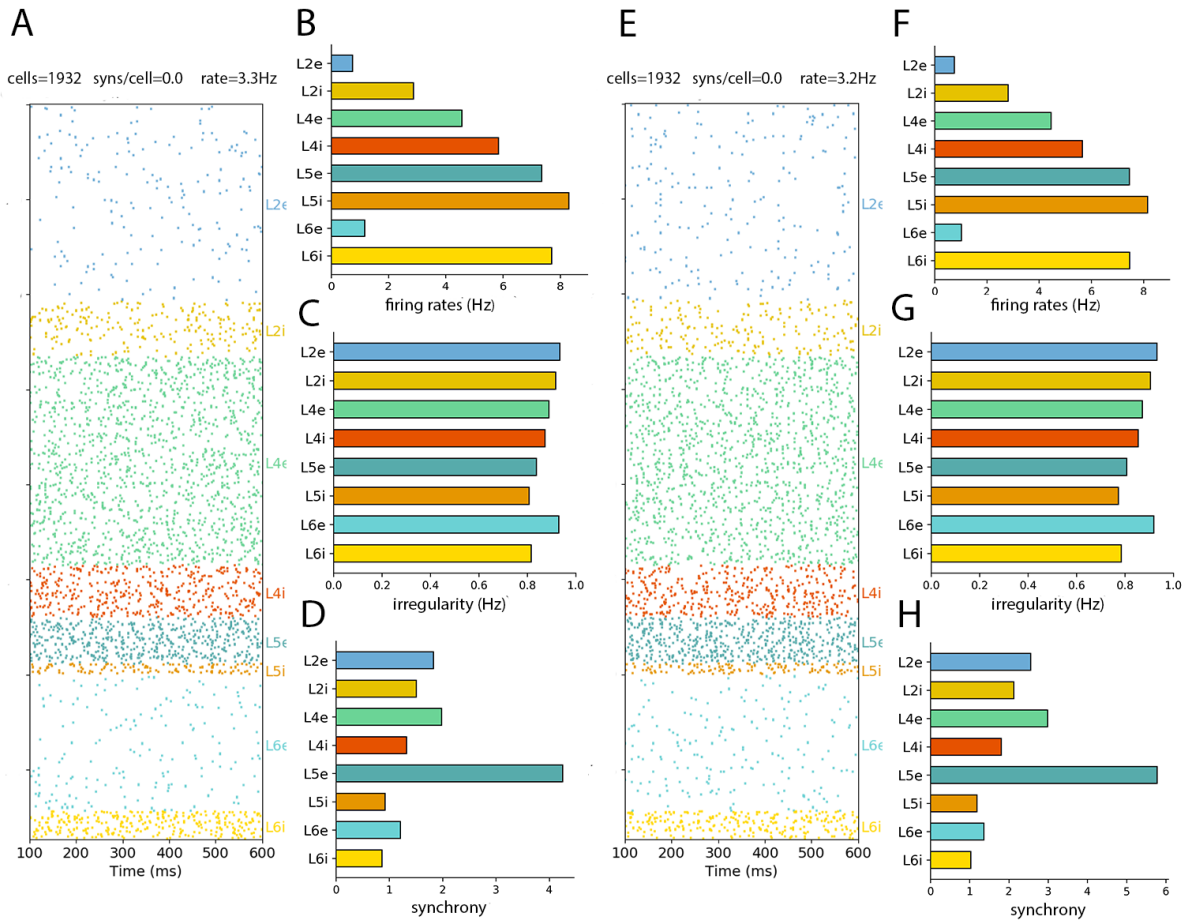


Figure 4: Network rescaled to 50% of the number of total neurons. (A) Raster plot and (B–D) statistics for external Poisson input. (E) raster plot and (F–H) statistics for external DC current. The simulation times and number of neurons sampled and plotted were chosen as in Figure 1.

For both Poisson and DC external inputs, the mean population firing rates of all rescaled versions are close to the original results (Potjans and Diesmann, 2014a). For Poisson inputs, even extreme downscaling to 1% resulted in mean firing rates within the ranges defined by the standard deviations calculated in the original article after 100 simulation trials (Table 4). For DC inputs, downscaling the network below 10% resulted in no firing activity due to insufficient spiking input (low standard deviation of the local mean-field potential). Nevertheless, the mean rates of the DC input models with downscaling above 10% also fall within the standard deviation intervals in the original article (Table 6).

A comparison of the mean population firing rates over the NetPyNE models with different degrees of rescaling presents a relatively restrained variability (Figure 8 and Tables 4 and 5). The populations with larger firing variability are L2e and L2i but even for them the relative deviations in comparison to the full-scale network are generally below 30%. In a comparison of input types, networks with Poisson inputs tend to exhibit larger mean firing variabilities at downscaling degrees below 10%, while networks with DC inputs display large variabilities

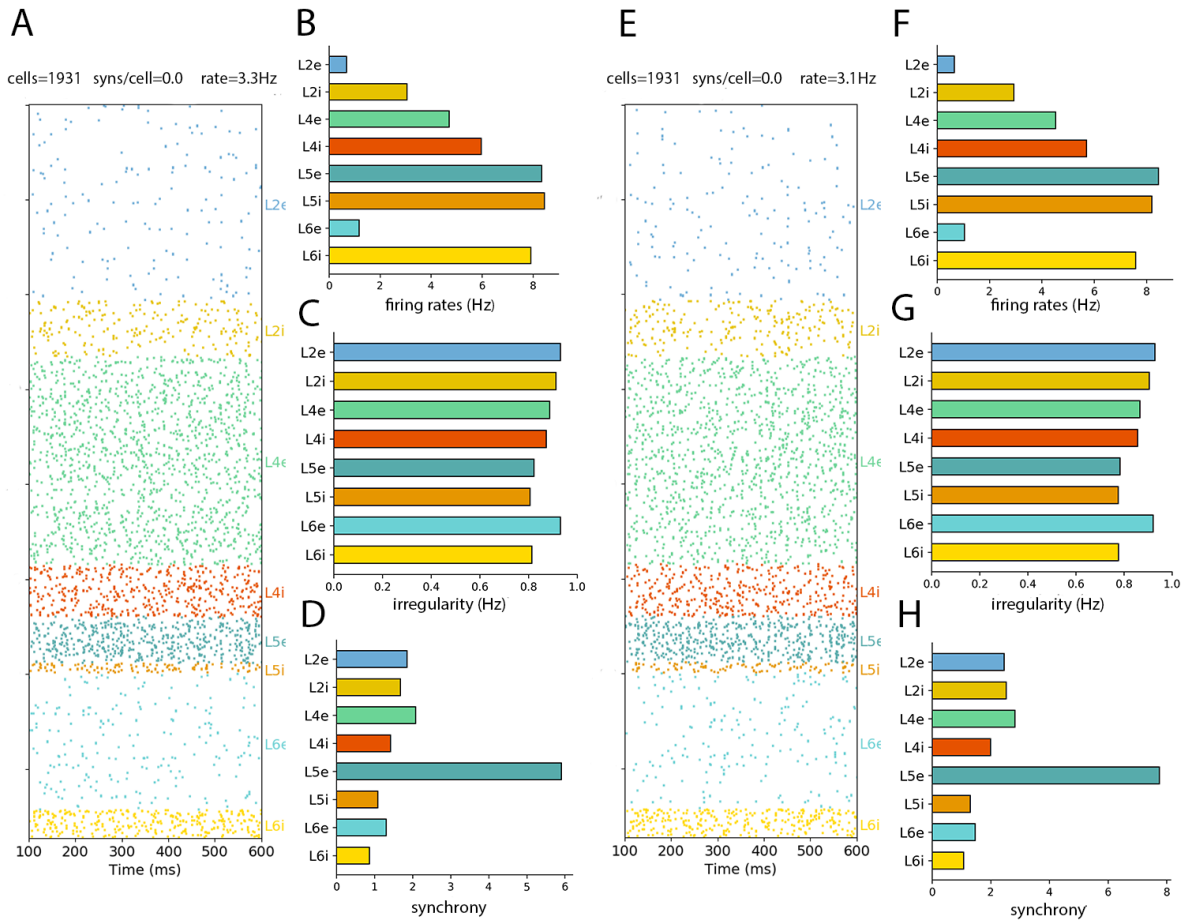


Figure 5: Network rescaled to 30% of the number of total neurons. (A) Raster plot and (B–D) statistics for external Poisson input. (E) raster plot and (F–H) statistics for external DC current. The simulation times and number of neurons sampled and plotted were chosen as in Figure 1.

already at downscalings of 40%.

The variability of the irregularity measure across the different levels of rescaling is much smaller than that of the mean firing rate, with relative deviations in relation to full scale around or below 1% (Figure 8 and Tables 6 and 7). The populations with larger irregularity variability in comparison to the full scale network are L5e and L5i.

Finally, we also compared the network synchrony after rescaling. For this, we first compared three different sampling approaches to illustrate the effect of sample size in the calculation:

1. To sample a fixed percentage of neurons per population, totaling 8,000 neurons (the fixed percentage is simply given by the ratio between 8,000 and 77,169, which is the number of neurons in the full size network);

2. To sample 1,000 neurons per population, as in the original article, totaling 8,000 neurons;

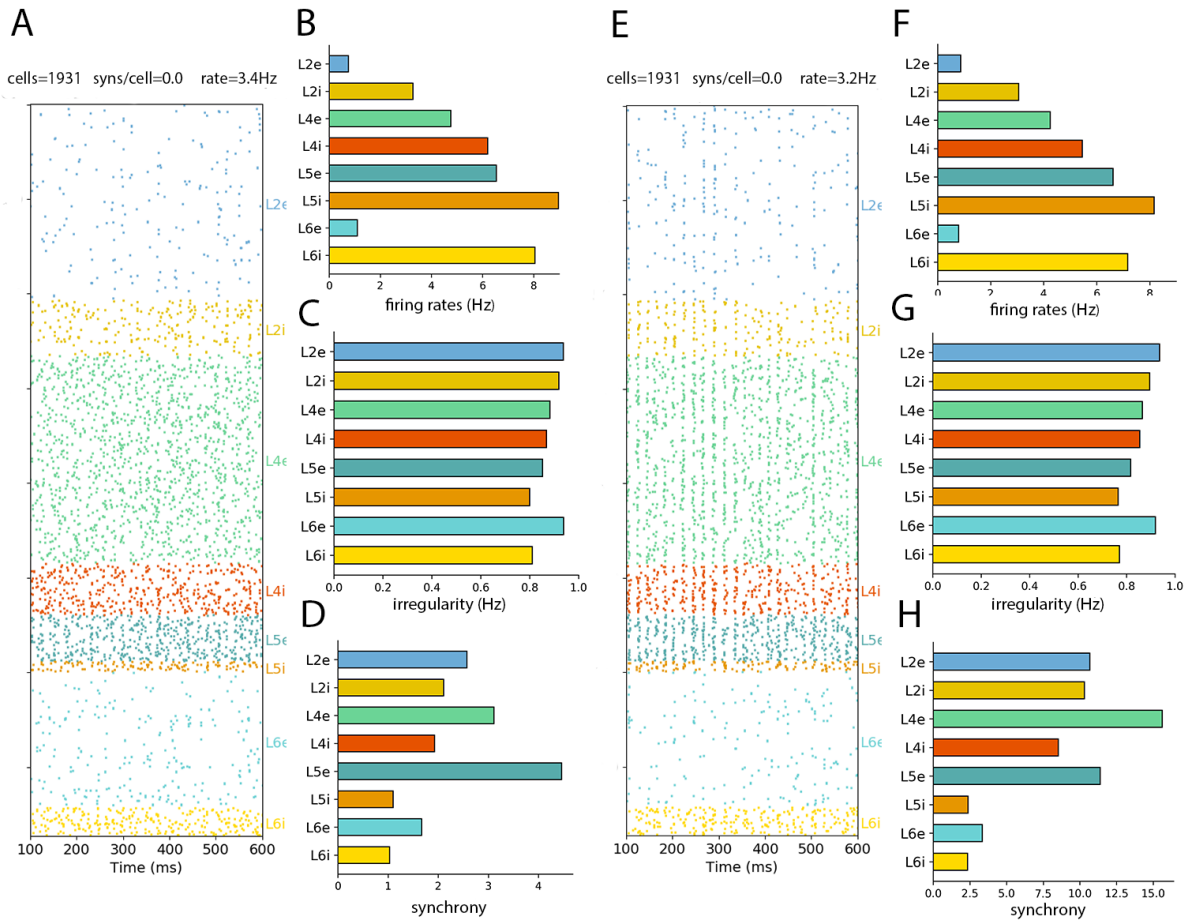


Figure 6: Network rescaled to 10% of the number of total neurons. (A) Raster plot and (B–D) statistics for external Poisson input. (E) raster plot and (F–H) statistics for external DC current. The simulation times and number of neurons sampled and plotted were chosen as in Figure 1.

3. To sample 2,000 neurons per population, totaling 16,000 neurons.

Differently from irregularity, synchrony depends on the sampling strategy adopted (see Table 3). The observed discrepancies in synchrony may be a consequence of sampling a different number of neurons or a different percentage of the population size (see the Discussion section). Interestingly, with the exception of L5i and L6i, synchrony appears to increase linearly with the number of sampled neurons from each population (Table 3). For comparison, we show in Figure 8 the synchrony of each population for the full-scale and scaled down NetPyNE implementations using the sampling strategy of the original article (1,000 neurons per layer) (more details are given in Tables 8 and 9).

1.3.3 Additional model analysis facilitated by NetPyNE

Converting the PDCM model to the NetPyNE standardized specifications has the added advantage of allowing the user to readily make use of the tool's many built-in analysis functions.

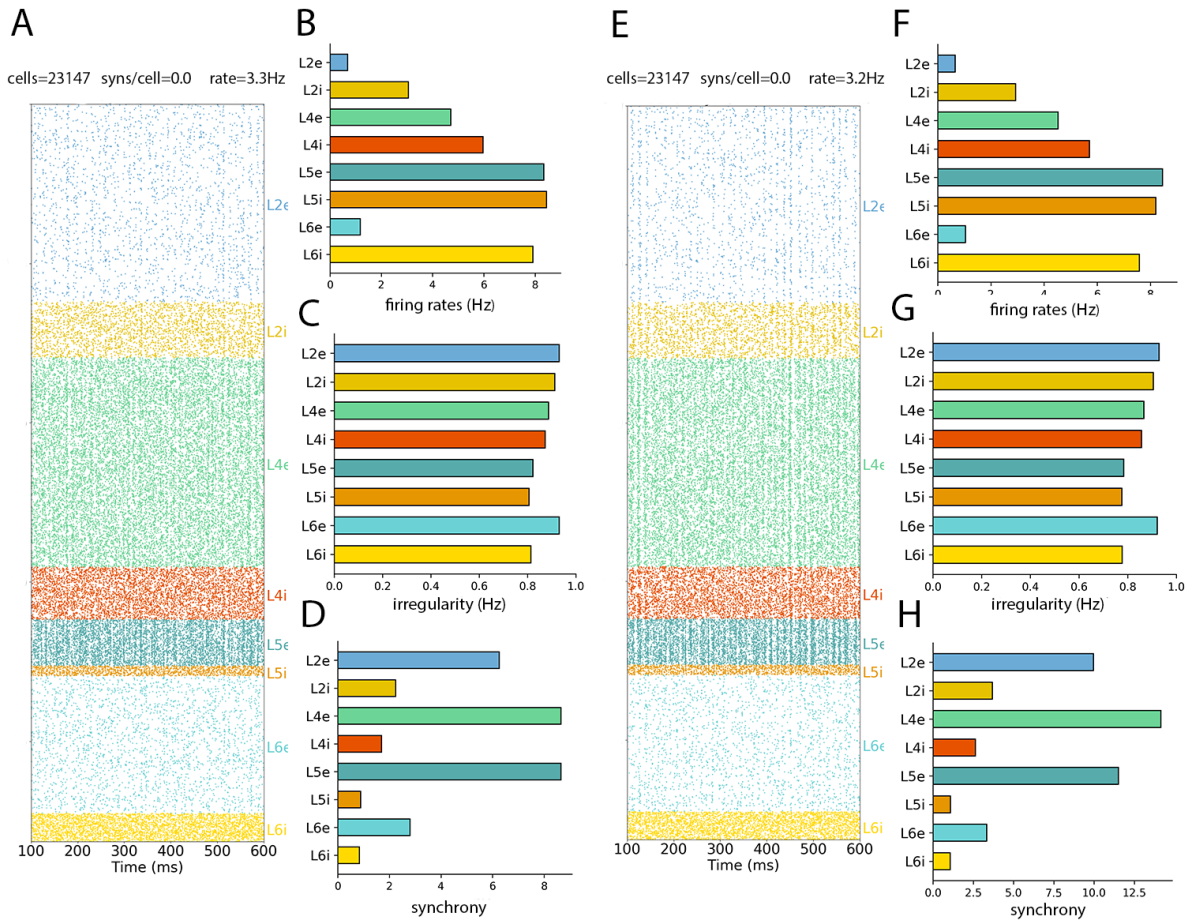


Figure 7: Network rescaled to 30% of the total number of neurons with full size sampling. (A) Raster plot and (B–D) statistics for external Poisson input. (E) Raster plot and (F–H) statistics for external DC current. Raster plots and statistics based on all the neurons in the network.

Population	L2e	L2i	L4e	L4i	L5e	L5i	L6e	L6i
Number of neurons	2,144	605	2,272	568	503	110	1492	306
Synchrony	5.1	1.5	5.7	1.4	2.5	1.2	1.4	1.0
Number of neurons	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
Synchrony	2.9	1.8	3.0	1.7	4.3	1.1	1.2	1.0
Number of neurons	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000
Synchrony	4.9	2.7	5.1	2.3	7.9	1.1	1.6	1.0

Table 3: Synchrony of multi-unit spike trains quantified by the normalized variance of spike count histogram with bin width of 3 ms, based on different numbers of neurons and recorded during 60 s simulations. In the first row, the percentage of sampled neurons per population is fixed. In the second row, the number of sampled neurons per population is fixed. In both cases the total number of sampled neurons is 8,000. The third row is similar to the second with the number of sampled neurons per population doubled, totaling 16,000 neurons. For each row (separated from the others by continuous lines), the corresponding synchrony measures are given in the second line.

These range from 2D visualization of the cell locations to different representations of network connectivity to spiking activity and information flow measures. Importantly, these are available

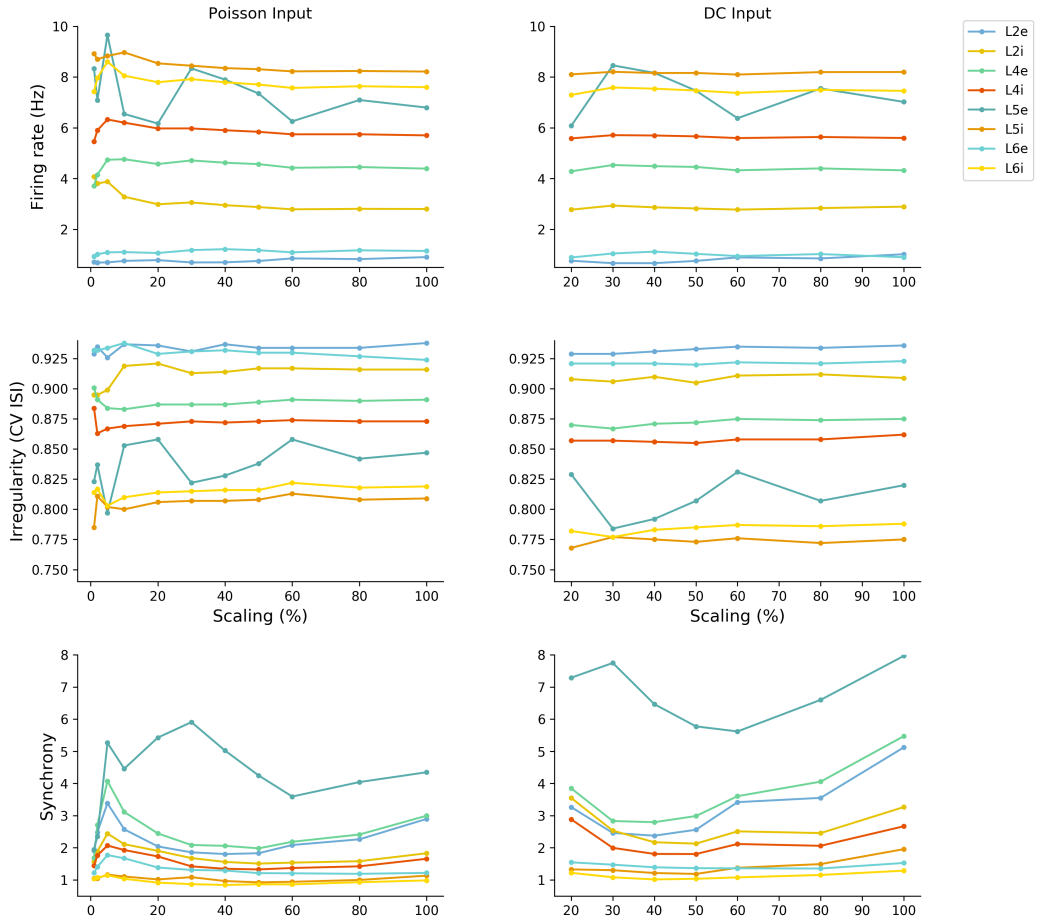


Figure 8: Mean population firing rate (top), irregularity (middle) and synchrony (bottom) of the 8 cell populations as a function of scaling for the NetPyNE reimplementation of the PDCM model, with Poisson (left) vs DC inputs (right). All results were calculated from 60-sec simulations and approximately 1000 neurons.

to the user through simple high-level function calls, which can be customized to include a specific time range, frequency range, set of populations, and visualization options.

We illustrate the range of NetPyNE's analysis capabilities using the PDCM model reimplementation in NetPyNE 9. All analyses were performed on the 10%-scaled version with 7713 cells and over 30M synapses, simulated for 4 biological seconds. First, we visualized the network cell locations from the top-down (Figure 9A and side (Figure 9B) views, which provided an intuitive representation of the cylindrical volume modeled, and the layer boundaries for each population. Next we plotted a stacked bar graph of convergence (Figure 9C), a measure of

connectivity that provides at-a-glance information on the average number and distribution of presynaptic inputs from each population. We then analyzed the spectrotemporal properties of the network's spiking activity through a Morlet wavelet-based spectrogram (Figure 9D); results depicted time-varying broad frequency peaks in the gamma range (40-80 Hz), consistent with the largely irregular and asynchronous network activity. Finally, we measured the spectral Granger causality between L4e and L2e cells and found stronger information flow from L4e to L2i than vice-versa, particularly at gamma range frequencies, consistent with the canonical microcircuit (Douglas et al., 1989). Information flow analysis can reveal functional circuit pathways, including those involving inhibitory influences, that are not always reflected in the anatomical connectivity.

1.4 Discussion

We reimplemented the PDCM model using the NetPyNE tool, a high-level interface to the NEURON simulator. The new model version reproduced the overall network dynamics of the original PDCM model, evaluated through population-specific average firing rates, irregularity and synchrony measures. The NetPyNE version also allows rescaling of the network size, while preserving the network statistics for most conditions. This feature can be used to study the effect of rescaling on network dynamics. For example, under certain conditions, network synchrony increased for smaller networks (see discussion below). Furthermore, the NetPyNE implementation (available in GitHub) provides a clear separation of model parameters and implementation, facilitates extension of the model, for example to include more biophysically-realistic multicompartment neuron models, and enables employing NetPyNE's analysis capabilities to gain further insights into the model. The latter was illustrated by visualizing the network's topology and connectivity, plotting the average firing rate spectrogram and calculating the spectral Granger causality (a measure of information flow) between two model populations (Figure 9).

1.4.1 Reproduction of original results

We were able to reproduce all the network statistics (mean firing rate, irregularity and synchrony) for the three types of external inputs: balanced Poisson, DC current, and unbalanced Poisson – compare Figure 1 with Figure 6 of the original article (Potjans and Diesmann, 2014a)), and Figure 3 with Figure 7 of the original article. Notably, in the unbalanced Poisson input condition, we can observe both the ceased activity in L6e and the rate frequency changes in the other populations. Therefore, the NetPyNE PDCM model is able to effectively reproduce

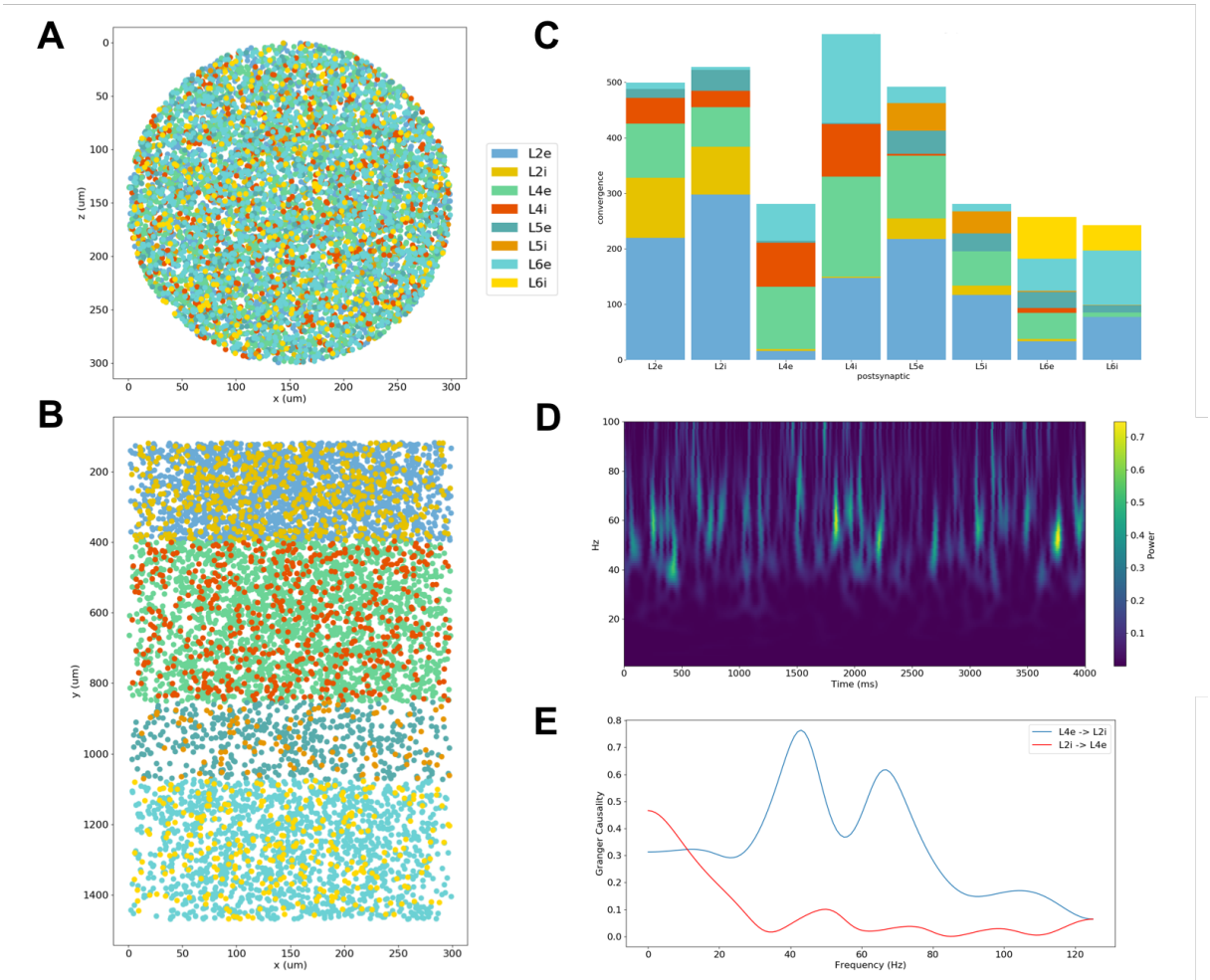


Figure 9: Analysis of model structure and simulation results facilitated by NetPyNE. (A) Top-down view (x-z plane) of network cell locations illustrates the diameter of the cylindrical volume modeled. (B) Side view (x-y plane) of network cell locations reveals cortical layer boundaries and populations per layer. Color indicates cell populations (see legend). (C) Stacked bar graph of the average connection convergence (number of presynaptic cells targeting each postsynaptic cell) for each population. (D) Spectrogram of average firing rate across all cells illustrating time-varying peaks in the gamma oscillation range. (E) Spectral Granger causality between L4e and L2i populations indicates stronger information flow from L4e to L2i than vice-versa.

the model in the original article without loss or changes in the statistics.

1.4.2 Preserved statistics in rescaled networks

Our rescaling method works by keeping the random inputs unchanged on average (Van Albada et al., 2015) and fixing the proportion between the firing threshold and the square root of the number of connections (Vreeswijk and Sompolinsky, 1998) (see parameters in Table 1). This method managed to approximately preserve the mean firing rate and irregularity for all populations across all scaling percentages, ranging from 90% to 1% (Figure 8 and Tables 4 to

7). The synchrony measure was similarly preserved for the Poisson external input condition (Table 8), but not for the DC input condition, as discussed below.

1.4.3 Multiple factors affect synchrony

The synchrony measure was dependent on the number of neurons used in its calculation, in general with a higher number of neurons resulting in higher synchrony values. Because of the different sampling strategies, comparing synchrony across populations and network models should be done with caution. For example, when we compared synchrony across populations sampling a fixed percentage of neurons per population (Table 3 top row) the two largest populations, namely L2e and L4e, exhibited the highest synchrony values. On the other hand, when we sampled a fixed number of neurons from each population, 1,000 as in the original article (Table 3 middle row) or 2,000 (Table 3 bottom row), the highest synchrony was displayed by population L5e. The strategy of sampling a fixed number of neurons per population also may lead to scaling distortions because a given fixed number corresponds to different percentages of the cell populations at each scaling degree. For example, in the full scale version 1,000 corresponds to almost 100% of L5e neurons but to less than 5% of L2e neurons. Besides, when the degree of downscaling is too strong ($\leq 20\%$) there may be not enough number of neurons in a population to include in the calculation.

In general, synchrony tends to decrease with the degree of downscale from the full size network (see plots at the bottom of Figure 8). This is due to the increase in the DC current that we provide to neurons to compensate for the decrease in the number of connections (see Section 1.2.3). This effect occurs up to a downscale level that depends on the population and external input type (between 40% and 60%). As we proceeded with the downscale past this point, we reached a situation in which the number of neurons was not sufficient to allow a reliable calculation of synchrony. For example, when we downscaled the networks to 10% of the original size, we had to replace 99% of the connections with DC inputs, and this resulted in large increases in synchrony (Figure 8, bottom plots).

Another characteristic of synchrony is that it depends on the number of neurons sampled to do the statistics. For example, the raster plot and synchrony for the Poisson-driven full-scale network indicate, both visually and numerically, a high degree of synchrony when all neurons ($\sim$80k) are sampled (Figure 2) but a very low degree of synchrony when only 2.3% of the neurons are sampled (Figure 1). The same phenomenon was observed in the rescaled implementations. For example, the 30% rescaled network displayed high synchrony when all neurons ($\sim$23k) were sampled (Figure 7) and low synchrony when a smaller subset of neurons

($\sim$2k) was sampled (Figure 5).

Synchrony was also dependent on the population average firing rate. The synchrony measure used (see Methods) increases with the heterogeneity of firing within the cell population (Pinsky and Rinzel, 1995), which for equal population sizes and fixed bin size is higher for cells with higher firing rates. This dependence may be a possible explanation for the high synchrony of L5e neurons (Figure 8 bottom plots; see also Tables 8 and 9).

Synchrony was generally higher under the DC input condition than the Poisson input condition (Figure 8 bottom plots; see also Tables 8 and 9). We hypothesize this is due to the two sources of randomization present in the Poisson-driven network: the Poisson inputs and the random pattern of connection. In the DC condition we removed the Poisson inputs, thus increasing the network synchrony. For very high downscaling, e.g. 10%, synchrony becomes visually perceptible in the raster plot for DC inputs but not in the one for Poisson inputs (compare Figures 6H and 6D). This effect is not seen for intermediate downscaling levels, cf. Figures 4D,H (50% downscaling) and 5D,H (30% downscaling) because the fraction of sampled cells is not high enough.

The rescaling method used here has the theoretical property of not adding synchrony or regularity to asynchronous irregular networks (Vreeswijk and Sompolinsky, 1998; Van Albada et al., 2015). In our study, we found that irregularity and synchrony did not appear to be affected by rescaling up to the limits for which mean firing rates were within the standard deviations of the original article (Potjans and Diesmann, 2014a), namely 1% for Poisson inputs and 10% for DC inputs.

Acknowledgments

This work was produced as part of the activities of NIH U24EB028998, R01EB022903, U01EB017695, NYS SCIRB DOH01-C32250GG-3450000, NSF 1904444; and FAPESP (S. Paulo Research Foundation) Research, Disseminations and Innovation Center for Neuromathematics 2013/07699-0, 2015/50122-0, 2018/20277-0. C.R. (grant number 88882.378774/2019-01) and F.A.N/ (grant number 88882.377124/2019-01) are recipients of PhD scholarships from the Brazilian Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES). A.C.R. is partially supported by a CNPq fellowship (grant 306251/2014-0).

References

Beul, S. F. and Hilgetag, C. C. (2015). Towards a "canonical" agranular cortical microcircuit. *Frontiers in Neuroanatomy*, 8:165.

Bezaire, M. J., Raikov, I., Burk, K., Vyas, D., and Soltesz, I. (2016). Interneuronal mechanisms of hippocampal theta oscillation in a full-scale model of the rodent ca1 circuit. *eLife*, 5:e18566.

Bos, H., Diesmann, M., and Helias, M. (2016). Identifying anatomical origins of coexisting oscillations in the cortical microcircuit. *PLoS Computational Biology*, 12(10):e1005132.

Brunel, N. (2000). Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. *Journal of Computational Neuroscience*, 8(3):183–208.

Cain, N., Iyer, R., Koch, C., and Mihalas, S. (2016). The computational properties of a simplified cortical column model. *PLoS Computational Biology*, 12(9):e1005045.

Carnevale, N. T. and Hines, M. L. (2006). *The NEURON Book*. Cambridge University Press.

de Kock, C. P. and Sakmann, B. (2009). Spiking in primary somatosensory cortex during natural whisking in awake head-restrained rats is cell-type specific. *Proceedings of the National Academy of Sciences (USA)*, 106(38):16446–16450.

Douglas, R. J., Martin, K. A., and Whitteridge, D. (1989). A canonical microcircuit for neocortex. *Neural Computation*, 1(4):480–488.

Dura-Bernal, S., Neymotin, S., Suter, B., GMG, S., and Lytton, W. (2018). Multiscale dynamics and information flow in a data-driven model of the primary motor cortex microcircuit. *bioRxiv*, (201707).

Dura-Bernal, S., Suter, B. A., Gleeson, P., Cantarelli, M., Quintana, A., Rodriguez, F., Kedziora, D. J., Chadderdon, G. L., Kerr, C. C., Neymotin, S. A., McDougal, R. A., Hines, M., Shepherd, G. M., and Lytton, W. W. (2019). Netpyne, a tool for data-driven multiscale modeling of brain circuits. *eLife*, 8:e44494.

Gewaltig, M.-O. and Diesmann, M. (2007). NEST (neural simulation tool). *Scholarpedia*, 2(4):1430.

Hines, M. L., Morse, T., Migliore, M., Carnevale, N. T., and Shepherd, G. M. (2004). Modeldb: a database to support computational neuroscience. *Journal of Computational Neuroscience*, 17(1):7–11.

Lapicque, L. (1907). Recherches quantitatives sur l'excitation electrique des nerfs traitee comme une polarization. *Journal de Physiologie et de Pathologie Generale*, 9:620–635.

Lee, J. H., Koch, C., and Mihalas, S. (2017). A computational analysis of the function of three inhibitory cell types in contextual visual processing. *Frontiers in Computational Neuroscience*, 11:28.

Lytton, W. W., Seidenstein, A. H., Dura-Bernal, S., McDougal, R. A., Schürmann, F., and Hines, M. L. (2016). Simulation neurotechnologies for advancing brain research: parallelizing large networks in neuron. *Neural Computation*, 28(10):2063–2090.

McDougal, R. A., Hines, M. L., and Lytton, W. W. (2013). Reaction-diffusion in the neuron simulator. *Frontiers in Neuroinformatics*, 7:28.

Newton, A. J. H., McDougal, R. A., Hines, M. L., and Lytton, W. W. (2018). Using neuron for reaction-diffusion modeling of extracellular dynamics. *Frontiers in Neuroinformatics*, 12:41.

Neymotin, S. A., Dura-Bernal, S., Lakatos, P., Sanger, T. D., and Lytton, W. (2016). Multitarget multiscale simulation for pharmacological treatment of dystonia in motor cortex. *Frontiers in Pharmacology*, 7(157).

Pinsky, P. F. and Rinzel, J. (1995). Synchrony measures for biological neural networks. *Biological Cybernetics*, 73(2):129–137.

Potjans, T. C. and Diesmann, M. (2014a). The cell-type specific cortical microcircuit: relating structure and activity in a full-scale spiking network model. *Cerebral Cortex*, 24(3):785–806.

Potjans, T. C. and Diesmann, M. (2014b). Spiking cortical network model – potjans and diesmann: http://www.opensourcebrain.org/projects /potjansdiesmann2014.

Ranjan, R., Khazen, G., Gambazzi, L., Ramaswamy, S., Hill, S. L., Schürmann, F., and Markram, H. (2011). Channelpedia: an integrative and interactive database for ion channels. *Frontiers in Neuroinformatics*, 5:36.

Sakata, S. and Harris, K. D. (2009). Laminar structure of spontaneous and sensory-evoked population activity in auditory cortex. *Neuron*, 64(3):404–418.

Schmidt, M., Bakker, R., Hilgetag, C. C., Diesmann, M., and van Albada, S. J. (2018). Multi-scale account of the network structure of macaque visual cortex. *Brain Structure and Function*, 223(3):1409–1435.

Schuecker, J., Schmidt, M., van Albada, S. J., Diesmann, M., and Helias, M. (2017). Fundamental activity constraints lead to specific interpretations of the connectome. *PLoS Computational Biology*, 13(2):e1005179.

Schwalger, T., Deger, M., and Gerstner, W. (2017). Towards a theory of cortical columns: From spiking neurons to interacting neural populations of finite size. *PLoS Computational Biology*, 13(4):e1005507.

Shimoura, R., Kamiji, N., Pena, R., Lima Cordeiro, V., Celis, C., Romaro, C., and Roque, A. C. (2018). [re] the cell-type specific cortical microcircuit: relating structure and activity in a full-scale spiking network model. *ReScience*, 4.

Sivagnanam, S., Majumdar, A., Yoshimoto, K., Astakhov, V., Bandrowski, A., Martone, M. E., and Carnevale, N. T. (2013). Introducing the neuroscience gateway. In *IWSG*, 993.

Swadlow, H. A. (1989). Efferent neurons and suspected interneurons in s-1 vibrissa cortex of the awake rabbit: receptive fields and axonal properties. *Journal of Neurophysiology*, 62(1):288–308.

Thomson, A. M., West, D. C., Wang, Y., and Bannister, A. P. (2002). Synaptic connections and small circuits involving excitatory and inhibitory neurons in layers 2–5 of adult rat and cat neocortex: triple intracellular recordings and biocytin labelling in vitro. *Cerebral Cortex*, 12(9):936–953.

Towns, J., Cockerill, T., Dahan, M., Foster, I., Gaither, K., Grimshaw, A., Hazlewood, V., Lathrop, S., Lifka, D., Peterson, G. D., et al. (2014). Xsede: accelerating scientific discovery. *Computing in Science & Engineering*, 16(5):62–74.

Van Albada, S. J., Helias, M., and Diesmann, M. (2015). Scalability of asynchronous networks is limited by one-to-one mapping between effective connectivity and correlations. *PLoS Computational Biology*, 11(9):e1004490.

Vreeswijk, C. v. and Sompolinsky, H. (1998). Chaotic balanced state in a model of cortical circuits. *Neural Computation*, 10(6):1321–1371.

Wagatsuma, N., Potjans, T. C., Diesmann, M., Sakai, K., and Fukai, T. (2013). Spatial and feature-based attention in a layered cortical microcircuit model. *PloS One*, 8(12):e80788.

West, D. C., Mercer, A., Kirchhecker, S., Morris, O. T., and Thomson, A. M. (2005). Layer 6 cortico-thalamic pyramidal cells preferentially innervate interneurons and generate facilitating epsps. *Cerebral Cortex*, 16(2):200–211.

Supplementary Material

Population Scaling	L2e	L2i	L4e	L4i	L5e	L5i	L6e	L6i
100%	0.90	2.80	4.39	5.70	6.80	8.22	1.14	1.14
80%	0.82 (9%)	2.81 (0%)	4.45 (1%)	5.74 (1%)	7.10 (4%)	8.24 (0%)	1.17 (3%)	1.17 (3%)
60%	0.85 (6%)	2.79 (0%)	4.42 (1%)	5.74 (1%)	6.25 (8%)	8.22 (0%)	1.09 (4%)	1.09 (4%)
50%	0.75 (17%)	2.87 (3%)	4.57 (4%)	5.84 (2%)	7.35 (8%)	8.31 (1%)	1.17 (3%)	1.17 (3%)
40%	0.70 (22%)	2.95 (5%)	4.62 (5%)	5.90 (4%)	7.90 (16%)	8.35 (2%)	1.22 (7%)	1.22 (7%)
30%	**0.69 (23%)**	3.06 (9%)	4.71 (7%)	5.98 (5%)	8.34 (23%)	8.45 (3%)	1.18 (4%)	1.18 (4%)
20%	0.78 (13%)	2.99 (7%)	4.57 (4%)	5.97 (5%)	6.17 (9%)	8.54 (4%)	1.06 (7%)	1.06 (7%)
10%	0.75 (17%)	3.28 (17%)	4.76 (8%)	6.20 (9%)	6.55 (4%)	**8.97 (9%)**	1.10 (4%)	1.10 (4%)
5%	**0.69 (23%)**	3.88 (39%)	4.74 (8%)	**6.33 (11%)**	**9.66 (42%)**	8.83 (7%)	1.09 (4%)	1.09 (4%)
2 %	**0.69 (23%)**	3.8 (36%)	4.15 (5%)	5.90 (4%)	7.08 (4%)	8.70 (6%)	1.02 (11%)	1.02 (11%)
1%	0.71 (21%)	**4.08 (46%)**	**3.71 (15%)**	5.46 (4%)	8.33 (23%)	8.92 (9%)	**0.93 (18%)**	**0.93 (18%)**
NEST-100 trials	1.11 ± 0.8	-	4.8 ± 1.1	-	11 ± 6.1	-	0.56 ± 0.9	-

Table 4: Population mean firing rates (Hz) for rescaled NetPyNE versions of the original PDCM model with Poisson external input. All results calculated from 60-sec simulations and approximately 1850 neurons. Relative deviations in relation to the full-scale NetPyNE version $\left(\left|f_{\mathrm{X}\%} - f_{100\%}\right| / f_{100\%}\right)$, are shown within parentheses, and their maxima in bold. For comparison, the last row shows mean and standard deviation firing rates from 100 runs of the NEST implementation with random numbers of external inputs to each layer (see (Potjans and Diesmann, 2014a) for details).

Population Scaling	L2e	L2i	L4e	L4i	L5e	L5i	L6e	L6i
100%	1.02	2.89	4.32	5.60	7.02	8.20	0.90	0.90
80%	0.85 (17%)	2.83 (2%)	4.40 (2%)	5.64 (1%)	7.55 (8%)	8.20 (0%)	1.02 (13%)	1.02 (13%)
60%	0.89 (13%)	2.77 (4%)	4.32 (0%)	5.59 (0%)	6.38 (9%)	8.10 (1%)	0.94 (4%)	0.94 (4%)
50%	0.75 (26%)	2.82 (2%)	4.46 (3%)	5.66 (1%)	7.46 (6%)	8.16 (0%)	1.03 (14%)	1.03 (14%)
40%	**0.66 (35%)**	2.86 (1%)	4.49 (4%)	5.70 (2%)	8.17 (16%)	8.16 (0%)	**1.12 (24%)**	**1.12 (24%)**
30%	**0.66 (35%)**	2.93 (1%)	**4.53 (5%)**	**5.71 (2%)**	**8.46 (21%)**	8.21 (0%)	1.05 (17%)	1.05 (17%)
20%	0.76 (25%)	2.77 (4%)	4.29 (1%)	5.58 (0%)	6.08 (13%)	**8.11 (1%)**	0.89 (1%)	0.89 (1%)
10%	0.87 (15%)	**3.06 (6%)**	4.25 (2%)	5.46 (2%)	6.62 (6%)	8.17 (0%)	0.80 (11%)	0.80 (11%)
NEST-100 trials	1.11 ± 0.8	-	4.8 ± 1.1	-	11 ± 6.1	-	0.56 ± 0.9	-

Table 5: Population firing rates (Hz) for rescaled NetPyNE versions of the original PDCM model with DC current input. All results calculated from 60-sec simulations and approximately 1850 neurons. Relative deviations in relation to the full-scale NetPyNE version $\left(\left|f_{\mathrm{X}\%} - f_{100\%}\right| / f_{100\%}\right)$ are shown within parentheses, and their maxima in bold. For comparison, the last row shows mean and standard deviation firing rates from 100 runs of the NEST implementation with random numbers of external inputs to each layer (see (Potjans and Diesmann, 2014a) for details).

Population Scaling	L2e	L2i	L4e	L4i	L5e	L5i	L6e	L6i
100%	0.938	0.916	0.891	0.873	0.847	0.809	0.924	0.819
80%	**0.934 (0.45%)**	0.916 (0.06%)	0.890 (0.17%)	0.873 (0.07%)	0.842 (0.63%)	0.808 (0.15%)	0.927 (0.29%)	0.818 (0.19%)
60%	**0.934 (0.45%)**	0.917 (0.04%)	0.891 (0.02%)	0.874 (0.12%)	0.858 (1.25%)	0.813 (0.48%)	0.930 (0.63%)	0.822 (0.27%)
50%	**0.934 (0.39%)**	0.917 (0.06%)	0.889 (0.19%)	0.873 (0.06%)	0.838 (1.12%)	0.808 (0.24%)	0.930 (0.68%)	0.816 (0.38%)
40%	0.937 (0.08%)	0.914 (0.31%)	0.887 (0.44%)	0.872 (0.12%)	0.828 (2.30%)	0.807 (0.36%)	0.932 (0.88%)	0.816 (0.40%)
30%	**0.931 (0.72%)**	0.913 (0.37%)	0.887 (0.45%)	0.873 (0.06%)	0.822 (2.95%)	0.807 (0.32%)	0.931 (0.72%)	0.815 (0.60%)
20%	0.936 (0.15%)	0.921 (0.50%)	0.887 (0.41%)	0.871 (0.28%)	0.858 (1.22%)	0.806 (0.41%)	0.929 (0.55%)	0.814 (0.64%)
10%	0.937 (0.05%)	0.919 (0.31%)	0.883 (0.87%)	0.869 (0.51%)	0.853 (0.62%)	0.800 (1.23%)	**0.938 (1.55%)**	0.810 (1.11%)
5%	**0.926 (1.21%)**	0.899 (1.90%)	0.884 (0.82%)	0.867 (0.76%)	**0.797 (5.99%)**	0.802 (0.97%)	0.934 (1.03%)	**0.803 (1.97%)**
2%	0.935 (0.32%)	**0.895 (2.34%)**	0.891 (0.03%)	**0.863 (1.21%)**	0.837 (1.23%)	0.811 (0.15%)	0.932 (0.81%)	0.817 (0.24%)
1%	**0.929 (0.92%)**	0.895 (2.30%)	**0.901 (1.12%)**	0.884 (1.16%)	0.823 (2.90%)	**0.785 (2.98%)**	0.932 (0.82%)	0.814 (0.66%)

Table 6: Irregularity of single-unit spike trains for rescaled NetPyNE versions of the original PDCM model with Poisson external input. All results calculated from 60-sec simulations and approximately 1000 neurons per population. Relative deviations in relation to the full-scale NetPyNE version $\left(\left|f_{\mathrm{X}\%} - f_{100\%}\right| / f_{100\%}\right)$ are shown within parentheses, and their maxima in bold.

Population Scaling	L2e	L2i	L4e	L4i	L5e	L5i	L6e	L6i
100%	0.936	0.909	0.875	0.862	0.820	0.775	0.923	0.788
80%	0.934 (0.21%)	0.912 (0.33%)	0.874 (0.11%)	0.858 (0.46%)	0.807 (1.59%)	0.772 (0.39%)	0.921 (0.22%)	0.786 (0.25%)
60%	0.935 (0.11%)	0.911 (0.22%)	0.875 (0.00%)	0.858 (0.46%)	0.831 (1.34%)	0.776 (0.13%)	0.922 (0.11%)	0.787 (0.13%)
50%	0.933 (0.32%)	0.905 (0.44%)	0.872 (0.34%)	**0.855 (0.81%)**	0.807 (1.59%)	0.773 (0.26%)	**0.920 (0.33%)**	0.785 (0.38%)
40%	0.931 (0.53%)	0.910 (0.11%)	0.871 (0.46%)	0.856 (0.70%)	0.792 (3.41%)	0.775 (0.00%)	0.921 (0.22%)	0.783 (0.63%)
30%	**0.929 (0.75%)**	0.906 (0.33%)	0.867 (0.91%)	0.857 (0.58%)	**0.784 (4.39%)**	0.777 (0.26%)	0.921 (0.22%)	0.777 (1.40%)
20%	**0.929 (0.75%)**	0.908 (0.11%)	0.870 (0.57%)	0.857 (0.58%)	0.829 (1.10%)	0.768 (0.90%)	0.921 (0.22%)	0.782 (0.76%)
10%	0.937 (0.11%)	**0.896 (1.43%)**	**0.866 (1.03%)**	**0.855 (0.81%)**	0.817 (0.37%)	**0.766 (1.16%)**	**0.920 (0.33%)**	**0.771 (2.16%)**

Table 7: Irregularity of single-unit spike trains for rescaled NetPyNE versions of the original PDCM model with DC current input. All results calculated from 60-sec simulations and approximately 1000 neurons per population. Relative deviations in relation to the full-scale NetPyNE version $\left(\left|f_{\mathrm{X}\%} - f_{100\%}\right| / f_{100\%}\right)$ are shown within parentheses, and their maxima in bold.

Layer Scaling	L2e	L2i	L4e	L4i	L5e	L5i	L6e	L6i
100%	2.9	1.8	3.0	1.7	4.4	1.1	1.2	1.0
80%	2.3 (21%)	1.6 (11%)	2.4 (20%)	1.4 (18%)	4.0 (9%)	1.0 (9%)	1.2 (0%)	0.9 (10%)
60%	2.1 (28%)	1.5 (17%)	2.2 (27%)	1.4 (18%)	3.6 (18%)	**0.9 (18%)**	1.2 (0%)	0.9 (10%)
50%	**1.8 (38%)**	1.5 (17%)	2.0 (33%)	1.3 (24%)	4.3 (2%)	**0.9 (18%)**	1.2 (0%)	0.9 (10%)
40%	**1.8 (38%)**	1.6 (11%)	2.1 (30%)	1.4 (18%)	5.0 (14%)	1.0 (9%)	1.3 (8%)	**0.8 (20%)**
30%	1.9 (34%)	1.7 (6%)	2.1 (30%)	1.4 (18%)	5.9 (34%)	1.1 (0%)	1.3 (8%)	0.9 (10%)
20%	2.0 (31%)	1.9 (6%)	2.4 (20%)	1.7 (0%)	5.4 (23%)	1.0 (9%)	1.4 (17%)	0.9 (10%)
10%	2.6 (10%)	2.1 (17%)	3.1 (3%)	1.9 (12%)	4.5 (2%)	1.1 (0%)	1.7 (42%)	1.0 (0%)
5%	3.4 (17%)	**2.4 (33%)**	4.1 (37%)	**2.1 (24%)**	5.3 (20%)	1.2 (9%)	**1.8 (50%)**	1.1 (10%)
2 %	2.5 (14%)	1.9 (6%)	2.7 (10%)	1.8 (6%)	2.4 (45%)	1.1 (0%)	1.4 (17%)	1.1 (10%)
1%	1.9 (34%)	1.6 (11%)	**1.7 (43%)**	1.4 (18%)	**2.0 (55%)**	1.1 (0%)	1.2 (0%)	1.0 (0%)

Table 8: Synchrony of multi-unit spike trains for rescaled NetPyNE versions of the original PDCM model with Poisson external input. All results calculated from 60-sec simulations and approximately 1000 neurons per population. Relative deviations in relation to the full-scale NetPyNE version $\left(\left|f_{\mathrm{X}\%} - f_{100\%}\right| / f_{100\%}\right)$ are shown within parentheses, and their maxima in bold.

Layer Scaling	L2e	L2i	L4e	L4i	L5e	L5i	L6e	L6i
100%	5.1	3.3	5.5	2.7	8.0	2.0	1.5	1.3
80%	3.6 (29%)	2.5 (24%)	4.1 (25%)	2.1 (22%)	6.6 (18%)	1.5 (25%)	1.4 (7%)	1.2 (8 %)
60%	3.4 (33%)	2.5 (24%)	3.6 (35%)	2.1 (22%)	5.6 (30%)	1.4 (30%)	1.4 (7%)	1.1 (15 %)
50%	2.6 (49%)	2.1 (36%)	3.0 (45%)	1.8 (33%)	5.8 (28%)	**1.2 (40%)**	1.4 (7%)	1.0 (23%)
40%	2.4 (53%)	2.2 (33%)	2.8 (49%)	1.8 (33%)	6.5 (19%)	**1.2 (40%)**	1.4 (7%)	1.0 (23%)
30%	2.5 (51%)	2.5 (24%)	2.8 (49%)	2.0 (26%)	7.8 (3%)	1.3 (35%)	1.5 (0%)	1.1 (15%)
20%	3.3 (35%)	3.6 (9%)	3.9 (29%)	2.9 (7%)	7.3 (9%)	1.3 (35%)	1.6 (7%)	1.2 (8%)
10%	**10.7 (110%)**	**10.3 (212%)**	**15.6 (184%)**	**8.5 (215%)**	**11.4 (43%)**	2.4 (20%)	**3.3 (120%)**	**2.3 (77%)**

Table 9: Synchrony of multi-unit spike trains for rescaled NetPyNE versions of the original PDCM model with DC current input. All results calculated from 60-sec simulations and approximately 1000 neurons per population. Relative deviations in relation to the full-scale NetPyNE version $\left(\left|f_{\mathrm{X}\%} - f_{100\%}\right| / f_{100\%}\right)$ are shown within parentheses, and their maxima in bold.

PART II

2 ADDING SPACE TO RANDOM NETWORKS OF SPIKING NEURONS: RESCALING AND BOUNDARY SOLUTION THAT KEEP FIRST- AND SECOND-ORDER STATISTICS

Abstract

There is a strong interdependence between the network size and the computational resources available to simulate it, which may hinder a computational neuroscience study. Rescaling the network size may be a solution but it must be done with caution to keep the properties of spiking dynamics unchanged. Additionally, adding space to a random network model to describe topographic arrangements of connections presents an extra challenge: how to deal with neurons at the spatial boundary to prevent spurious behavior? Such behavior could be, for example, spurious oscillations introduced by periodic boundary conditions, or unbalanced neuronal spiking due to lack/excess of connections. Here, we describe a network rescaling method which preserves first and second-order statistical properties of spiking activity, and introduce a boundary solution method for networks with added spatial extension which prevents the occurrence of spurious behavior.

2.1 Introduction

The past decade has seen the appearance of a number of large-scale spiking network models based on point neurons of the integrate-and-fire type (Izhikevich and Edelman, 2008; Potjans and Diesmann, 2014a; Hagen et al., 2016; Joglekar et al., 2018), (Schmidt et al., 2018; Igarashi et al., 2019; Merkt et al., 2019; Pena et al., 2020). The promise of these models is to provide explanatory mechanisms for how spiking neurons, when interconnected in specific ways, give rise to observable network dynamics (Lansner and Diesmann, 2012; Bassett et al., 2018; Einevoll et al., 2019). However, lack of or insufficient high-performance computational resources (Lytton et al., 2016; Jordan et al., 2018) may hinder the breadth and diversity of studies that could be done using these or similar models. Consequently there seems to be a compromise between the increase in detail or the size of spiking network models and the computational resource available. To make more detailed simulations computationally feasible, a possible strategy would be to reduce the size of the network.

Rescaling the network to decrease or increase its size is, however, a challenging task. For example, as we reduce the number of neurons, an increase in the number of connections or in the synaptic weights is needed to balance the external inputs. However, this can lead to undesired spiking synchrony and regularity (Brunel, 2000; Van Albada et al., 2015; Iyer et al., 2013).

An additional challenge arises if the modeller wants to add spatial extension to the model to take into account the spatial structure of the neuronal connectivity in the network to be modelled. An example is given by models of the primary somatossensory cortex (S1) which aim to describe somatotopy, i.e. the topographically ordered representation of the body surface in S1 (Woolsey et al., 1942). The topographic pattern of connection is interrupted at the network edges, changing the activity at the network boundary (Markram et al., 2015; Mazza et al., 2004). A classic solution to this problem is the adoption of periodic boundary conditions (toroidal connectivity in 2D) (Mehring et al., 2003; Yger et al., 2011; Voges and Perrinet, 2012; Rosenbaum and Doiron, 2014) but this introduces undesired oscillations to the network (Veltz and Sejnowski, 2015; Senk et al., 2018b,a).

In this work, we introduce a method to deal with the boundary conditions of two-dimensional spatially extended networks of spiking neurons that prevents the occurrence of oscillatory artefacts. The method is based on a rescaling technique that preserves the first and second order statistics of network activity (Romaro et al., 2020a).

This paper is organized as follows. In section 2.2, which is subdived in three subsections, we present the rescaling method. In subsection 2.2.1 we show the algorithm; in subsection

2.2.2 we give some application examples; and in subsection 2.2.3, we present the mathematical explanation and discuss model requirements as well as the limitations of the method. In Section 2.3, which also is divided in three subsections, we present the boundary solution method. In subsection 2.3.1 we describe the algorithm; in subsection 2.3.2 we give application examples; and, finally, in subsection 2.3.3 we discuss the sufficient conditions that the network model must satisfy for the method to be applied.

2.2 Rescaling Method

A neural network structure can be defined by the number of neurons N, a function of connection $\mathscr{F}(o_{pre}, o_{post})$ between the pre-synaptic neurons o_{pre} and the post-synaptic neurons o_{post}, and the synaptic strengths $w_{pre,post}$. The network can correspond to a given brain region, e.g. a cortical column, with neurons organized in populations and/or layers. Other model-dependent parameters, which characterize single neurons, such as firing threshold V_{th}, reset potential after spike V_{res} and absolute refractory period τ_{ref}, or synapses, such as synaptic time constant τ_{syn} and synaptic transmission delay Δ_{syn}, can integrate the model. These latter parameters definitely affect the neural network activity and behavior but do not constitute the network structural parameters.

Our rescaling method is dependent on a single parameter k positive in the interval $]0, \infty[$, which is used to resize down ($k \in]0, 1[$) or up ($k \in]1, \infty[$) the numbers of neurons, connections, external inputs, and synaptic weights, while maintaining fixed the function of connection $\mathscr{F}(o_{pre}, o_{post})$ and the proportions of cells per subpopulation of neurons.

The method is able to preserve the first and second-order statistics of network activity, by which we mean the population (or layer)-specific average firing rates, the synchrony level, the irregularity features of the spike trains, so that the overall behavior of the resized network is similar to the one of the full version. This happens essentially because the method keeps fixed (i) the probabilities and pattern of connections, (ii) the average random external input to the network (Van Albada et al., 2015), and (iii) the ratio between the firing threshold and the square root of the number of connections (Vreeswijk and Sompolinsky, 1998).

2.2.1 Rescaling Method algorithm

The algorithm of the rescaling method is available on GitHub (`https://github.com/ceciliaromaro/recoup-the-first-and-second-order-statistics-of-neuron-ne -network-dynamics`). It was also previously described in a conference (Romaro et al., 2018) and appeared as an arXiv preprint (Romaro et al., 2020a). The algorithm has four steps:

- **Step 1:** Decrease the number of neurons and the external input per neuron by multiplying them by the scale factor while keeping the proportions of cells per population fixed;
- **Step 2:** Decrease the number of connections per population by multiplying them by the square of the scale factor while keeping the functions of connections (probabilities) between populations unchanged;
- **Step 3:** Increase the synaptic weights by dividing them by the square root of the scale factor;
- **Step 4:** Provide each cell with a DC input current with a value corresponding to the total input lost due to rescaling.

The first three steps keep the proportional balance of neurons, external inputs and synaptic weights among the network populations. The fourth step changes the threshold to guarantee the neuron/population activity.

Network models are usually subdivided into more than one population or layer of neurons. Below, we give the algorithms for when the model has only one population or layer, and when it has more than one population or layer.

2.2.1.1 Rescaling method for 1 population/layer model

Algorithm 1 Rescaling method for model with 1 set of neurons

1: N the number of neurons.
2: C the probability of connection ($\mathscr{F}(o_{pre}, o_{post}) = C$).
3: X the total number of connections ($x = X/N$ the average number of connections per neuron).
4: X_{ext} the average number of external neurons connected to each of the N neurons.
5: w (pA or mV) the synaptic weight.
6: k the rescaling factor.
7: f_{ext} (Hz) the average firing rate of the external input.
8: f (Hz) the average firing rate of the set of neurons.
9: τ_{syn} (ms) synaptic time constant.
1.NUMBER OF NEURONS
10: $N' \leftarrow k * N$
11: $X'_{ext} \leftarrow k * X_{ext}$
2.NUMBER OF CONNECTIONS
12: $C' \leftarrow C$
13: $X' \leftarrow k^2 * X$ ▷ corolary of $X = C * N_{pre} * N_{pos}$
3. SYNAPTIC STRENGHT
14: $w' \leftarrow w/\sqrt{k}$
4. THRESHOLD ADJUSTMENT
15: $q_{sum} = w * f * x$
16: $q_{ext} = w * f_{ext} * X_{ext}$
17: $I'_{DC} = \tau_{syn} * ((1 - \sqrt{k}) * (q_{sum} + q_{ext}))$ ▷ Extra DC (pA or mV) input to compensate resize
18: Done! ▷ Notice that step 4 uses parameters without resizing.

Notice that if w is given in mV, it is not necessary to multiply the DC input by τ_{syn} in step 17. Instead, 17: $V'_{DC} = (1 - \sqrt{k}) * (q_{sum} + q_{ext})$ ▷ Extra DC (mV) input to compensate resizing. Notice that τ_m and C_m are neuronal parameters, not network parameters.

2.2.1.2 Rescaling method for a model with n layers or populations

The same idea used for a 1 layer or population model is recurrently applied for a many layers or populations model. One only has to pay attention to calculate the compensation threshold current correctly: a weighted average of number of connections, firing rate and weight of each presynaptic layer.

Algorithm 2 Rescaling method for model with n sets of neurons

n the number of layers/sets of neurons.
N_i the number of presynaptic neurons in layer i. ($i \in n$).
N_j the number of postsynaptic neurons in layer j. ($j \in n$).
C_{ij} the probability of connection from layer i to layer j ($\mathscr{F}(i,j) = C_{ij}$).
X_{ij} the total number of connections between layer i and layer j. ($x_j = X_{ij}/N_j$ the average number of received connections per neuron).
$X_{ext,j}$ the average number of external neurons connected to each neuron of layer j.
w_{ij} (pA or mV) the average synaptic weight of synapses from neurons in layer i to neurons in layer j.
$w_{ext,j}$ (pA or mV) the average synaptic weight of synapses from $X_{ext,j}$ to layer j.
k the rescaling factor.
$f_{ext,j}$ (Hz) the average firing rate of the external input to the set of neurons j.
f_i (Hz) the average firing rate of the presynaptic neurons.
τ_{syn} (ms) synaptic time constant.

1.NUMBER OF NEURONS

for each j in n **do**
    $N'_j \leftarrow k * N_j$
    $X'_{ext,j} \leftarrow k * X_{ext,j}$

2.NUMBER OF CONNECTIONS

for each j in n **do**
    for each i in n **do**
        $C'_{ij} \leftarrow C_{ij}$
        $X'_{ij} \leftarrow k^2 * X_{ij}$ ▷ corolary of X_{ij} = $C_{pre,pos}$*N_i*N_j

3. SYNAPTIC STRENGHT

for each j in n **do**
    for each i in n **do**
        $w'_{ij} \leftarrow w_{ij}/\sqrt{k}$
    $w'_{ext,j} \leftarrow w_{ext,j}/\sqrt{k}$

4. THRESHOLD ADJUSTMENT

for each j in n **do**
    $q_{sum_j} = \sum_{i=1}^{n} w_{ij} * f_i * X_{ij}/N_j$
    $q_{ext,j} = w_{ext,j} * f_{ext,j} * X_{ext,j}$
    $I'_{DC_j} = \tau_{syn} * ((1 - \sqrt{k}) * (q_{sum_j} + q_{ext,j}))$ ▷ DC (pA or mV) input to compensate resize

Done!

2.2.2 Rescaling Applied

The objective of this section is to present some application examples of the rescaling method to random networks and show that the method preserves first- and second-order statistical properties of the original model.

All applications of the rescaling method presented here were implemented in Python (with Brian2 or NetPyNE) and can be found on GitHub (`https://github.com/ceciliaromaro/recoup-the-first-and-second-order-statistics-of-neuron-network-dynamics`). The neuron model is the leaky integrate-and-fire (LIF) model.

2.2.2.1 Sparsely connected random network of inhibitory neurons

For any pre-synaptic neuron i and any post-synaptic neuron j in the network, a fixed probability p of connection $i \rightarrow j$ will be called random intra connectivity. For a probability p lower than 0.1 we can say the intra connectivity is sparse (Vreeswijk and Sompolinsky, 1998). The first illustrative application of the rescaling method will be for a network of inhibitory neurons with sparse random intra connectivity ($p << 1$) and Poisson external input of parameter f_{ext} (Brunel, 2000).

In this application, we rescale the network to different levels (120%, 100%, 80%, 50%, 30%, 20%, 10%, 5%, 1%). The network parameters before and after rescaling to 1% are shown in Table 10. Figure 10 presents the raster plots for the full network (10A) and for the network scaled down to 1% (10B). It also presents, for the different rescale levels, the average network firing rate (a first-order statistic) (10C) and the coefficient of variation of the inter spike interval (irregularity) (a second-order statistic) (10D). The differences among the average frequencies are less than 8.5%, and the differences among the irregularities are less than 1%.

Parameter description	Variable	Full scale	Rescaling
Factor of rescaling	k	-	0.01
Number of inhibitory neurons	N_{-}	10^4	10^2
Number of external inputs to each neuron	X_{ext}	2300	23
Total number of intra connections	X	$1,000,000$	100
Weight of external excitatory synapse	$w \pm \delta w$ (pA)	30$\pm$3	300 $\pm$30
Probability of connection	p	0.05	0.05
Absolute refractory period	τ_{ref} (ms)	2	2
Synaptic time constant	τ_{syn} (ms)	0.5	0.5
Membrane time constant	τ_m (ms)	10	10
Synaptic transmission delay	$\Delta_t \pm \delta\Delta_t$ (ms)	1.5 ± 0.75	1.5 ± 0.75
Membrane capacitance	C_m (pF)	250	250
Inhibitory synaptic strength	g	-2	-2
Reset potential (mV)	V_{ret}	-65	-65
Fixed firing threshold (mV)	V_{th}	-45	-45
Average firing rate of the external input	f_{ext} (Hz)	8	8

Table 10: Specification of the sparsely connected random network of inhibitory neurons before and after rescaling down to 1%: parameters and metrics.

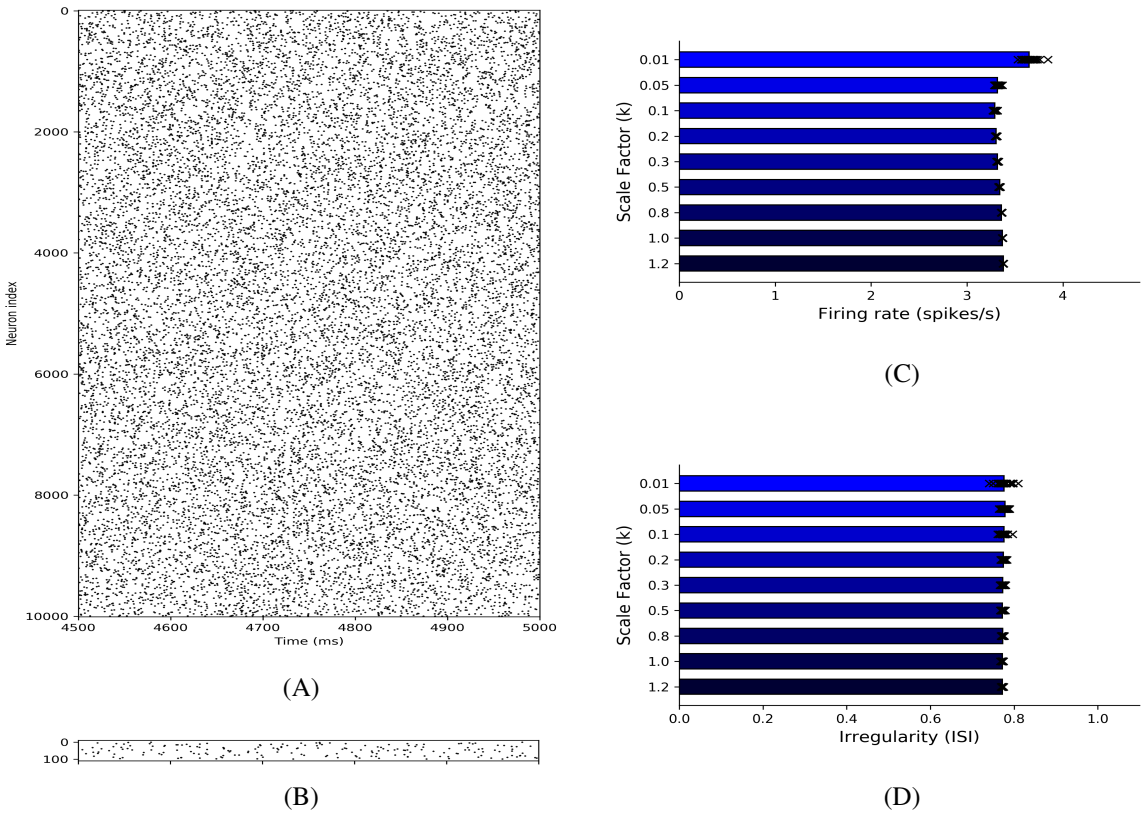


Figure 10: Sparsely connected random network of inhibitory neurons model. (A) Raster plot for the full network. (B) Raster plot for the network rescaled to 1% of the number of total neurons. (C) Average firing rate and (D) Irregularity of network neurons for different scale factors k (120%, 100%, 80%, 50%, 30%, 20%, 10%, 5%, 1%). The 'x' is value for each simulation run and the bar is the average of the set run.

2.2.2.2 Sparsely connected random network of excitatory and inhibitory neurons: avalanches

A neuronal avalanche can be defined as the rise of spiking activity above some basal or threshold level (Beggs and Plenz, 2003). The rise can be triggered by the activation of a few or a single neuron due to activity fluctuations within a randomly connected network (Dalla Porta and Copelli, 2019), producing a cascade of spikes that returns below threshold after some time. This process can have particular statistical properties like power law distributions of size and duration.

In other words, the avalanche is a quick rise in the network activity, locally or systemic, followed by a sudden decay of activity back to the previous activity level. It is a phenomenon that occurs spontaneously in networks of recurrently connected neurons (Hahn et al., 2010).

The second application of the rescaling method to be shown here is to a network composed of two populations of neurons, one excitatory and the other inhibitory. Neurons in the network, independently of being excitatory or inhibitory, are connected with sparse random intra con-

nectivity ($p << 1$) and receive Poisson external inputs. The weight of the excitatory synapses is w and the weight of the inhibitory synapses is gw, where g is a negative number ($|g|$ is the relative inhibitory synaptic weight (Brunel, 2000)). This network is similar to the first one with the difference that now part of the neurons are excitatory. The presence of excitatory neurons with recurrent connections is able to produce avalanches.

In this application, we rescale the network down to 50% of its original size. The network parameters before and after rescaling are given in Table 11. Figure 11 shows the raster plots and spike histograms for the full network and for the network rescaled to 50%. It is possible to see avalanches in both cases, indicating that rescaling does not hinders the occurrence of avalanches.

Parameter description	Variable	Full scale	Rescaling
Factor of rescaling	k	-	0.5
Number of excitatory neurons	N_+	4000	2000
Number of inhibitory neurons	N_-	1000	500
Number of external inputs to each neuron	X_{ext}	50	25
Total number of intra connections	X	250k	62.5k
Weight of excitatory synapse	$w \pm \delta w$ (pA)	400±40	566 ±57
Probability of connection	p	0.01	0.01
Absolute refractory period	τ_{ref} (ms)	2	2
Synaptic time constant	τ_{syn} (ms)	0.5	0.5
Membrane time constant	τ_m (ms)	10	10
Synaptic transmission delay	$\Delta_t \pm \delta\Delta_t$ (ms)	1.5 ± 0.75	1.5 ± 0.75
Membrane capacitance	C_m (pF)	250	250
Relative inhibitory synaptic weight	g	-4	-4
Reset potential (mV)	V_{ret}	-65	-65
Fixed firing threshold (mV)	V_{th}	-50	-50
Average firing rate of neurons	f (Hz)		
Average firing rate of the external input	f_{ext} (Hz)	8	8

Table 11: Specification of the sparsely connected random network of excitatory and inhibitory neurons before and after rescaling down to 50%: parameters and metrics.

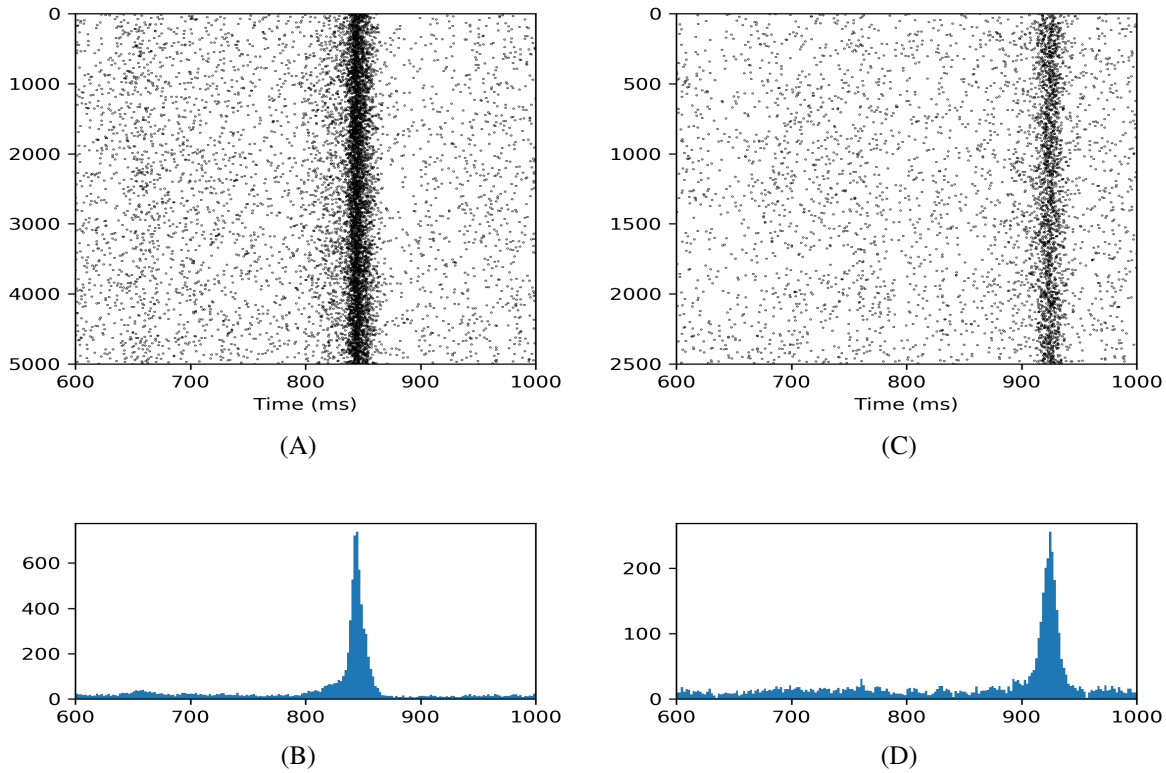


Figure 11: Network of excitatory and inhibitory interconnected neurons with avalanches. (A) Raster plot of the full network. (B) Spike histogram for the full network. (C) Raster plot of the network rescaled to 50% of its original size. (D) Spike histogram for network rescaled to 50% of its original size.

2.2.2.3 Sparsely connected random network of excitatory and inhibitory neurons: activity states

In this subsection we continue with the application of the rescaling method to a sparsely connected random network of excitatory and inhibitory neurons but the parameters are different from the one presented in the previous subsection (see below). Random networks of excitatory and inhibitory neurons are known to exhibit different activity states depending on the relative inhibitory synaptic strength and the magnitude of external excitatory input (Brunel, 2000; Vogels et al., 2005). Here, we will investigate whether the rescaled versions of these networks also exhibit a similar parameter dependency.

Inspired by the network introduced by Brunel (2000), the sparse random intra connectivity of the network considered here will be $p = 0.1$. The remaining network parameters are also based on the Brunel network and are shown in Table 12. We will assume that each neuron receives a fixed external DC input, which will be given in units of voltage (by incorporating the membrane resistance) as a multiple m of the threshold voltage for firing: $\Theta = mV_{th}$. The

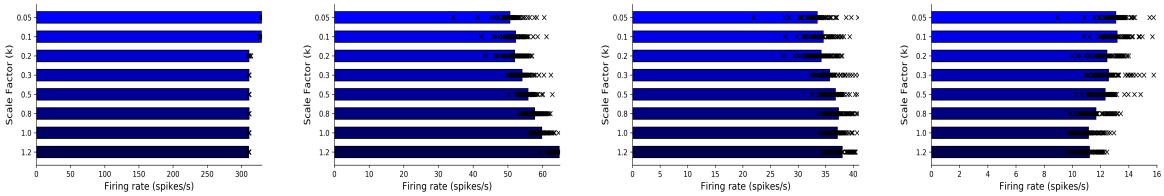
network behavior depends on the relative inhibitory synaptic strength g and the DC external input Θ. Depending on the combination of these two parameters, the network displays different activity states with distinct characteristics: average firing rate, synchrony and irregularity.

Parameter description	Variable	Full scale	Rescaling
Factor of rescaling	k	-	0.05
Number of excitatory neurons	N_+	10000	500
Number of inhibitory neurons	N_-	2500	125
Number of external inputs to each neuron	X_{ext}	1	1
Total number of intra connections	X	15.6M	39k
Weight of excitatory synapse	$w \pm \delta w$ (mV)	0.1	0.45
Probability of connection	p	0.1	0.1
Absolute refractory period	τ_{ref} (ms)	2	2
Membrane time constant	τ_m (ms)	20	20
Membrane capacitance	C_m (pF)	250	250
Synaptic transmission delay	$\Delta_t \pm \delta\Delta_t$ (ms)	1.5	1.5
Reset potential (mV)	V_{ret}	10	10
Fixed firing threshold (mV)	V_{th}	20	20

Table 12: Specification of the sparsely connected random network of excitatory and inhibitory neurons inspired by the Brunel model before and after rescaling down to 5%: parameters and metrics.

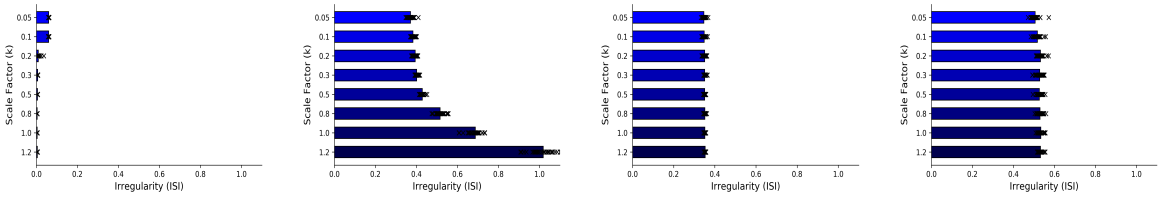
Figure 12 shows the average network firing rate and Figure 13 shows the coefficient of variation of the inter-spike intervals of network neurons (irregularity) for varying degrees of network rescaling (120%, 100%, 80%, 50%, 30%, , 20%, 10%, 5%) and four combinations of parameters (g, Θ): (A) $(g = 3, \Theta = 2V_{th})$; (B) $(g = 6, \Theta = 4V_{th})$, (C) $(g = 5, \Theta = 2V_{th})$, and (D) $(g = 4.5, \Theta = 1.001V_{th})$. In general, the variability of the average network firing rate with rescaling is low. The highest variabilities occur for the parameter sets $(g = 6, \Theta = 4V_{th})$, with relative deviation in relation to the full-scale value of 15% (Figure 12B), and $(g = 4.5, \Theta = 1.001V_{th})$, with relative deviation in relation to the full-scale value of 14.5% (Figure 12D). Regarding the changes in irregularity caused by rescaling, they are generally very small or non existent, with the exception of the parameter set $(g = 6, \Theta = 4V_{th})$. In this case the relative deviation in relation to the full-scale irregularity is of 43% (Figure 13B). Remarkably, rescaling the network to less than 50% of its original size decreases the irregularity from values greater

than 0.5 (a usual sign of irregular firing (Feng and Brown, 1999)) to values smaller than 0.5, which usually indicates regular spiking (Feng and Brown, 1999).



(A) (B) (C) (D)

Figure 12: Average firing rate of the sparsely connected random network for different rescaling sizes (120%, 100%, 80%, 50%, 30%, , 20%, 10%, 5%) and parameters (g, Θ). (A) $g = 3$ and $\Theta = 2V_{th}$; (B) $g = 6$ and $\Theta = 4V_{th}$, (C) $g = 5$ and $\Theta = 2V_{th}$, and (D) $g = 4.5$ and $\Theta = 1.001V_{th}$. The 'x' is value for each simulation run, which lasted for 10s, and the bar is the average of the set run, which was comprised of 30 runs.



(A) (B) (C) (D)

Figure 13: Average CV of ISI (irregularity) of the sparsely connected random network for different rescaling sizes (120%, 100%, 80%, 50%, 30%, , 20%, 10%, 5%). (A) $g = 3$ and $\Theta = 2V_{th}$; (B) $g = 6$ and $\Theta = 4V_{th}$, (C) $g = 5$ and $\Theta = 2V_{th}$, and (D) $g = 4.5$ and $\Theta = 1.001V_{th}$. The 'x' is value for each simulation run, which lasted for 10s, and the bar is the average of the set run, which was comprised of 30 runs.

Figure 14 shows the raster plots and spike count histograms for the full network and the rescaled network to 25% of original size for two sets of (g, Θ) parameters: $(g = 6, \Theta = 4V_{th})$, and $(g = 5, \Theta = 2V_{th})$. The raster plots allow a view of the single-cell behavior while the spike count histograms allow a view of the global network activity for each of these parameter sets. For the parameter set $(g = 6, \Theta = 4V_{th})$, the full scale network displays fast global oscillatory activity (Figure 14B) and irregular neuronal spiking (Figure 14A; see also Figure 13B). This corresponds to the state called "fast, synchronous irregular" (fast SI) by Brunel (Brunel, 2000). When the network with this parameter set is rescaled to 25% of the original size, the global oscillatory activity does not seem as synchronous as in the full scale network, at least not visually (Figure 14C). On the other hand, for the parameter set $(g = 5, \Theta = 2V_{th})$, the full scale network displays asynchronous global activity (Figure 14F) and irregular neuronal spiking (Figure 14E). This corresponds to the state called "asynchronous irregular" (AI) by

Brunel (Brunel, 2000). This activity state is preserved in the network when its size is rescaled to 25% of the original size (Figures 14G and 14G).

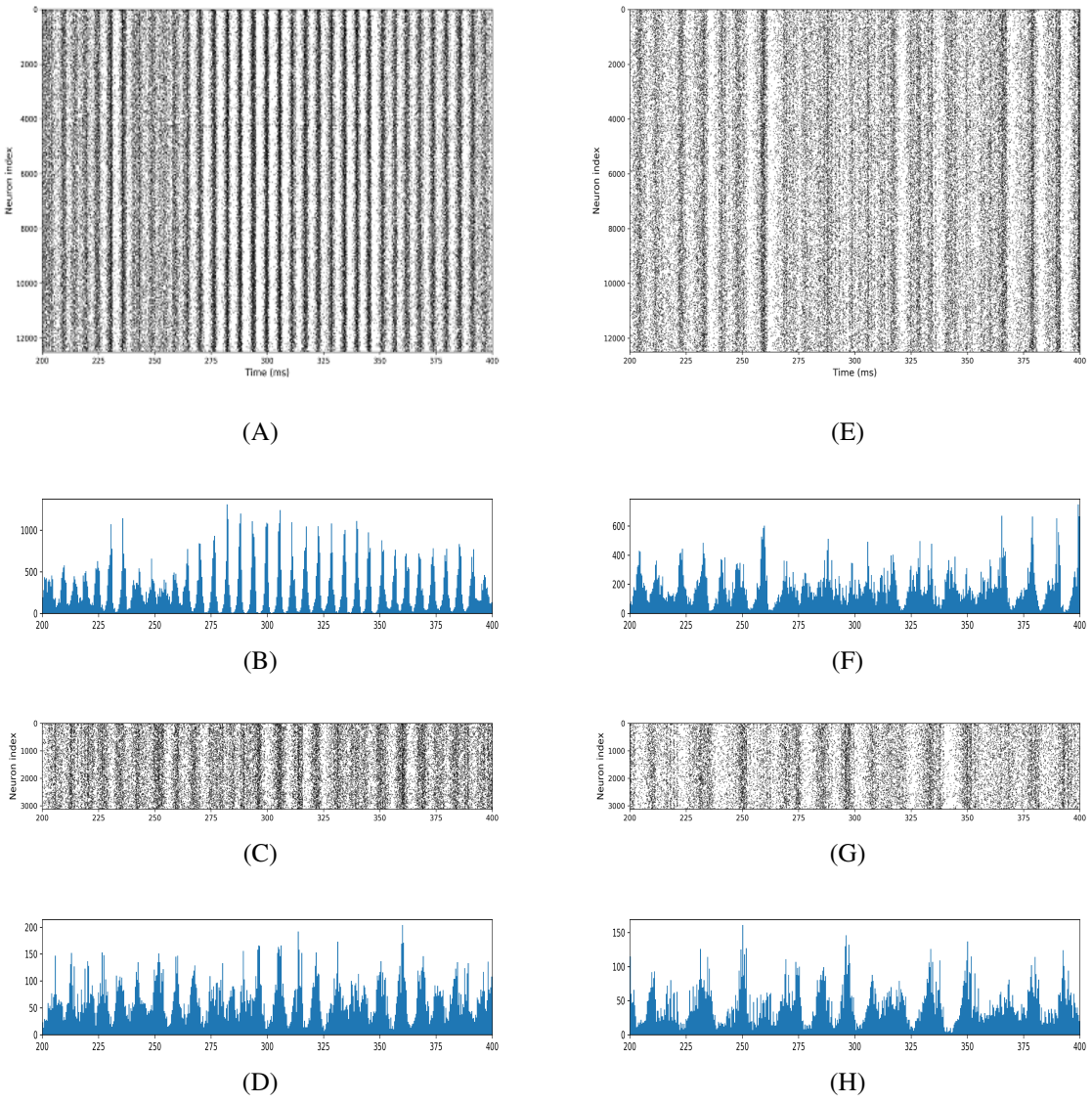


Figure 14: Distinct network activity states for two different sets of (g, Θ) parameters. Left: $(g = 6, \Theta = 4V_{th})$. Right: $(g = 5, \Theta = 2V_{th})$. (A) Raster plot for the full scale network. (B) Spike count histogram for the full scale network. (C) Raster plot for the network rescaled to 25% of the original size. (D) Spike count histogram for the network rescaled to 25% of the original size. (E) Raster plot for the full scale network. (F) Spike count histogram for the full scale network. (G) Raster plot for the network rescaled to 25% of the original size. (H) Spike count histogram for the network rescaled to 25% of the original size.

2.2.2.4 The Potjans-Diesmann network: eight populations of excitatory and inhibitory interconnected neurons

The Potjans-Diesmann (PD) model reproduces well the local connectivity of early sensory cortices (Potjans and Diesmann, 2014a). The model has four layers with populations of excitatory and inhibitory neurons in each of them, comprising eight cell populations (indicated as L2e, L2i, L4e, L4i, L5e, L5i, L6e and L6i). Based on experimental evidence, the model gives the amount of neurons per population and an 8×8 probability matrix which specifies the probability of a synaptic contact from a neuron of one population to a neuron of another population. Neurons from each population receive population-specific numbers of external inputs modeled as Poissonian spike trains with rate f_{ext}. The PD model is able to reproduce the layer-specific average firing rates of early sensory cortex observed in *vivo*.

The application of the rescaling method to the PD network model has been discussed elsewhere (Romaro et al., 2020a). Here, we will give a summary of the main results. Table 13 shows the parameters used in the construction of the model and for the rescaling to 30% of the original size.

Parameter description	Variable	Full scale	Rescaling
Factor of rescaling	k	-	0.3
Number of excitatory neurons	N_+	$\approx$ 62k*	$\approx$18.5k
Number of inhibitory neurons	N_-	$\approx$ 15k*	$\approx$4.5k
Number of external inputs to each neuron	X_{ext}	$\approx$ 2k*	$\approx$600
Total number of intra connection	X	$\approx$300M	$\approx$27M
Weight of excitatory synapse	$w \pm \delta w$ (pA)	87.8$\pm$8.8	160 $\pm$16
Probability of connection	p	C*	C
Absolute refractory period	τ_{ref} (ms)	2	2
Synaptic time constant	τ_{syn} (ms)	0.5	0.5
Membrane time constant	τ_m (ms)	10	10
Synaptic transmission delays	$\Delta_t \pm \delta\Delta_t$ (ms)	Δ_t*	Δ_t
Membrane capacitance	C_m (pF)	250	250
Relative inhibitory synaptic weight	g	-4	-4
Reset potential (mV)	V_{ret}	-65	-65
Fixed firing threshold (mV)	V_{th}	-50	-50
Average firing rate of neurons	f (Hz)	f*	f
Average firing rate of the external input	f_{ext} (Hz)	8	8

Table 13: Specification of the PD model and its rescaled version to 30% of the full size: parameters and metrics. (*) The synaptic delay depends on the layer and neuron type. The values for all layers and neuron types are given in Table 5 of the original article by Potjans and Diesmann (Potjans and Diesmann, 2014a).

Figure 15 shows the raster plots, the average network firing rate per layer (a first order statistic), and the irregulariry per layer (a second order statistic) for both the full scale network and the rescaled network to 30% of the original size. The rescaled model to 30% of original size has less than 10% of the total number of original connections (k^2X), and the network behavior and the first- and second-order statistics per layer were maintained. In (Romaro et al., 2020a), rescalings to different factors of the original network size, down to 1%, were studied and the overal network behavior, including the first- and second-order statistics, were generally maintained.

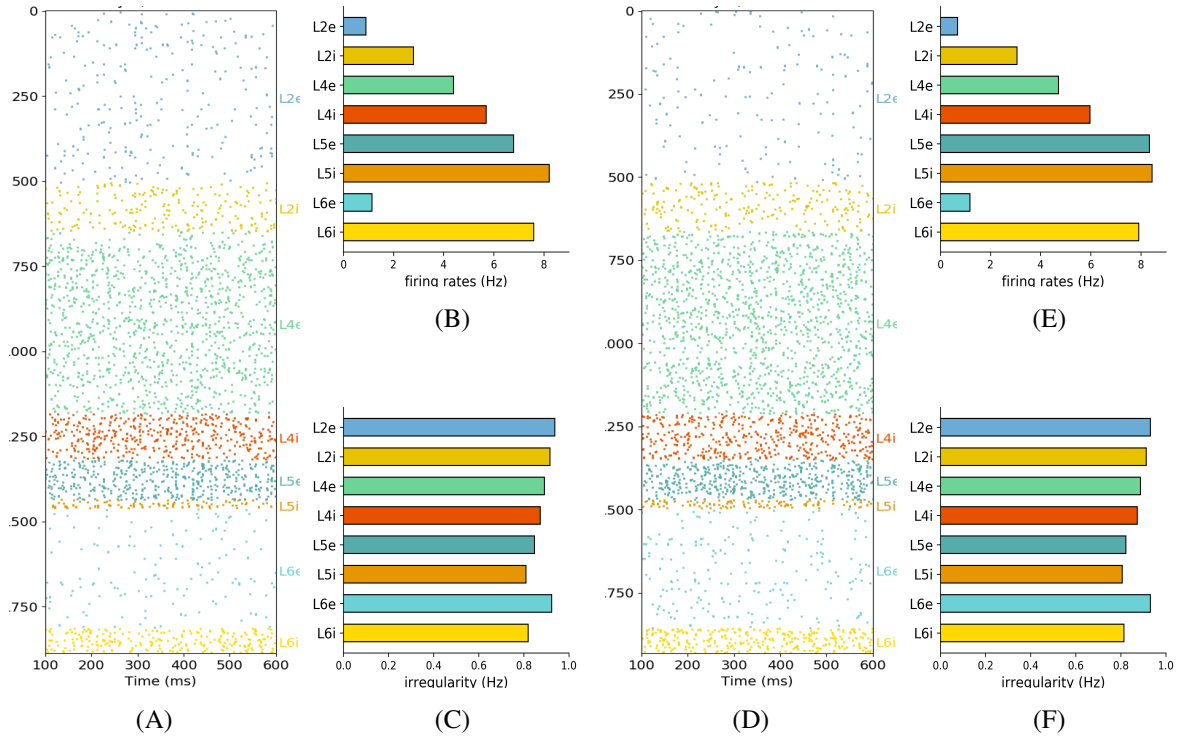


Figure 15: Behavior of the PD model in its full-scale version and its version rescaled to 30% of the original size. (A) Raster plot of spiking activity for the eight neuron populations (indicated by different colors as shown in the plot). The number of neurons shown per layer is proportional to the relative number of neurons per layer in the full scale model, resulting in a total number of 1850 neurons plotted. (B) Average firing rates of the spiking activity shown in (A). (C) Irregularity of single-unit spike trains over 60 seconds. (D) Raster plot, (E) Average firing rate and (F) Irregularity, as in (A,B,C) for the model rescaled to 30% of original size. The simulation times and number of neurons shown in the plots were chosen as in the full-scale model.

2.2.3 Model requirements, mathematical explanation and method limitations

The rescaling method works in any neural network that satisfies the following requirements (Brunel, 2000):

- the synaptic weight, which represents the postsynaptic potential (PSP) amplitude, is small compared to the voltage amplitude from voltage reset to firing threshold ($w << V_{th} - V_{ret}$);
- the probability of connection is small ($p << 1$).

The mathematical reason for these requirements is that, for these networks, the second-order statistics are dependent on the number of received connections x and the square of the synaptic weight w^2 (Vreeswijk and Sompolinsky, 1998; Van Albada et al., 2015).

The rescaling method gives:

$$w' = \frac{w}{\sqrt{k}}, \tag{2.2.1}$$

$$x' = kx = \frac{k^2 X}{kN}. \tag{2.2.2}$$

This maintains xw^2 and, therefore, the second order statistics.

The first order statistics depends on the number of received connections, $x = X/N$, and the synaptic weight, w. So, the fourth step of the method (see **Step 4** in section 2.2.1) provides a DC input to compensate for the loss of $(1 - \sqrt{k})$ in the first order statistic.

More formally, this method works in any sparsely connected random network model where the input $u(t)$ to a neuron can be written as a mean term plus a fluctuating Gaussian term (Brunel, 2000):

$$u(t) = \mu(t) + \sigma\eta(t), \tag{2.2.3}$$

where $\mu(t)$ is the mean term, σ is the standard deviation and $\eta(t)$ is an uncorrelated Gaussian white noise with zero mean, so that $\sigma\eta(t)$ is the fluctuating term. Furthermore, the mean and fluctuating terms are composed of internal and external contributions:

$$\mu(t) = \mu_{int}(t) + \mu_{ext}(t), \tag{2.2.4}$$

$$\sigma^2(t) = \sigma^2_{int}(t) + \sigma^2_{ext}(t). \tag{2.2.5}$$

2.2.3.1 Sparsely connected random network of excitatory and inhibitory neurons

For a sparsely connected random network of excitatory and inhibitory neurons (see Section 2.2.2.3), the average number of received intra connections per neuron, x, is given by

$$x = x_e + x_i = \frac{X}{N}, \tag{2.2.6}$$

where x_e is the average number of received excitatory connections per neuron and x_i is the average number of received inhibitory connections per neuron.

$$X/N = x = x_e + x_i, \tag{2.2.7}$$

The mean $\mu(t)$ and variance $\sigma^2(t)$ of the neuron input are given by:

$$\mu(t) = xw\left(1 - g\frac{x_i}{x_e}\right)\tau f(t - \Delta_t) + X_{ext}w\tau f_{ext}, \tag{2.2.8}$$

$$\sigma^2(t) = xw^2\left(1 + g^2\frac{x_i}{x_e}\right)\tau f(t - \Delta_t) + X_{ext}w^2\tau f_{ext}. \tag{2.2.9}$$

Substituting (2.2.1) and (2.2.2) in (2.2.9), one obtains:

$$\sigma'^2(t) = kx\left(\frac{w}{\sqrt{k}}\right)^2(1 + g^2x_i/x_e)\tau f(t - \Delta_t) + kX_{ext}\left(\frac{w}{\sqrt{k}}\right)^2\tau f_{ext} = \sigma^2(t). \tag{2.2.10}$$

In words, the input variance after rescaling is equal to the input variance before rescaling. This guarantees that the second-order statistics is kept.

To deal with the first-order statistics, we will start by substituting (2.2.1) and (2.2.2) in (2.2.8):

$$\mu'(t) = kx\left(\frac{w}{\sqrt{k}}\right)\left(1 - g\frac{x_i}{x_e}\right)\tau f(t - \Delta_t) + kX_{ext}\left(\frac{w}{\sqrt{k}}\right)\tau f_{ext}. \tag{2.2.11}$$

According to the fourth step of the rescaling method, a DC input is added to each neuron of the rescaled network. This current is given by

$$I'_{DC} = (1 - \sqrt{k})\left[xw\left(1 - g\frac{x_i}{x_e}\right)\tau f(t - \Delta_t) + X_{ext}w\tau f_{ext}\right]. \tag{2.2.12}$$

Thus, the new $\mu'(t)$ is:

$$\begin{aligned}\mu'(t) &= kx\left(\frac{w}{\sqrt{k}}\right)\left(1 - g\frac{x_i}{x_e}\right)\tau f(t - \Delta_t) + kX_{ext}\left(\frac{w}{\sqrt{k}}\right)\tau f_{ext} \\ &\quad + (1 - \sqrt{k})\left[xw\left(1 - g\frac{x_i}{x_e}\right)\tau f(t - \Delta_t) + X_{ext}w\tau f_{ext}\right] &(2.2.13)\\ &= \mu(t). &(2.2.14)\end{aligned}$$

The mean input is the same, before and after rescaling. Hence, the first-order statistics is also preserved.

2.2.3.2 Any network: Excitatory and inhibitory neurons interconnected

For any network with n sets of neurons, where each set may be made of inhibitory ($w_{pre} < 0$) or excitatory ($w_{pre} > 0$) neurons:

$$\mu_{post}(t) = \sum_{pre=1}^{n} x_{pre,post} w_{pre,post} f_{pre,post} \tau_{pre,post} + X_{ext,post} w_{ext,post} f_{ext,post} . \tau_{ext,post}, \tag{2.2.15}$$

$$\sigma^2_{post}(t) = \sum_{pre=1}^{n} x_{pre,post} w^2_{pre,post} f_{pre,post} \tau_{pre,post} + X_{ext,post} w^2_{ext,post} f_{ext,post} \tau_{ext,post}. \tag{2.2.16}$$

Hence, replacing (2.2.1) and (2.2.2) in (2.2.16),

$$\begin{aligned} \sigma'^2_{post}(t) = & \\ & = \sum_{pre=1}^{n} k x_{pre,post} \left(\frac{w_{pre,post}}{\sqrt{k}} \right)^2 \tau_{pre,post} \\ & + k X_{ext,post} \left(\frac{w_{ext,post}}{\sqrt{k}} \right)^2 f_{ext,post} \tau_{ext,post} \\ & = \sigma^2_{post}(t), \end{aligned} \tag{2.2.17}$$

which means that the second order statistics is preserved.

Now, replacing (2.2.1) and (2.2.2) in (2.2.15):

$$\mu'_{post}(t) = \sum_{pre=1}^{n} k x_{pre,post} \frac{w_{pre,post}}{\sqrt{k}} f_{pre,post} \tau_{pre,post}) + k X_{ext,post} \frac{w_{ext,post}}{\sqrt{k}} f_{ext,post} \tau_{ext,post}. \tag{2.2.18}$$

The fourth step of the rescaling method adds a DC input to each neuron of the rescaled network,

$$
\begin{aligned}
I'_{DC_{post}} &= \\
&= (1-\sqrt{k})\Bigg(\sum_{pre=1}^{n} x_{pre,post} w_{pre,post} f_{pre,post} \tau_{pre,post} \\
&+ X_{ext,post} w_{ext,post} f_{ext,post} \tau_{ext,post}\Bigg),
\end{aligned} \tag{2.2.19}
$$

Thus, the new $\mu'(t)$ is given by:

$$
\begin{aligned}
\mu'_{post}(t) = &\sum_{pre=1}^{n} k x_{pre,post} \frac{w_{pre,post}}{\sqrt{k}} f_{pre,post} \tau_{pre,post}) \\
&+ k X_{ext,post} \frac{w_{ext,post}}{\sqrt{k}} f_{ext,post} \tau_{ext,post} \\
&+ (1-\sqrt{k}) \sum_{pre=1}^{n} x_{pre,post} w_{pre,post} f_{pre,post} \tau_{pre,post} \\
&+ (1-\sqrt{k}) X_{ext,post} w_{ext,post} f_{ext,post} \tau_{ext,post} \\
&= \mu_{post}(t).
\end{aligned} \tag{2.2.20}
$$

So, the first order statistics is also preserved by the rescaling method.

2.2.3.3 Rescaling limit and oscillation

The size limit of rescaling happens when w becomes so large that the first requirement of the method ($w << V_{th} - V_{rt}$) is no longer satisfied.

In case the rescaled model does not behave as the original one at a smaller scale, a possible solution is to increase the random input, which means, to artificially add an external random input and compensate it on the threshold. A massive random external input guarantees the network operation at a stable point because it reduces the perturbation caused by sub or over input to a neuron generated by the internal connectivity. It reduces the ratio between the mean internal input (μ_{int}) or the variance of the internal input (σ^2_{int}) and the total mean input (μ) or total variance of the input (σ^2). It avoids changing the previous balance point of the network activity.

Additionally, the rescaling method does not introduce any resonance or oscillation. Instead, it tends to prevent oscillations such as in the application to a sparsely connected random network of inhibitory neurons (Section 2.2.2.1). This is due to the reduction of the ratio between the

mean internal input (μ_{int}) and the total variance of the input (σ^2).

More formally, let us recall Equation (30) from the article by Brunel (Brunel, 2000), which, in the notation adopted here, reads:

$$\begin{aligned} G &= \frac{xwf\tau(gx_i/x_e - 1)}{\sigma} = \frac{-\mu_{int}}{\sigma}, \\ H &= \frac{xw^2f\tau(g^2x_i/x_i + 1)}{\sigma^2} = \frac{\sigma_{int}^2}{\sigma^2}, \end{aligned} \tag{2.2.21}$$

where μ_{int} is the mean internal input and σ_{int}^2 is the variance of the internal input.

The synchronous oscillation in Figure 14B (and in Figure 8B in (Brunel, 2000)) is due to $H \to 1$ and $G \sim \sqrt{\frac{\tau}{\Delta_t}}$. Once the rescaling method does not change the ratio indicated by H in (2.2.21) nor σ but decreases μ_{int}, the ratio G in (2.2.21) is not satisfied anymore and, consequently, the synchronous oscillatory activity is reduced as seen in Figure 14D.

2.3 Boundary Correction Method

The key to this method is to apply the rescaling idea to each neuron o in the network (including the ones at the boundary). The rescaling factor k_o for each neuron o aims to compensate for the number of connections lost due to the existence of the boundary in comparison to the case in which the boundary did not exist (i.e. the network was infinite). The rescaling factor is calculated as a normalized connection density.

The normalized connection density is calculated in the following way: for a given neuron o, divide the actual number of connections received by the neuron by the total number of connections it would receive if the network had no boundaries (were infinite). For example, for a two-dimensional square lattice with nearest neighbor coupling, the normalized connection density for one of the neurons at the corners is $1/4 = 0.25$; for one of the neurons along the sides is $2/4 = 0.50$; and for a neuron in the middle of the lattice is $4/4 = 1$.

2.3.1 Boundary correction method algorithm

The boundary correction method numerically estimates the normalized connection density function in the first step, then weights each neuron connection based on this density, and finally balances the threshold to keep the neuron/layer activity.

The laborious part of this method is to calculate the normalized connection density func-

tion, which depends on the specific pattern of connections in each model. Below we give an algorithm that will work for any pattern of connection. However, if the normalized connection density function is analytically known, one can use it and start the boundary correction algorithm at step 2.

The algorithm of the boundary correction method can be found in GitHub (`https://github.com/ceciliaromaro/recoup-the-first-and-second-order-statistics-of-neuron-network-dynamics`) and its usage is illustrated in the application examples of Section 2.3.2 below. The algorithm has the following three steps:

- **Step 1:** Calculate the rescaling factor for any neuron o in the network as the normalized connection density;
- **Step 2:** Increase (or keep unchanged) the synaptic weights by dividing them by the square root of the rescaling factor;
- **Step 3:** Provide each cell with a DC input current with a value corresponding to the total input lost due to the network edge (in comparison to an infinite network).

2.3.1.1 Boundary correction method for a network with n sets/layers of neurons

More formally, the method can be described by the following pseudo-algorithm:

Algorithm 3 Boundary correction rescaling method for a network model with n sets of neurons

n the number of sets of neurons in the model.
N_j the (finite) set of postsynaptic neurons ($j \in n$).
N_i the (finite) set of presynaptic neurons ($i \in n$).
$\overline{X_{ij}}$ the average number of connections between N_i and one neuron in N_j if the model had no boundary.
x_{oj} the number of synapses to neuron o in N_j.
w_{oij} (pA or mV) the average synaptic weight of synapses from neurons in the set N_i to neuron o in N_j.
k'_{oj} the rescaling factor for neuron o from the set N_j (to be calculated).
$\overline{f_i}$ (Hz) the average firing rate of the set of neurons N_i.
τ_{syn} (ms) the synaptic time constant.

1. CALCULATION OF THE NORMALIZED CONNECTION DENSITY

for each set j in n **do**
    for each neuron o in N_j **do**
        $k'_{oj} \leftarrow x_{oj} / \sum_{i=1}^{n} (\overline{X_{ij}})$

2. SYNAPTIC STRENGHT

for each set j in n **do**
    for each neuron o in N_j **do**
        $w'_{oj} \leftarrow w_{oj} / \sqrt{k'_{oj}}$

3. THRESHOLD ADJUSTMENT

for each set j in n **do**
    for each neuron o in N_j **do**
        $c_{sum_{oj}} = w_{oj} * \sum_{i=1}^{n} \overline{f_i} * \overline{X_{ij}}$
        $I'_{DC_{oj}} = \tau_{syn} * (1 - \sqrt{k'_{oj}}) * c_{sum_{oj}}$ ▷ DC (pA or mV) input to compensate resize

Done!

2.3.2 Boundary correction applied

We applied the boundary correction method to the models presented in Sections 2.2.2.1, 2.2.2.3 and 2.2.2.4 with spatial extensions. In order to create a boundary problem for these models, one first needs to convert them to models with topographic patterns of connections. To do so, we assigned a spatial position to each neuron, then we connected neurons using a distance-dependent connectivity scheme in which the probability of connection decreases monotonically

as a function of distance following a Gaussian distribution with standard deviation σ_g, and, finally, we simulated the network with and without the boundary solution.

All applications of the boundary correction method presented here were implemented in Python (with Brian2) and can be found on GitHub (`https://github.com/ceciliaromaro/recoup-the-first-and-second-order-statistics-of-neuron-network-dynamics`

2.3.2.1 Sparsely connected random network of inhibitory neurons

All neurons from the model presented in Section 2.2.2.1 were homogeneously distributed on an area of 1 mm^2, constituting a two-dimensional square grid, and the connection probability between neuron pairs decayed with distance according to a Gaussian distribution with $\sigma_g = 0.25$ mm. Figure 16 shows the average firing rate of each neuron in the network grid before and after the boundary correction. Figure 16 also shows, before and after the boundary correction, the mean firing rate of neurons located at the boundary (around 50% of all neurons) and of neurons not located at the boundary (these will be called core neurons here), which correspond to about 50% of all neurons, and the network irregularity. Visually, the boundary neurons spike more than core neurons in the model without boundary correction because neurons at the border receive less inhibitory input than core neurons.

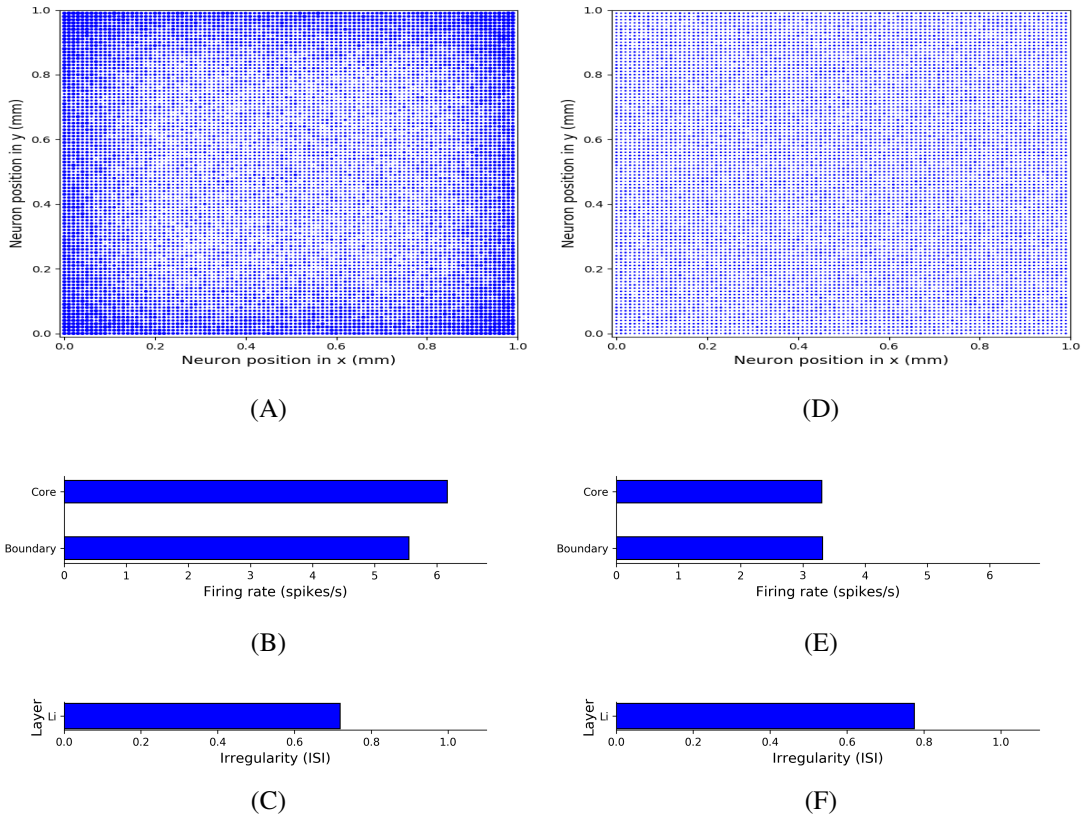


Figure 16: Sparsely connected random network of inhibitory neurons with spatial extension, with and without boundary correction. (A) Spatial network grid where each dot corresponds to a neuron and the dot size is proportional to the average firing rate, without boundary correction. (B) Average firing rate of core and boundary neurons without boundary correction. (C) Irregularity of neuronal spike trains without boundary correction. (D) Same grid plot as in (A) but with boundary correction. (E) Average firing rate of core and boundary neurons with boundary correction. (F) Irregularity of neuronal spike trains with boundary correction.

2.3.2.2 Sparsely connected random network of excitatory and inhibitory neurons

Excitatory and inhibitory neurons from the full version of the model presented in Section 2.2.2.3 were homogeneously distributed on two layers of equal area of 1 mm^2 (a layer of excitatory neurons and a layer of inhibitory neurons), constituting two bidimensional square grids, and the connection probability between neuron pairs decayed with distance according to Gaussian distributions with the same standard deviation of $\sigma_g = 0.150$ mm. We simulated the model, with and without the boundary correction, for the following parameter sets: ($g = 6, \Theta = 4V_{th}$) (Figure 17), ($g = 5, \Theta = 2V_{th}$) (Figure 18), ($g = 3, \Theta = 2V_{th}$) (Figure 19, left-hand side), and ($g = 4, \Theta = 1.001V_{th}$) (Figure 19, right-hand side).

Figures 17 and 18 show, for their corresponding parameter sets, the excitatory and inhibitory two-dimensional grids, where each dot corresponds to a neuron position and its size is proportional to the average firing rate of the neuron, for the situations without boundary correction

(Figures 17A and 17C; and Figures 18A and 18C) and with boundary correction (Figures 17B and 17D; and Figures 18B and 18B).

Figures 17 and 18 also show, for their corresponding parameter sets, the average firing rates of excitatory and inhibitory core and boundary neurons with and without boundary correction (Figures 17E, 17F and 17H; and Figures 18E, 18F and 18H), and the network irregularity with and without boundary correction (Figures 17G and 17I; and Figures 18G and 18I).

The raster plot of spiking activity and spike count histogram for the networks after the boundary correction are shown in Figures 17J and 17K (parameter set ($g = 6, \Theta = 4V_{th}$)) and Figures 18J and 18K (parameter set ($g = 5, \Theta = 2V_{th}$)).

Notice that the boundary correction method preserves the synchronous oscillatory state displayed by the original network (without spatial extension) with parameter set ($g = 6, \Theta = 4V_{th}$) (compare Figures 14B and 17K). This phenomenon will be explained in Section 2.3.3.

Figures 19A, 19B and 19C show the average firing-rate of core and boundary neurons, the irregularity and the spike-count histogram for the network with parameter set ($g = 3, \Theta = 2V_{th}$) with boundary correction. The same graphs are shown for the network with parameter set ($g = 4, \Theta = 1.001V_{th}$) in Figures 19D, 19E and 19F. However, in this latter case the external DC input Θ was replaced by an equivalent Poisson input.

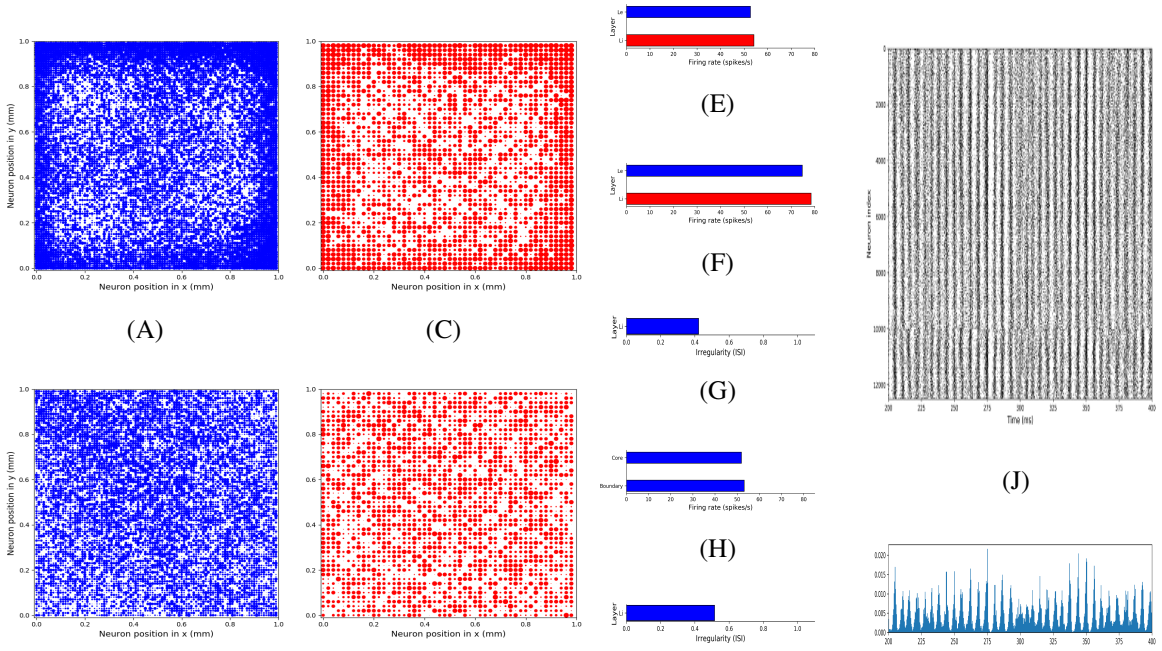


Figure 17: Sparsely connected random network of excitatory and inhibitory neurons with spatial extension, without and with boundary correction. Network with parameter set ($g = 6, \Theta = 4V_{th}$). (A) Grid of excitatory neurons without boundary correction. Each dot corresponds to a neuron position and the dot size is proportional to the firing rate. (B) Grid of excitatory neurons with boundary correction. Dot positions and sizes as in (A). (C) Grid of inhibitory neurons without boundary correction. Dot positions and sizes as in (A). (D) Grid of inhibitory neurons with boundary correction. Dot positions and sizes as in (A). (E) Average firing rate of excitatory (blue) and inhibitory (red) core neurons without boundary correction. (F) Average firing rate of excitatory (blue) and inhibitory (red) boundary neurons without boundary correction. (G) Irregularity of the single-unit spike trains without boundary correction. (H) Average firing rate of core (excitatory and inhibitory neurons lumped together) and boundary (excitatory and inhibitory neurons lumped together) neurons with boundary correction. (I) Irregularity of the single-unit spike trains with boundary correction. (J) Raster plot and (K) spike-count histogram of the network for 200 ms simulation. All others graphs showed results for 5 s simulation.

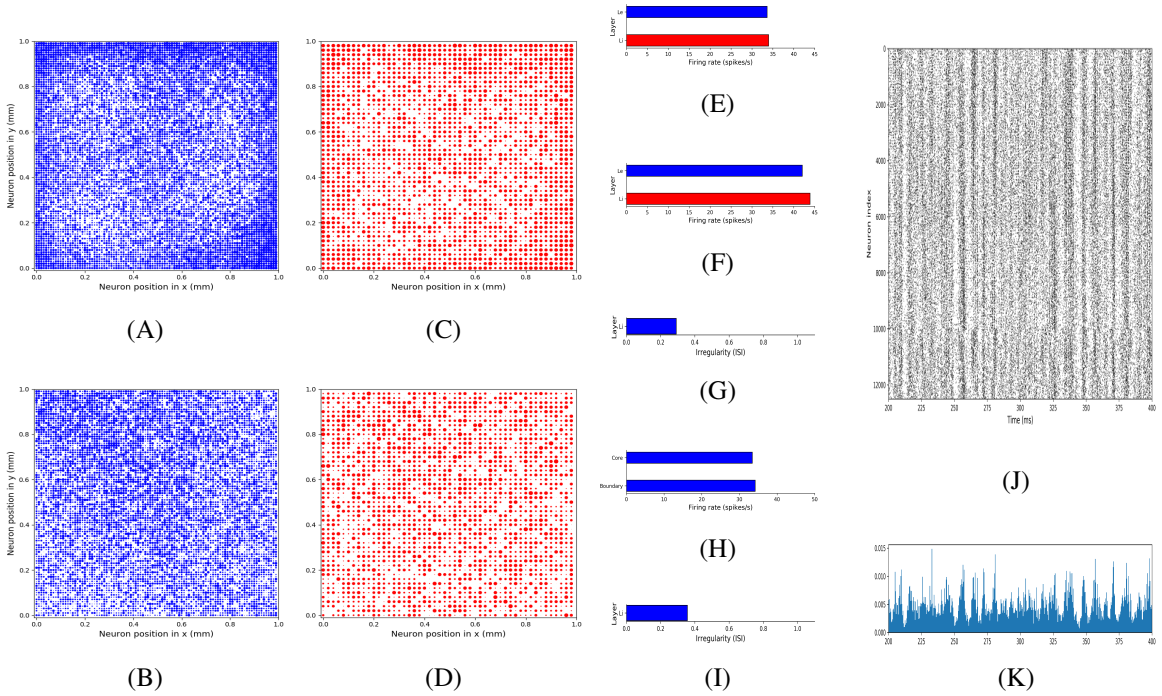


Figure 18: Sparsely connected random network of excitatory and inhibitory neurons with spatial extension, without and with boundary correction. Network with parameter set ($g = 5, \Theta = 2V_{th}$). (A) Grid of excitatory neurons without boundary correction. Each dot corresponds to a neuron position and the dot size is proportional to the firing rate. (B) Grid of excitatory neurons with boundary correction. Dot positions and sizes as in (A). (C) Grid of inhibitory neurons without boundary correction. Dot positions and sizes as in (A). (D) Grid of inhibitory neurons with boundary correction. Dot positions and sizes as in (A). (E) Average firing rate of excitatory (blue) and inhibitory (red) core neurons without boundary correction. (F) Average firing rate of excitatory (blue) and inhibitory (red) boundary neurons without boundary correction. (G) Irregularity of the single-unit spike trains without boundary correction. (H) Average firing rate of core (excitatory and inhibitory neurons lumped together) and boundary (excitatory and inhibitory neurons lumped together) neurons with boundary correction. (I) Irregularity of the single-unit spike trains with boundary correction. (J) Raster plot and (K) spike-count histogram of the network for 200 ms simulation. All others graphs showed results for 5 s simulation.

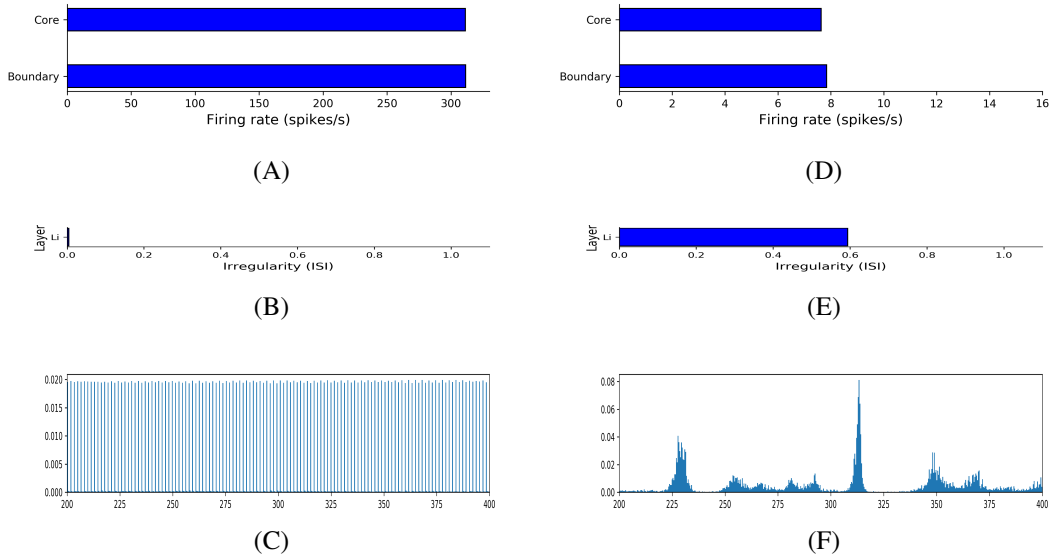


Figure 19: Sparsely connected random network of excitatory and inhibitory neurons with spatial extension and boundary correction. Network with parameter set ($g = 3, \Theta = 2V_{th}$) on the left-hand side and network with parameter set ($g = 4, \Theta = 1.001V_{th}$) on the right-hand side. (A) Average firing rate of core and boundary neurons (excitatory and inhibitory neurons lumped together). (B) Irregularity of spike-trains. (C) Spike-count histogram. (D), (E) and (F): Similar graphs of (A), (B) and (C) for the network with parameter set ($g = 4, \Theta = 1.001V_{th}$). Average firing rate and irregularity calculated for 5 s simulation and spike-count histogram calculated for 200 ms simulation.

2.3.2.3 Potjans-Diesmann model with spatial extension: somatosensory cortex model

For each one of the eight cell populations of the full-scale PD model (see Section 2.2.2.4), all neurons were homogeneously distributed on a layer of area of 1 mm^2 (so, at the end we had eight bidimensional square grids, one for each population), and the connection probability between neuron pairs decayed with distance according to Gaussian distributions with the same standard deviation of $\sigma_g = 0.275$ mm. To study the effect of the rescaling method on a network with spatial extension, this model was also submitted to a rescaling to 50% of its original size.

Figure 20 shows the effect of the boundary correction method on the full-scale PD model with spatial extension. Figure 20A shows the average firing-rates of L2e neurons (indicated by the sizes of the dots in the two-dimensional grid) without boundary correction, and Figure 20B shows the same plot for L2e after the boundary correction. Figures 20C and 20D show the same for L5e neurons. Figures 20E and 20F present the average firing rate of core neurons of all populations (around 50 % of all neurons) without boundary correction and with boundary correction, respectively. Figures 20G and 20H present the average firing rate of boundary neurons of all populations without boundary correction and with boundary correction, respectively. Figure 21 presents the same plots of Figure 20 for the network rescaled in 50% of original size.

The results depicted in Figures 20 and 21 show that it is possible to combine both the rescaling and the boundary correction methods. In all cases, the boundary neurons spike more than the core neurons in the model without boundary correction due to the lack of inhibitory input.

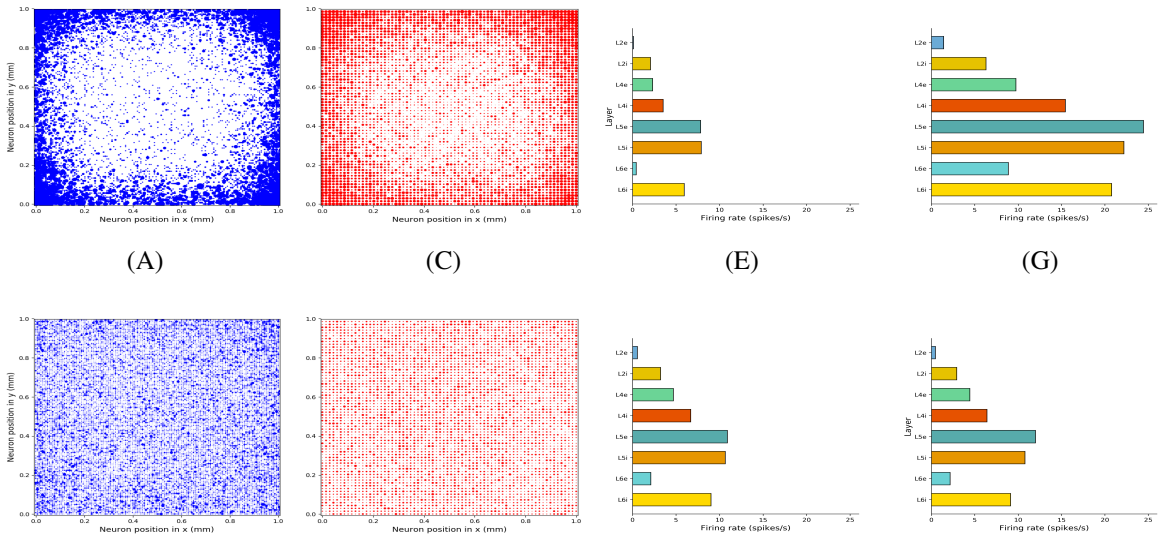
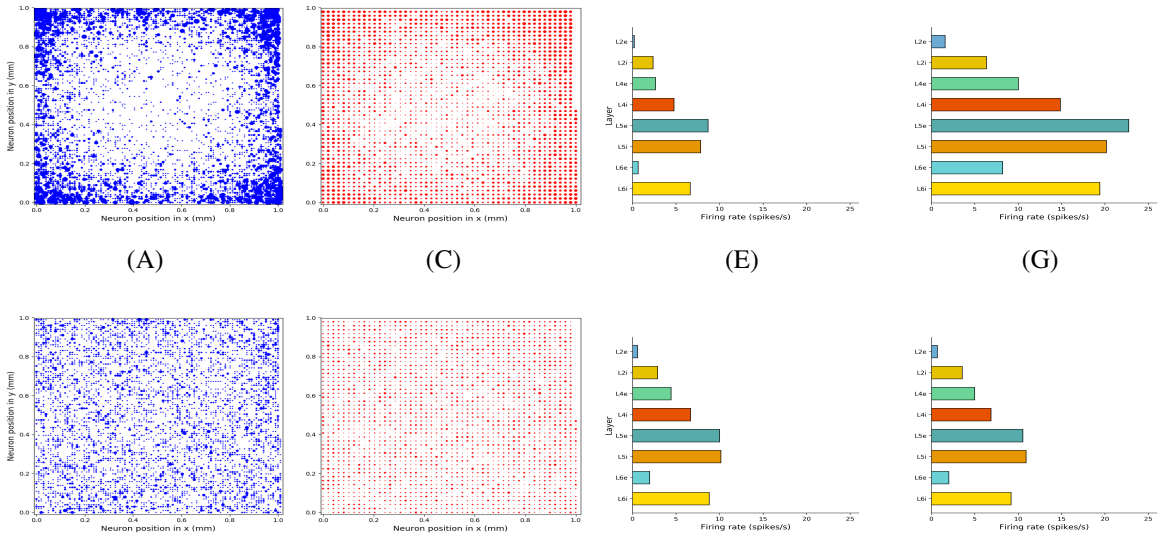


Figure 20: Full-scale PD model with spatial extension, with and without boundary correction. (A) Grid of L2 excitatory neurons without boundary correction. Each dot corresponds to a neuron position and the dot size is proportional to the firing rate. (B) Grid of L2 excitatory neurons with boundary correction. (C) Grid of L5 excitatory neurons without boundary correction. (D) Grid of L5 excitatory neurons with boundary correction. (E) Average firing rate of core neurons from all populations without boundary correction. (F) Average firing rate of core neurons from all populations with boundary correction. (G) Average firing rate of boundary neurons from all populations without boundary correction. (H) Average firing rate of boundary neurons from all populations with boundary correction.



(A) (C) (E) (G)

(B) (D) (F) (H)

Figure 21: PD model with spatial extension rescaled to 50 % of its original size, with and without boundary correction. (A) Grid of L2 excitatory neurons without boundary correction. Each dot corresponds to a neuron position and the dot size is proportional to the firing rate. (B) Grid of L2 excitatory neurons with boundary correction. (C) Grid of L5 excitatory neurons without boundary correction. (D) Grid of L5 excitatory neurons with boundary correction. (E) Average firing rate of core neurons from all populations without boundary correction. (F) Average firing rate of core neurons from all populations with boundary correction. (G) Average firing rate of boundary neurons from all populations without boundary correction. (H) Average firing rate of boundary neurons from all populations with boundary correction.

2.3.3 Model requirements, mathematical explanation and method limitations

The boundary correction method works in any network that satisfies the rescaling condition (Section 2.2.3) for a size reduction of down to 25% of the original size, which is the minimum rescaling to which a corner neuron can be submitted. This is a sufficient condition, though not a necessary condition.

The boundary solution method was able to retrieve the following properties of the original models which were lost when spatial extension was added to them:

2.3.3.1 Firing rate

1. Average firing rate of the full-scale random network model of inhibitory neurons. Compare the original firing rate in Figure 10C (for $k = 1$) with the firing rates for the model with spatial extension and no boundary correction (Figure 16B) and the firing rates for the model with spatial extension and boundary correction (Figure 16E).

2. Average firing rate of the full-scale random network of excitatory and inhibitory neurons (parameter set ($g = 6, \Theta = 4V_{th}$)). Compare the original firing rate in Figure 12B (for $k = 1$) with the firing rates for the model with spatial extension and no boundary correction (Figures 17E and 17F) and the firing rates for the model with spatial extension and boundary correction (Figure 17H).

3. Average firing rate of the full-scale random network of excitatory and inhibitory neurons (parameter set ($g = 5, \Theta = 2V_{th}$)). Compare the original firing rate in Figure 12C (for $k = 1$) with the firing rates for the model with spatial extension and no boundary correction (Figures 18E and 18F) and the firing rates for the model with spatial extension and boundary correction (Figure 18H).

4. Average firing rate of the full-scale random network of excitatory and inhibitory neurons (parameter set ($g = 3, \Theta = 2V_{th}$)). Compare the original firing rate in Figure 12C (for $k = 1$) with the firing rates for the model with spatial extension and boundary correction (Figure 19A).

5. Average firing rate of the full-scale random network of excitatory and inhibitory neurons (parameter set ($g = 4, \Theta = 1.001V_{th}$)). Compare the original firing rate in Figure 12D (for $k = 1$) with the firing rates for the model with spatial extension and boundary correction (Figure 19D).

6. Population-specific average firing rates of the full-scale PD model. Compare the original firing rates in Figure 15B with the firing rates for the model with spatial extension and no boundary correction (Figures 20E and 20G) and the firing rates for the model with spatial extension and boundary correction (Figures 20F and 20H).

7. Population-specific average firing rates of the rescaled PD model. Compare the original firing rates in Figure 15E with the firing rates for the model with spatial extension and no boundary correction (Figures 21E and 21G) and the firing rates for the model with spatial extension and boundary correction (Figures 21F and 21H).

2.3.3.2 Irregularity

1. Irregularity of the full-scale random network model of inhibitory neurons. Compare the original irregularity in Figure 10D (for $k = 1$) with the irregularity of the model with spatial extension and no boundary correction (Figure 16C) and the irregularity of the model with spatial extension and boundary correction (Figure 16F).

2. Irregularity of the full-scale random network of excitatory and inhibitory neurons (parameter set ($g = 5, \Theta = 2V_{th}$)). Compare the original irregularity in Figure 13C (for $k = 1$) with the irregularity of the model with spatial extension and no boundary correction (Figure 18G) and the irregularity of the model with spatial extension and boundary correction (Figure 18I).

3. Irregularity of the full-scale random network of excitatory and inhibitory neurons (parameter set ($g = 3, \Theta = 2V_{th}$)). Compare the original irregularity in Figure 13A (for $k = 1$) with the irregularity of the model with spatial extension and boundary correction (Figure 19B).

4. Irregularity of the full-scale random network of excitatory and inhibitory neurons (parameter set ($g = 4, \Theta = 1.001V_{th}$)). Compare the original irregularity in Figure 13D (for $k = 1$) with the irregularity of the model with spatial extension and boundary correction (Figure 19E).

It is possible that even networks that could lose some synchrony with the application of the rescaling to 25% of the original network size (see Figure 14D) could have the synchrony retrieved with the application of the boundary correction method (see Figures 18K and 14B). This is due to the weighting of neurons in networks that had the internal part of the average input μ_{int} (see equation (2.2.4)) reduced. If this term is low enough to do not perturb the ratio G (see Equation (2.2.21)), the network oscillatory proprieties remain approximately the same.

Acknowledgments

This work was produced as part of the activities of FAPESP Research, Disseminations and Innovation Center for Neuromathematics (Grant 2013/07699-0, S. Paulo Research Foundation). The author is the recipient of PhD scholarships from the Brazilian Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES). The author is thankful to Mauricio Girardi-Schappo, who encourage her to explain the method in a paper and author's little sister, Cinthia Romaro, who found time to comment on the manuscript even during her Harvard MBA.

References

Bassett, D. S., Zurn, P., and Gold, J. I. (2018). On the nature and use of models in network neuroscience. *Nature Reviews Neuroscience*, 19(9):566—578.

Beggs, J. M. and Plenz, D. (2003). Neuronal avalanches in neocortical circuits. *Journal of Neuroscience*, 23(35):11167–11177.

Brunel, N. (2000). Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. *Journal of Computational Neuroscience*, 8(3):183–208.

Dalla Porta, L. and Copelli, M. (2019). Modeling neuronal avalanches and long-range temporal correlations at the emergence of collective oscillations: continuously varying exponents mimic M/EEG results. *PLoS Computational Biology*, 15(4):e1006924.

Einevoll, G. T., Destexhe, A., Diesmann, M., Grün, S., Jirsa, V., de Kamps, M., Migliore, M., Ness, T. V., Plesser, H. E., and Schürmann, F. (2019). The scientific case for brain simulations. *Neuron*, 102(4):735–744.

Feng, J. and Brown, D. (1999). Coefficient of variation of interspike intervals greater than 0.5. how and when? *Biological Cybernetics*, 80(5):291–297.

Hagen, E., Dahmen, D., Stavrinou, M., Lindén, H., Tetzlaff, T., van Albada, S.J., G. S., Diesmann, M., and Einevoll, G. (2016). Hybrid scheme for modeling local field potentials from point-neuron networks. *Cerebral Cortex*, 26:4461–4496.

Hahn, G., Peterman, T., Havenith, M. N., Yu, S., Singer, W., Plenz, D., and Nikolić, D. (2010). Neuronal avalanches in spontaneous activity in vivo. *Journal of Neurophysiology*, 104(6):3312–3322.

Igarashi, J., Yamaura, H., and Yamazaki, T. (2019). Large-scale simulation of a layered cortical sheet of spiking network model using a tile partitioning method. *Frontiers in Neuroinformatics*, 13:71.

Iyer, R., Menon, V., Buice, M., Koch, C., and Mihalas, S. (2013). The influence of synaptic weight distribution on neuronal population dynamics. *PLoS Computational Biology*, 9(10):e1003248.

Izhikevich, E. and Edelman, G. (2008). Large-scale model of mammalian thalamocortical systems. *Proceedings of the National Academy of Sciences (USA)*, 105(9):3593–3598.

Joglekar, M. R., Mejias, J. F., Yang, G. R., and Wang, X.-J. (2018). Inter-areal balanced amplification enhances signal propagation in a large-scale circuit model of the primate cortex. *Neuron*, 98(1):222–234.

Jordan, J., Ippen, T., Helias, M., Kitayama, I., Sato, M., Igarashi, J., Diesmann, M., and Kunkel, S. (2018). Extremely scalable spiking neuronal network simulation code: from laptops to exascale computers. *Frontiers in Neuroinformatics*, 12:2.

Lansner, A. and Diesmann, M. (2012). Virtues, pitfalls, and methodology of neuronal network modeling and simulations on supercomputers. In Novére, N. L., editor, *Computational Systems Neurobiology*, chapter 10, pages 283–315. Dordrecht: Springer.

Lytton, W. W., Seidenstein, A. H., Dura-Bernal, S., McDougal, R. A., Schürmann, F., and Hines, M. L. (2016). Simulation neurotechnologies for advancing brain research: parallelizing large networks in neuron. *Neural Computation*, 28(10):2063–2090.

Markram, H., Muller, E., Ramaswamy, S., Reimann, M. W., Abdellah, M., Sanchez, C. A., Ailamaki, A., Alonso-Nanclares, L., Antille, N., Arsever, S., et al. (2015). Reconstruction and simulation of neocortical microcircuitry. *Cell*, 163(2):456–492.

Mazza, M., De Pinho, M., Piqueira, J. R. C., and Roque, A. C. (2004). A dynamical model of fast cortical reorganization. *Journal of Computational Neuroscience*, 16(2):177–201.

Mehring, C., Hehl, U., Kubo, M., Diesmann, M., and Aertsen, A. (2003). Activity dynamics and propagation of synchronous spiking in locally connected random networks. *Biological Cybernetics*, 88(5):395–408.

Merkt, B., Schüßler, F., and Rotter, S. (2019). Propagation of orientation selectivity in a spiking network model of layered primary visual cortex. *PLoS Computational Biology*, 15(7):e1007080.

Pena, R. F. O., Lima, V., Shimoura, R., Novato, J. a. P., and Roque, A. C. (2020). Optimal interplay between synaptic strengths and network structure enhances activity fluctuations and information propagation in hierarchical modular networks. *Brain Sciences*, 10(4):228.

Potjans, T. C. and Diesmann, M. (2014a). The cell-type specific cortical microcircuit: relating structure and activity in a full-scale spiking network model. *Cerebral Cortex*, 24(3):785–806.

Romaro, C., Najman, F., Dura-Bernal, S., and Roque, A. C. (2018). Implementation of the potjans-diesmann cortical microcircuit model in netpyne/neuron with rescaling option. *BMC Neuroscience*, (64):sup–2.

Romaro, C., Najman, F. A., Lytton, W. W., Roque, A. C., and Dura-Bernal, S. (2020a). NetPyNE implementation and rescaling of the Potjans-Diesmann cortical microcircuit model. *arXiv:2005.03764*.

Rosenbaum, R. and Doiron, B. (2014). Balanced networks of spiking neurons with spatially dependent recurrent connections. *Physical Review X*, 4(2):021039.

Schmidt, M., Bakker, R., Hilgetag, C. C., Diesmann, M., and van Albada, S. J. (2018). Multi-scale account of the network structure of macaque visual cortex. *Brain Structure and Function*, 223(3):1409–1435.

Senk, J., Hagen, E., van Albada, S. J., and Diesmann, M. (2018a). Reconciliation of weak pairwise spike-train correlations and highly coherent local field potentials across space. *arXiv preprint arXiv:1805.10235*.

Senk, J., Korvasová, K., Schuecker, J., Hagen, E., Tetzlaff, T., Diesmann, M., and Helias, M. (2018b). Conditions for traveling waves in spiking neural networks. *arXiv preprint arXiv:1801.06046*.

Van Albada, S. J., Helias, M., and Diesmann, M. (2015). Scalability of asynchronous networks is limited by one-to-one mapping between effective connectivity and correlations. *PLoS Computational Biology*, 11(9):e1004490.

Veltz, R. and Sejnowski, T. J. (2015). Periodic forcing of inhibition-stabilized networks: nonlinear resonances and phase-amplitude coupling. *Neural Computation*, 27(12):2477–2509.

Vogels, T. P., Rajan, K., and Abbott, L. F. (2005). Neural network dynamics. *Annu. Rev. Neurosci.*, 28:357–76.

Voges, N. and Perrinet, L. (2012). Complex dynamics in recurrent cortical networks based on spatially realistic connectivities. *Frontiers in Computational Neuroscience*, 6:41.

Vreeswijk, C. v. and Sompolinsky, H. (1998). Chaotic balanced state in a model of cortical circuits. *Neural Computation*, 10(6):1321–1371.

Woolsey, C., Marshall, W. H., and Bard, P. (1942). Representation of cutaneous tactile sensibility in the cerebral cortex of the monkey as indicated by evoked potentials. *Johns Hopkins Hospital Bulleti*, 70:399–441.

Yger, P., El Boustani, S., Destexhe, A., and Frégnac, Y. (2011). Topologically invariant macroscopic statistics in balanced networks of conductance-based integrate-and-fire neurons. *Journal of Computational Neuroscience*, 31(2):229—245.

PART III

3 SOMATOTOPIC ORGANIZATION IN THE CELL-TYPE SPECIFIC CORTICAL MICROCIRCUIT MODEL AND INPUT SPATIAL RESOLUTION

Abstract

The Potjans-Diesmann (PD) model reproduces the cortical microcircuit under a 1 mm^2 surface area of early sensory cortex in a 1×1 scale. The model, originally built in the neural simulator NEST, specifies fixed numbers of excitatory and inhibitory neurons per layer across four different cortical layers, connections between these neuronal populations, and external inputs to each cell population. The PD model uses a uniform distribution of probabilities to connect neurons from two populations and, therefore, is not able to represent spatial connectivity. In this work, we re-implement the PD model in the neural simulator Brian and introduce spatial structure to the model, allowing the modeling of topographic mapping between the input (thalamic) space and the cortical layers. We assign spatial positions to neurons across eight different layers and connect them using a distance-dependent Gaussian probability function. Instead of adopting periodic boundary conditions at the layers borders, we use a boundary correction method that maintains neuronal activity within physiological range. The spatial resolution of the topographic map is studied by stimulating different regions of the thalamic space and measuring the cortical response. The spatial pattern of connections allows a rescaling of the modeled cortical area and, consequently, a change in the number of neurons without affecting the network behavior.

3.1 Introduction

The Potjans-Diesmann (PD) model (Potjans and Diesmann, 2014a) is a landmark model that firstly reproduced with statistical fidelity the local connectivity structure of early sensory cortex and the corresponding cell-type specific activity behavior in good agreement with ex-

perimental evidence. It has been broadly used to study the emergence of macroscopic cortical activity patterns, such as layer specific oscillations (Van Albada et al., 2015; Bos et al., 2016) and effects on cortical functionality resulting from inter-layer or inter-columns communication (Cain et al., 2016; Schwalger et al., 2017; Schmidt et al., 2018). Other use examples can be found in (Schuecker et al., 2017; Lee et al., 2017; Beul and Hilgetag, 2015).

The PD model represents the network of neurons under a 1 mm^2 surface area of early sensory cortex in a 1×1 scale. The network consists of around 80,000 leaky integrate-and-fire (LIF) neurons divided in eight cell populations representing excitatory and inhibitory neurons in cortical layers 2/3, 4, 5 and 6. External input is provided by thalamic and cortico-cortical afferents. The model generates spontaneous activity with population-specific average firing rates and both synchrony and irregularity characteristics similar to the ones observed experimentally. It allows the study of the propagation of thalamic inputs from layers 4 and 5 through all layers. The network, originally built in NEST (Gewaltig and Diesmann, 2007), specifies a fixed numbers of excitatory and inhibitory neurons per layer, the number of connections among these neuronal populations and the number of external inputs to each cell population. These numbers are based on experimental data.

Although the PD model is based on data from the primary visual and somatosensory cortices (V1 and S1), it does not have spatial structure, which means it cannot describe topographic patterns of connection or activity. Despite being a good model for the representation of a functional group of neurons with similar characteristic responses, as a rodent cortical barrel (de Kock and Sakmann, 2009) or a group of V1 neurons with the same preferred orientation (Wagatsuma et al., 2013), the PD model is not suited to model properties that depend on spatial extension like cortical topographic maps.

In this work, we implemented a version of the PD model in Brian (Goodman and Brette, 2009) with spatial extension and used a boundary correction method to deal with the problem of activity change at the spatial border of the model (Romaro et al., 2020b) to study the spatial activity resolution. The spatially extended version of the PD model enables the representation and study of phenomena involving spatial patterns of connections.

This paper is organized as follows: In Section 3.2 we present the three steps followed to introduce spatial extension to the PD model: the spatial positioning of neurons in subsection 3.2.1, the conversion of the probability of connection from uniformly distributed to distance-dependently distributed and its calculation in Subsection 3.2.2, and, lastly, the boundary activity correction method in Subsection 3.2.3. In Subsection 3.2.4, the network upscaling (increase in size) method is exposed. In the final subsection (3.2.5) the spatial resolution study method

is described. In Section 3.3 we present the application results corresponding to the methods presented in the above subsections (3.3.1 to 3.3.5). In Section 3.4 we discuss the relevant results and end with the conclusion in Section 3.5.

3.2 Methods

Three steps were used to transform a network without spatial extension to a network with spatial extension. They are:

- **Step 1:** All neurons were distributed on two-dimensional layers and received a position on the x-y coordinate axis.
- **Step 2:** The original pattern of connections was transformed to a distance-dependent pattern of connections.
- **Step 3:** The problem of neural activity change at the spatial border due to the reduction of connections to boundary neurons as a consequence of the introduction of space was solved.

3.2.1 Spatial positioning of neurons

Initially, the PD model (Potjans and Diesmann, 2014a) was re-implemented in Brian as in the original article (Shimoura et al., 2018). Then, for each of the eight cell populations, all of its neurons were homogeneously distributed on a two-dimensional layer of 1 mm^2 and received a position on the x-y coordinate axis. This created a stack of eight cell layers, one for each neuronal population of the PD model. Since the population sizes are different, the densities of neurons in the eight layers are different.

3.2.2 Conversion from uniform distributed to topography

Cortical connectivity maps derived from electrophysiological data show a high density of connections in a small area of radius 100 μm around the somata of neurons (Thomson et al., 2002). On the other hand, anatomical data show a lower density of cortical connections in areas farther away from the somata (Binzegger et al., 2004). This suggests that the connection probability decays monotonically with the distance from the soma for cortical neurons. Thus, a two-dimensional Gaussian probability density function may be a good approximation for this connection probability profile.

We will assume here that the probability of connection from a neuron of one population to a neuron of another population, which may be the same population of the first neuron, is given by a two-dimensional Gaussian probability density function,

$$C(r) = C_0 e^{-\frac{r^2}{\sigma^2}}, \tag{3.2.1}$$

where C_0 is a normalization constant, r is the distance from the presynaptic neuron to the postsynaptic neuron, and σ is the standard deviation.

Symmetrically, each neuron from the postsynaptic set has the same Gaussian distribution of connection probability as the presynaptic set.

Computing the total probability given by the Gaussian distribution under a circle of radius R, an equivalent uniform probability distribution C_m modeling the area A of the circle can be defined:

$$C_m = \frac{1}{A} \int_0^{2\pi} \int_0^R C(r) r dr d\theta. \tag{3.2.2}$$

Substituting (3.2.1) in (3.2.2):

$$C_m = \frac{1}{A} \int_0^{2\pi} \int_0^R C_0 e^{-\frac{r^2}{\sigma^2}} r dr d\theta. \tag{3.2.3}$$

Accordingly, for a given network with uniform probability of connection C_m between pre- and postsynaptic neurons, the transformation to the distance-dependent probability distribution (given by equation (3.2.1)) is given by C_0 as:

$$C_0 = \frac{AC_m}{\int_0^{2\pi} \int_0^R e^{-\frac{r^2}{\sigma^2}} r dr d\theta}. \tag{3.2.4}$$

For the radius R going to infinity ($R \to \infty$), all connections given by the uniform distribution model are spread to the Gaussian distribution model with:

$$C_0 = \frac{AC_m}{2\pi\sigma^2}. \tag{3.2.5}$$

Since this uniform-to-Gaussian transformation allows for the increase and decrease of the area modeled and, hence, the number of neurons, the transformation in (3.2.5) gives a better fit than the one for a finite R given by (3.2.4) to avoid overconnecting the network. However, both transformations work.

The standard deviation σ^2 may depend on the cortex part modeled and contributes to spatial network resolution as will be presented further in this chapter.

The PD model (Potjans and Diesmann, 2014a) represents the connectivity under a surface area of 1 mm^2, therefore $A = 1$ (in units of mm^2), and the spread of biological data, characterized by σ^2, is around 200 to 300 μm (de Kock and Sakmann, 2009; Meyer et al., 2010).

3.2.3 Boundary Activity Correction

Once the network received a topographic pattern of connection, the neurons on the boundary network will present hight or low activity due to lack of inhibitory or excitatory connections respectively. There are few approaches to face this issue, such as the torus solution, compensating for the lack of connections, and the disregard of boundary neurons in the network. Each approach has pros and cons. The torus solution, for example, has an easier implementation but introduces undesired oscillations. Disregarding boundary neurons increases the computational power required once a lot of computational resources will be allocated for neurons which will be disregarded. Compensating the lack of connections is not an easy task once it may change the neuron activity and the local first or second order (Van Albada et al., 2015) statistics of network. To solve this issue we implemented the boundary correction method presented in (Romaro et al., 2020b). This method recoups the first and second-order statistics.

The method basically numerically estimates the normalized density function of connection in the first step, then weights each neuron connection based on it, and finally balances the threshold to grant the neuron/layer activity.

The algorithm of boundary correction method consists in:

- **Step 1:** Calculating the scale factor for any neuron o in network based in the normalized connection density;
- **Step 2:** Increasing the synaptic weights by dividing them by the square root of the scale factor;
- **Step 3:** Providing each cell with a DC input current with a value corresponding to the total input lost due to network edge (boundary cut).

3.2.4 Network upscaling

Once the Gaussian probability rules the connection distribution of the model, it is possible to increase the number of neurons in the network, and, consequently, the represented cortex area, without increasing significantly the number of connections per neuron once the original network size is already larger than double of the Gaussian standard deviation, σ. In other words, it is expected that the number of connections increase lineally with the number of neurons and that the network behavior remain unchanged regardless of upscaling adjustments.

3.2.5 The spatial resolution study

To find the minimal stimulation area that leads to an increase in network activity through layers we started by calculating the minimum input (number of synapses received per time) to activate a neuron in the network. The neuron in the network is governed by the following differential equations:

$$\frac{\delta V}{\delta t} = \frac{-V + V_{reset}}{\tau_m} + \frac{I}{C_m},\ (if\ V < V_{th}) \tag{3.2.6}$$

$$\frac{\delta I}{\delta t} = \frac{-I}{\tau_{syn}}, \tag{3.2.7}$$

Hence, for the network parameters ($I = 87.8pA$; $V_{reset} = -65mV$; $V_{th} = -50mV$; $\tau_m = 100ms$; $C_m = 250pF$; $\tau_{syn} = 0.5ms$), one synapse input results in a rise of approximately $15mV$ in membrane potential, and around 100 inputs are necessary to rise the potential from V_{reset} to V_{th} leading to a spike.

The number of connections between pre-synaptic population i and post-synaptic population j can be calculated as:

$$C_{i,j} = 1 - (\frac{1}{N_{pre}.N_{post}})^k, \tag{3.2.8}$$

and than, the number of connections that a neuron in j receives from population i is:

$$k = \frac{ln(1 - C_{i,j})}{ln(1 - 1/N_{pre})}, \tag{3.2.9}$$

Each neuron from layers L4e, L4i, L5e, L5i receives approximately 93, 58, 47, 18 connec-

tions respectively from thalamus (composed by 902 neurons). Therefore, one spike from each thalamus neuron should be enough to lead to an extra spike in layer L4e, but not necessarily an extra spike in other layers. For the others layers a temporal sum with connections from those layers is necessary to result in an increase on spike activity.

One spike from each Thalamus neuron can be obtained from stimulus like one second with 1 Hz frequency or 100ms with 10Hz frequency or 10ms input with 100Hz. The third option will be utilized in this paper.

Once the somatotopic organization changes the probability function of connection, besides the relation of time and frequency of stimulation, a relation between the Gaussian standard deviation σ of the probability function and stimulus area is expected as presented in Table 14 and from the equation:

$$C_m = C_o.2\pi.\sigma^2.(1 - e^{-R^2/2\sigma^2}), \tag{3.2.10}$$

the frequency of thalamus stimulation input with radius R related with the full area stimulation frequency f_o is:

$$f_R = f_o/(1 - e^{-0.5(R/\sigma)^2}), \tag{3.2.11}$$

R/σ	Activate area	Compensated Frequency
$2:1$	0.87	116 Hz
$1:1$	0.40	254 Hz
$1:2$	0.12	851 Hz
$1:3$	0.06	1.67 kHz
$1:4$	0.03	3.20 kHz

Table 14: Thalamus active area and frequency input to lead to an extra spike in layer L4e for different ratios of stimulus areas with radius R and standard deviation σ of Gaussian probability function of connections.

Once previous publications showed a maximum fire rate for a single neuron, we ran for each ratio R/σ the correspondent thalamus frequency for lower frequencies compatible with biological results. Therefore, the somatotopic network was stimulated with 48 sets of 10ms of thalamus inputs, following frequency data in Table 15, with intervals of 10ms of thalamus silence. The firing rate was calculated for each layer separately and the result are presented in section 3.3.

R/σ	Activate area	Frequencies
1 : 1	0.40	254 Hz
1 : 2	0.12	851 Hz - 425 Hz - 212 Hz
1 : 3	0.06	1667 Hz - 851 Hz - 300 Hz

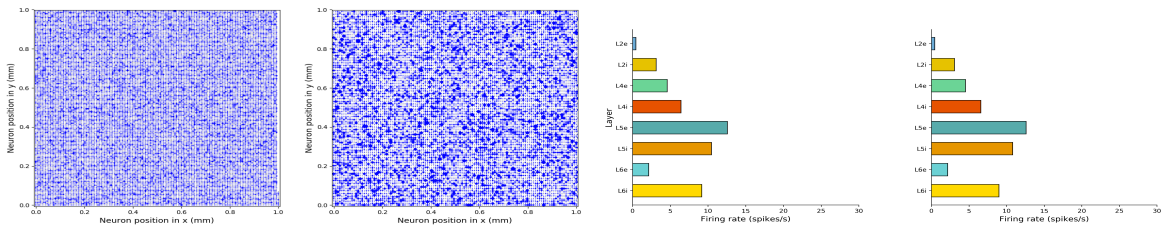
Table 15: Thalamus active area and frequency applied input and its ratios of stimulus areas radius R and standard deviation σ of Gaussian probability function of connections.

3.3 Results

3.3.1 Geographic position

All neurons from the full version of the model presented on (Potjans and Diesmann, 2014a) were homogeneously distributed on $1mm^2$ keeping the homogeneous probability density function of connections. Note that, even if each neuron has a geographic position, the pattern of connection disregards that, resulting in a lack of both a topographic structural organisation and topographic activity organisation.

Figures 22-A and B present the average firing-rate per neuron. Each dot represents the position of the neuron and the size of the dot is proportional to the average firing rate of that neuron. Figures 22 - A and B correspond to excitatory layer L4 and L6 without boundary correction. Figures 22-C and D present the core (around 50 % of all neurons) and the boundary (the complementary 50 % of all neurons) layers average firing rate respectively



(A) (B) (C) (D)

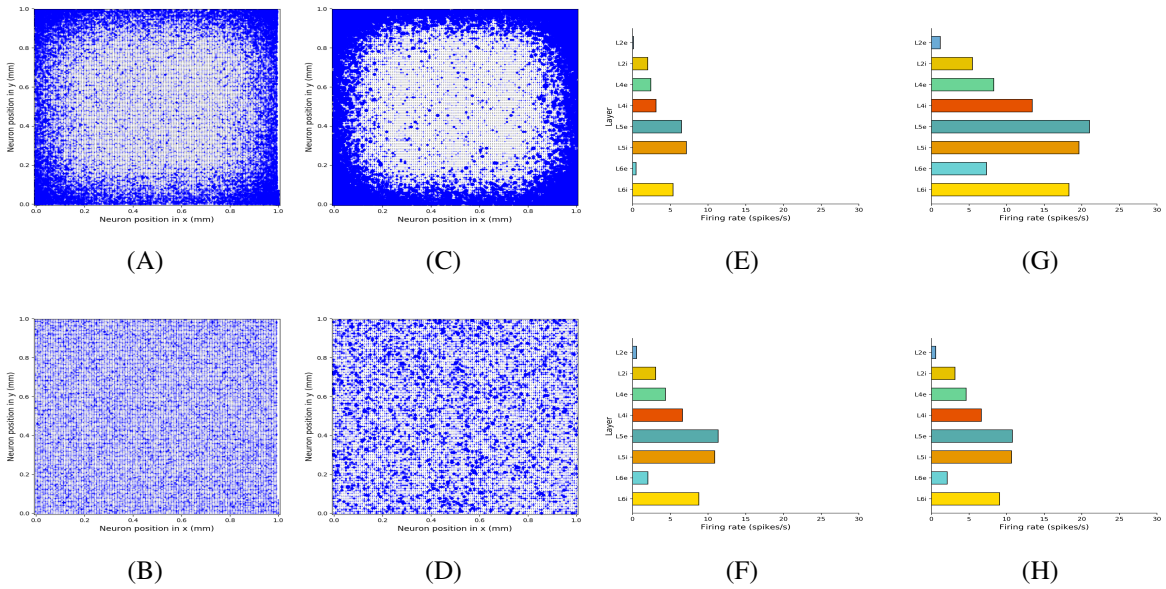
Figure 22: Reproduction of the full scale PD average firing-rate per neuron and per layer. Neurons from (A) L4 and (B) L6 excitatory. Each dot represents the position of a neuron and the size of the dot is proportional to the average firing rate of that neuron. (C) Core (around 50 % of all neurons) layers average firing rate. (D) Boundary (complementary 50 % of all neurons) layers average firing rate.

3.3.2 Conversion from uniform distributed to topography

Figure 23 presents the same of the Figure 22 for the homogeneous probability function of connections replaced by a Gaussian with standard deviation $\sigma_g = 0.275mm$. In this case, boundary neurons spike visually more than the core ones due to the leak of inhibitory connections.

3.3.3 Boundary Activity Correction

Figure 23-B presents the same of the Figure 23-A, for the homogeneous probability function of connections replaced by a Gaussian with standard deviation $\sigma_g = 0.275mm$, but with boundary correction. In this case, boundary neurons turn over to spike as much as core neurons and as the previous model in order to compensate the leak of inhibitory connections.



(A) (C) (E) (G)

(B) (D) (F) (H)

Figure 23: Full scale PD average firing-rate per neuron and per layer. Neurons from L4 excitatory (A)without and (B) with boundary correction. Neurons in L6 excitatory (C) without and (D) with boundary correction. Each dot represents the position of a neuron and the size of the dot is proportional to the average firing rate of that neuron. Core (around 50 % of all neurons) layers average firing rate (E) without boundary correction and (F) with boundary correction. Boundary (complementary 50 % of all neurons) layers average firing rate (G) without boundary correction and (H) with boundary correction.

3.3.4 Network upscaling

Figure 24 presents the same of the Figure 23, a Gaussian distribution of connection probability with standard deviation $\sigma_g = 0.275mm$ and with the boundary correction but the number of neurons on network was up scaled to 4 times the number of neurons in original network. Neurons behavior is similar to one the previous size model (Figure 23).

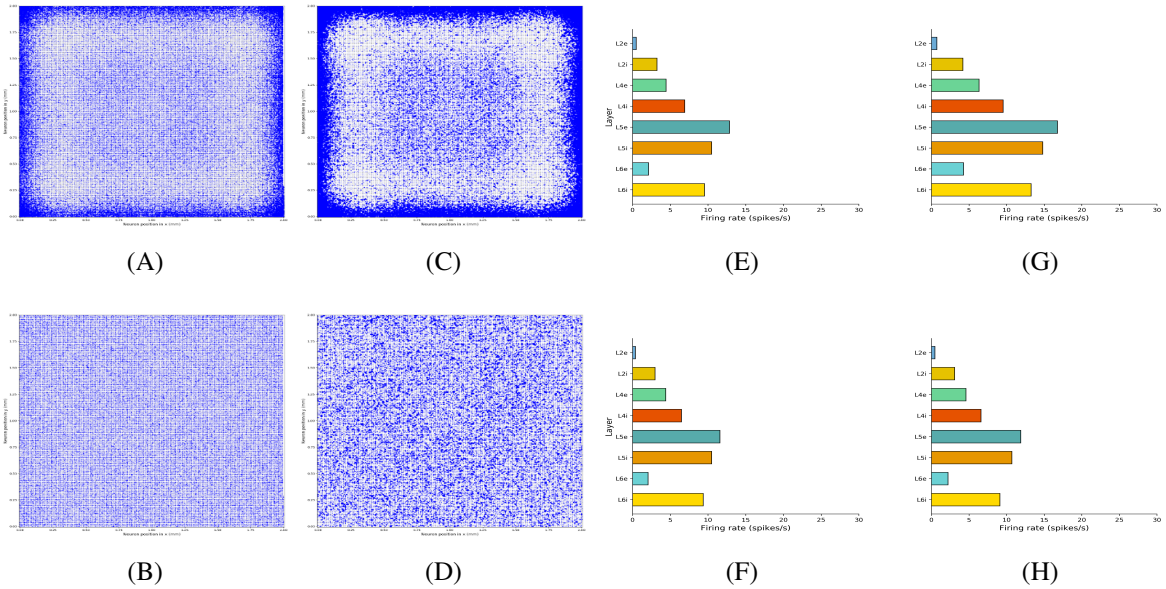
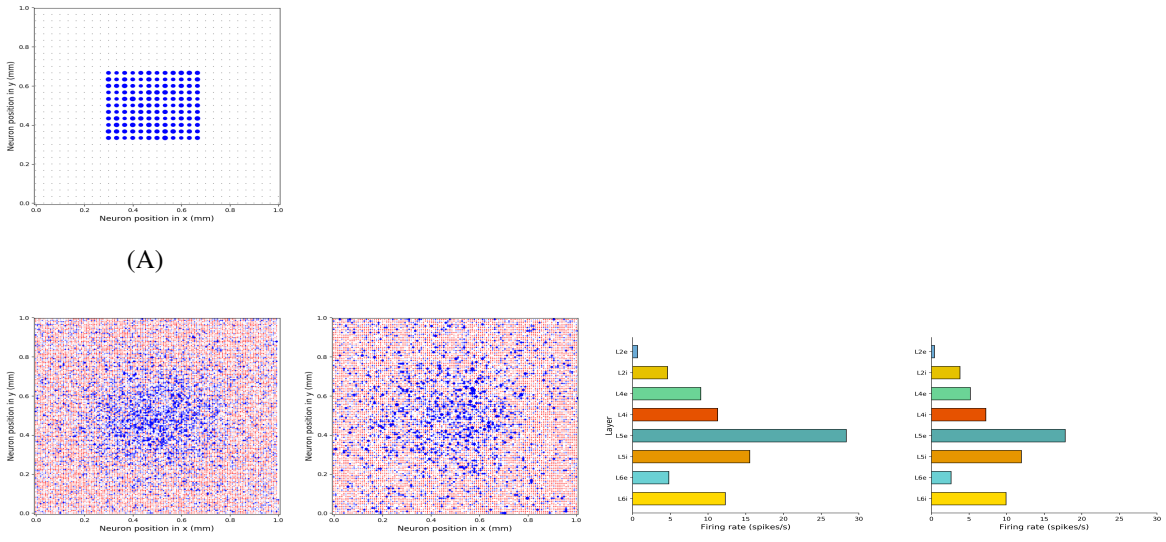


Figure 24: Average firing-rate per neuron and per layer of PD up-size model to $4mm^2$. Neurons from L4 excitatory (A)without and (B) with boundary correction. Neurons in L5 excitatory (C) without and (D) with boundary correction. Each dot represents the position of a neuron and the size of the dot is proportional to the average firing rate of that neuron. Core (around 50 % of all neurons) layers average firing rate (E) without boundary correction and (F) with boundary correction. Boundary (complementary 50 % of all neurons) layers average firing rate (G) without boundary correction and (H) with boundary correction.

3.3.5 The spatial resolution study

Figure 25 presents results for ratio $R/\sigma = 1$, i.e., a stimulus with square side of 2x the Gaussian standard deviation (Figure 25 - A). Figures 25 - B and C present the neuron location (position) and the neuron average firing (size of the dot) from layers L4 and L6 excitatory. Neurons with fire rate above the layer fire rate are represented in blue dots. Neurons with fire rate below the layer fire rate are represented in red dots. It is possible to see that the neurons located in the middle of the represented area have a higher average firing rate than the boundary ones for both, layer L4, directly connected with thalamus, and layer L6, the process layer not directly connected with thalamus. Comparing Figure 25 - D and E, it is possible to see that this

behavior is present in all neuron populations.



(A) (B) (C) (D) (E)

Figure 25: Thalamus input influence. (A) Activated thalamus neurons. Neurons (B) from L4 excitatory and (C) from L6 excitatory. Each dot represents the position of a neuron and the size of the dot is proportional to the difference between average firing rate of that neuron and that of the layer (all neurons in that layer). Neurons with fire rate above the layer fire rate are represented in blue dots. Neurons with fire rate below the layer fire rate are represented in red dots. (D) Core (same area of activated thalamus) layers average firing rate. (E) Boundary (complementary area of activated thalamus) layers average firing rate.

Figure 26 presents results for ratio $R/\sigma = 0.5$, i.e., a stimulus with square side of the Gaussian standard deviation (Figure 26 - A). Figures 26 - B, E, and H present the neuron location (position) and the neuron average firing (size of the dot) from layer L4 excitatory and thalamus input of 851, 425 and 212 Hz, respectively. Once more it is possible to see that the neurons located in the middle of the represented area have a higher average firing rate than the boundary ones. However this effect tends to become less evident with the decrease in thalamus input. The core/boundary firing-rate difference in layer L4 (Figure 26 - H) is almost indistinguishable, but can be confirmed by Figures 26 - I and J. It is possible to see that the cortex area which had increased in firing rate (blue dots in Figure 26 - B) is higher than the thalamus activated area (blue dots in Figure Figure 26 - A). Comparing Figures 26 - C to D, F to G and I to J, it is possible to see that this behavior is present in all neuron populations, although it tends to vanish in lower thalamus frequencies.

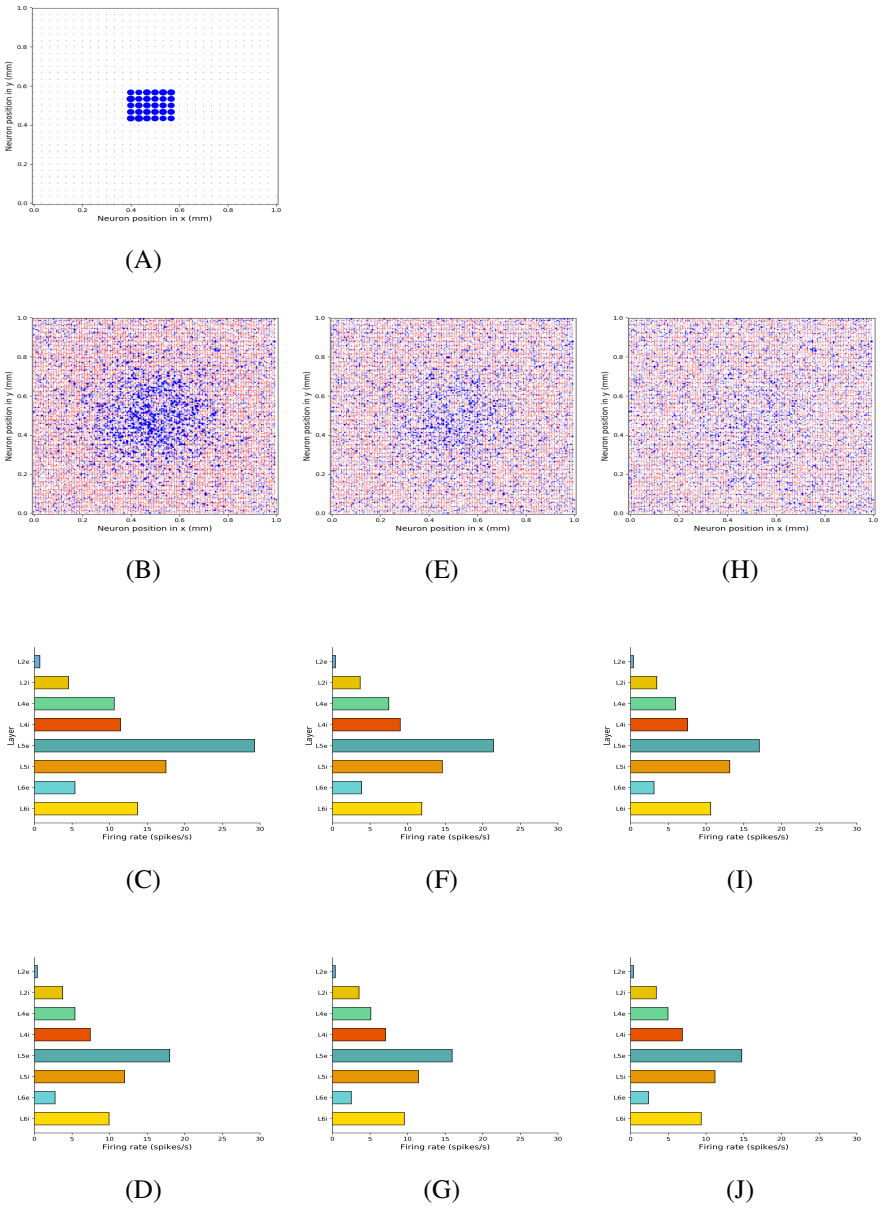


Figure 26: Thalamus input influence for stimulus with ratio $R/\sigma = 0.5$ and deferments frequencies. (A) Activated thalamus neurons. Neurons (B,E,H) from L4 excitatory receiving thalamus input of 851, 425 and 212 Hz respectively. Each dot represents the position of a neuron and the size of the dot is proportional to the difference between average firing rate of that neuron and that of the layer (all neurons in that layer). Neurons with fire rate above the layer fire rate are represented in blue dots. Neurons with fire rate below the layer fire rate are represented in red dots. (C, F, I) Core (same area of activate thalamus (mesma area do talamo ativado)) layers average firing rate. (D, G, J) Boundary (complementary area of active thalamus) layers average firing rate.

Figure 27 presents results for ratio $R/\sigma = 0.33$. Figures 27 - B, E, and H present the neuron location (position) and the neuron average firing rate (size of the dot) from layer L4 excitatory and thalamus input of 1667, 851, and 300 Hz, respectively. The neurons located in the middle of the represented area have a higher average firing rate than the boundary ones just for 1667 and 851 Hz thalamus frequencies. The core/boundary firing-rate difference in layer L4 (Figure 27 - H) is indistinguishable and cannot be confirmed by Figures 27 - I and J. Comparing Figures 27 - C to D and F to G, it is possible to see that, differently from (opposite to) previous results and all other layers, the thalamus input leads to a decrease in layer L5 inhibitory average firing rate in the corresponding activated thalamus area. The firing rate of all other layers tend to increase in the corresponding activated thalamus area. Comparing the Figures 27 - I to J, average firing rate in the corresponding activated thalamus area increases in some layers but decreases in others.

Figure 28 presents the results of a thalamus input influence for stimulus with ratio $R/\sigma = 2$ ($\sigma = 0.2$) on the left and $R/\sigma = 0.5$ ($\sigma = 0.275$) on the right. Comparing Figure 28-B with all other B Figures (25, 26, 27), it is possible to see that the activated area is higher than all others, and its diameter is twice as large as the standard deviation σ.

Comparing activated area (in blue dots) in Figures 25, 26, 27 Bs it is possible to see that independently of the thalamus activated area size (Figures 25, 26, 27 As), the activated area in layer L4e is a circle with diameter twice as large as the Gaussian standard deviation σ. The activated area can be even large if the thalamus activated area is bigger than that (Figure 28-B), but cannot be smaller than that (Figures 25, 26, 27 Bs).

A decrease in thalamus firing rate leads to lack of response in cortex layers (Figure 28-H), but without (or non-perceivable) influence on cortex activated area.

Additionally, the Figure 28-F has a thalamus input of $R/\sigma = 0.5$ as does the Figure 26-B, but has an larger activated diameter than the one one the Figure 26-B.

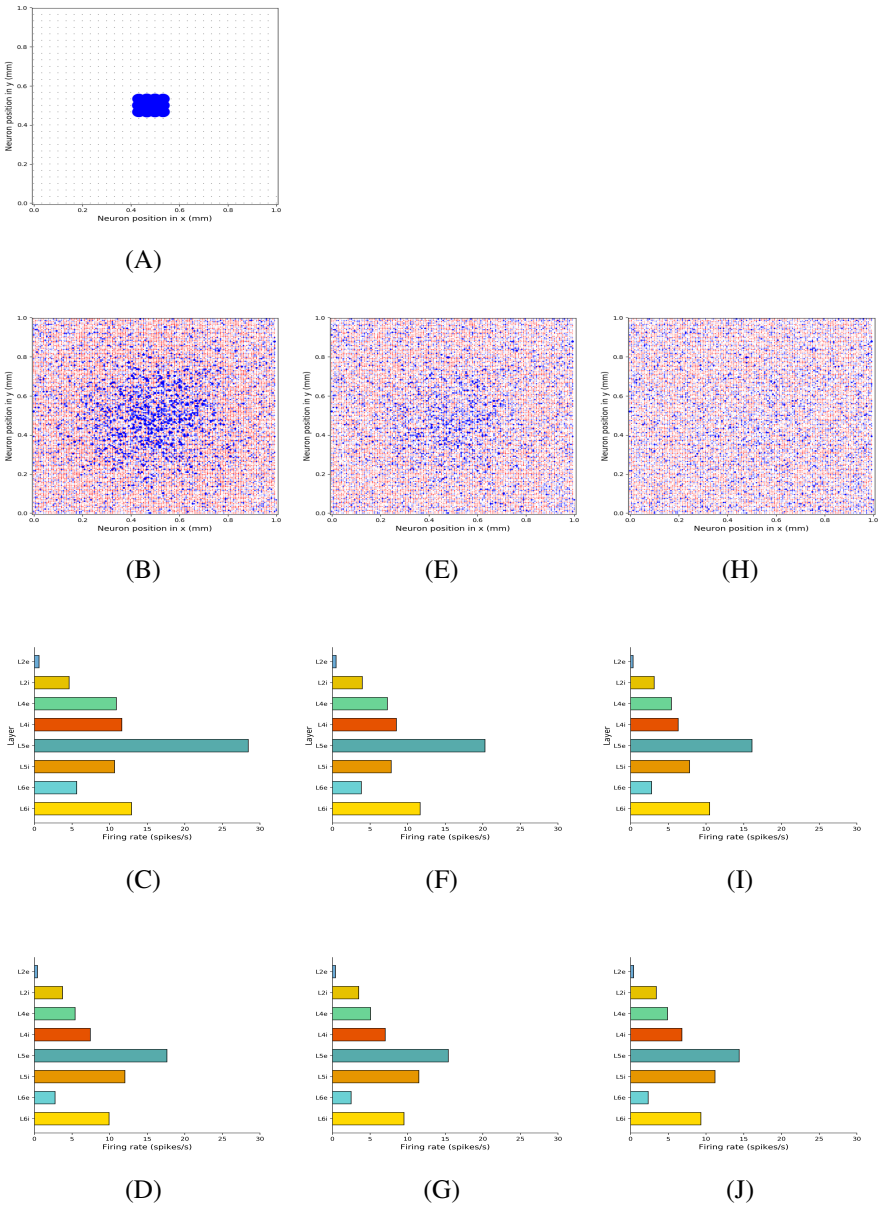


Figure 27: Thalamus input influence for stimulus with ratio $R/\sigma = 0.33$ and deferments frequencies. (A) Activated thalamus neurons. Neurons (B,E,H) from L4 excitatory receiving thalamus input of 1667, 833, and 300 Hz respectively. Each dot represents the position of a neuron and the size of the dot is proportional to the difference between average firing rate of that neuron and that of the layer (all neurons in that layer). Neurons with fire rate above the layer fire rate are represented in blue dots. Neurons with fire rate below the layer fire rate are represented in red dots. (C, F, I) Core (same area of activated thalamus (mesma area do talamo ativado)) layers average firing rate. (D, G, J) Boundary (complementary area of activated thalamus) layers average firing rate.

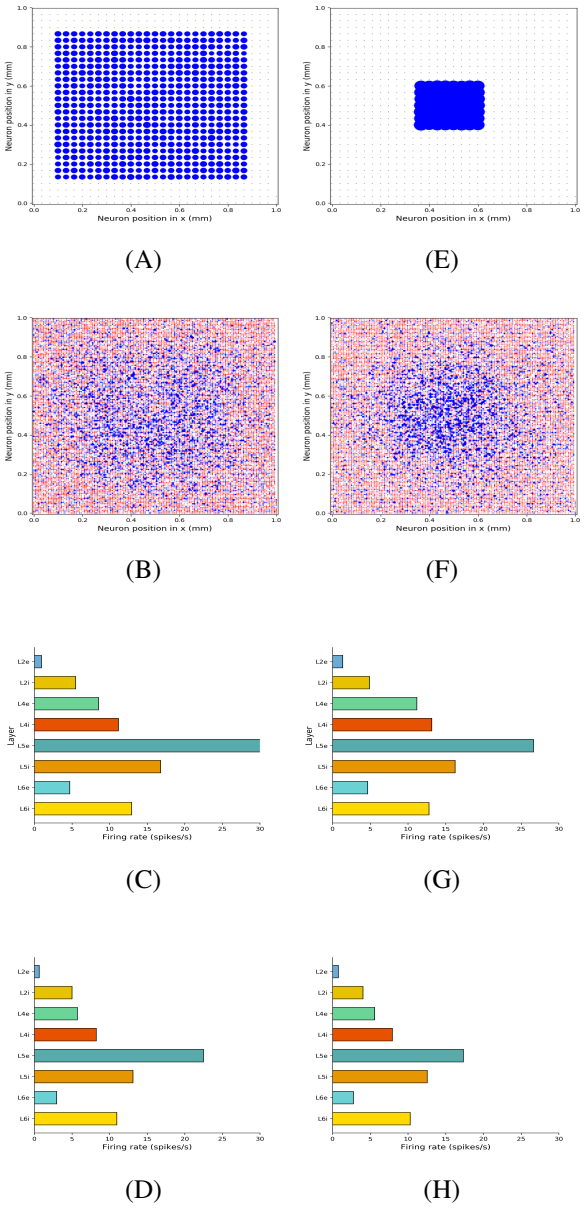


Figure 28: Thalamus input influence for stimulus with ratio $R/\sigma = 2$ ($\sigma = 0.2$) on the left and $R/\sigma = 0.5$ ($\sigma = 0.275$) on the right. (A, E) Activated thalamus neurons. (B) Neurons from L4 excitatory receiving thalamus input of 200 and 851 Hz respectively. Each dot represents the position of a neuron and the size of the dot is proportional to the difference between average firing rate of that neuron and that of the layer (all neurons in that layer). Neurons with fire rate above the layer fire rate are represented in blue dots. Neurons with fire rate below the layer fire rate are represented in red dots. (C) Core (same area of activated thalamus) layers average firing rate. (D) Boundary (complementary area of activated thalamus) layers average firing rate.

3.4 Discussion

The original network was not able to present topographic organization neither in structure nor in activity once the connection probability was a homogeneous function. The first step was to give a position in space for each neuron, which was not enough to generate a spatial organization, although it was a required step. The statistics of this model are like the original ones once introducing a position in space does not change the network pattern of connection. Therefore, this is in a sophisticated reproduction of the model.

The second step was to transform the homogeneous connection probability function into a neuron-distance-dependent connection probability function. The distribution used could be a Gamma distribution, Poisson, Beta or a linear combination of any of them, or any other function distance-dependent. We chose the Gaussian distribution because it presents a good approximation for the biological pattern of connection (Potjans and Diesmann, 2014a). The structural topographic organization was done using only this function, however it is not granted that this is enough for a topographic organization in activity. Actually, this change leads to a side effect in activity: the neurons in the edge change their activity due to a loss of connections, which can lead to an increase or decrease in activity, depending on the loss of excitatory or inhibitory connections. The Figure 23-A presents this noticeable difference in effect.

Solving for this change in edge activity is not trivial. In literature, alternatives such as the Torus solution (with the likely introduction of oscillation as one artefact) or neglecting neurons on the edge (increasing the computational resources required) are adopted. We utilized a method to equalized the boundary-neurons activity (Romaro et al., 2020b) and the results are notorious in Figure 23-B.

Figure 24 presented the results of the network resized to 4 times the original one and maintaining the statistics. This is a not trivial effect, networks with non-distance-dependent and even some kinds of distance-dependent functions do not always allow for a resize in such as easy way: recouping all parameters and activities just increasing the number of neurons, and consequently, the represented area.

This model is an alternative to studying the cortex area with topographic representation or to when the barre structure is not well defined, and thus, the connection between layers is not a good fit.

Next step is to calculate the minimum activated thalamus area to lead to a notorious increase in the firing rate of the somatotopic cortex correspondent area. A compromise between the thalamus area and the standard deviation of the Gaussian distribution of connection probability

was presented.

Independently of the thalamus activated area size, the activated area in layer L4e is a circle with radius equal to the Gaussian standard deviation σ. The activated area can present a radius larger than the Gaussian standard deviation, if the thalamus activated area is larger than that (Figure 28 - left), but cannot be smaller (Figures 25, 26, 27 Bs and Figure 28-F).

The decrease in thalamus firing rate leads to a lack of response in cortex layers (Figure 27-H), but without (or non-perceivable) influence on cortex activated area, what lead us to conclude that the standard deviation σ of patter of connection is indeed the resolution of these network.

This computational experiment raised a biological question: is the representation area correlated with the connection-space-ratio concentration? It is known that the ratio between some body parts' size and their representation areas in the somatosensory cortex are higher than others (Ramachandran, 1998; Merzenich et al., 1984; Bear et al., 2020), and hence, the sensitivity of some areas is higher as their ratios get smaller. It is expected that the standard deviation of distance-dependent distribution of connection is related with the body size species and their somatotopic brain area. However, are these relations governed by one law throughout all species? And does all somatosensory cortex area inside one brain have distance-dependent distribution of connection with the same average and standard deviation?

It is assumed that the pyramidal cells play a determinant role in defining the diameter of functional columns in brain and specially in the somatosensory cortex. The horizontal spread of axonal and dendritic arbours of those cells is around $150\mu m$ to $250\mu m$, which comprehends around 80 % of all connections onto neocortical neurons (Wang et al., 2004). Once this pattern of connections is continuous through the somatocortex area, the lateral inhibition extended beyond the local recurrent axonal network is the principal supposition to hold the activity constricted onto an area. This consists in a different standard deviation connection function for inhibitory neurons high enough to the signal not spread. Furthermore, this model uses the same standard deviation for the Gamma connection function both for inhibitory and excitatory neurons, and even then, the activity is held in a concerned area. Thus, it seems that the fact that inhibitory connections are distinct of excitatory ones can be a contribution factor to this behavior, but not a necessary one. We conjecture that the interlayer connection structure and the higher inhibitory synapse strength higher than excitatory strength are the two main factors avoiding a spread in activity.

Other instructing result is the network ability to restaining previous balanced activity even with the introduction of an extra external input. A new thalamus input is perceived and processed on network by a increase in flow activity through the layers. But if the thalamic input

is hold on for a while, the network reestablish the new balance despite the input. This seems as a somatossensory adaptation like how humans wear clothes but do not constantly feel their touch. We conjecture that the adaptation may be not only on skin sensors but in cortex inter layer connections too.

A further step is to introduce a plasticity and train the network to even larger σ, so the network is able to distinguish the representation of different areas, such as, different fingers in one hand. In other words, a study of correlation between the plasticity parameter and σ would intermediate the study of the cross-columnar inhibitiory and excitatory changes to delimit boundaries representation on homunculus (Ramachandran, 1998; Merzenich et al., 1984; Bear et al., 2020). The answer to the question previously raised may lay here.

3.5 Conclusion

The PD model with topographic organization maintains the features of the original model, such as firing rate and signal propagation through layers at the same time that it introduces a spatial organization in structure and activity.

This model accepts scaling up while maintaining network features. Therefore it is an alternative to studies of cortex regions where the pattern of connection is topographic and not barrel (de Kock and Sakmann, 2009; Wagatsuma et al., 2013), as in that, the side barrel connections to represent large areas is not a good model.

A correlation between spatial resolution in activity and the standard deviation of the Gaussian distribution was utilized to rise the structural organization is presented.

Acknowledgments

This work was produced as part of the activities of FAPESP Research, Disseminations and Innovation Center for Neuromathematics (Grant 2013/07699-0, S. Paulo Research Foundation). The author is the recipient of PhD scholarships from the Brazilian Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

References

Bear, M., Connors, B., and Paradiso, M. A. (2020). *Neuroscience: Exploring the brain*. Jones & Bartlett Learning, LLC.

Beul, S. F. and Hilgetag, C. C. (2015). Towards a "canonical" agranular cortical microcircuit. *Frontiers in Neuroanatomy*, 8:165.

Binzegger, T., Douglas, R. J., and Martin, K. A. (2004). A quantitative map of the circuit of cat primary visual cortex. *Journal of Neuroscience*, 24(39):8441–8453.

Bos, H., Diesmann, M., and Helias, M. (2016). Identifying anatomical origins of coexisting oscillations in the cortical microcircuit. *PLoS Computational Biology*, 12(10):e1005132.

Cain, N., Iyer, R., Koch, C., and Mihalas, S. (2016). The computational properties of a simplified cortical column model. *PLoS Computational Biology*, 12(9):e1005045.

de Kock, C. P. and Sakmann, B. (2009). Spiking in primary somatosensory cortex during natural whisking in awake head-restrained rats is cell-type specific. *Proceedings of the National Academy of Sciences (USA)*, 106(38):16446–16450.

Gewaltig, M.-O. and Diesmann, M. (2007). NEST (neural simulation tool). *Scholarpedia*, 2(4):1430.

Goodman, D. F. and Brette, R. (2009). The brian simulator. *Frontiers in Neuroscience*, 3:26.

Lee, J. H., Koch, C., and Mihalas, S. (2017). A computational analysis of the function of three inhibitory cell types in contextual visual processing. *Frontiers in Computational Neuroscience*, 11:28.

Merzenich, M. M., Nelson, R. J., Stryker, M. P., Cynader, M. S., Schoppmann, A., and Zook, J. M. (1984). Somatosensory cortical map changes following digit amputation in adult monkeys. *Journal of Comparative Neurology*, 224(4):591–605.

Meyer, H. S., Wimmer, V. C., Hemberger, M., Bruno, R. M., de Kock, C. P., Frick, A., Sakmann, B., and Helmstaedter, M. (2010). Cell type–specific thalamic innervation in a column of rat vibrissal cortex. *Cerebral Cortex*, 20(10):2287–2303.

Potjans, T. C. and Diesmann, M. (2014a). The cell-type specific cortical microcircuit: relating structure and activity in a full-scale spiking network model. *Cerebral Cortex*, 24(3):785–806.

Ramachandran, V. S. (1998). Consciousness and body image: lessons from phantom limbs, capgras syndrome and pain asymbolia. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 353(1377):1851–1859.

Romaro, C., Roque, A. C., and Piqueira, J. R. C. (2020b). Boundary solution based on rescaling method: recoup the first and second-order statistics of neuron network dynamics. *arXiv preprint arXiv:2002.02381*.

Schmidt, M., Bakker, R., Hilgetag, C. C., Diesmann, M., and van Albada, S. J. (2018). Multi-scale account of the network structure of macaque visual cortex. *Brain Structure and Function*, 223(3):1409–1435.

Schuecker, J., Schmidt, M., van Albada, S. J., Diesmann, M., and Helias, M. (2017). Fundamental activity constraints lead to specific interpretations of the connectome. *PLoS Computational Biology*, 13(2):e1005179.

Schwalger, T., Deger, M., and Gerstner, W. (2017). Towards a theory of cortical columns: From spiking neurons to interacting neural populations of finite size. *PLoS Computational Biology*, 13(4):e1005507.

Shimoura, R., Kamiji, N., Pena, R., Lima Cordeiro, V., Celis, C., Romaro, C., e Roque, A. C. (2018). [re] the cell-type specific cortical microcircuit: relating structure and activity in a full-scale spiking network model. *ReScience*, 4.

Thomson, A. M., West, D. C., Wang, Y., and Bannister, A. P. (2002). Synaptic connections and small circuits involving excitatory and inhibitory neurons in layers 2–5 of adult rat and cat neocortex: triple intracellular recordings and biocytin labelling in vitro. *Cerebral Cortex*, 12(9):936–953.

Van Albada, S. J., Helias, M., and Diesmann, M. (2015). Scalability of asynchronous networks is limited by one-to-one mapping between effective connectivity and correlations. *PLoS Computational Biology*, 11(9):e1004490.

Wagatsuma, N., Potjans, T. C., Diesmann, M., Sakai, K., and Fukai, T. (2013). Spatial and feature-based attention in a layered cortical microcircuit model. *PloS One*, 8(12):e80788.

Wang, Y., Toledo-Rodriguez, M., Gupta, A., Wu, C., Silberberg, G., Luo, J., e Markram, H. (2004). Anatomical, physiological and molecular properties of martinotti cells in the somatosensory cortex of the juvenile rat. *The Journal of Physiology*, 561(1):65–90.

PART IV

4 A NUMERICAL STUDY OF THE TIME OF EXTINCTION IN A CLASS OF SYSTEMS OF SPIKING NEURONS

Abstract

In this paper we present a numerical study of a mathematical model of spiking neurons introduced by Ferrari et al. (2018). In this model we have a countable number of neurons linked together in a network, each of them having a membrane potential taking value in the integers, and each of them spiking over time at a rate which depends on the membrane potential through some rate function ϕ. Beside being affected by a spike each neuron can also be affected by leaking. At each of these leak times, which occurs for a given neuron at a fixed rate γ, the membrane potential of the neuron concerned is spontaneously reset to 0. A wide variety of versions of this model can be considered by choosing different graph structures for the network and different activation functions. It was rigorously shown that when the graph structure of the network is the one-dimensional lattice with a hard threshold for the activation function, this model presents a phase transition with respect to γ, and that it also presents a metastable behavior. By the latter we mean that in the sub-critical regime the re-normalized time of extinction converges to an exponential random variable of mean 1. It has also been proven that in the super-critical regime the renormalized time of extinction converges in probability to 1. Here, we investigate numerically a richer class of graph structures and activation functions. Namely we investigate the case of the two dimensional and the three dimensional lattices, as well as the case of a linear function and a sigmoid function for the activation function. We present numerical evidence that the result of metastability in the sub-critical regime holds for these graphs and activation functions as well as the convergence in probability to 1 in the super-critical regime.

Keywords: Interacting point processes, Stochastic model of spiking neurons, Metastability, Numerical simulation

4.1 Introduction

This article presents and discusses a simulation study of a system of interacting point processes modeling a biologically inspired neural network; focusing on the time of extinction of a sustained activity exhibited by the network in the absence of external input. The use of point processes, like Hawkes processes (Hawkes, 1971; Chevallier et al., 2015; Chornoboy et al., 1988; Hansen et al., 2015) or Wold processes (Kass and Ventura, 2001), as models for biological neural networks has by now a pretty long history. The model considered here is closer to the one introduced by Brillinger (1988), since the process mimicking the neuron membrane potential is reset to a fixed value after each spike emitted by the neuron; it is informally speaking a stochastic version of the integrate and fire model. While network dynamics has been the subject of many studies both from a point process (*i.e.* stochastic) and from a deterministic (Brunel, 2000; Vreeswijk and Sompolinsky, 1998; Millman et al., 2010) perspective, the concern for a *rigorous* characterization of the endogenous behavior (that is, in the absence of input) of the first kind of networks is, to the best of our knowledge, recent (Duarte and Ost, 2016; Galves et al., 2019; Ferrari et al., 2018; André, 2019c,a). These recent studies used Galves-Löcherbach-like models (Galves and Löcherbach, 2013).

The exact model we consider here is as in Ferrari et al. (2018), which consists in an infinite number of neurons, each of them being affected by two different point processes representing the spiking times and the leaking times, the latter being an homogeneous Poisson process. This choice of a non-continuous potential loss is a clearly peculiar feature of our model and is introduced for technical convenience. It contrasts with the continuous leakage of the integrate and fire model (in the deterministic family) and of the original Galves-Löcherbach one (in the stochastic family).

Within the framework of our model, a series of rigorous results were established and are reviewed in Section 4.3 of this paper. Considering a one dimensional lattice together with a hard threshold activation function – the neuron's spiking rate switches from zero to one when the membrane potential becomes positive – (Ferrari et al., 2018) showed that the network exhibits a phase transition with respect to the leakage rate (Theorem 4.3.1 in Section 4.3); namely, there is a phase in which every neuron stops spiking in finite time with probability one, while in the other, each neuron continues spiking infinitely many times with positive probability. Considering a *finite* version of the same model, André (2019c) showed that, in the continuously spiking phase of Ferrari et al. (2018) – the sub-critical phase, – the re-normalized time of extinction is asymptotically (with respect to the number of neurons) exponentially distributed (Theorem 4.3.2 in Section 4.3), a property that we refer to as *metastability*. Informally, the network will

present an unpredictably long period of pseudo-stationary activity before being kicked out to the only equilibrium (where all neurons are quiescent) by a sudden fluctuation. André (2019a) also showed that in the other phase, the re-normalized time of extinction goes to one asymptotically (see Theorem 4.3.3 in Section 4.3). Informally, the network activity will exhibit a sharp membrane potential loss leading to a comparatively fast and predictable extinction time.

The notion of metastability has acquired more and more attention during the last years in the field of neuroscience – see for example (Deco et al., 2017) and (Roberts et al., 2019)–, especially to explain the brain ability to switch periodically between different functional networks. The most common approach considers large scale models to keep track of brainwaves. The approach we adopt here is to the best of our knowledge the first one – along with (André, 2019c) – to give evidence of metastability in a theoretical model operating at the neuron level. It is inspired by statistical physics and takes its roots in (Cassandro et al., 1984), where a formal framework was developed for metastability in interacting particles systems, and where some models such as the Curie-Weiss mean field dynamics and the Harris contact process were proven to be metastable in the sense of this framework. The key point in this framework is to show that under suitable conditions (which here means in the sub-critical phase) the time of extinction of the system is asymptotically memory-less, or, in other words, exponentially distributed (Theorem 4.3.2). The result proven in André (2019c), as well as the results proven in André (2019a) and Ferrari et al. (2018), are nonetheless dependent on the technical assumptions that the structure of the network is a one-dimensional lattice, and that the activation function is a hard threshold, which is convenient mathematically but unrealistic biologically. Here, we develop a numerical study aimed to show that the results related to the time of extinction hold in lattices of higher dimensions $\mathbb{Z}^d$ ($d = 2$ and $d = 3$) and for two other commonly used activation functions.

The choice of such structure for the network was motivated by the fact that lattices of various dimensions have been previously used in the literature that is concerned with mathematical modeling of neural networks – see for example (Miranda and Herrmann, 1991), (Makarenkov and Kirillov, 1991) and (Sporns and Kötter, 2004)–, and also by the fact that experimental works show that some cortical regions, such as specific regions of the primate visual cortex, present high similarity with multidimensional lattices – see (Rockland and Lund, 1983) and (Yoshioka et al., 1992)). We also deliberately chose non-random graphs in order to avoid a double source of stochasticity, which would uselessly complicate the analysis. For each of these multidimensional lattices we investigate the effect of changing the activation function to a linear function and to a sigmoid function. Both activation functions have been used in some form in stochastic models of biological neural networks – see for example (Brochini et al., 2016) and (Kinouchi

et al., 2019) – as well as in artificial neural networks – see for example (Maass, 1997).

This paper is organized as follows. In Section 4.2 we give a formal definition of the model. In Section 4.3 we give a formal presentation of the rigorous results that were proven in previous papers. In Section 4.4 we give a description of the algorithm used for the simulations. In Section 4.5 we give a precise characterization of the instantiations that we considered. In Section 4.6 we present the results of our simulations. Finally in Section 4.7 we discuss these results.

4.2 Definition of the model

Informally the model we consider is as follows. I is a countable set representing the neurons, and to each $i \in I$ is associated a set $\mathbb{V}_i$ of *presynaptic neurons*. If you consider the elements of I as vertices, and draw and edge from j to i whenever $j \in \mathbb{V}_i$, then you obtain the graph structure of the network. The *membrane potential* of neuron i is a stochastic process denoted $(X_i(t))_{t\geq 0}$ taking value in the set $\mathbb{N}$ of non-negative integers. Moreover, we associate to each neuron a Poisson process $(N_i^\dagger(t))_{t\geq 0}$ of some parameter γ, representing the *leak times*. At any of these leak times the membrane potential of the neuron concerned is reset to 0. Another point process $(N_i(t))_{t\geq 0}$ representing the *spiking times* is also associated to each neuron, which rate at time t is given by $\phi(X_i(t))$, where ϕ is the rate function. Whenever a neuron spikes its membrane potential is also reset to 0 and the membrane potential of all of its post-synaptic neurons is increased by one (i.e. the neurons of the set $\{j : i \in \mathbb{V}_j\}$). All the point processes involved are assumed to be are mutually independent.

More formally, beside asking that $(N_i^\dagger(t))_{t\geq 0}$ be a Poisson process of some parameter γ, this is the same as saying that $(N_i(t))_{t\geq 0}$ is the point process characterized by the following equation

$$\mathbb{E}(N_i(t) - N_i(s)|\mathscr{F}_s) = \int_s^t \mathbb{E}(\phi(X_i(u))|\mathscr{F}_s)du$$

where

$$X_i(t) = \sum_{j\in\mathbb{V}_i} \int_{]L_i(t),t[} dN_j(s),$$

$L_i(t)$ being the time of the last event affecting neuron i before time t, that is,

$$L_i(t) = \sup\Big\{s \leq t : N_i(\{s\}) = 1 \text{ or } N_i^\dagger(\{s\}) = 1\Big\}.$$

$(\mathscr{F}_t)_{t\geq 0}$ is the standard filtration for the point processes involved here. See Ferrari et al. (2018) for more details.

4.3 Rigorous results

In Ferrari et al. (2018), André (2019c), and André (2019a) a specific version of the model above was studied. The graph structure chosen there was the one-dimensional lattice, i.e. $I = \mathbb{Z}$ with $\mathbb{V}_i = \{i-1, i+1\}$. Moreover the activation function was chosen to be a hard threshold of the form $\phi(x) = \mathbb{1}_{x>0}$. In such a paradigm the rate of the point processes representing spiking times can only take two values: 0 or 1, depending on whether the membrane potential is positive or null. In this context whenever the membrane potential of a neuron is different from 0 we say that the neuron is *active*, otherwise we say that it is *quiescent*. More generally, we will only consider here the function ϕ satisfying $\phi(0) = 0$ and $\phi(x) > 0$ for $x > 0$, so that we will keep this distinction between active and quiescent neurons.

In Ferrari et al. (2018) it was proven that in the case of the one-dimensional lattice with hard threshold the following theorem holds

Theorem 4.3.1. *Suppose that for any $i \in \mathbb{Z}$ we have $X_i(0) \geq 1$. There exists a critical value γ_c for the parameter γ, with $0 < \gamma_c < \infty$, such that for any $i \in \mathbb{Z}$*

$$\mathbb{P}\Big(N_i([0,\infty[) \ = \infty\Big) > 0 \text{ if } \gamma < \gamma_c$$

and

$$\mathbb{P}\Big(N_i([0,\infty[) \ < \infty\Big) = 1 \text{ if } \gamma > \gamma_c.$$

It means that there is a critical point for the parameter γ, such that below this critical point each neuron stays active forever (with positive probability), and above it each neuron becomes quiescent if you wait long enough. These two mutually exclusive type of behavior for the system are called the sub-critical regime (or phase) and the super-critical regime respectively.

The process as a whole of course never dies completely because of the fact that there is an infinite number of neurons, so that it doesn't makes sense to consider the extinction time. Nonetheless, nothing prevents us to consider a finite version of this model. Suppose we're still in the case $\mathbb{V}_i = \{i-1, i+1\}$ and $\phi(x) = \mathbb{1}_{x>0}$, and for any $N \geq 0$ consider the system defined on the finite set $I_N = \{-N, \dots N\}$ instead of the whole lattice. Then you can define the process $(\xi_N(t))_{t\geq 0}$ of the spiking rates of the system, that is, the process taking value in $\{0,1\}^{I_N}$ defined by $\xi_N(t)_i = \mathbb{1}_{X_i(t)>0}$. This is a Markov process, belonging to the category of interacting particle systems, and by classical results on Markov processes we know that it will

always reach the state 0^{I_N} - where all neurons are quiescent - in finite time. We can therefore consider the extinction time of this finite model, which we denote σ_N, and it is natural to ask about its distribution in each of the two phases distinguished by the theorem above. It was proven in André (2019c) that the following holds.

Theorem 4.3.2. *There exists γ_c' such that if $\gamma < \gamma_c'$, then we have the following convergence*

$$\frac{\sigma_N}{\mathbb{E}(\sigma_N)} \xrightarrow[N\to\infty]{\mathscr{L}} \mathscr{E}(1).$$

In words, the re-normalized time of extinction converges in distribution to an exponential random variable of mean 1.

We know that $\gamma_c' < \gamma_c$, and the fact that the theorem is stated for some γ_c' and not for the critical value γ_c comes from essentially technical reasons related to the way the proof is built. We believe that this metastable result holds for the whole sub-critical region but it is not yet proven. Moreover, it was proven in André (2019a) that we also have the following.

Theorem 4.3.3. *Suppose that $\gamma > 1$. Then the following convergence holds*

$$\frac{\sigma_N}{\mathbb{E}(\sigma_N)} \xrightarrow[N\to\infty]{\mathbb{P}} 1.$$

We know that $\gamma_c < 1$, so that the result concerns a sub-region of the super-critical region, and as for the previous result, while to the best of our knowledge the result has been proven only for $\gamma > 1$, it is reasonable to expect that it holds in the whole super-critical region.

As stated in the introduction the choice of a one-dimensional lattice for the graph of interaction and of a hard threshold for the activation function were initially essentially motivated by mathematical conveniency, and we're interested in checking that the results hold for a richer class of instantiations of the model. Namely, we are aimed to establish numerically that the results related to the asymptotical distribution of the extinction time stated for the one-dimensional case in Theorem 4.3.2 and Theorem 4.3.3 remain valid for lattices of dimension 2 and 3. We also investigate the effect of changing the activation function to a linear function and to a sigmoid function on these results.

4.4 Simulation algorithm

All simulations were done in Python. The algorithm used to simulate our system of spiking neurons can be informally described as follows:

- **Initial configuration:** The network start with all neurons actives (by default membrane potential equal 1), and for each neuron an initial spiking value is sampled from an exponential random variable of parameter 1 (corresponding to the first atom of $(N_i(t))_{t\geq 0}$) and an initial leaking time value is sampled for each neuron from an another independent exponential random variable of parameter γ (corresponding to the first atom of $(N_i^\dagger(t))_{t\geq 0}$).
- **Interaction:** The current time is set to be the smallest of the previously sampled values, the neuron i associated with this value is found and the value of its membrane potential is set to 0. In case the event considered is a spike the membrane potential of the neurons in the set $\{j : i \in \mathbb{V}_j\}$ (the set of post-synaptic neurons) is increased by one. The membrane potential of neuron i being equal to 0 we set the next spiking time for neuron i to be infinite until further notice. If the event was a leaking then we sample an exponential random variable of parameter γ and add it to the current time to get the next leaking time for neuron i. If the event was a spike then we sample an exponential random variable of parameter $\phi(X_j)$ for all post-synaptic neuron j and add this value to the current time to get the next spiking time for these neurons.
- **Stopping condition:** The previous operation is iterated until all neurons are quiescent.

More formally, our simulation algorithm can be described by the following pseudo-algorithm.

All codes are available at `https://github.com/ceciliaromaro/Finite_Spike`.

Algorithm 4 Simulate the system of spiking neurons and return the extinction time

I the (finite) set of neurons.
ϕ the activation function.
γ the rate of the leaking point processes.
t the current time.
$\mathbb{V}_i$ the set of presynaptic neurons for neuron i.
X_i the membrane potential of neuron i at the current time.
$\sigma_i^\dagger$ the time of the next leaking for neuron i.
σ_i the time of the next spike for neuron i.

INITIALIZATION

$t \leftarrow 0$
for each i in I **do**
    $X_i \leftarrow 1$
for each i in I **do**
    $\sigma_i^\dagger \leftarrow \mathscr{E}(\gamma)$ ▷ $\mathscr{E}$ denotes the realization of an exponential random variable
    $\sigma_i \leftarrow \mathscr{E}(\phi(X_i))$

SIMULATION

while $\sum_{i \in I} X_i \neq 0$ **do**
    $MinLeaking \leftarrow \min_{i \in I} \sigma_i^\dagger$
    $MinSpiking \leftarrow \min_{i \in I} \sigma_i$
    if $MinLeaking < MinSpiking$ **then**
        $t = MinLeaking$
        $i \leftarrow \operatorname{argmin}_{j \in I} \sigma_j^\dagger$
        $X_i \leftarrow 0$
        $\sigma_i \leftarrow \infty$
        $\sigma_i^\dagger \leftarrow t + \mathscr{E}(\gamma)$
    else
        $t = MinSpiking$
        $i \leftarrow \operatorname{argmin}_{j \in I} \sigma_j$
        $X_i \leftarrow 0$
        $\sigma_i \leftarrow \infty$
        for each j such that $i \in \mathbb{V}_j$ **do**
            $X_j \leftarrow X_j + 1$
            $\sigma_j \leftarrow t + \mathscr{E}(\phi(X_j))$
$\sigma_N \leftarrow t$
return σ_N

4.5 Models investigated

In this section we specify the structure of the graph of interaction and the activation function we're interested in.

4.5.1 Multi-dimensional lattices

For the graph of the network we consider the lattices $\mathbb{Z}^1$, $\mathbb{Z}^2$ and $\mathbb{Z}^3$. For any $d \in \{1, 2, 3\}$, let $\|\cdot\|$ be the norm on $\mathbb{Z}^d$ given for any $j \in \mathbb{Z}^d$ by

$$\|j\| = \sum_{k=1}^{d} |j_k|,$$

where j_k is the k-th coordinate of j. The structure of the network is then given by $I = \mathbb{Z}^d$ and $\mathbb{V}_i = \{j \in I^d : \|i - j\| = 1\}$ for $i \in I$.

Notice that by defining the set of presynaptic neurons as the set of the nearest neighbours we actually have $j \in \mathbb{V}_i$ if and only if $i \in \mathbb{V}_j$. In other words, for a given neuron the set of the presynaptic neurons and the set of the postsynaptic neurons are equal. For this specific choice the graph of interaction is therefore actually undirected.

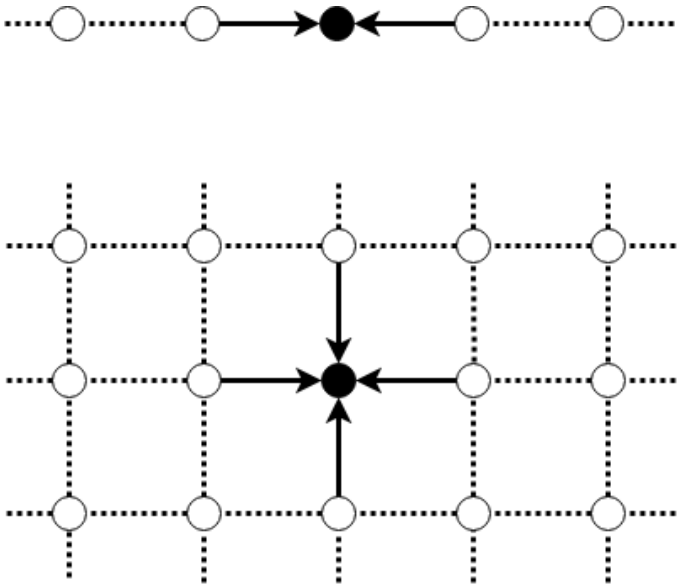


Figure 29: One-dimensional and two-dimensional lattices. A directed arrow is drawn toward the black neuron from each of its presynaptic neurons.

4.5.2 Linear and sigmoid activation functions

The activation function considered in André (2019c), Ferrari et al. (2018) and André (2019a) was the hard threshold $\phi(x) = \mathbb{1}_{x>0}$. This choice is convenient mathematically as the system then becomes an additive interacting particle system where any neuron can only have two possible values for the spiking rate at any time: 0 or 1. Nonetheless, from a biological point of view, a hard threshold is a rough choice, and we would like to consider smoother options.

The first option we consider is a linear function of the simplest form: $\phi(x) = x$.

The second option we consider is a somewhat more sophisticated sigmoid function of the following form

$$\phi(x) = \begin{cases} (1 + e^{-3x+6})^{-1} & \text{if } x > 0, \\ 0 & \text{if } x = 0. \end{cases}$$

Notice that we need to have $\phi(0) = 0$, in order to avoid spontaneous spiking (neuron with null membrane potential that spikes nonetheless). This is the reason why we force this value for the sigmoid function.

4.6 Results

We run simulations for instantiations of the system of spiking neurons consisting of all the possible combinations between the graphs and activation functions described above.

4.6.1 Simulations with a fixed number of neurons

4.6.1.1 Multidimensional lattices and hard threshold

For each of the three lattices the Algorithm 4 described in Section 4.4 was run, with an activation function of the form $\phi(x) = \mathbb{1}_{x>0}$. Each of the simulations were run for two different values of γ, 10,000 times for each of these values, using a number of neurons of the order of 100. The mean of the time of extinction σ_N was then computed using these data, and used to build the re-normalized histogram in each of these cases. The exact values for the size of the network and for the parameter γ can be found in Table 16.

The resulting histograms are presented in Figure 30 and 31.

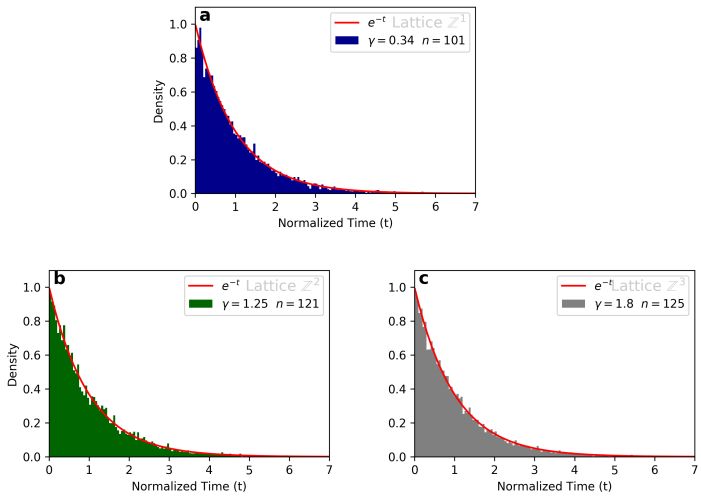


Figure 30: Histogram of the re-normalized time of extinction σ_N for small values of gamma, and an activation function of the form $\phi(x) = \mathbb{1}_{x>0}$. In **a**, **b** and **c** the blue, green and gray bars are the histograms for the time of extinction in the one-dimensional lattice, two-dimensional lattice and three-dimensional lattice respectively. The red line is the exponential function $t \mapsto e^{-t}$, which corresponds to the density of an exponential law of parameter 1. The parameter n corresponds to the number of neurons.

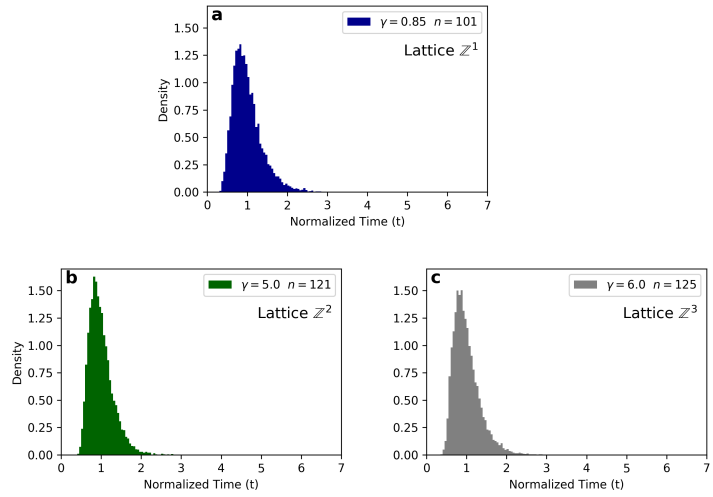


Figure 31: Histogram of the re-normalized time of extinction σ_N for high values of gamma, and an activation function of the form $\phi(x) = \mathbb{1}_{x>0}$. In **a**, **b** and **c** the blue, green and gray bars are the histograms for the time of extinction in the one-dimensional lattice, two-dimensional lattice and three-dimensional lattice respectively. The parameter n corresponds to the number of neurons.

Lattice	Number of neurons	Value of γ	Figure
$\mathbb{Z}$	101	0.34	30
		0.85	31
$\mathbb{Z}^2$	121	1.25	30
		5.00	31
$\mathbb{Z}^3$	125	1.80	30
		6.00	31

Table 16: Values of the total number of neurons and of the parameter γ used in the simulation for each of the three lattices.

4.6.1.2 Multidimensional lattices, linear function and sigmoid function

The routine described above was repeated with the two other activation functions. Only the values of γ change, which was necessary as changing the activation function must change the critical value of the system. These values are given in Table 17.

Lattice	Number of neurons	Activation function	Value of γ	Figure
$\mathbb{Z}$	101	Linear	0.42	32
		Sigmoid	0.028	33
		Linear	1	32
		Sigmoid	0.85	33
$\mathbb{Z}^2$	121	Linear	1.70	32
		Sigmoid	0.2	33
		Linear	5.00	32
		Sigmoid	1.7	33
$\mathbb{Z}^3$	125	Linear	1.90	32
		Sigmoid	0.09	33
		Linear	6.00	32
		Sigmoid	1.8	33

Table 17: Values of the total number of neurons and of the parameter γ used in the simulation for each of the three lattices.

The resulting histograms are presented in figures 32 and 33.

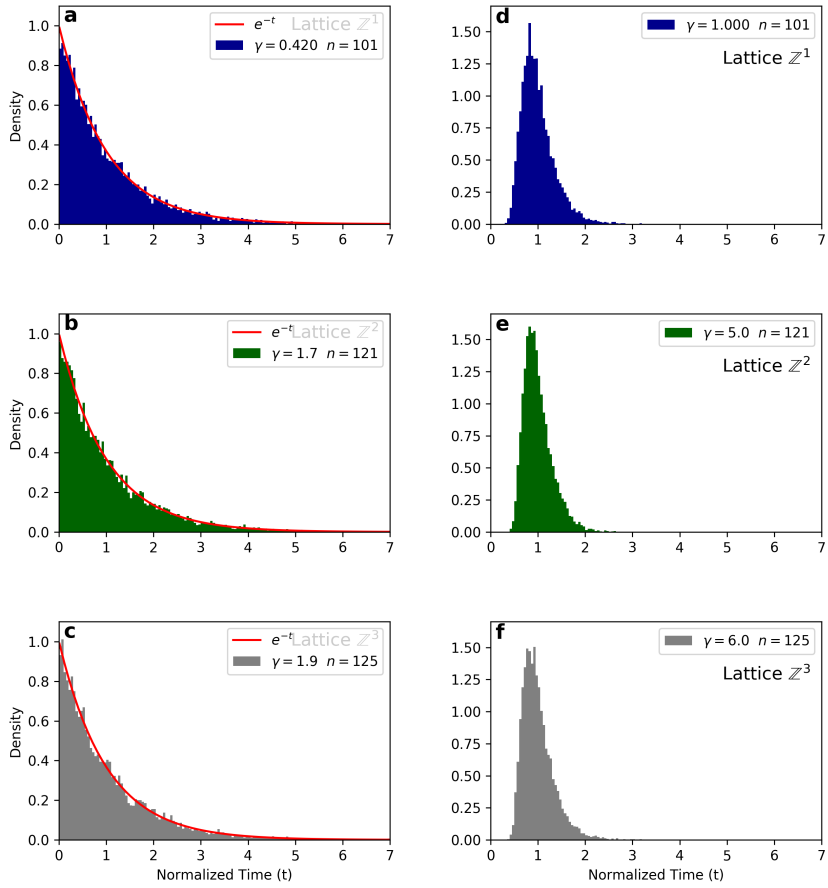


Figure 32: Histogram of the re-normalized time of extinction σ_N for a linear activation function for each of the three lattices. On the left side are the histograms for small values of γ and on the right side the histograms for high values of γ. The red line on the left side is the exponential function $t \mapsto e^{-t}$, which corresponds to the density of an exponential law of parameter 1. The parameter n corresponds to the number of neurons.

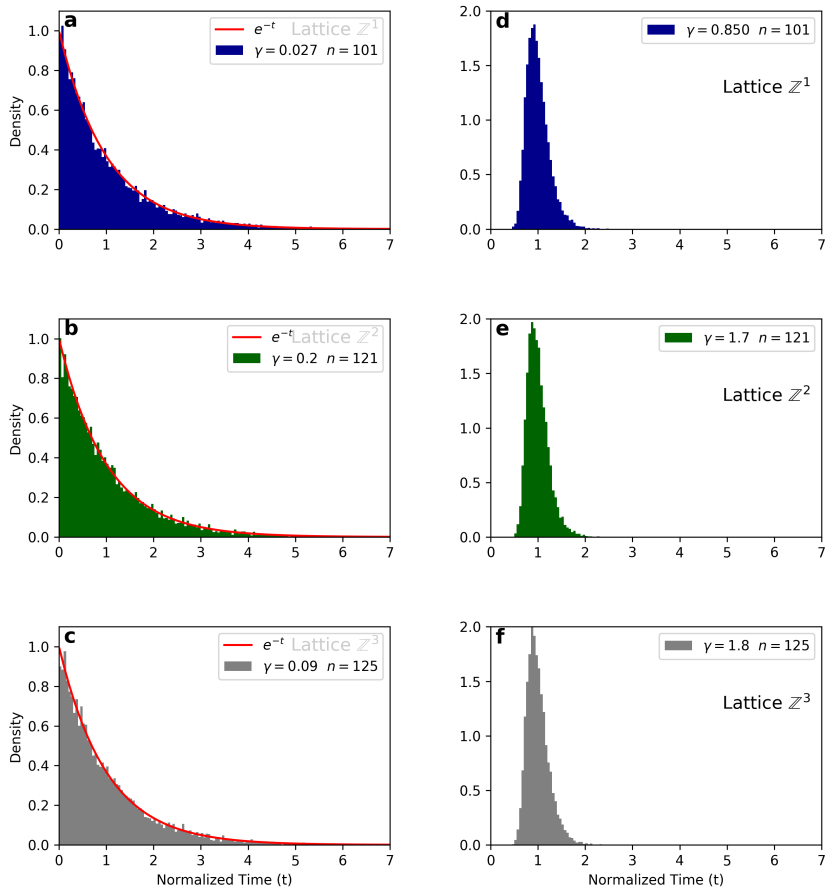


Figure 33: Histogram of the re-normalized time of extinction σ_N for a sigmoid activation function for each of the three lattices. On the left side are the histograms for small values of γ and on the right side the histograms for high values of γ. The red line on the left side is the exponential function $t \mapsto e^{-t}$, which corresponds to the density of an exponential law of parameter 1. The parameter n corresponds to the number of neurons.

4.6.2 Simulations for a varying number of neurons

To further investigate the behavior of the time of extinction in the super-critical regime, we've run a set of simulation for a (fixed) high value of γ and for a varying number of neurons. Each element of the set consists in 1000 repetitions with $\gamma = 4$. The value of the size of the network varies from 11 to 2000. For each of these values we've estimated the mean and variance of the extinction time σ_N, and the variance of the re-normalized extinction time $\sigma_N/\mathbb{E}(\sigma_N)$. These simulations have been done in the one-dimensional lattices for all of the three activation functions

The results of these simulations are presented in Figure 34, Figure 35 and Figure 36.

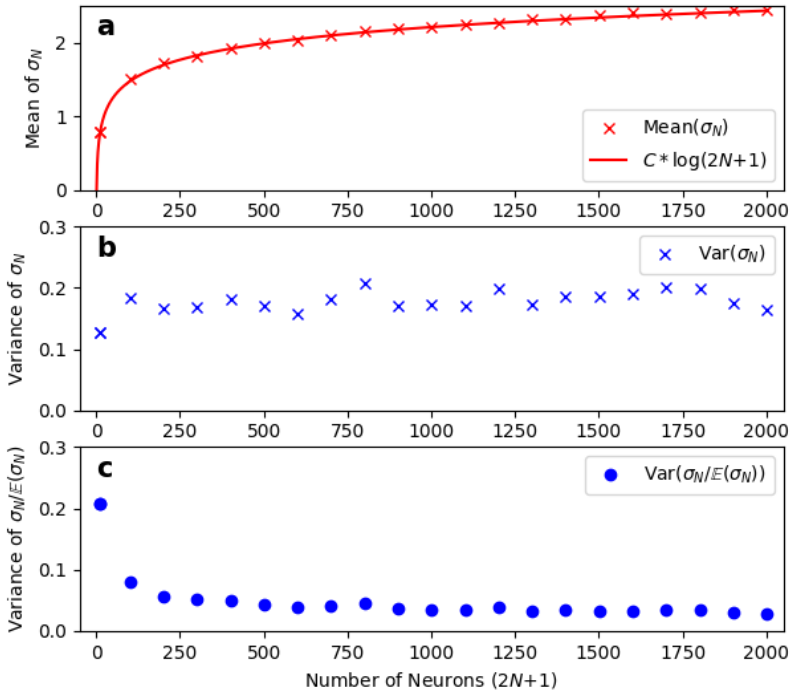


Figure 34: Mean and variance of σ_N and variance of the renormalized extinction time $\sigma_N/\mathbb{E}(\sigma_N)$ for a hard-threshold activation function. In **a** the red dots represents the values of the estimated mean of σ_N for varying numbers of neurons and the red line a logarithmic function fitted over the values of the mean ($C = 0.32$). In **b** the blue crosses represent the variance of σ_N as the number of neurons increases. The blue dots in **c** represent the variance of the renormalized extinction time $\sigma_N/\mathbb{E}(\sigma_N)$ for a varying number of neurons. These simulations were run with $\gamma = 4$.

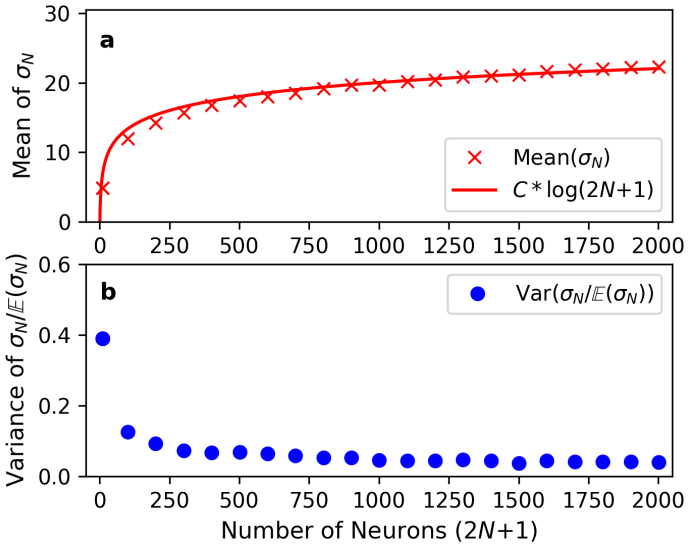


Figure 35: Mean of σ_N and variance of the renormalized extinction time $\sigma_N/\mathbb{E}(\sigma_N)$ for a linear activation function. In **a** the red dots represents the values of the estimated mean of σ_N for varying numbers of neurons and the red line a logarithmic function fitted over the values of the mean ($C = 2.9$). The blue dots in **b** represent the variance of the renormalized extinction time $\sigma_N/\mathbb{E}(\sigma_N)$ for a varying number of neurons. These simulations were run with $\gamma = 1$.

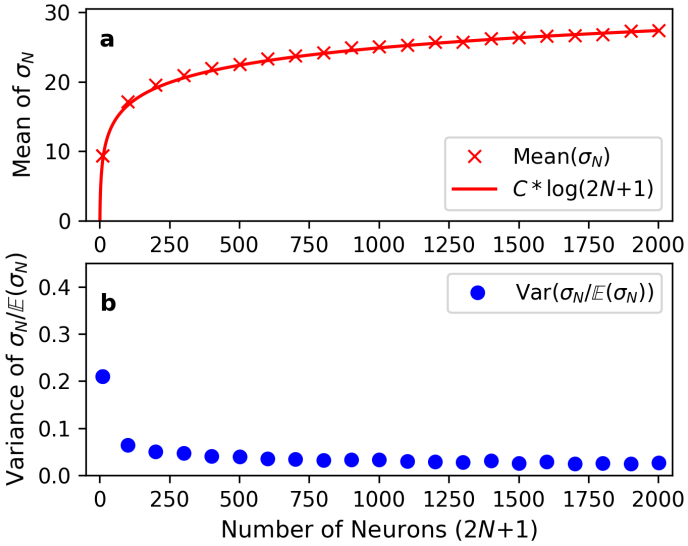


Figure 36: Mean of σ_N and variance of the renormalized extinction time $\sigma_N/\mathbb{E}(\sigma_N)$ for a sigmoid activation function. In **a** the red dots represents the values of the estimated mean of σ_N for varying numbers of neurons and the red line a logarithmic function fitted over the values of the mean ($C = 3.6$). The blue dots in **b** represent the variance of the renormalized extinction time $\sigma_N/\mathbb{E}(\sigma_N)$ for a varying number of neurons. These simulations were run with $\gamma = 0.34$.

4.7 Discussion

4.7.1 Sub-critical regime

The histogram built from the simulations for which γ is small (Figure 30 and left side of Figure 32 and Figure 33) closely approximates the density of a mean 1 exponential random variable. The fact that the result of the simulations remains identical in all the cases investigated (dimension one, two and three, with hard-threshold, linear function and sigmoid function) suggests that Theorem 4.3.2 doesn't merely hold for the specific instantiation of the model for which is was proven, but for a wide class of systems.

In the one dimensional case with hard threshold the fact that the histogram approximates the density of an exponentially distributed random variable is of course not a surprise as this is what Theorem 4.3.2 predicts asymptotically. Nonetheless, it gives us evidence that the approximation by an exponential law holds for relatively small networks (in the simulation concerned the number of neuron in the system is 101). The number of neurons in animals varies from hundreds (White et al., 1986) to billions (Lent et al., 2012), so that in our model the approximation by an exponential law is observed for all possible biologically realistic number of neurons. This indicates that, in networks where the randomness of the connections between neurons is not of interest, the model presented here might be an interesting choice for the investigation of metastable behaviors in dense cortical regions.

4.7.2 Super-critical regime

The histograms built from the simulations with high values of γ are visibly not approximating any exponential. Instead they show a distribution that is reminiscent of a gamma distribution with mass concentrated around 1 (See Figure 31 and right side of Figure 32 and Figure 33).

Moreover, the evolution of the variance of the renormalized time of extinction (last graph in Figure 34, Figure 35 and Figure 36) is seemingly converging toward 0 as the number of neurons grows for all of the three instantiations investigated.

For the simulation of the system with hard-threshold, these facts are not a surprise neither, as this is what Theorem 4.3.3 predicts. Again the fact that the simulations of the systems with a linear and a sigmoid function for ϕ show similar results suggests that Theorem 4.3.3 isn't only satisfied for $\phi(x) = \mathbb{1}_{x>0}$, but for a wide class of activation functions.

Furthermore, the first graph in Figure 34, Figure 35 and Figure 36 gives us strong evidence that in each of the three instantiations the expectation of the time of extinction grows approximately like a logarithm (up to a multiplicative constant) with respect to the number of neurons. This fact is interesting in itself as the proof of Theorem 4.3.3 (which can be found in André (2019a)) relies on the fact that for the hard-threshold instantiation of the model we have the following convergence in the super-critical region (at least for $\gamma > 1$)

$$\frac{\mathbb{E}(\tau_N)}{\log(2N+1)} \underset{N\to\infty}{\longrightarrow} C, \tag{4.7.1}$$

where C is a strictly positive (and finite) constant. This is an additional hint that the behavior of the time of extinction should be qualitatively identical in the super-critical region as well for any of the choices we proposed here for the activation function.

Acknowledgments

This work was produced as part of the activities of FAPESP Research, Disseminations and Innovation Center for Neuromathematics (Grant 2013/07699-0, S. Paulo Research Foundation). Morgan André is supported by a FAPESP scholarship (grant number 2017/02035-7), Cecilia Romaro (grant number 88882.378774/2019-01) and Fernando Araujo Najman (grant number 88882.377124/2019-01) are the recipient of PhD scholarships from the Brazilian Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

References

André, M. (2019a). Asymptotically deterministic time of extinction for a stochastic system of spiking neurons. *arXiv:1910.00055v2*.

André, M. (2019c). A result of metastability for an infinite system of spiking neurons. *Journal of Statistical Physics*, 177(5):984–1008.

Brillinger, D. R. (1988). Maximum likelihood analysis of spike trains of interacting nerve cells. *Biological Cybernetics*, 59(3):189–200.

Brochini, L., de Andrade Costa, A., Abadi, M., Roque, A. C., Stolfi, J., e Kinouchi, O. (2016). Phase transitions and self-organized criticality in networks of stochastic spiking neurons. *Scientific Reports*, 6:35831.

Brunel, N. (2000). Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. *Journal of Computational Neuroscience*, 8(3):183–208.

Cassandro, M., Galves, A., Olivieri, E., and Vares, M. E. (1984). Metastable behavior of stochastic dynamics: A pathwise approach. *Journal of Statistical Physics*, 35(5-6):603–634.

Chevallier, J., Caceres, M. J., Doumic, M., and Reynaud-Bouret, P. (2015). Microscopic approach of a time elapsed neural model. *Mathematical Models and Methods in Applied Sciences*, 25(14):2669–2719.

Chornoboy, E. S., Schramm, L. P., and Karr, A. F. (1988). Maximum likelihood identification of neural point process systems. *Biological Cybernetics*, 59(4-5):265–275.

Deco, G., Kringelbach, M. L., Jirsa, V. K., and Ritter, P. (2017). The dynamics of resting fluctuations in the brain: metastability and its dynamical cortical core. *Scientific Reports*, 7(1):1–14.

Duarte, A. and Ost, G. (2016). A model for neural activity in the absence of external stimuli. *Markov Processes Related Fields*, 22(1):35–52.

Ferrari, P., Galves, A., Grigorescu, I., and Löcherbach, E. (2018). Phase transition for infinite systems of spiking neurons. *Journal of Statistical Physics*, 172(6):1564–1575.

Galves, A. and Löcherbach, E. (2013). Infinite systems of interacting chains with memory of variable length—a stochastic model for biological neural nets. *Journal of Statistical Physics*, 151(5):896–921.

Galves, A., Löcherbach, E., and Pouzat, C. (2019). A system of interacting neurons with short term synaptic facilitation. *Journal of Statistical Physics* 178(4):869–892.

Hansen, N. R., Reynaud-Bouret, P., and Rivoirard, V. (2015). Lasso and probabilistic inequalities for multivariate point processes. *Bernoulli*, 21(1):83–143.

Hawkes, A. G. (1971). Point spectra of some mutually exciting point processes. *Journal of the Royal Statistical Society*, 33(3):438–443.

Kass, R. E. and Ventura, V. (2001). A spike-train probability model. *Neural Computation*, 13(8):1713–1720.

Kinouchi, O., Brochini, L., Costa, A. A., Campos, J. G. F., and Copelli, M. (2019). Stochastic oscillations and dragon king avalanches in self-organized quasi-critical systems. *Scientific Reports*, 9(1):3874.

Lent, R., Azevedo, F. A., Andrade-Moraes, C. H., and Pinto, A. V. (2012). How many neurons do you have? some dogmas of quantitative neuroscience under revision. *European Journal of Neuroscience*, 35(1):1–9.

Maass, W. (1997). Networks of spiking neurons: the third generation of neural network models. *Neural Networks*, 10(9):1659–1671.

Makarenkov, V. I. and Kirillov, A. (1991). Self-organized criticality in neural networks. In *Applications of Artificial Neural Networks II*, volume 1469, pages 843–845. International Society for Optics and Photonics.

Millman, D., Mihalas, S., Kirkwood, A., and Niebur, E. (2010). Self-organized criticality occurs in non-conservative neuronal networks during 'up' states. *Nature Physics*, 6(10):801.

Miranda, E. and Herrmann, H. (1991). Self-organized criticality with disorder and frustration. *Physica A: Statistical Mechanics and its Applications*, 175(3):339–344.

Roberts, J. A., Gollo, L. L., Abeysuriya, R. G., Roberts, G., Mitchell, P. B., Woolrich, M. W., and Breakspear, M. (2019). Metastable brain waves. *Nature Communications*, 10(1):1–17.

Rockland, K. S. and Lund, J. S. (1983). Intrinsic laminar lattice connections in primate visual cortex. *Journal of Comparative Neurology*, 216(3):303–318.

Sporns, O. and Kötter, R. (2004). Motifs in brain networks. *PLoS Biology*, 2(11):e369.

Vreeswijk, C. v. and Sompolinsky, H. (1998). Chaotic balanced state in a model of cortical circuits. *Neural Computation*, 10(6):1321–1371.

White, J. G., Southgate, E., Thomson, J. N., and Brenner, S. (1986). The structure of the nervous system of the nematode caenorhabditis elegans. *Philos Trans R Soc Lond B Biol Sci*, 314(1165):1–340.

Yoshioka, T., Levitt, J. B., and Lund, J. S. (1992). Intrinsic lattice connections of macaque monkey visual cortical area v4. *Journal of Neuroscience*, 12(7):2785–2802.

PART V

5 STOCHASTIC NEURAL NETWORK MODEL BASED ON THE MICROCIRCUITRY OF SOMATOSENSORY CORTEX OF JUVENILE RAT

Abstract

Markram and collaborators produced a highly biologically constrained reconstruction of the microcircuitry of the somatosensory cortex of juvenile rat. In this work, we statistically processed the connectivity of the reconstructed network and, based on the obtained results, constructed a 5-layer network model of the somatosensory cortex of juvenile rat. The model is composed of excitatory and inhibitory neurons described by the stochastic model of Galves and Löcherbach. Three biological data from the reconstruction were used in the construction of our model: the number of connections between layers, the number of synapses per connection, and the average strength of the synapses. The model outcomes were biologically plausible behaviors such as self-sustained average firing rate per layer and thalamic input response. The model also presented dynamic features like asynchronous activity state, irregular neuronal firing and a phase transition that was quickly evoked by thalamic input. This work leads do some layer-connections load conclusions and also proposes explanations based on connectome for phenomena such as the ion channel-conductance, and the reported concentrations differences between patch clamp data and in-vivo-like models.

5.1 Introduction

“If our brain were simple enough for us to study it, we would not be able to do it" (Unknown Author). The challenge is not in the inaccessibility of the brain but in its complex mechanisms. Modeling in neuroscience has been typically used to understand the neurons (Lapicque, 1907; Hodgkin and Huxley, 1952) and neuronal systems (Tsodyks et al., 1998; Millman et al., 2010; Kriener et al., 2014). The cortical network features, cognition, code and decoding processing might be either in neuronal intracellular, synaptic strength or synapse location, arrangement of

neurons or synapses, connections between cortical locations or any combinations of the above possibilities. The neuroscience current focus on the connectome requires not only the mapping of all neuronal connections, but also the studying and understanding of features correlated to the organization of neural interactions.

As part of this effort, Markram et al. (2015) compiled several years of data collected by his team and himself in the reconstruction of the microcircuitry of the somatosensory cortex of juvenile rat. The reconstruction is based on anatomical and physiological data and the network corresponds to a volume of about 0.30 mm^3 comprising six cortical layers. It contains about 31,000 neurons of different layer-specific morphologies with circa 8 million connections and 37 million synapses. The neurons are modeled according to Rall's compartmental approach and the Hodgkin-Huxley formalism. The model was used to simulate spontaneous activity under in vitro and in vivo like conditions, which were mimicked by controlling the simulated extracellular calcium concentration (the in vivo concentration is lower than the in vitro one).

The model by Markram et al. (2015) provides a bridge between cellular and synaptic mechanisms and emergent network activity states, however the number of details included in the model is so large that it obscures the understanding of the underlying mechanisms responsible by cortical dynamics. Focusing on clarifying how local network dynamics arises from network connectivity (independently from neuronal morphology), we statistically processed the publicly-available connectivity data for the 31 thousand neurons in the Markram et al. model and condensed them into connectivity data for 9 neuronal populations.

Based in the reconstruction made by Markram et al. (2015), we calculated the weighted average of the number of connections between the nine populations, the synapses per connection for each pair of populations, the synaptic strength for each pair of populations, and the probability of connection for each pair of populations.

Then, we built, a 5-layer network model of excitatory and inhibitory neurons (4 layers with both excitatory and inhibitory neurons and one layer with exclusively inhibitory neurons). The neuron model used to describe these neurons was the stochastic model of Galves and Löcherbach (2013) (GL). Simulations of the resulting model produced self-sustained activity states with average firing rate per layer and thalamic input response in agreement with biological evidence. The network model also presented an asynchronous and irregular activity state and a phase transition that was quickly evoked by thalamic input.

The purpose of this work is to use the pattern of connections between inhibitory and excitatory neurons and the respective synaptic strengths given in the model of Markram et al. (2015), which we can call the *connectome* of the Markram et al. (2015) model, to construct a model

based on point neurons (i.e. without morphology) able to reproduce some dynamic cortical behaviors indicative of being originated from the pattern of connections.

5.1.1 Paper organization

This paper is organized as follows: Section 5.2 starts by giving a description of the database features and the connectome analysis. In Subsection 5.2.1 we present the results of the connectome analysis and in Subsection 5.2.2 we discuss the connection balance. Section 5.3 starts by giving a description of the somatosensory model and its construction algorithm. In Subsection 5.3.1 we present the network behavior and statistics for different initial condition of the network state, and, finally, in Subsection 5.3.2, we discuss the network properties and statistics, and make a parallel with the model of Potjans and Diesmann (2014a) (PD). A quick conclusion is presented in Section 5.4 and 5.5.

5.2 Connectome analysis

The reconstruction (Markram et al., 2015) contains 55 morphological types of neurons and 10 excitatory and 1 inhibitory electrophysiological types, which result in 207 possible morpho-electrical sub-type combinations (because not every morphology type is compatible with every electrophysiology type). A cortical volume of 0.29 mm^3 was reconstructed, which is equivalent to a column under an area of 0.14 mm^2 containing about 31 thousand neurons distributed in 6 layers with 8 million connections, totaling 37 million synapses in the reconstructed area. Given the morphological characteristic of the neuron and its location in the cortical layer, synapses were classified in 1941 types of multi-synaptic connection. These synapses are estimated to correspond to only 20% of the existing synapses in the reconstructed region. The other connections come from other regions of the nervous system, including neighboring unreconstructed areas.

The reconstruction database (Ramaswamy et al., 2015) provides data for 1933 of the types of multi-synaptic connections. The information of the 7 missing types was requested to the database owner; however we have not received any answer yet. For each type of connection, among other morphological information, the following are available: the total number of synapses, the size of the post-synaptic potential (PSP) and the average number of synapses per connection and their standard deviations. The database also gives the number of neurons for each morphological class for each layer of the cortical column.

The information reported above allows the distribution of the 31 thousand neurons into

5 layers, which we named L1, L2/3, L4, L5 and L6, and 9 neuronal populations, named L1i, L2/3e, L2/3i, L4e, L4i, L5e, L5i, L6e, L6i, where the letters e/i indicate excitatory and inhibitory neurons, respectively. Note that layer 1 has only inhibitory neurons.

The information also allows, by using weighted averages, the calculation of the connection probabilities between neurons from one population to neurons from another population populations, totaling 81 probabilities, and the number of synapses per connection for each of these 81 probabilities, as well as their corresponding synaptic weight and their respective standard deviations. This primary source of processed information is available at 5.2.1 and will be used to build the model. The combinations of the tables which help in the data analyses are presented in the supplementary material (Section 5.5).

It is important to point out that a connection means that a pair of neurons – the pre-synaptic neuron and the post-synaptic neuron – are connected to each other by at least one synapse. The same pair of neurons cannot have more than one connection – there is an unlikely probability that the same pair of neurons are connected twice in the same pre- and post-synaptic order –, but one connection can have many synapses.

Lastly, the average innervation number from a single thalamic neuron and the synapse/connection ratio for L4 and L5 neurons were estimated (data from article, not database) using the weighted average. Table 22 will present the average innervation ratio from a single thalamic neuron.

5.2.1 Results

Before presenting the results it is relevant to point out that we will be modeling here chemical synapses, which means that neurons can be classified based on the type of neurotransmitter they release, either excitatory or inhibitory. This means that an excitatory neuron makes synapses in both excitatory and inhibitory neurons, and an inhibitory neuron makes synapses in both excitatory and inhibitory neurons.

Table 18 shows the distribution of neurons over the 9 populations in absolute numbers and in percentage. One can see that the overall number of excitatory neurons is higher than the overall number of inhibitory neurons, even inside the same layer. The only exception is the layer L1, which has only inhibitory neurons.

Layer	L1i	L2/3e	L2/3i	L4e	L4i	L5e	L5i	L6e	L6i	Total
Number of neurons	338	5877	1647	4178	478	5050	1064	11462	1189	31283
Proportion per layer (%)	1.1	18.8	5.3	13.4	1.5	16.1	3.4	36.6	3.8	100.0

Table 18: Distribution of neurons by population.

The numbers of connections and synapses per connection, and the synaptic strengths for each pair of populations are shown in Tables 19-21. They were obtained by weighted average of the 1933 types of multi-synaptic connections and the absolute number of connections per pair of population.

In Table 19, one can see that, for a given population (fix a row), with exception of the population L1i (first row), the connections **from** any layer (pair of columns) are majorly excitatory, i.e. there are more excitatory connections than inhibitory connections from this layer. The same result can be observed in supplementary Table 29: for a given neuron in a population (fix a row), with exception of neurons in population L1i (first row), the average connection from any layer (pair of rows) are majorly excitatory, i.e. the average number of excitatory connections is higher than the average number of inhibitory connections from this layer. The opposite occurs for population L1i, the connections **from** any layer (pair of columns) are majorly inhibitory, i.e. there are more inhibitory connections than excitatory connections targeting this population.

Still in Table 19, one can also see that for a given population (fix a column), the connections **to** any layer (pair of rows) target majorly excitatory neurons, i.e. there are more excitatory neurons targeted than inhibitory neurons targeted on the average. Note that the connection nature (excitatory/inhibitory) is given by population (column) and not the neuron target (row). The same result can be observed in supplementary Table 29: for a given population (fix a column), and a receiver layer (a pair of row), one excitatory neuron on this layer receives on the average more connections from the population than one inhibitory neuron on this layer. On the average, one excitatory neuron is targeted more times than one inhibitory neuron for a neuronal population. It is important to highlight that given a pre-synaptic population (excitatory or inhibitory) and a post-synaptic layer (with excitatory and inhibitory neurons), on the average, one excitatory neuron receives more connections than an inhibitory neuron. In other words, the probability of connection to an excitatory neuron is higher than the probability of connection to an inhibitory neuron, independently from the excitatory or inhibitory characteristic of the connection (see supplementary Table 30). Therefore, the total number of excitatory connections is higher than the total number of inhibitory connections, both when the target neuron is excitatory or inhibitory. The probability of connection to an excitatory neuron is higher than

to an inhibitory neuron but nothing can be said about the probability of excitatory or inhibitory connection.

		From								
		L1i	L2/3e	L2/3i	L4e	L4i	L5e	L5i	L6e	L6i
	L1i	1569	2614	3434	94	355	223	666	15	168
	L2e	26876	557875	119282	150312	21313	21313	32312	10350	19
	L2i	4547	68614	30935	17498	4553	4553	7590	1781	1004
To	L4e	6796	673860	68686	548389	42143	42143	50221	152270	15548
	L4i	62	20958	3646	25716	3828	3828	4424	6427	1360
	L5e	32913	1031685	138820	886544	46497	46497	125840	801180	47093
	L5i	36	33012	3202	34488	3238	3238	7356	29259	5577
	L6e	564	54432	28557	656952	30420	30420	64528	3178676	153232
	L6i	0	7620	392	10376	457	457	875	99002	11771

Table 19: Average number of connections per pair of populations.

The number of excitatory connections is higher than the number of inhibitory connections. However, each inhibitory connection tends to have a higher number of synapses than an excitatory connection.

In Table 20, one can see that for a given layer (fix the pair of columns), inhibitory connections have a higher number of synapses per connection than excitatory connections. It is also possible to see that, for a given population (Table 20, fix a pair of row), the excitatory connections when targeting inhibitory neurons have a higher number of synapses per connection than when targeting excitatory neurons. Parallel to that, the inhibitory connections when targeting excitatory neurons tend to have a higher number of synapses per connection than when targeting inhibitory neurons. Exceptions occur for connections from L2/3i and L6i to L4, and for connections from L5i and L6i to L2. The first three tend to be equal and the last one tends to be the opposite. Thus, inhibitory neurons tend to establish higher number of synapses per connection, but the number of excitatory synapses per connection is higher when targeting inhibitory neurons, and the number of inhibitory synapses per connection is higher when targeting excitatory neurons.

In Table 21, one can see that for a given layer (fix and compare the pair of columns line by line), the strength of the excitatory synapses is higher than the strength of the inhibitory synapses. It is possible to see on same Table 21 that for a given population (fix a pair of rows), the synaptic strength is higher when targeting inhibitory neurons than when targeting excita-

tory neurons, regardless of the excitatory or inhibitory aspect of the synapse. Thus, excitatory synapses tend to be stronger, but inhibitory neurons tend to receive stronger synapses.

		From								
		L1i	L2/3e	L2/3i	L4e	L4i	L5e	L5i	L6e	L6i
	L1i	15.84	4.41	11.01	2.85	11.13	3.63	11.85	3.30	7.67
	L2e	11.96	2.80	14.58	2.71	12.68	2.29	10.53	2.20	2.70
	L2i	11.33	5.63	13.75	5.36	12.50	4.62	10.69	4.40	7.58
	L4e	6.57	2.61	10.16	3.20	16.16	2.51	12.67	2.30	9.17
To	L4i	5.25	6.89	10.88	6.14	15.92	6.47	12.49	5.35	9.37
	L5e	13.14	3.69	13.91	4.22	18.03	5.73	17.65	3.10	17.43
	L5i	4.17	6.00	7.74	7.31	11.52	6.76	12.68	5.66	11.31
	L6e	4.09	2.39	7.08	2.47	9.11	3.23	9.49	3.05	13.49
	L6i	-	4.77	6.05	5.21	6.70	7.63	7.55	7.21	12.00

Table 20: Weighted average number of synapses per connection per pair of populations.

In absolute value, the weights of synapses that reach inhibitory neurons are larger than the ones that reach excitatory neurons, independently of their origin (compare pairs of rows in supplementary Table 31).

		From								
		L1i	L2/3e	L2/3i	L4e	L4i	L5e	L5i	L6e	L6i
	L1i	2.44	5.57	1.98	1.76	1.95	3.95	1.93	3.97	1.90
	L2e	3.36	1.90	0.54	1.97	0.49	2.28	0.44	1.50	0.15
	L2i	1.16	3.24	1.29	3.52	1.19	2.78	1.12	4.72	1.05
	L4e	1.25	1.56	0.41	2.44	0.47	1.49	0.43	1.55	0.40
To	L4i	0.68	2.58	0.96	2.60	1.15	2.27	1.07	2.24	0.91
	L5e	0.62	0.50	0.10	0.65	0.16	1.23	0.18	0.60	0.17
	L5i	0.45	1.24	0.92	1.65	0.99	1.59	1.04	1.29	0.99
	L6e	0.19	1.16	0.16	1.04	0.14	1.83	0.24	1.92	0.46
	L6i	-	2.12	0.93	2.11	0.85	2.62	0.86	2.57	1.04

Table 21: Weighted average of synaptic strength (PSP in mV) per connection per pair of populations.

Table 22 presents the number of connections and synapses that a single thalamic neuron innervates. On the average, the synapse/connection ratio for L5 cells is higher than for L4 cells.

This means that one thalamic cell makes stronger connections to L5 neurons than to L4 neurons.

Connections	Synapses	Synapses/connections L4	Synapses/connections L5
903	7314	6.7	11.2

Table 22: Innervation ratio of a single thalamic neuron.

5.2.2 Discussion

The number of excitatory neurons is higher than the number of inhibitory neurons (Table 18. As expected, the number of excitatory connections is higher than the number of inhibitory connections (19), once there are more excitatory neurons to make connections. However, it was not so expected that the probability of connection to an excitatory neuron be higher than to an inhibitory neuron. This results in a single excitatory neuron being targeted on the average more times than an inhibitory neuron, i.e. the probability of connection to an excitatory neurons is higher than to an inhibitory neuron. Actually, in a single layer the probability of an inhibitory connection in an excitatory neuron is higher than the probability of an excitatory connection in an inhibitory neuron. This is probably due to the size of target neurons: excitatory neurons tend to be bigger than inhibitory neurons.

Neurons in population L1i are the only ones that receive more inhibitory than excitatory connections. The strength of inhibitory synapses is larger than the strength of excitatory synapses. This is probably the reason for the lack of sustained activity in this population (see below).

For any single neuron, except for those in layer L1i, the number of excitatory connections is higher than the number of inhibitory connections, but each inhibitory connection tends to have a higher number of synapses than an excitatory connection. It is expected that the pool of inhibitory synapses of a single connection and many of the less-effective excitatory connections act together, keeping the system balanced – a temporal summation – as well as in our model.

In absolute value, the weights of synapses made on inhibitory neurons are stronger than the weights of synapses made on excitatory neurons, independently of their origin. This means that (supplementary Table 29) excitatory neurons receive more connections but (supplementary Table 31) with smaller weight (strength of the synapse multiplied by the average of number of synapses per connection), while inhibitory neurons receive less connections but with higher weight. This should compensate the low number of connections (comparing to the number of connections received by the excitatory neurons), so that, the average firing rates of inhibitory and excitatory neurons are similar

In the original work of Markram et al. (2015), adjustments in ionic conductances were required to reproduce in-vivo-like activity. Why was this necessary?

It is estimated that only about 20% of synapses are intrinsic from the volume of 0.3 mm^3, i.e. generated internally to the network. The other synapses, an estimated total of 148 million, come from other regions of the cortex (extrinsic synapses). We hypothesize that the loss of this extrinsic input may be one of the two responsible factors for the differences reported between the simulated and the in vivo neuronal activity. Temporal and spatial summations are required to explain the network activity with such pattern of connection. The mismatch between patch clamp data and the conductance set required to reproduce in vivo-like activity in the model could be explained as a result of unbalanced neurons due to the loss of extrinsic synapses. In other words, the loss of the homeostatic plasticity adjustment. Probably, if an adaptive term was provided to these neurons, they would be able to recover the homeostatic plasticity adjustment resulting in a neuronal activity in vivo-like even after being removed from cortex.

The in vivo-like activity is usually reproduced in in vivo-like models by adjustment of changing ion conductances, as in the original work of Markram et al. (2015). The loss of the extrinsic input may be responsible for the differences reported between the simulated and the in vivo neuronal activity but temperature could also have some effect. It is known that the temperature has an influence on the conductance channel, and so, on synaptic strength. The increase in the temperature increases the channel conductance, which increases the synaptic strength and, for a short period of time, the network activity given that less synapses are necessary to activate the neurons. Once the synaptic weight of inhibitory neurons are higher and effective, the balance is reestablished putting activity close to the initial level. Reverting the reasoning line, decreasing the temperature will decrease the channel conductance, which decreases the synaptic strength until the point that inhibitory synapses start to lose effectiveness and the network activity increases. The number of connections becomes more significant than the weights. The continuous decrease of the temperature will reach a point in which the number of connections also lose power and the neuronal network loses activity. Through this reasoning, it is possible that the patch clamp data cannot keep track of temperature.

Both the effects of the temperature and the synapses from other cortical regions (extrinsic synapses) may be simulated in our model. Other conditions such as thalamic input were also simulated.

5.3 Somatosensory cortical column model

Our somatosensory cortical column model is based on the following biological data taken from the reconstruction made by Markram et al. (2015):

- The number of excitatory and inhibitory neurons per layer – resulting in 9 populations of neurons (Table 18);
- The probability of connections per pair of populations (Table 30) – calculated by the average number of connections per pair of populations (Table 19) and the number of neurons per population (Table 18);
- The weighted average number of synapses per connection per pair of populations (Table 20);
- The weighted average of synaptic strength (PSP in mV) per connection per pair of populations (Table 21).

These parameters define the network structure. The neuron model adopted is the GL neuron model (Galves and Löcherbach, 2013).

Informally, the GL neuron model is a stochastic neuron with probability to spike as its parameter. This probability is a function that can be a constant in time, decrease in time (due to leakage), increase or decrease given an external event (inhibitory or excitatory synapse: spike from other neuron), present a steep change (reset after spike), or even be any other kind of function. This could be seen as a time-varying function.

More formally, it is a Poisson process in which the parameter λ is the mean firing rate. The time differences between events of the counting process are exponential variables with mean $1/\lambda$, where λ may be a time varying function.

A function translating the neuron membrane potential into a spike probability could be drawn as a numerical approximation with any desired level of accuracy. This means that it is possible to find a transfer function to translate a LIF (leaky integrate-and-fire) (Lapicque, 1907) neural network into a GL (Galves and Löcherbach, 2013) neural network. The method to do so will remain future endeavor.

In this work, we will apply the simplest membrane potential translation: a linear function. More specifically, a synapse with PSP of Δx (mV) leads to an increase of $\alpha \Delta x$ (mHz) in the parameter λ (mHz).

The linear function is given by:

$$\lambda(x) = \alpha x + \beta, \tag{5.3.1}$$

where α is the angular coefficient and β is the linear coefficient,

When relaxing the first axiom of Kolmogorov (mathematics of probability theory) and accepting a negative value for the probability, the λ parameter function is given by:

$$\lambda(x) = \alpha x + \beta, \text{ if } (\alpha x + \beta) > 0 \tag{5.3.2}$$

and

$$\lambda(x) = 0, \text{ if } (\alpha x + \beta) \leq 0. \tag{5.3.3}$$

Once the neuron spikes, the spike parameter λ goes to λ_0 instantaneously.

Note that it is not necessary to find values for β and $\lambda(0)$, and also for $\{x|\lambda(x) = \lambda_0\}$. Just α and λ_0 are sufficient to simulate the network model in a computer.

No leakage (traditionally represented by an exponential decay of the membrane potential) will be considerate on the firing rate probability in this model version, but some spikes may be needed before $\lambda(x)$ becomes higher than zero. Some points questioning the intent of this leakage modeling will be raised during the discussion.

Clearly the membrane potential does not reflect a linear function on spike probability. Previous works suggest it looks more like an exponential function (Cordeiro, 2019) than a linear function.

Also clearly, for a Poisson point process with a mean firing rate parameter of λ_o, once the neuron spiked (an event) the parameter is reset to λ_o again so that the average firing rate remains λ_o. Nevertheless, an input process of rate q per second added to the neuron does not increase the average firing rate by q (after a spike, the firing rate parameter is reset to λ_o again until a new input of the process q happens).

Given that, the linearization was made near the firing rate target. A single α was estimated for all network in order to preserve the synaptic weight rate per population and a value of λ_o below zero was set for each population. The idea is to build a network with self-sustained activity due to its pattern of connections while for the ability to change state – and even stop the self-sustained activity – the λ_o must be below zero, otherwise some network layers would

present activity independently of the pattern of connections.

Table 23 presents mean firing rates recorded from specific layers of the somatosensory cortex of the rat (de Kock and Sakmann, 2009). Based on these firing rates and on Tables 18 to 21, the average Δx received in 1 second was estimated for each neuron population and is presented in Table 24. In the same table, estimations of the $\alpha_e \Delta x$ increase in the λ parameter – if no event occurs – were estimated for different values of α_e.

After an event (spike), the neuron does not keep its previous λ. So after picking one α_e value that could result in a good increase in λ, we multiply α_e by 10. This is the reason for $\alpha = 6$ in this model. Note that any other value of α_e could be used, as well as any other multiplying factor higher or lower than 10.

Layer	L1	L2/3	L4	L5	L6
Firing rate (Hz)	-	0.3	1.4	2-3	0.5

Table 23: In vivo mean firing rates recorded from the somatosensory cortex of the rat (de Kock and Sakmann, 2009).

	$\Delta x(V)$	$\alpha_e.\Delta x(Hz)$			
Population	Absolute value	0.5	0.6	0.7	6
L1i	-0.132	-.07	-.08	-.09	-.79
L2e	0.380	.19	.23	.27	2.28
L2i	0.410	.21	.25	.29	2.46
L4e	2.210	1.11	1.33	1.55	13.26
L4i	2.099	1.05	1.26	1.47	12.59
L5e	4.747	2.37	2.85	3.32	28.48
L5i	2.503	1.25	1.50	1.75	15.02
L6e	2.662	1.33	1.60	1.86	15.98
L6i	2.268	1.13	1.36	1.59	13.61

Table 24: Estimates of weight received in 1 second per population and of λ increment for some α values in 1 second per population supposing no spike.

Table 25 presents network parameters and structure. The network simulation schematic steps are presented in Algorithm 5. Note that this network has just 8 parameters, 5 of them based on biological data and 3 of them, including the network initialization condition, mathematically calculated due to the lack of biological data.

The populations are the combination of layers and a neuron type (inhibitory/excitatory).

Their features are based on biological data. The number of neurons N per population is the number of excitatory and inhibitory per layer in a column under the somatosensory cortical area. The probability of connection $\mathbb{P}(o_{pre}, o_{post})$ (supplementary Table 30) between a pre-synaptic neuron o_{pre} and a post-synaptic neuron o_{post} is the total number of connections between pre and post populations (Table 19) divided by the product of the numbers of neurons in populations pre (Table 18) and post (Table 18):

$$\mathbb{P}(o_{pre}, o_{post}) = \frac{\text{Total number of connections}_{(pre \to pos)}}{N_{pre} N_{pos}}$$

The number of synapses per connection is the weighted average of the number of synapses that a single pre-synaptic neuron o_{pre} makes on a post-synaptic neuron o_{pos}, once they are connected (i.e. have at least 1 synapse). The synaptic strength is the average voltage change – increase or decrease – in the post-synaptic neuron o_{pos} potential (PSP) due to a single synapse from the pre-synaptic neuron o_{pre}, also calculated by a weighted average (Table 21). All these parameters are biologically based. Therefore, the connection weight $w_{pre,post}$ is given by the number of synapses per connection multiplied by the synaptic strength (supplementary Table 31).

The neuron model is a Poisson point process neuron with a single variable parameter λ. The λ parameter is set to λ_o when the neuron spikes and all post-synaptic neurons connected to this neuron receive an increment $\alpha w_{pre,post}$ if the pre-synaptic neuron is excitatory, or a decrement $-\alpha w_{pre,post}$ if the pre-synaptic neuron is inhibitory.

The λ_o is the reset parameter after a neuron spike, and α is the angular coefficient of the linearization of the membrane potential change (Volt) and probability of spike (Hz). These two parameters are mathematically based.

The network was simulated in discrete time, with time step $\Delta t = 0.1$ ms. The increment $\alpha w_{pre,post}$ in λ leads to an increment of $\alpha w_{pre,post} \Delta t$ in the probability of spike in next time step. The model could also be run in continuous time. The somatosensory model was implemented in Python (with Brian2) and the code can be found on GitHub (`https://github.com/ceciliaromaro/recoup-the-first-and-second-order-statistics-of-neuron-ne neuron-network-dynamics`).

The network simulation was run several times with several different initial conditions $\lambda(t = 0)$ (listed in Table 26). The results are shown in Section 5.3.1.

Network Parameters	Value	Source
Populations:	Table 18	Biological
Number of neurons N per populations:	Table 18	Biological
Probability of connections:	Tables 18 and 19	Biological
Number of synapses per connections:	Table 20	Biological
Synapse strength (PSP):	Table 21	Biological
Angular coefficient α:	$6\ (V/Hz)$	Mathematical
Poisson process reset parameter λ_0:	Table 26	Mathematical
Poisson process parameter initialization $\lambda(t = t_x)$:	Table 26	Mathematical

Table 25: Parameters of the model of the juvenile rat somatosensory cortical column and initial conditions used for the model simulations.

Layer	L1i	L2/3e	L2/3i	L4e	L4i	L5e	L5i	L6e	L6i
Reset parameter: λ_o	-1	-6	-6	-5	-5	-4	-1	-30	-30
Initial condition									
A: $\lambda(t = 0)$	0.0	0.3	0.3	1.4	1.4	2.5	2.5	0.5	0.5
B: $\lambda(t = 0)$	10	10	10	10	10	10	10	10	10
C: $\lambda(t = 0)$	0	0	0	0	0	0	0	0	0
D: $\lambda(t = 2)$	–	–	–	10	–	10	–	–	–
E:	–	–	–	0	–	0	–	–	–

Table 26: After spike reset parameter λ_o and the different initial conditions $\lambda(t = t_x)$ used in simulations.

Algorithm 5 Network interaction algorithm

N number of neurons in network.

$w_{(ij)}$ connection weight from pre-synaptic neuron i to post-synaptic neuron j.

$s_{(i)}$ synapse state: $1 =$ activated; $0 =$ not activated.

$\lambda_{(i)}$ Poisson parameter of neuron i.

$\lambda_{o(i)}$ Reset Poisson parameter of neuron i.

α Angular coefficient (V/Hz).

INITIAL CONDITIONS

for each neuron i in N **do**

    $\lambda_{(i)} \leftarrow \lambda_o(t = t_x)$

    $s_{(i)} \leftarrow 0$

Step 1. NEURON SPIKE

for each neuron i in N **do**

    if $\lambda_{(i)}.\Delta t > rand$ **then**

        $\lambda_{(i)} \leftarrow \lambda_{o(i)}$

        $s_{(i)} \leftarrow 1$

Step 2. SYNAPSE RECEIVE

for each neuron i in N **do**

    if $s_{(i)} == 1$ **then**

        for each neuron j connected with i **do**

            $\lambda_{(j)} \leftarrow \lambda_{(j)} + \alpha * w_{ij}$

    $s_{(i)} \leftarrow 0$

Repeat

5.3.1 Results

The network simulation was run 101 times during 10 seconds and 1 time during 1 hour for the initial condition 'A' in Table 26. The mean fire rates are presented in Table 27. Comparing this table with Table 23, it is possible to see that the model is able to reproduce the mean fire rate with relative deviation of up to 15% of the biological data, for each layer, both for inhibitory and excitatory neurons.

Layer	L1i	L2/3e	L2/3i	L4e	L4i	L5e	L5i	L6e	L6i
Mean firing rates 1 trial 1 hour	0.008 (–)	0.306 (2%)	0.321 (7%)	1.480 (6%)	1.415 (1%)	2.927 (0%)	2.750 (0%)	0.489 (2%)	0.423 (15%)
Mean firing rates 1 trial 5s	0.009 (–)	0.316 (5%)	0.330 (10%)	1.493 (7%)	1.442 (3%)	2.948 (0%)	2.780 (0%)	0.510 (2%)	0.441 (12%)
Mean firing rates of 101 trials 5s	0.009 ± 0.002	0.301 ± 0.002	0.319 ± 0.003	1.474 ± 0.006	1.428 ± 0.010	2.920 ± 0.011	2.757 ± 0.011	0.488 ± 0.002	0.423 ± 0.003

Table 27: Mean firing rates per population. Mean firing rates per population for 1 trial – recorded for 5 seconds and 1 hour and its relative deviation from biological data from Table 23 –, and 101 trials – recorded for 5 seconds and their standard deviations. The network was run for 10 seconds and the last 5 seconds were used for the calculations.

In order to study possible influences of the initial condition and transient state, the network with initial condition 'A' was simulated once for 15 seconds and its statistics were calculated for the first 5 seconds, the middle 5 seconds, and the last 5 seconds separately. The results are presented with the corresponding raster plot in Figure 37.

The calculated statistics are mean firing rate, single unit and population irregularity and population synchrony. The firing rates per neuron and the mean firing rates per population are presented in Figures 37B (first 5 seconds), 37E (middle 5 seconds), and 37H (last 5 seconds). The irregularity per neuron was calculated as the coefficient of variation of the interspike interval (ISI) for a single unit and is presented in Figures 37C, 37F and 37I. The population averages of the irregularity are also presented in the same figures. Synchrony per population, calculated as the variance of the spike histogram divided by its mean, is presented in Figures 37D (first 5 seconds), 37G (middle 5 seconds), and 37J (last 5 seconds). In order to avoid bias in the average firing rate per population, the bin was calculated as 10 ms divided by the population average fire rate.

It is possible to see synchrony in layers L2 and L6 in the first seconds of the raster plot (Figure 37), which tends to vanish over time. This visual observation is confirmed by the decay of the synchrony measure over time, by comparing Figures 37D, 37G and 37J. Notice the change of scale in the x-axis of Figure 37D in comparison with the scales of Figures 37G and 37H. For example, the variance of the spike histogram of layer L6 is higher than 100% of the mean only during the first 5 seconds; after that this variance becomes smaller than 50% of the mean. The variance of the spike histogram of layer L2/3 is higher than 80% of the mean only during the first 5 seconds; after that the variance drops below 40% of the mean.

It is also possible to visually see the delays in the avalanche starting points between layer L6 excitatory and inhibitory neurons (Figure 37A), even when the network and neuron delay parameters are absent.

Additionally, both the mean firing rate and the irregularity measure remain constant over time. These are observable in Figures 37B, 37E and 37H for the mean firing rate, and in Figures 37C, 37D and 37E for the irregularity measure.

The mean of irregularity remains constant over time, but each single neuron may exhibit irregularity above or below the mean irregularity inside the same time period. Observing the irregularity per neuron (black vertical bars) in Figures 37F and 37I, one can see that, for example, for L5i, the standard deviation of the ISI for a single neuron can be below 20% or above 80% of its mean.

A similar behavior can be seen for the mean firing rates per neuron (black Xs) in Figures 37G and 37J. Although the population means remain constant over time, each single neuron may exhibit a mean firing rate up to 150% higher than the layer's mean or up to 100% lower than the layer's mean firing rate (see supplementary Table 32).

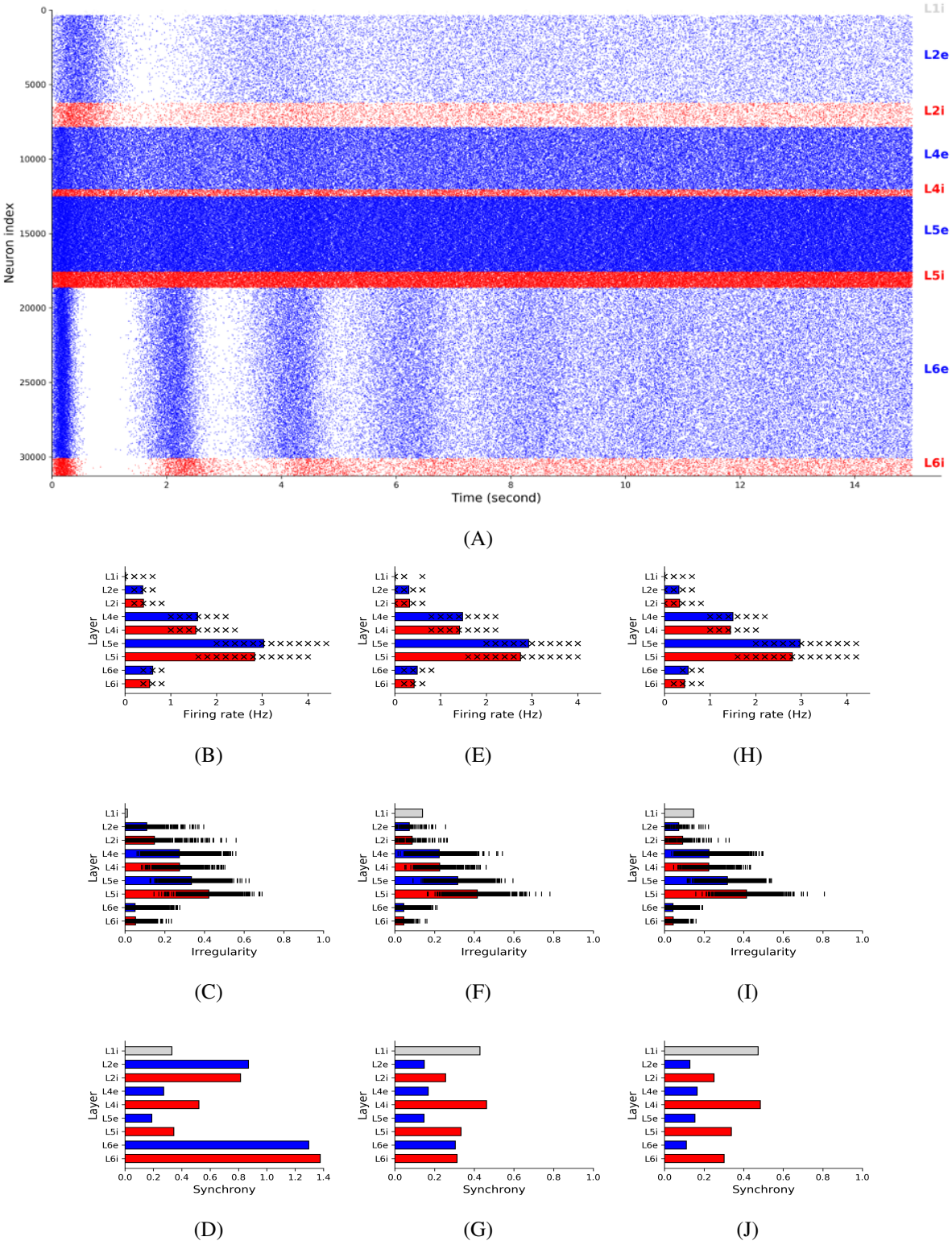
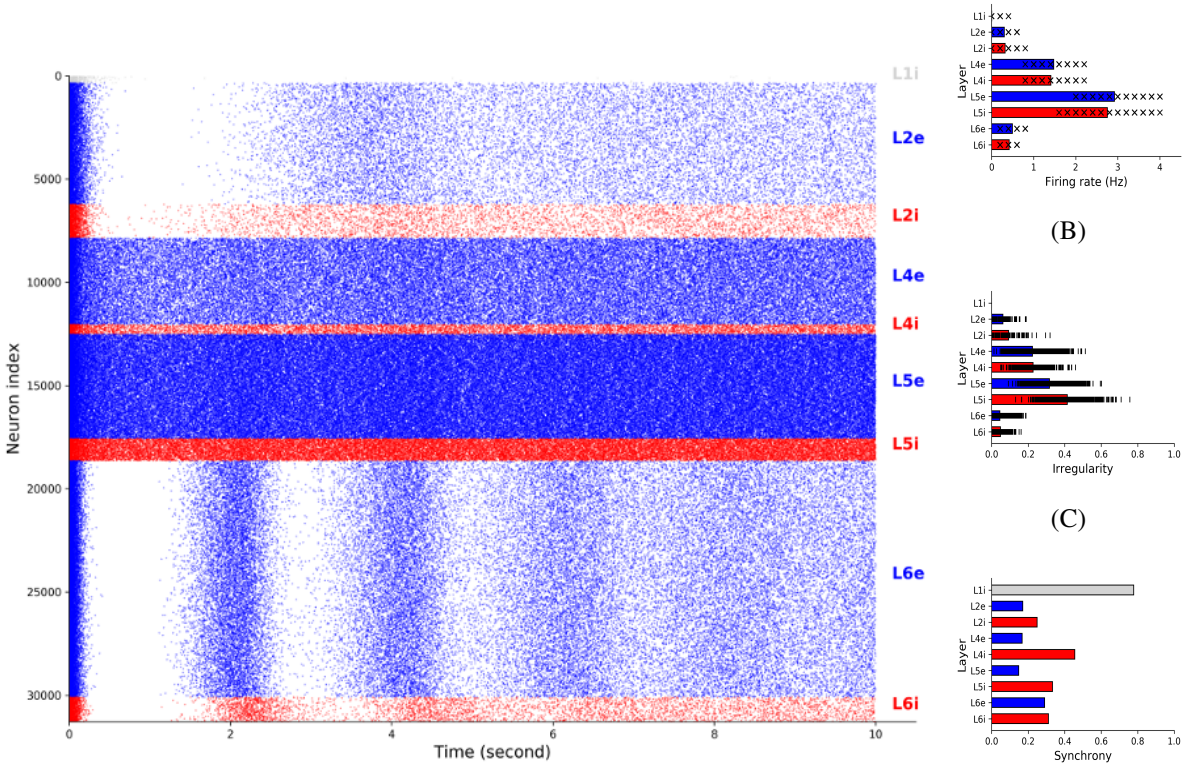


Figure 37: Raster plot and statistics for the 15-second simulation run. (A) Raster plot of one trial ran for 15 seconds. Each line represents the spikes of a neuron (indicated by dots). Spikes of excitatory neurons are represented in blue and pikes of inhibitory neurons are represented in red or gray (the latter for L1i neurons). (B-D) Statistics for the first 5 seconds run. (B) Mean firing rate per neuron (black X) and per population (bars) for the first 5 seconds of the raster plot. (C) Irregularity per neuron (CV of the ISI for a single unit spikes) represented by black vertical lines and mean irregularity calculated over the population presented by bars. (D) Synchrony per population (variance of spike histogram divided by its mean) with bins of 10 ms divided by population mean firing rate. (E-G) Same as (B-D) for the middle 5 seconds. (H-J) same as (B-D) for last 5 seconds.

Figure 38A presents the raster plot for a 10-second simulation run of the network with 'B' initial conditions. Figures 38B-38D present the statistics of the raster plot for the last 5 seconds of simulation. Comparing Figures 38B-38D with Figures 37E-37G and 37H-37J, it is not possible to see any apparent difference in network behavior among them: mean firing rate, irregularity, and synchrony per population, all appear to be similar to each other independently of the initial condition used, as long as it is sufficient to activate the network.



(A) (B) (C) (D)

Figure 38: Raster plot for the 10-second simulation for initial condition 'B', and raster plot statistics for the last 5 seconds of simulation. (A) Raster plot of one simulation trial for 10 seconds. Each line represents the spikes of a neuron (indicated by dots). Spikes of excitatory neurons are represented in blue and spikes of inhibitory neurons are represented in red or gray (the latter for L1i neurons) (B-D) Statistics for the last 5 seconds of simulation. (B) Mean firing rate per neuron (black X) and per population (bars). (C) Irregularity per neuron (CV of the ISI for a single unit spikes) represented by black vertical lines and mean irregularity calculated over the population presented by bars. (D) Synchrony per population (variance of spike histogram divided by its mean) with bins of 10 ms divided by population mean firing rate.

Figure 39A presents the raster plot for a 12-second simulation run of the network with 'C' initial condition followed by a thalamic input stimulation at time $t = 2$ seconds (instant change of the λ parameter of L4e and L5e neurons to 10) and an additional 10-second simulation. Notice that after the neuron spikes, λ is reset to λ_o below zero. One can see that the network remains inactive for the initial condition 'C' of $\lambda = 0$ – zero probability of spike for each neuron – until some forced change in the spike probabilities of L4e and L5e neurons occur (at

time $t = 2$ seconds). Otherwise, the network would remain inactive indefinitely. One can also observe that the one-time change in the λ parameter of L4e and L5e neurons to 10 was enough to put the network in an active state, even with the subsequent return of the spike-rate parameter λ of these neurons to a value below zero λ_o.

Figures 39B-39D present the statistics of the raster plot during the last 5 seconds of the simulation. Comparing Figures 39B-39D to Figures 38B-38D, Figures 37E-37G and Figures 37H-37J, it is not possible to see any substantial difference in the network behavior: the mean firing rate, irregularity, and synchrony per population appear to be similar. This can also be confirmed by the supplementary information presented in Table 32, which presents the average values of data from Figures 37 to 39.

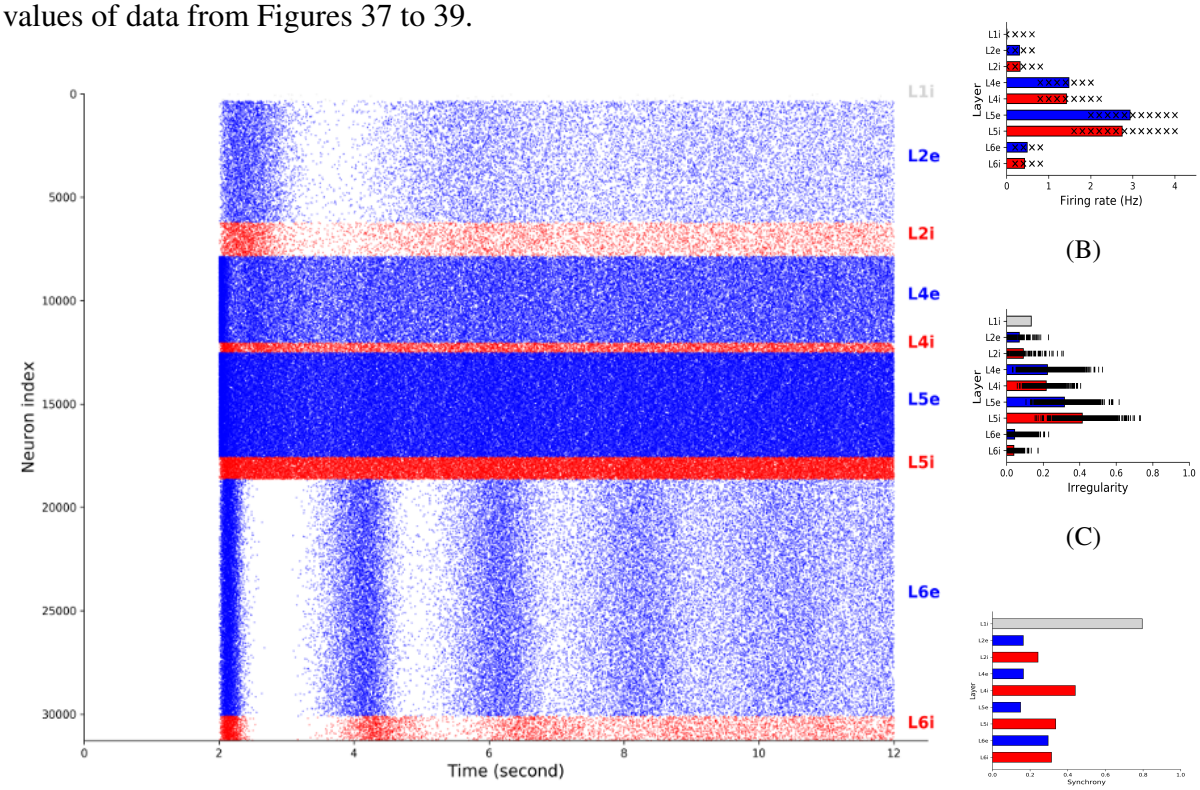
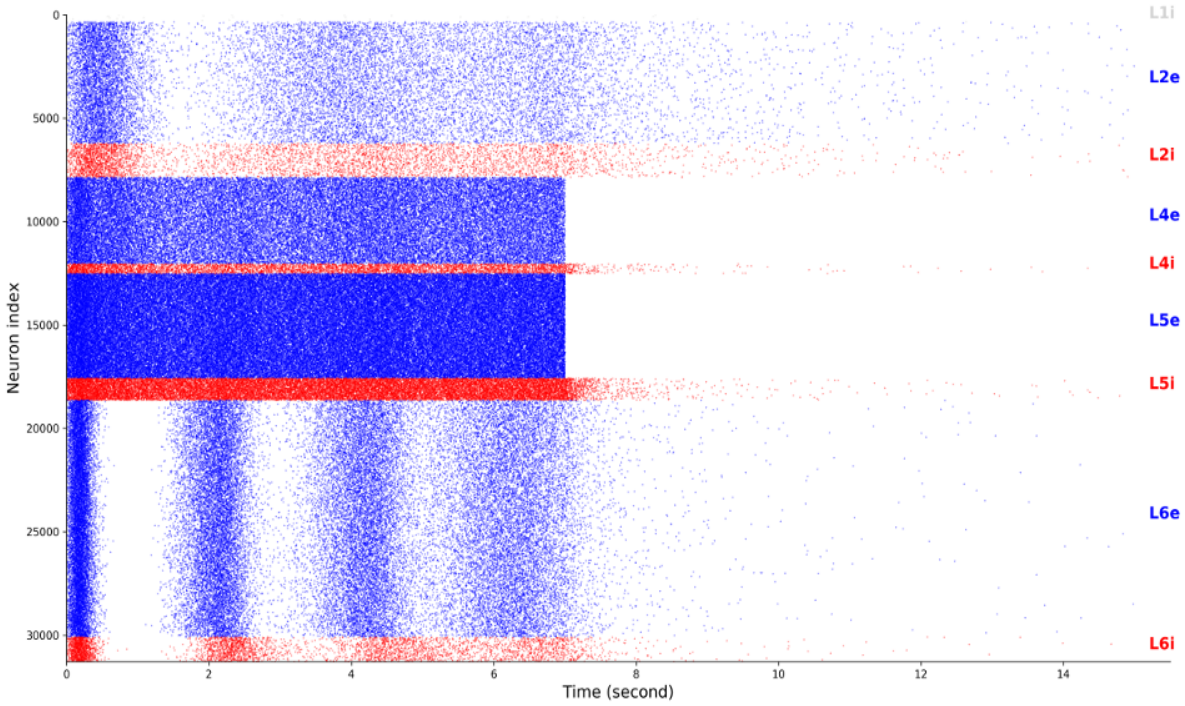


Figure 39: Raster plot of the 12-second simulation for initial condition 'C', and a thalamic input stimulation at time $t = 2$ seconds (instant change of the λ parameter of L4e and L5e neurons to 10), and the statistics of the raster plot for the last 5 seconds of simulation. (A) Raster plot of one 12-second simulation trial. Each line represents the spikes of a neuron (indicated by dots). Spikes of excitatory neurons are represented in blue and spikes of inhibitory neurons are represented in red or gray (the latter for L1i neurons). (B-D) Statistics for the last 5 seconds of simulation. (B) Mean firing rate per neuron (black X) and per population (bars). (C) Irregularity per neuron (CV of the ISI for a single unit spikes) represented by black vertical lines and mean irregularity calculated over the population presented by bars. (D) Synchrony per population (variance of spike histogram divided by its mean) with bins of 10 ms divided by population mean firing rate.

Figure 40 presents the raster plot of a 15-second simulation run with 'A' initial conditions followed by an inhibitory thalamic input stimulation at time $t = 7$ seconds (instant change of the λ parameter of L4e and L5e neurons to -10) and an additional 8 second simulation. Once the neuronal spikes happen, the λ parameter is reset to the default λ_o below zero. One can see that this one-time change in L4e and L5e, λ was enough to change the network equilibrium point from an active state to a near zero activity (quiescent) state.



(A)

Figure 40: Raster plot of the 15-second simulation run for 'A' initial conditions and an inhibitory thalamic input stimulation at time $t = 7$ seconds (instant change of the λ parameter of L4e and L5e neurons to -10). (A) Raster plot of one 15-second simulation trial. Each line represents the spikes of a neuron (indicated by dots). Spikes of excitatory neurons are represented in blue and spikes of inhibitory neurons are represented in red or gray (the latter for L1i neurons)

5.3.2 Discussion

This somatosensory cortex model (Figures 37 - 39) was able to reproduce the experimentally observed mean firing rate per layer for the somatosensory cortex of the rat (de Kock and Sakmann, 2009; Sakata and Harris, 2009; Swadlow, 1989). This network also presents at least two equilibrium points: an active state, which reproduces the rat's somatosensory cortex mean firing rate per layer; and a quiescent state, in which the network activity is close to zero. The existence of other equilibrium points is possible.

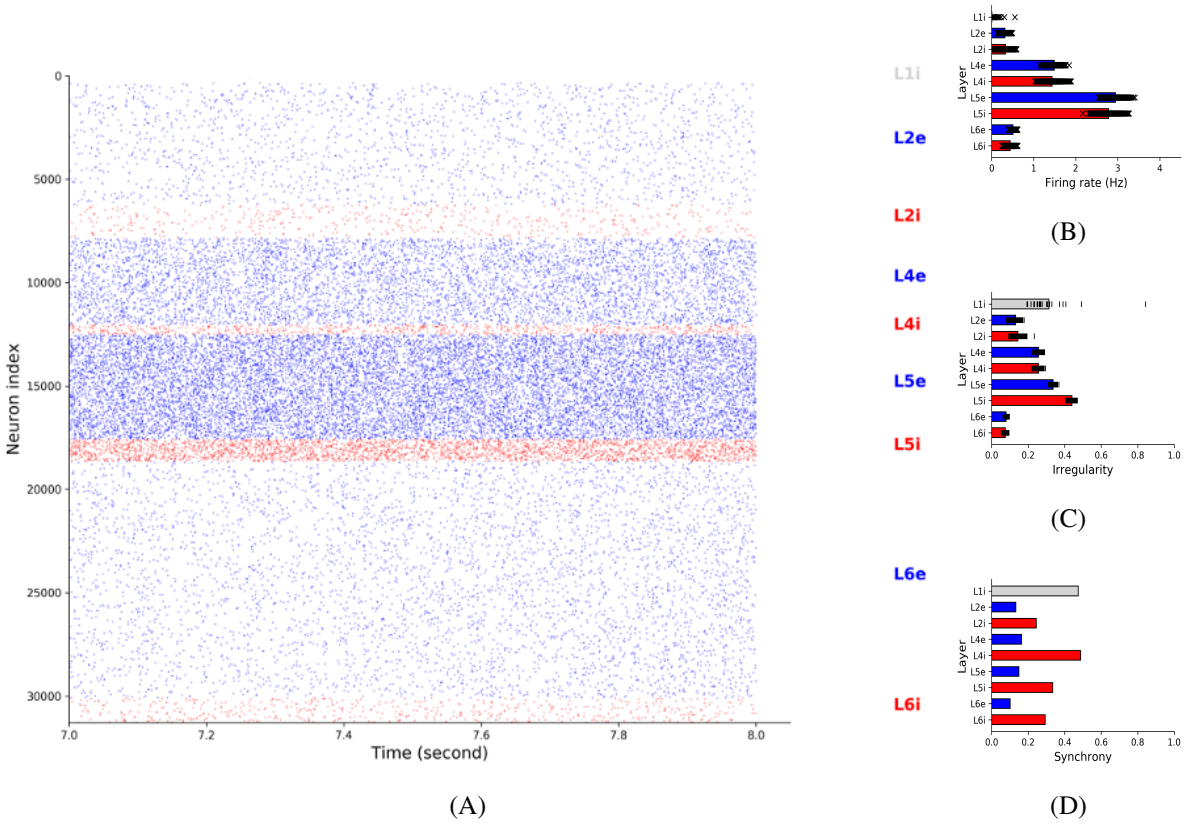
The switch between these two states was possible due to the perturbation in L4e and L5e neurons caused by the thalamic input.

It is possible that the active state of the network presents a metastable behavior (a phase transitions to the quiescent state (Romaro et al., 2019), once the network satisfies the condition of absence of external input). However, during a 1 hour simulation (Figure 41 and Table 27) the network was able to self-sustain its active state.

The active state in the cortical column network could correspond to the activation of the representation of a given body surface region in S1, which could be selected depending on the thalamic input. The network is a model of of a single column of the rat somatosensory cortex. Thus, turning "on" or "off" the column could be a mechanism to identify the stimulation or not of a body surface area or whisker.

The network presents the desired asynchrony property observed in the awake state (Goldman et al., 2019). The initial synchrony, visually perceived in layers L2/3 and L6, tends to vanish (Figures 37 to 40) as seen in the synchrony measures (Figures 37D, 37G and 37J). This may be an influence of the stochastic neuron model used. However, even for a network based on a stochastic neuron model there are combinations of connection patterns and connection strengths that could lead to synchronous activity states in the network.

A positive aspect of the model is the relatively small number of parameters required by it in comparison to networks based on the LIF (Lapicque, 1907) model. The Potjans-Diesmann (PD) network model, which uses the LIF neuron model (Potjans and Diesmann, 2014a), requires the adoption of many non-biologically-based parameters, such as membrane time constant, absolute refractory period, post-synaptic current time constant, membrane capacitance, inhibitory synaptic transition delay, excitatory synaptic transition delay and its standard deviation. The standard deviation could be playing a role – yet lacking complete explanation – in the asynchronous states observed in LIF-based networks (Potjans and Diesmann, 2014a; Brunel, 2000). None of these parameters are mandatory in GL-based (Galves and Löcherbach, 2013) network, although they could be adopted if desired.



(A) (B) (C) (D)

Figure 41: Statistics of a 1-hour simulation for initial condition 'A'. (A) Raster plot of one simulation trial for 1 second. Each line represents the spikes of a neuron (indicated by dots). Spikes of excitatory neurons are represented in blue and spikes of inhibitory neurons are represented in red or gray (the latter for L1i neurons) (B-D) Statistics for the full 1-hour simulation run. (B) Mean firing rate per neuron (black X) and per population (bars). (C) Irregularity per neuron (CV of the ISI for a single unit spikes) represented by black vertical lines and mean irregularity calculated over the population presented by bars. (D) Synchrony per population (variance of spike histogram divided by its mean) with bins of 10 ms divided by population mean firing rate.

It is impossible to create a cortex model that reproduces the average firing rate per layer and not make at least a rough comparison with the classic PD model (Potjans and Diesmann, 2014a) cortex model. The PD model, which first reproduced the average firing rate per cortical layer, has around 80 thousand LIF neurons representing the neuronal network under 1 mm^2 of sensory cortex area. Our somatosensory model has around 31 thousand GL neurons representing the neuronal network under 0.31 mm^2 of somatosensory cortex area. The PD network is essentially an inhibitory network with excitatory external input. In other words, nullifying the connections between neurons inside the network, keeping external input, the neuronal activity soars to 155 Hz. Our somatosensory network has absence of external input, and therefore is essentially an excitatory network. Due to the neuron model used in each network, our model requires just 2 non biologically- based parameters, unlike the PD network which requires all the non biologically-based parameters mentioned above.

Our somatosensory network model has the advantage of presenting less synchrony than

the PD model (Potjans and Diesmann, 2014a), but it has the disadvantage of presenting less irregularity. Irregularity in the spikes of a single neuron as well as asynchrony in the population activity are desired features in a model that attempts to model the awake state of the brain (Goldman et al., 2019).

An observation on the synchrony measure used in this work in comparison to the one used in the PD model (Potjans and Diesmann, 2014a) is worth being made. The PD model uses the variance of the spike count histogram with a fixed bin of 3 ms divided by the mean, which leads to a measure dependent on the number of neurons and the mean firing-rate. Our work uses a bin size equal to 10 ms divided by the average firing rate of the layer, thus reducing such bias. Given that, the synchrony of L5e was calculated with the same bin size for both networks, and our model presented a lower synchrony value (Figures 37 to 41) and did not display the visually perceived synchrony that can be seen in the PD model for any layer. In contrast, the PD model presented higher irregularity than our somatosensory cortex model for all layers.

The irregularity is calculated by the ISI standard deviation divided its mean, as in the PD model. Once the ISI is pushed to a Gaussian distribution, due to the linearization process of Equation 5.3.1 in the stochastic neuron, a low value for the ISI standard deviation is expected.

The irregularity of our model could be increased by changing the linear function 5.3.1 to a sigmoidal, for example. Or by changing the linear function 5.3.1 to any better approximation for the conversion function from membrane potential to probability of spike.

Introducing a random component in the constant reset value λ_o will also lead to an increase in the irregularity measurement (see supplementary Figure 43), despite the author opposition to this method. This ad hoc way of introducing irregularity in the network would be as questionable as to adapt many LIF neuron parameters without biological base. We would end up without knowing which property was caused from biological modeling and which was caused by the ad hoc measure.

The average irregularity of neurons per layer is always below 0.5. Even so, inside a single layer it is possible to find neurons with irregularities higher than 0.8 and neurons with irregularities lower than 0.2 (Layer L5i in Figure 37I and supplementary Table 32). Interestingly, inside one single layer, and using the same neuron model and same kind of connections, it is possible to find neurons with regular behavior and neurons with irregular behavior, meaning that the regularity or irregularity of these neurons is caused exclusively by the pattern of connections – the connectome.

5.4 Future Works

This work raised some mathematical questions not formally solved, such as a Poisson point process biasing the parameter of another Poisson point process (see Section 5.3), and the possibility of equating the first and second order statistics of LIF and GL models. Just to show the possibility of such agreement, Figure 42 presents results for a model with the graph of connections of the PD network with the original LIF neurons replaced by GL neurons. This implementation used a linear function 5.3.1 (parameters in Table 28) able to keep first order statistics and the L5 activity death with unbalanced input. This implementation was able to reproduced these features, namely L5 layer activity death and firing rate, presented in the original PD model (Potjans and Diesmann, 2014a) with the LIF neuron. The Poisson external input has remained unchanged but it is possible to incorporate it in the GL neuron model.

The solutions to these questions will be let for future works.

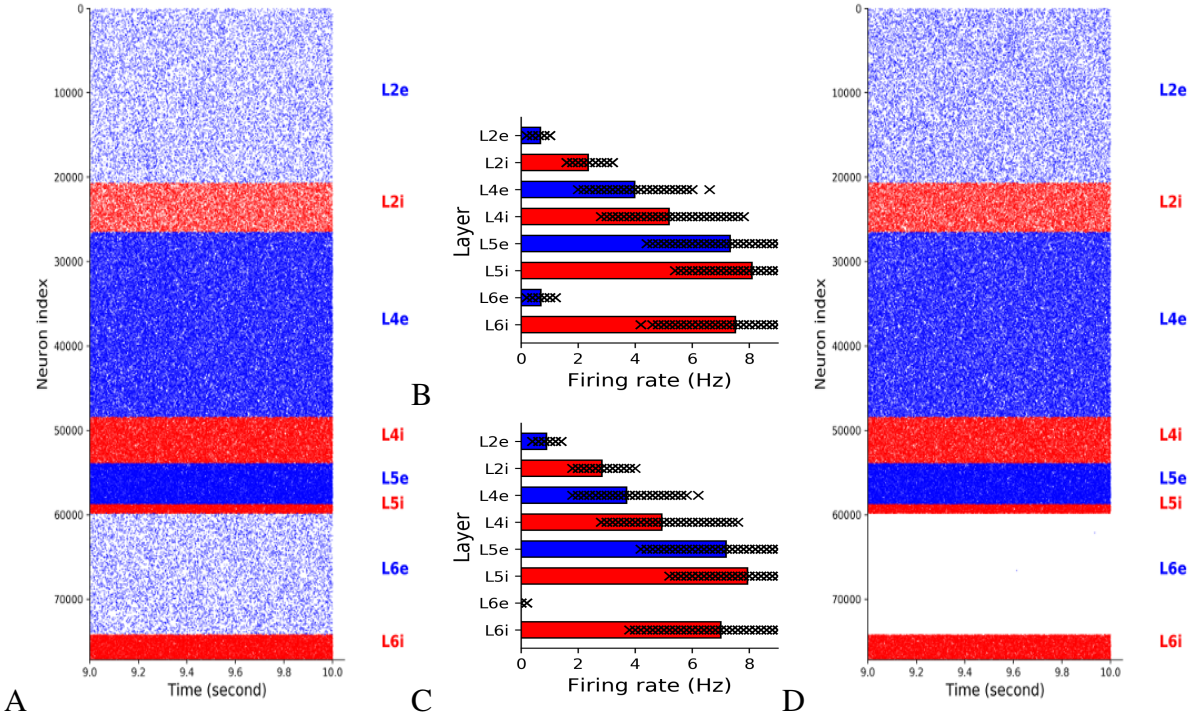


Figure 42: Raster plot of the 10-second simulation run for the model with PD graph and GL neurons, and the average layer-specific firing rates for the last 5 seconds run. (A-B) Balanced input. (C-D) Unbalanced input. (A) Raster plot of one 10-second simulation trial. Each line represents the spikes of a neuron (indicated by dots). Spikes of excitatory neurons are represented in blue and spikes of inhibitory neurons are represented in red. (B-D) Statistics for the last 5 seconds of simulation. (B) Average firing rate per neuron (black x) and per population (bars). (D-C) same as (B-D) for unbalanced input.

Angular Coefficient: α (Hz/nA) = 4								
Layer	L2/3e	L2/3i	L4e	L4i	L5e	L5i	L6e	L6i
Reset parameter: λ_o	-21	-4	0	2	4	5	-22	5

Table 28: After spike reset parameter λ_o and angular coefficient α. The relative inhibitory synaptic strength g (w_-/w_+ or $\alpha g/\alpha$) remains equal to -4.

5.5 Conclusion

The somatosensory cortex model was constructed with 7 parameters plus the initial condition. Five of these parameters are biologically based, and just 2 of them are mathematically adjusted. The model was able to reproduce the experimentally measured average firing rates per layer, a self-sustained activity without external input, and two balance states and the shift between them by an one-step interference in Layers L4e and L5e. The network also presented asynchronous activity and neurons with regular and irregular behaviour inside the same layer.

A comparison study of LIF (Lapicque, 1907) neuron networks and GL (Galves and Löcherbach, 2013) neuron networks using the networks with the same connectivity graph was not the purpose of this work. However, this work definitely aroused an interest in such comparison, which will be left for future works.

Acknowledgments

This work was produced as part of the activities of FAPESP Research, Disseminations and Innovation Center for Neuromathematics (Grant 2013/07699-0, S. Paulo Research Foundation). The author is the recipient of PhD scholarships from the Brazilian Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES). The author is thankful to Cinthia Romaro, author's little sister who found time to comment on the manuscript even during her Harvard MBA.

References

Brunel, N. (2000). Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. *Journal of Computational Neuroscience*, 8(3):183–208.

Cordeiro, V. L. (2019). Modelos de redes neurais com neurônios estocásticos e diferentes topologias: construção e análise. Master's thesis, FFCLRP-Universidade de São Paulo, Ribeirão Preto.

de Kock, C. P. and Sakmann, B. (2009). Spiking in primary somatosensory cortex during natural whisking in awake head-restrained rats is cell-type specific. *Proceedings of the National Academy of Sciences (USA)*, 106(38):16446–16450.

Galves, A. and Löcherbach, E. (2013). Infinite systems of interacting chains with memory of variable length—a stochastic model for biological neural nets. *Journal of Statistical Physics*, 151(5):896–921.

Goldman, J. S., Tort-Colet, N., di Volo, M., Susin, E., Bouté, J., Dali, M., Carlu, M., Nghiem, T.-A., Górski, T., and Destexhe, A. (2019). Bridging single neuron dynamics to global brain states. *Frontiers in Systems Neuroscience*, 13:75.

Hodgkin, A. L. and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500–544.

Kriener, B., Helias, M., Rotter, S., Diesmann, M., and Einevoll, G. T. (2014). How pattern formation in ring networks of excitatory and inhibitory spiking neurons depends on the input current regime. *Frontiers in computational neuroscience*, 7:187.

Lapicque, L. (1907). Recherches quantitatives sur l'excitation electrique des nerfs traitee comme une polarization. *Journal de Physiologie et de Pathologie Generale*, 9:620–635.

Markram, H., Muller, E., Ramaswamy, S., Reimann, M. W., Abdellah, M., Sanchez, C. A., Ailamaki, A., Alonso-Nanclares, L., Antille, N., Arsever, S., et al. (2015). Reconstruction and simulation of neocortical microcircuitry. *Cell*, 163(2):456–492.

Millman, D., Mihalas, S., Kirkwood, A., and Niebur, E. (2010). Self-organized criticality occurs in non-conservative neuronal networks during 'up' states. *Nature)hysics*, 6(10):801.

Potjans, T. C. and Diesmann, M. (2014a). The cell-type specific cortical microcircuit: relating structure and activity in a full-scale spiking network model. *Cerebral Cortex*, 24(3):785–806.

Ramaswamy, S., Courcol, J.-D., Abdellah, M., Adaszewski, S. R., Antille, N., Arsever, S., Atenekeng, G., Bilgili, A., Brukau, Y., Chalimourda, A., et al. (2015). The neocortical microcircuit collaboration portal: a resource for rat somatosensory cortex. *Frontiers in Neural Circuits*, 9:44.

Romaro, C., Najman, F. A., and André, M. (2019). A numerical study of the time of extinction in a class of systems of spiking neurons. *arXiv:1911.02609*.

Tsodyks, M., Pawelzik, K., and Markram, H. (1998). Neural networks with dynamic synapses. *Neural Computation*, 10(4):821–835.

Sakata, S. and Harris, K. D. (2009). Laminar structure of spontaneous and sensory-evoked population activity in auditory cortex. *Neuron*, 64(3):404–418.

Swadlow, H. A. (1989). Efferent neurons and suspected interneurons in s-1 vibrissa cortex of the awake rabbit: receptive fields and axonal properties. *Journal of Neurophysiology*, 62(1):288–308.

Supplementary Material

		From								
		L1i	L2/3e	L2/3i	L4e	L4i	L5e	L5i	L6e	L6i
To	L1i	4.64	7.73	10.16	0.28	1.05	0.66	1.97	0.04	0.50
	L2e	4.57	94.93	20.30	25.58	3.63	13.66	5.50	1.76	0.00
	L2i	2.76	41.66	18.78	10.62	2.76	5.84	4.61	1.08	0.61
	L4e	1.63	161.29	16.44	131.26	10.09	86.20	12.02	36.45	3.72
	L4i	0.13	43.85	7.63	53.80	8.01	30.89	9.25	13.45	2.85
	L5e	6.52	204.29	27.49	175.55	9.21	231.92	24.92	158.65	9.33
	L5i	0.03	31.03	3.01	32.41	3.04	78.11	6.91	27.50	5.24
	L6e	0.05	45.75	2.49	57.32	2.65	114.13	5.63	277.32	13.37
	L6i	0.00	6.41	0.33	8.73	0.38	28.44	0.74	83.27	9.90

Table 29: Average number of connections from a population to a single neuron. (colocar unidade)

		From								
		L1i	L2/3e	L2/3i	L4e	L4i	L5e	L5i	L6e	L6i
To	L1i	0.0137	0.0013	0.0062	0.0001	0.0022	0.0001	0.0019	0.0000	0.0004
	L2e	0.0135	0.0162	0.0123	0.0061	0.0076	0.0027	0.0052	0.0002	0.0000
	L2i	0.0082	0.0071	0.0114	0.0025	0.0058	0.0012	0.0043	0.0001	0.0005
	L4e	0.0048	0.0274	0.0100	0.0314	0.0211	0.0171	0.0113	0.0032	0.0031
	L4i	0.0004	0.0075	0.0046	0.0129	0.0168	0.0061	0.0087	0.0012	0.0024
	L5e	0.0193	0.0348	0.0167	0.0420	0.0193	0.0459	0.0234	0.0138	0.0078
	L5i	0.0001	0.0053	0.0018	0.0078	0.0064	0.0155	0.0065	0.0024	0.0044
	L6e	0.0001	0.0078	0.0015	0.0137	0.0056	0.0226	0.0053	0.0242	0.0112
	L6i	0.0000	0.0011	0.0002	0.0021	0.0008	0.0056	0.0007	0.0073	0.0083

Table 30: The probability of connections per pair of population

		From								
		L1i	L2/3e	L2/3i	L4e	L4i	L5e	L5i	L6e	L6i
To	L1i	38.72	24.59	21.83	5.01	21.71	14.33	22.92	13.10	14.60
	L2e	40.26	5.32	7.91	5.35	6.21	5.21	4.63	3.30	0.40
	L2i	13.12	18.26	17.68	18.89	14.87	12.82	11.96	20.75	7.95
	L4e	8.23	4.07	4.14	7.83	7.62	3.75	5.47	3.58	3.69
	L4i	3.56	17.80	10.46	15.95	18.24	14.67	13.38	12.01	8.56
	L5e	8.17	1.85	1.44	2.75	2.88	7.03	3.19	1.85	3.02
	L5i	1.89	7.44	7.11	12.08	11.35	10.74	13.20	7.30	11.15
	L6e	0.76	2.78	1.12	2.57	1.27	5.91	2.30	5.86	6.26
	L6i	0.00	10.14	5.65	11.00	5.71	20.03	6.52	18.49	12.77

Table 31: Average weight of connection per pair of populations. (colocar unidade)

Firing rate

Figure37:	L1i	L2/3e	L2/3i	L4e	L4i	L5e	L5i	L6e	L6i
First 5 seconds: mean (bar)	0.015	0.391	0.405	1.589	1.550	3.030	2.836	0.603	0.538
max (black x)	0.600 (3956%)	0.600 (54%)	0.800 (97%)	2.200 (38%)	2.400 (55%)	4.400 (45%)	4.000 (41%)	0.800 (33%)	0.800 (49 %)
min (black x)	0.000 (100%)	0.200 (49%)	0.200 (51%)	1.000 (37%)	1.000 (35%)	2.000 (34%)	1.600 (44%)	0.400 (34%)	0.400 (26 %)
Middle 5 seconds: mean (bar)	0.008	0.306	0.321	1.480	1.415	2.927	2.750	0.489	0.423
max (black x)	0.600 (7143%)	0.600 (96%)	0.600 (87%)	2.200 (49%)	2.200 (55%)	4.000 (37%)	4.000 (45%)	0.800 (64%)	0.600 (42 %)
min (black x)	0.000 (100%)	0.000 (100%)	0.000 (100%)	0.800 (46%)	0.800 (43%)	2.000 (32%)	1.600 (42%)	0.200 (59%)	0.200 (53 %)
Last 5 seconds: mean (bar)	0.009	0.315	0.329	1.500	1.448	2.973	2.806	0.518	0.442
max (black x)	0.600 (6660%)	0.600 (91%)	0.800 (143%)	2.200 (47%)	2.000 (38%)	4.200 (41%)	4.200 (50%)	0.800 (55%)	0.800 (81 %)
min (black x)	0.000 (100%)	0.000 (100%)	0.000 (100%)	1.000 (33%)	1.000 (31%)	2.000 (33%)	1.600 (43%)	0.400 (23%)	0.200 (55 %)
Figure38: mean (bar)	0.005	0.303	0.320	1.475	1.409	2.923	2.751	0.493	0.420
max (black x)	0.400 (7411%)	0.600 (98%)	0.800 (150%)	2.200 (49%)	2.200 (56%)	4.000 (37%)	4.000 (45%)	0.800 (62%)	0.600 (43 %)
min (black x)	0.000 (100%)	0.000 (100%)	0.000 (100%)	0.800 (46%)	0.800 (43%)	2.000 (32%)	1.600 (42%)	0.200 (59%)	0.200 (52 %)
Figure39: mean (bar)	0.009	0.305	0.324	1.481	1.426	2.934	2.748	0.492	0.425
max (black x)	0.600 (6238%)	0.600 (97%)	0.800 (147%)	2.000 (35%)	2.200 (54%)	4.000 (36%)	4.000 (46%)	0.800 (63%)	0.800 (88 %)
min (black x)	0.000 (100%)	0.000 (100%)	0.000 (100%)	0.800 (46%)	0.800 (44%)	2.000 (32%)	1.600 (42%)	0.200 (59%)	0.200 (53 %)

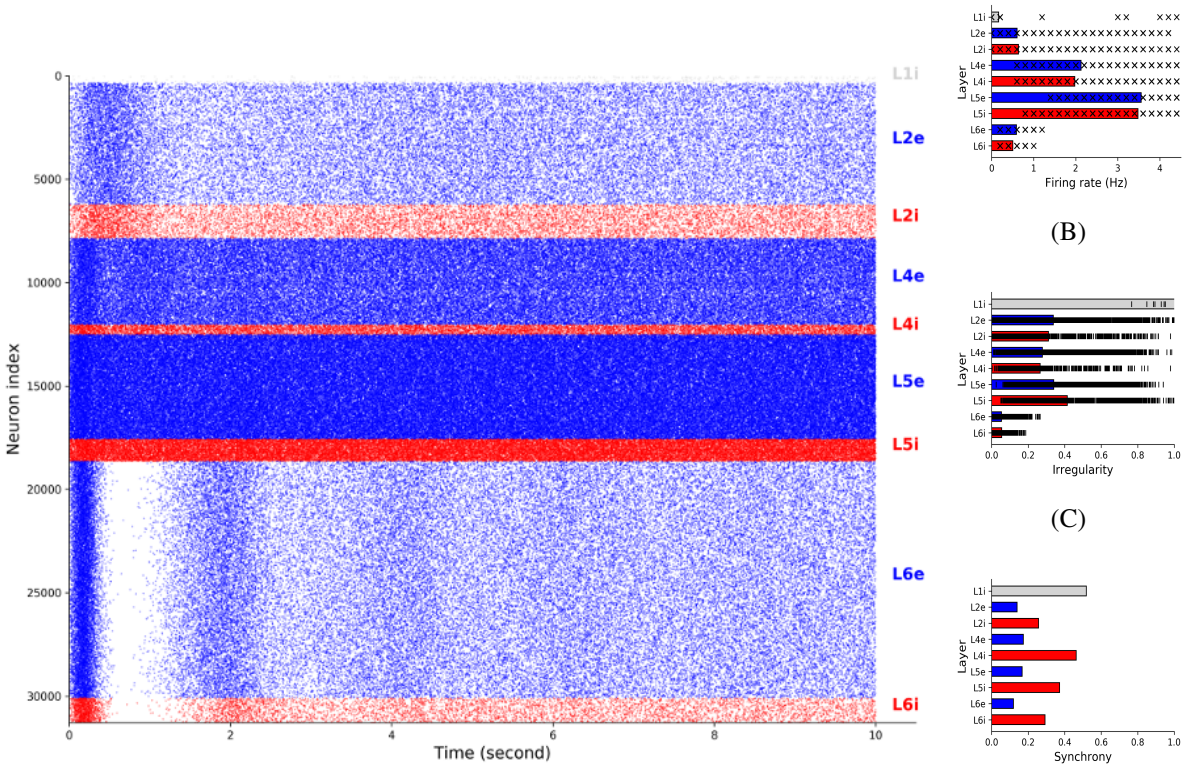
Irregularity

Figure37:	L1i	L2/3e	L2/3i	L4e	L4i	L5e	L5i	L6e	L6i
First 5 seconds: mean (bar)	0.012	0.109	0.148	0.274	0.275	0.335	0.422	0.051	0.053
max (vertical scratches)	0.012 (0%)	0.397 (263%)	0.559 (277%)	0.558 (104%)	0.503 (83%)	0.625 (87%)	0.690 (63%)	0.276 (443%)	0.234 (340 %)
min (vertical scratches)	0.012 (0%)	0.000 (100%)	0.003 (98%)	0.061 (78%)	0.083 (70%)	0.126 (62%)	0.146 (65%)	0.000 (100%)	0.000 (100 %)
Middle 5 seconds: mean (bar)	0.140	0.073	0.087	0.224	0.226	0.317	0.416	0.045	0.045
max (vertical scratches)	0.140 (0%)	0.256 (253%)	0.265 (206%)	0.542 (142%)	0.459 (103%)	0.595 (88%)	0.781 (88%)	0.213 (371%)	0.161 (257 %)
min (vertical scratches)	0.140 (0%)	0.002 (98%)	0.001 (99%)	0.016 (93%)	0.051 (77%)	0.092 (71%)	0.163 (61%)	0.000 (100%)	0.000 (100 %)
Last 5 seconds: mean (bar)	0.147	0.072	0.092	0.225	0.224	0.318	0.415	0.043	0.044
max (vertical scratches)	0.147 (0%)	0.222 (208%)	0.327 (257%)	0.498 (121%)	0.435 (94%)	0.541 (70%)	0.808 (95%)	0.193 (344%)	0.159 (258 %)
min (vertical scratches)	0.147 (0%)	0.000 (99%)	0.002 (98%)	0.037 (84%)	0.047 (79%)	0.115 (64%)	0.159 (62%)	0.000 (100%)	0.001 (99 %)
Figure38: :mean (bar)	0.000	0.062	0.092	0.224	0.227	0.317	0.415	0.045	0.048
max (vertical scratches)	0.000 (–)	0.190 (208%)	0.321 (247%)	0.513 (129%)	0.459 (103%)	0.602 (90%)	0.757 (82%)	0.188 (318%)	0.161 (236 %)
min (vertical scratches)	1.000 (–)	0.000 (100%)	0.003 (97%)	0.013 (94%)	0.050 (78%)	0.092 (71%)	0.132 (68%)	0.000 (100%)	0.000 (100 %)
Figure39: mean (bar)	0.135	0.070	0.092	0.224	0.217	0.317	0.415	0.045	0.039
max (vertical scratches)	0.135 (0%)	0.229 (228%)	0.310 (237%)	0.525 (135%)	0.405 (87%)	0.617 (94%)	0.732 (76%)	0.230 (416%)	0.172 (338 %)
min (vertical scratches)	0.135 (0%)	0.000 (100%)	0.001 (99%)	0.032 (86%)	0.059 (73%)	0.106 (67%)	0.156 (62%)	0.000 (100%)	0.000 (100 %)

Synchrony

Figure37:	L1i	L2/3e	L2/3i	L4e	L4i	L5e	L5i	L6e	L6i
First 5 seconds	0.330	0.870	0.815	0.274	0.521	0.189	0.344	1.296	1.377
Middle 5 seconds	0.429	0.148	0.256	0.168	0.462	0.147	0.334	0.305	0.314
Last 5 seconds	0.474	0.128	0.251	0.164	0.484	0.154	0.338	0.110	0.302
Figure38:	0.778	0.170	0.249	0.168	0.455	0.149	0.333	0.291	0.312
Figure39:	0.796	0.164	0.242	0.165	0.440	0.150	0.336	0.296	0.314

Table 32: Average values of data from Figures 37, 38 and 39.



(A) (B) (C) (D)

Figure 43: Raster plot of the 10-second simulation for initial condition 'A' with a Gaussian white noise of $-20 * (rand - 0.4)$ added to the parameter reset λ_o, and the statistics of the raster plot for the last 5 seconds of simulation. (A) Raster plot of one 10-second simulation trial. Each line represents the spikes of a neuron (indicated by dots). Spikes of excitatory neurons are represented in blue and spikes of inhibitory neurons are represented in red or gray (the latter for L1i neurons). (B-D) Statistics for the last 5 seconds of simulation. (B) Mean firing rate per neuron (black X) and per population (bars). (C) Irregularity per neuron (CV of the ISI for a single unit spikes) represented by black vertical lines and mean irregularity calculated over the population presented by bars. (D) Synchrony per population (variance of spike histogram divided by its mean) with bins of 10 ms divided by population mean firing rate.

CONCLUSION

This thesis focused on modeling neuronal networks and some mathematical tools and proprieties needed to do so.

The thesis proposed new approaches to known, yet limited, topics such as rescaling the network size while maintaining the first and second order statistics with the objective of decreasing machine time or increasing network details, and solving the boundary problem without loss of activity for spatially extended networks with topographic pattern of connections (Chapters 1 and 2).

A correlation between spatial resolution in activity and the standard deviation of the Gaussian distribution was utilized to propose a scheme for the construction of a network with topographic pattern of connections. This is the subject of Chapter 3.

The thesis also shows that the metastable phase transition depends on the spike-waste rate, and not on the network connectivity pattern (Chapter 4).

Lastly, the thesis presented a network model for the primary somatosensory cortex based on the GL stochastic neuron (Galves and Löcherbach, 2013). The network model was able to reproduce the biological average firing rate per layer, self-sustained activity without external input, and two activity states and the shift between them by a one-step interference in Layers L4e and L5e. This network also presented asynchronous activity and neurons with regular and irregular behaviour inside the same layer.

WORKS RESULTING FROM THIS THESIS

Articles submitted or placed in ArXiv

Romaro, C., Najman, F. A., Lytton, W. W., Roque, A. C., and Dura-Bernal, S. (2020). NetPyNE implementation and rescaling of the Potjans-Diesmann cortical microcircuit model. arXiv preprint arXiv:2005.03764. (Neural Computation)

Romaro, C., Najman, F. A., and André, M. (2019). A Numerical Study of the Time of Extinction in a Class of Systems of Spiking Neurons. arXiv preprint arXiv:1911.02609. (Journal of Statistical Physics)

Romaro, C., Roque, A. C., and Piqueira, J. R. C. (2020). Boundary solution based on rescalingmethod: recoup the first and second-order statistics of neuron network dynamics. arXivpreprint arXiv:2002.02381

Articles in preparation

Somatotopic organization in the cell-type specific cortical microcircuit and input spacial resolution. (Chapter 3)

Specific cortical microcircuit model based on somatossensory cortex connectome of juvenile rat. (Chapter 5)

Conference abstracts

Compensating method for the lack of connection on topographic neuron network edge (Submitted to CNS*2020)

Romaro, C., Najman, F. A., and André, M. (2019). Dynamical phase transitions study in simulations of finite neurons network. (In: Bojak, I. and Nowotny, T., eds. (2019) 28th Annual Computational Neuroscience Meeting: CNS*2019. BMC Neuroscience, pp190. doi: https://**doi.org/10.1186/s12868-019-0538-0** Available at http://centaur.reading .ac.uk/88633/)

Romaro, C., Najman, F., Dura-Bernal, S., and Roque, A. C. (2018). Implementation of the Potjans-Diesmann cortical microcircuit model in NetPyNE/NEURON with rescaling option.

BMC Neuroscience, (64), sup-2. **This work won the Best Poster Award** (In: 27th Annual Computational Neuroscience Meeting (CNS*2018): Part One. BMC Neurosci 19, 64 (2018). https://**doi.org/10.1186/s12868-018-0452-x**)

Romaro, C., Roque, A. C. (2017). Network model of the primary somatosensory cortex: a study of the connection pattern. SBNeC (2018). (In: XL SBNeC - Brazilian Society of Neurosciences and Behavior. (Presentation) UNESP – São Paulo State University, Santos, SP, Brazil.)

Romaro, C., Shimoura, R. O., Roque, A. C. (2017). Somatotopic representation in microcircuitry modeling of the mammalian cortex. (In: XL SBNeC - Brazilian Society of Neurosciences and Behavior. (Presentation) UNESP – São Paulo State University, Araraquara, SP, Brazil.)

Romaro, C., Shimoura, R. O., Kamiji, N. L., Roque, A. C. (2017). Reorganização somatotópica em um modelo de microcircuito local do córtex de mamífero. (In: XXII CBFM - Brazilian Congress of Medical Physics, 3399.)

REFERENCES

André, M. (2019b). A result of metastability for an infinite system of spiking neuron. *Journal of Statistical Physics*, 177:984–1008.

Antunes, G., Roque, A. C., and De Souza, F. M. S. (2016). Modelling intracellular competition for calcium: kinetic and thermodynamic control of different molecular modes of signal decoding. *Scientific Reports*, 6:23730.

Bear, M., Connors, B., and Paradiso, M. A. (2020). *Neuroscience: Exploring the brain*. Jones & Bartlett Learning, LLC.

Bower, J. M. and Beeman, D. (2012). *The book of GENESIS: exploring realistic neural models with the GEneral NEural SImulation System*. Springer Science & Business Media.

Brunel, N. (2000). Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. *Journal of Computational Neuroscience*, 8(3):183–208.

Carnevale, N. T. and Hines, M. L. (2006). *The NEURON Book*. Cambridge University Press.

de Kock, C. P. and Sakmann, B. (2009). Spiking in primary somatosensory cortex during natural whisking in awake head-restrained rats is cell-type specific. *Proceedings of the National Academy of Sciences (USA)*, 106(38):16446–16450.

De Schutter, E. (2008). Why are computational neuroscience and systems biology so separate? *PLoS Computational Biology*, 4(5):e1000078.

Diekman, C. O., Fall, C. P., Lechleiter, J. D., and Terman, D. (2013). Modeling the neuroprotective role of enhanced astrocyte mitochondrial metabolism during stroke. *Biophysical Journal*, 104(8):1752–1763.

Dura-Bernal, S., Suter, B. A., Gleeson, P., Cantarelli, M., Quintana, A., Rodriguez, F., Kedziora, D. J., Chadderdon, G. L., Kerr, C. C., Neymotin, S. A., et al. (2019b). Netpyne, a tool for data-driven multiscale modeling of brain circuits. *Elife*, 8:e44494.

Ferrari, P., Galves, A., Grigorescu, I., and Löcherbach, E. (2018). Phase transition for infinite systems of spiking neurons. *Journal of Statistical Physics*, 172(6):1564–1575.

Galves, A. and Löcherbach, E. (2013). Infinite systems of interacting chains with memory of variable length—a stochastic model for biological neural nets. *Journal of Statistical Physics*, 151(5):896–921.

Gewaltig, M.-O. and Diesmann, M. (2007). NEST (neural simulation tool). *Scholarpedia*, 2(4):1430.

Goodman, D. F. and Brette, R. (2009). The brian simulator. *Frontiers in Neuroscience*, 3:26.

Hodgkin, A. L., Huxley, A. F., and Katz, B. (1952). Measurement of current-voltage relations in the membrane of the giant axon of loligo. *Journal of Physiology*, 116(4):424–448.

Jones, A. R., Overly, C. C., and Sunkin, S. M. (2009). The allen brain atlas: 5 years and beyond. *Nature Reviews Neuroscience*, 10(11):821–828.

Lapicque, L. (1907). Recherches quantitatives sur l'excitation electrique des nerfs traitee comme une polarization. *Journal de Physiologie et de Pathologie Generale*, 9:620–635.

Lisman, J. and Raghavachari, S. (2006). A unified model of the presynaptic and postsynaptic changes during ltp at ca1 synapses. *Science's STKE*, 2006(356):re11–re11.

Markram, H. (2012). The human brain project. *Scientific American*, 306(6):50–55.

Markram, H., Muller, E., Ramaswamy, S., Reimann, M. W., Abdellah, M., Sanchez, C. A., Ailamaki, A., Alonso-Nanclares, L., Antille, N., Arsever, S., et al. (2015). Reconstruction and simulation of neocortical microcircuitry. *Cell*, 163(2):456–492.

Mazza, M., De Pinho, M., Piqueira, J. R. C., and Roque, A. C. (2004). A dynamical model of fast cortical reorganization. *Journal of Computational Neuroscience*, 16(2):177–201.

Morrison, A., Mehring, C., Geisel, T., Aertsen, A., and Diesmann, M. (2005). Advancing the boundaries o high-connectivity network simulation with distributed computing. *Neural Computation*, 17:1776–1801.

Potjans, T. C. and Diesmann, M. (2014a). The cell-type specific cortical microcircuit: relating structure and activity in a full-scale spiking network model. *Cerebral Cortex*, 24(3):785–806.

Ramachandran, V. S. (1998). Consciousness and body image: lessons from phantom limbs, capgras syndrome and pain asymbolia. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 353(1377):1851–1859.

Romaro, C., Najman, F., Dura-Bernal, S., and Roque, A. C. (2018). Implementation of the potjans-diesmann cortical microcircuit model in netpyne/neuron with rescaling option. *BMC Neuroscience*, (64):sup–2.

Romaro, C., Najman, F. A., and André, M. (2019). A numerical study of the time of extinction in a class of systems of spiking neurons. *arXiv:1911.02609*.

Romaro, C., Najman, F. A., Lytton, W. W., Roque, A. C., and Dura-Bernal, S. (2020a). NetPyNE implementation and rescaling of the Potjans-Diesmann cortical microcircuit model. *arXiv:2005.03764*.

Sakata, S. and Harris, K. D. (2009). Laminar structure of spontaneous and sensory-evoked population activity in auditory cortex. *Neuron*, 64(3):404–418.

Swadlow, H. A. (1989). Efferent neurons and suspected interneurons in s-1 vibrissa cortex of the awake rabbit: receptive fields and axonal properties. *Journal of Neurophysiology*, 62(1):288–308.

Towns, J., Cockerill, T., Dahan, M., Foster, I., Gaither, K., Grimshaw, A., Hazlewood, V., Lathrop, S., Lifka, D., Peterson, G. D., et al. (2014). Xsede: accelerating scientific discovery. *Computing in Science & Engineering*, 16(5):62–74.

Yin, X. and Wang, R. (2016). The computational simulation of the effect of dopamine on short-term and long-term memory. In *Advances in Cognitive Neurodynamics (V)*, pages 645–652. Springer.